

Package ‘wpa’

September 1, 2021

Type Package

Title Tools for Analysing and Visualising Workplace Analytics Data

Version 1.6.1

Description Opinionated functions that enable easier and faster analysis of Workplace Analytics data. There are three main types of functions in 'wpa': (i) Standard functions create a 'ggplot' visual or a summary table based on a specific Workplace Analytics metric; (2) Report Generation functions generate HTML reports on a specific analysis area, e.g. Collaboration; (3) Other miscellaneous functions cover more specific applications (e.g. Subject Line text mining) of Workplace Analytics data. This package adheres to 'tidyverse' principles and works well with the pipe syntax. 'wpa' is built with the beginner-to-intermediate R users in mind, and is optimised for simplicity.

URL <https://github.com/microsoft/wpa/>

BugReports <https://github.com/microsoft/wpa/issues/>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.1.2)

Imports dplyr, stats, utils, tidyr, tidyselect (>= 1.0.0), magrittr, purrr, reshape2, ggplot2, ggrepel, scales, htmltools, markdown, rmarkdown, networkD3, DT, tidytext, ggraph, igraph, proxy, ggwordcloud, methods, data.table

RoxygenNote 7.1.1

VignetteBuilder knitr

Suggests knitr, extrafont, leiden, lifecycle, fst, glue, flexdashboard, lmtest, portes, sandwich

Language en-US

NeedsCompilation no

Author Martin Chan [aut, cre],
Carlos Morales [aut],

Mark Powers [ctb],
 Ainize Cidoncha [ctb],
 Rosamary Ochoa Vargas [ctb],
 Tannaz Sattari [ctb],
 Lucas Hogner [ctb],
 Jasminder Thind [ctb],
 Simone Liebal [ctb],
 Aleksey Ashikhmin [ctb],
 Ellen Trinklein [ctb],
 Microsoft Corporation [cph]

Maintainer Martin Chan <martin.chan@microsoft.com>

Repository CRAN

Date/Publication 2021-09-01 09:00:02 UTC

R topics documented:

| | |
|--------------------------------|----|
| afterhours_dist | 5 |
| afterhours_fizz | 7 |
| afterhours_line | 8 |
| afterhours_rank | 10 |
| afterhours_summary | 11 |
| afterhours_trend | 12 |
| anonymise | 14 |
| calculate_IV | 15 |
| camel_clean | 15 |
| capacity_report | 16 |
| check_inputs | 17 |
| check_query | 18 |
| coaching_report | 19 |
| collaboration_area | 20 |
| collaboration_dist | 21 |
| collaboration_fizz | 23 |
| collaboration_line | 25 |
| collaboration_rank | 26 |
| collaboration_report | 28 |
| collaboration_sum | 29 |
| collaboration_trend | 31 |
| combine_signals | 32 |
| comma | 33 |
| connectivity_report | 33 |
| copy_df | 34 |
| create_bar | 35 |
| create_bar_asis | 37 |
| create_boxplot | 39 |
| create_bubble | 41 |
| create_dist | 43 |
| create_dt | 45 |

| | |
|-------------------------------------|-----|
| create_fizz | 46 |
| create_hist | 47 |
| create_ITSA | 49 |
| create_IV | 51 |
| create_line | 52 |
| create_line_asis | 54 |
| create_period_scatter | 56 |
| create_rank | 58 |
| create_rank_combine | 60 |
| create_sankey | 61 |
| create_scatter | 63 |
| create_stacked | 64 |
| create_tracking | 66 |
| create_trend | 68 |
| cut_hour | 69 |
| dv_data | 70 |
| email_dist | 73 |
| email_fizz | 74 |
| email_line | 76 |
| email_rank | 77 |
| email_summary | 79 |
| email_trend | 80 |
| em_data | 81 |
| export | 84 |
| external_network_plot | 85 |
| extract_date_range | 86 |
| extract_hr | 87 |
| flag_ch_ratio | 88 |
| flag_em_ratio | 89 |
| flag_extreme | 90 |
| flag_outlooktime | 91 |
| flex_index | 92 |
| g2g_data | 95 |
| generate_report | 96 |
| generate_report2 | 98 |
| heat_colours | 99 |
| hrvar_count | 100 |
| hrvar_count_all | 101 |
| hrvar_trend | 102 |
| hr_trend | 103 |
| identify_churn | 105 |
| identify_holidayweeks | 106 |
| identify_inactiveweeks | 108 |
| identify_nkw | 109 |
| identify_outlier | 110 |
| identify_privacythreshold | 111 |
| identify_query | 112 |
| identify_shifts | 113 |

| | |
|-----------------------|-----|
| identify_shifts_wp | 114 |
| identify_tenure | 116 |
| import_to_fst | 118 |
| import_wpa | 119 |
| internal_network_plot | 120 |
| is_date_format | 121 |
| IV_by_period | 122 |
| IV_report | 123 |
| jitter_metrics | 125 |
| keymetrics_scan | 125 |
| map_IV | 127 |
| maxmin | 128 |
| meetingtype_dist | 129 |
| meetingtype_dist_ca | 130 |
| meetingtype_dist_mt | 131 |
| meetingtype_summary | 132 |
| meeting_dist | 134 |
| meeting_extract | 135 |
| meeting_fizz | 136 |
| meeting_line | 138 |
| meeting_quality | 139 |
| meeting_rank | 141 |
| meeting_skim | 143 |
| meeting_summary | 144 |
| meeting_tm_report | 145 |
| meeting_trend | 146 |
| mgrcoatt_dist | 147 |
| mgrrel_matrix | 148 |
| mt_data | 150 |
| network_describe | 152 |
| network_g2g | 153 |
| network_leiden | 156 |
| network_louvain | 158 |
| network_p2p | 160 |
| one2one_dist | 163 |
| one2one_fizz | 165 |
| one2one_freq | 166 |
| one2one_line | 168 |
| one2one_rank | 169 |
| one2one_sum | 171 |
| one2one_trend | 173 |
| p2p_data_sim | 174 |
| pairwise_count | 175 |
| period_change | 176 |
| personas_hclust | 177 |
| plot_flex_index | 179 |
| plot_WOE | 180 |
| p_test | 181 |

| | |
|-------------------------------------|-----|
| read_preamble | 182 |
| remove_outliers | 182 |
| rgb2hex | 183 |
| sq_data | 184 |
| standardise_pq | 186 |
| subject_classify | 187 |
| subject_scan | 188 |
| subject_validate | 190 |
| subject_validate_report | 191 |
| theme_wpa | 192 |
| theme_wpa_basic | 193 |
| tm_clean | 193 |
| tm_cooc | 194 |
| tm_freq | 195 |
| tm_wordcloud | 196 |
| totals_bind | 198 |
| totals_col | 199 |
| totals_reorder | 200 |
| track_HR_change | 201 |
| tstamp | 202 |
| us_to_space | 203 |
| validation_report | 203 |
| wellbeing_report | 205 |
| workloads_dist | 207 |
| workloads_fizz | 208 |
| workloads_line | 210 |
| workloads_rank | 211 |
| workloads_summary | 213 |
| workloads_trend | 214 |
| workpatterns_area | 215 |
| workpatterns_classify | 217 |
| workpatterns_classify_bw | 221 |
| workpatterns_classify_pav | 223 |
| workpatterns_hclust | 225 |
| workpatterns_rank | 227 |
| workpatterns_report | 228 |
| wrap | 229 |

Index**231**

| | |
|-----------------|--|
| afterhours_dist | <i>Distribution of After-hours Collaboration Hours as a 100% stacked bar</i> |
|-----------------|--|

Description

Analyse the distribution of weekly after-hours collaboration time. Returns a stacked bar plot by default. Additional options available to return a table with distribution elements.

Usage

```
afterhours_dist(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  cut = c(1, 2, 3)
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |
| cut | A vector specifying the cuts to use for the data, accepting "default" or "range-cut" as character vector, or a numeric value of length three to specify the exact breaks to use. e.g. c(1, 3, 5) |

Details

Uses the metric `After_hours_collaboration_hours`. See `create_dist()` for applying the same analysis to a different metric.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#),

keymetrics_scan(), meeting_dist(), meeting_fizz(), meeting_line(), meeting_quality(), meeting_rank(), meeting_summary(), meeting_trend(), meetingtype_dist_ca(), meetingtype_dist_mt(), meetingtype_dist(), meetingtype_summary(), mgrcoatt_dist(), mgrrel_matrix(), one2one_dist(), one2one_fizz(), one2one_freq(), one2one_line(), one2one_rank(), one2one_sum(), one2one_trend(), period_change(), workloads_dist(), workloads_fizz(), workloads_line(), workloads_rank(), workloads_summary(), workloads_trend(), workpatterns_area(), workpatterns_rank()

Other After-hours Collaboration: afterhours_fizz(), afterhours_line(), afterhours_rank(), afterhours_summary(), afterhours_trend()

Examples

```
# Return plot
afterhours_dist(sq_data, hrvar = "Organization")

# Return summary table
afterhours_dist(sq_data, hrvar = "Organization", return = "table")

# Return result with a custom specified breaks
afterhours_dist(sq_data, hrvar = "LevelDesignation", cut = c(4, 7, 9))
```

afterhours_fizz *Distribution of After-hours Collaboration Hours (Fizzy Drink plot)*

Description

Analyze weekly after-hours collaboration hours distribution, and returns a 'fizzy' scatter plot by default. Additional options available to return a table with distribution elements.

Usage

```
afterhours_fizz(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> "plot" "table" See Value for more information. |

Details

Uses the metric `After_hours_collaboration_hours`. See `create_fizz()` for applying the same analysis to a different metric.

Value

A different output is returned depending on the value passed to the `return` argument:

- `"plot"`: 'ggplot' object. A jittered scatter plot for the metric.
- `"table"`: data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other After-hours Collaboration: `afterhours_dist()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`

Examples

```
# Return plot
afterhours_fizz(sq_data, hrvar = "LevelDesignation", return = "plot")

# Return summary table
afterhours_fizz(sq_data, hrvar = "Organization", return = "table")
```

afterhours_line

After-hours Collaboration Time Trend - Line Chart

Description

Provides a week by week view of after-hours collaboration time, visualized as line charts. By default returns a line chart for after-hours collaboration hours, with a separate panel per value in the HR attribute. Additional options available to return a summary table.

Usage

```
afterhours_line(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Details

Uses the metric `After_hours_collaboration_hours`.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A faceted line plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

[create_line\(\)](#) for applying the same analysis to a different metric.

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other After-hours Collaboration: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#)

Examples

```
# Return a line plot
afterhours_line(sq_data, hrvar = "LevelDesignation")

# Return summary table
afterhours_line(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|-----------------|--|
| afterhours_rank | <i>Rank groups with high After-Hours Collaboration Hours</i> |
|-----------------|--|

Description

This function scans a Standard Person Query for groups with high levels of After-Hours Collaboration. Returns a table with a all of groups (across multiple HR attributes) ranked by hours of After-Hours Collaboration Hours.

Usage

```
afterhours_rank(data, hrvar = extract_hr(data), mingroup = 5, return = "table")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> "table" (default) "plot" See Value for more information. |

Details

Uses the metric After_hours_collaboration_hours. See create_rank() for applying the same analysis to a different metric.

Value

When 'table' is passed in return, a summary table is returned as a data frame.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other After-hours Collaboration: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_summary()`, `afterhours_trend()`

afterhours_summary *Summary of After-Hours Collaboration Hours*

Description

Provides an overview analysis of after-hours collaboration time. Returns a bar plot showing average weekly after-hours collaboration hours by default. Additional options available to return a summary table.

Usage

```
afterhours_summary(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

```
afterhours_sum(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|-----------------------|--|
| <code>data</code> | A Standard Person Query dataset in the form of a data frame. |
| <code>hrvar</code> | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| <code>mingroup</code> | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| <code>return</code> | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> "plot" "table" |

See Value for more information.

Details

Uses the metric `After_hours_collaboration_hours`.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other After-hours Collaboration: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_trend()`

Examples

```
# Return a ggplot bar chart
afterhours_summary(sq_data, hrvar = "LevelDesignation")

# Return a summary table
afterhours_summary(sq_data, hrvar = "LevelDesignation", return = "table")
```

afterhours_trend

After-Hours Time Trend

Description

Provides a week by week view of after-hours collaboration time. By default returns a week by week heatmap, highlighting the points in time with most activity. Additional options available to return a summary table.

Usage

```
afterhours_trend(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |

Details

Uses the metric After_hours_collaboration_hours.

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other After-hours Collaboration: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#)

Examples

```
# Run plot
afterhours_trend(sq_data)

# Run table
```

```
afterhours_trend(sq_data, hrvar = "LevelDesignation", return = "table")
```

anonymise

Anonymise a categorical variable by replacing values

Description

Anonymize categorical variables such as HR variables by replacing values with dummy team names such as 'Team A'. The behaviour is to make 1 to 1 replacements by default, but there is an option to completely randomise values in the categorical variable.

Usage

```
anonymise(x, scramble = FALSE, replacement = NULL)
```

```
anonymize(x, scramble = FALSE, replacement = NULL)
```

Arguments

| | |
|-------------|--|
| x | Character vector to be passed through. |
| scramble | Logical value determining whether to randomise values in the categorical variable. |
| replacement | Character vector containing the values to replace original values in the categorical variable. The length of the vector must be at least as great as the number of unique values in the original variable. Defaults to NULL, where the replacement would consist of "Team A", "Team B", etc. |

See Also

jitter

Examples

```
unique(anonymise(sq_data$Organization))

rep <- c("Manager+", "Manager", "IC")
unique(anonymise(sq_data$Layer), replacement = rep)
```

| | |
|--------------|--|
| calculate_IV | <i>Calculate Weight of Evidence (WOE) and Information Value (IV) between a single predictor and a single outcome variable.</i> |
|--------------|--|

Description

Calculates Weight of Evidence (WOE) and Information Value (IV) between a single predictor and a single outcome variable. This function implements the common Information Value calculations whilst maintaining the minimum reliance on external dependencies. Use `map_IV()` for the equivalent of `Information::create_infotables()`, which performs calculations for multiple predictors and a single outcome variable.

Usage

```
calculate_IV(data, outcome, predictor, bins)
```

Arguments

| | |
|-----------|---|
| data | Data frame containing the data. |
| outcome | String containing the name of the outcome variable. |
| predictor | String containing the name of the predictor variable. |
| bins | Numeric value representing the number of bins to use. |

Details

The approach used mirrors the one used in `Information::create_infotables()`.

Value

A data frame is returned as an output.

| | |
|-------------|--|
| camel_clean | <i>Convert "CamelCase" to "Camel Case"</i> |
|-------------|--|

Description

Convert a text string from the format "CamelCase" to "Camel Case". This is used for converting variable names such as "LevelDesignation" to "Level Designation" for the purpose of prettifying plot labels.

Usage

```
camel_clean(string)
```

Arguments

string A string vector in 'CamelCase' format to format

Value

Returns a formatted string.

See Also

Other Support: [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
camel_clean("NoteHowTheStringIsFormatted")
```

| | |
|-----------------|---|
| capacity_report | <i>Generate a Capacity report in HTML</i> |
|-----------------|---|

Description

The function generates an interactive HTML report using the Standard Person Query data as an input. The report contains a series of summary analysis and visualisations relating to key **capacity** metrics in Workplace Analytics, including length of week and time in after-hours collaboration.

Usage

```
capacity_report(
  data,
  hrvar = "Organization",
  mingroup = 5,
  path = "capacity report",
  timestamp = TRUE
)
```

Arguments

data A Standard Person Query dataset in the form of a data frame.

hrvar String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes).

mingroup Numeric value setting the privacy threshold / minimum group size. Defaults to 5.

| | |
|-----------|---|
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . For example, "capacity report". |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

See Also

Other Reports: [IV_report\(\)](#), [coaching_report\(\)](#), [collaboration_report\(\)](#), [connectivity_report\(\)](#), [generate_report\(\)](#), [meeting_tm_report\(\)](#), [read_preamble\(\)](#), [subject_validate_report\(\)](#), [validation_report\(\)](#), [workpatterns_report\(\)](#)

| | |
|--------------|--|
| check_inputs | <i>Check whether a data frame contains all the required variable</i> |
|--------------|--|

Description

Checks whether a data frame contains all the required variables. Matching works via variable names, and used to support individual functions in the package. Not used directly.

Usage

```
check_inputs(input, requirements, return = "stop")
```

Arguments

| | |
|--------------|--|
| input | Pass a data frame for checking |
| requirements | A character vector specifying the required variable names |
| return | A character string specifying what to return. The default value is "stop". Also accepts "names" and "warning". |

Value

The default behaviour is to return an error message, informing the user what variables are not included. When return is set to "names", a character vector containing the unmatched variable names is returned.

See Also

Other Support: [camel_clean\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
# Return error message
## Not run:
check_inputs(iris, c("Sepal.Length", "mpg"))

## End(Not run)

#' # Return warning message
check_inputs(iris, c("Sepal.Length", "mpg"), return = "warning")

# Return variable names
check_inputs(iris, c("Sepal.Length", "Sepal.Width", "RandomVariable"), return = "names")
```

check_query

Check a query to ensure that it is suitable for analysis

Description

Prints diagnostic data about the data query to the R console, with information such as date range, number of employees, HR attributes identified, etc.

Usage

```
check_query(data, return = "message", validation = FALSE)
```

Arguments

| | |
|------------|--|
| data | A person-level query in the form of a data frame. This includes: <ul style="list-style-type: none"> • Standard Person Query • Ways of Working Assessment Query • Hourly Collaboration Query <p>All person-level query have a PersonId column and a Date column.</p> |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "message" (default) • "text" <p>See Value for more information.</p> |
| validation | Logical value to specify whether to show summarized version. Defaults to FALSE. To hide checks on variable names, set validation to TRUE. |

Details

This can be used with any person-level query, such as the standard person query, Ways of Working assessment query, and the hourly collaboration query. When run, this prints diagnostic data to the R console.

Value

A different output is returned depending on the value passed to the return argument:

- "message": a message is returned to the console.
- "text": string containing the diagnostic message.

See Also

Other Data Validation: [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
check_query(sq_data)
```

coaching_report

Generate a Coaching report in HTML

Description

The function generates an interactive HTML report using Standard Person Query data as an input. The report contains a series of summary analysis and visualisations relating to key **coaching** metrics in Workplace Analytics, specifically relating to the time spent between managers and their direct reports.

Usage

```
coaching_report(  
  data,  
  hrvar = "LevelDesignation",  
  mingroup = 5,  
  path = "coaching report",  
  timestamp = TRUE  
)
```

Arguments

| | |
|-------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |

| | |
|-----------|---|
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . For example, "coaching report". |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

See Also

Other Reports: [IV_report\(\)](#), [capacity_report\(\)](#), [collaboration_report\(\)](#), [connectivity_report\(\)](#), [generate_report\(\)](#), [meeting_tm_report\(\)](#), [read_preamble\(\)](#), [subject_validate_report\(\)](#), [validation_report\(\)](#), [workpatterns_report\(\)](#)

| | |
|--------------------|--|
| collaboration_area | <i>Collaboration - Stacked Area Plot</i> |
|--------------------|--|

Description

Provides an overview analysis of Weekly Digital Collaboration. Returns an stacked area plot of Email and Meeting Hours by default. Additional options available to return a summary table.

Usage

```
collaboration_area(data, hrvar = NULL, mingroup = 5, return = "plot")
```

```
collab_area(data, hrvar = NULL, mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. A Ways of Working assessment dataset may also be provided, in which Unscheduled call hours would be included in the output. |
| hrvar | HR Variable by which to split metrics, defaults to NULL, but accepts any character vector, e.g. "LevelDesignation". If NULL is passed, the organizational attribute is automatically populated as "Total". |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Details

Uses the metrics Meeting_hours, Email_hours, Unscheduled_Call_hours, and Instant_Message_hours.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked area plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Collaboration: [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#)

Examples

```
# Return plot with total (default)
collaboration_area(sq_data)

# Return plot with hrvar split
collaboration_area(sq_data, hrvar = "Organization")

# Return summary table
collaboration_area(sq_data, return = "table")
```

collaboration_dist *Distribution of Collaboration Hours as a 100% stacked bar*

Description

Analyze the distribution of Collaboration Hours. Returns a stacked bar plot by default. Additional options available to return a table with distribution elements.

Usage

```

collaboration_dist(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  cut = c(15, 20, 25)
)

collab_dist(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  cut = c(15, 20, 25)
)

```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |
| cut | A numeric vector of length three to specify the breaks for the distribution, e.g. c(10, 15, 20) |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked bar plot for the metric.
- "table": data frame. A summary table for the metric.

Metrics used

The metric `Collaboration_hours` is used in the calculations. Please ensure that your query contains a metric with the exact same name.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Collaboration: `collaboration_area()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`

Examples

```
# Return plot
collaboration_dist(sq_data, hrvar = "Organization")

# Return summary table
collaboration_dist(sq_data, hrvar = "Organization", return = "table")
```

`collaboration_fizz` *Distribution of Collaboration Hours (Fizzy Drink plot)*

Description

Analyze weekly collaboration hours distribution, and returns a 'fizzy' scatter plot by default. Additional options available to return a table with distribution elements.

Usage

```
collaboration_fizz(data, hrvar = "Organization", mingroup = 5, return = "plot")

collab_fizz(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|--------------------|--|
| <code>data</code> | A Standard Person Query dataset in the form of a data frame. |
| <code>hrvar</code> | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |

| | |
|----------|--|
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A jittered scatter plot for the metric.
- "table": data frame. A summary table for the metric.

Metrics used

The metric `Collaboration_hours` is used in the calculations. Please ensure that your query contains a metric with the exact same name.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Collaboration: `collaboration_area()`, `collaboration_dist()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`

Examples

```
# Return plot
collaboration_fizz(sq_data, hrvar = "Organization", return = "plot")

# Return summary table
collaboration_fizz(sq_data, hrvar = "Organization", return = "table")
```

collaboration_line *Collaboration Time Trend - Line Chart*

Description

Provides a week by week view of collaboration time, visualised as line charts. By default returns a line chart for collaboration hours, with a separate panel per value in the HR attribute. Additional options available to return a summary table.

Usage

```
collaboration_line(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

```
collab_line(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A faceted line plot for the metric.
- "table": data frame. A summary table for the metric.

Metrics used

The metric Collaboration_hours is used in the calculations. Please ensure that your query contains a metric with the exact same name.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Collaboration: `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`

Examples

```
# Return a line plot
collaboration_line(sq_data, hrvar = "LevelDesignation")

# Return summary table
collaboration_line(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|--------------------|------------------------------|
| collaboration_rank | <i>Collaboration Ranking</i> |
|--------------------|------------------------------|

Description

This function scans a standard query output for groups with high levels of 'Weekly Digital Collaboration'. Returns a table with a all of groups (across multiple HR attributes) ranked by hours of digital collaboration.

Usage

```
collaboration_rank(
  data,
  hrvar = extract_hr(data),
  mingroup = 5,
  mode = "simple",
  plot_mode = 1,
  return = "plot"
)
```

```
collab_rank(
  data,
  hrvar = extract_hr(data),
  mingroup = 5,
  mode = "simple",
  plot_mode = 1,
  return = "plot"
)
```

Arguments

| | |
|-----------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| mode | String to specify calculation mode. Must be either: <ul style="list-style-type: none"> • "simple" • "combine" |
| plot_mode | Numeric vector to determine which plot mode to return. Must be either 1 or 2, and is only used when return = "plot". <ul style="list-style-type: none"> • 1: Top and bottom five groups across the data population are highlighted • 2: Top and bottom groups <i>per</i> organizational attribute are highlighted |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "table" (default) • "plot" See Value for more information. |

Details

Uses the metric Collaboration_hours. See create_rank() for applying the same analysis to a different metric.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bubble plot where the x-axis represents the metric, the y-axis represents the HR attributes, and the size of the bubbles represent the size of the organizations. Note that there is no plot output if mode is set to "combine".
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Collaboration: `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_sum()`, `collaboration_trend()`

Examples

```
# Return rank table
collaboration_rank(
  data = sq_data,
  return = "table"
)

# Return plot
collaboration_rank(
  data = sq_data,
  return = "plot"
)
```

`collaboration_report` *Generate a Collaboration Report in HTML*

Description

The function generates an interactive HTML report using Standard Person Query data as an input. The report contains a series of summary analysis and visualisations relating to key **collaboration** metrics in Workplace Analytics, including email and meeting hours.

Usage

```
collaboration_report(
  data,
```

```

  hrvar = "AUTO",
  mingroup = 5,
  path = "collaboration report",
  timestamp = TRUE
)

```

Arguments

| | |
|-----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . For example, "collaboration report". |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

See Also

Other Reports: [IV_report\(\)](#), [capacity_report\(\)](#), [coaching_report\(\)](#), [connectivity_report\(\)](#), [generate_report\(\)](#), [meeting_tm_report\(\)](#), [read_preamble\(\)](#), [subject_validate_report\(\)](#), [validation_report\(\)](#), [workpatterns_report\(\)](#)

| | |
|-------------------|------------------------------|
| collaboration_sum | <i>Collaboration Summary</i> |
|-------------------|------------------------------|

Description

Provides an overview analysis of 'Weekly Digital Collaboration'. Returns a stacked bar plot of Email and Meeting Hours by default. Additional options available to return a summary table.

Usage

```

collaboration_sum(data, hrvar = "Organization", mingroup = 5, return = "plot")

collab_sum(data, hrvar = "Organization", mingroup = 5, return = "plot")

collaboration_summary(
  data,

```

```

  hrvar = "Organization",
  mingroup = 5,
  return = "plot"
)

collab_summary(data, hrvar = "Organization", mingroup = 5, return = "plot")

```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |

Details

Uses the metrics Meeting_hours, Email_hours, Unscheduled_Call_hours, and Instant_Message_hours.

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Collaboration: [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_trend\(\)](#)

 collaboration_trend *Collaboration Time Trend*

Description

Provides a week by week view of collaboration time. By default returns a week by week heatmap, highlighting the points in time with most activity. Additional options available to return a summary table.

Usage

```
collaboration_trend(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot"
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

Metrics used

The metric `Collaboration_hours` is used in the calculations. Please ensure that your query contains a metric with the exact same name.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#),

[email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#),
[external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#),
[keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#),
[meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#),
[meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#),
[one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#),
[period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#),
[workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)
 Other Collaboration: [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#),
[collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#)

combine_signals

Combine signals from the Hourly Collaboration query

Description

Takes in an Hourly Collaboration Data, and for each hour sums and aggregates the signals (e.g. Emails_sent and IMs_sent) in Signals_sent. This is an internal function used in the Working Patterns functions.

Usage

```
combine_signals(data, hr, signals = c("Emails_sent", "IMs_sent"))
```

Arguments

| | |
|---------|---|
| data | Hourly Collaboration query containing signal variables (e.g. Emails_sent_00_01) |
| hr | Numeric value between 0 to 23 to iterate through |
| signals | Character vector for specifying which signal types to combine. Defaults to c("Emails_sent", "IMs_sent"). Other valid values include "Unscheduled_calls" and "Meetings". |

Details

combine_signals uses string matching to aggregate columns.

Value

Returns a numeric vector that represents the sum of signals sent for a given hour.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
# Demo using simulated variables
sim_data <-
  data.frame(Emails_sent_09_10 = sample(1:5, size = 10, replace = TRUE),
            Unscheduled_calls_09_10 = sample(1:5, size = 10, replace = TRUE))

combine_signals(sim_data, hr = 9, signals = c("Emails_sent", "Unscheduled_calls"))
```

| | |
|-------|--|
| comma | <i>Add comma separator for thousands</i> |
|-------|--|

Description

Takes a numeric value and returns a character value which is rounded to the whole number, and adds a comma separator at the thousands. A convenient wrapper function around `round()` and `format()`.

Usage

```
comma(x)
```

Arguments

x A numeric value

Value

Returns a formatted string.

| | |
|---------------------|---|
| connectivity_report | <i>Generate a Connectivity report in HTML</i> |
|---------------------|---|

Description

The function generates an interactive HTML report using Standard Person Query data as an input. The report contains a series of summary analysis and visualisations relating to key **connectivity** metrics in Workplace Analytics, including external/internal network size vs breadth.

Usage

```
connectivity_report(
  data,
  hrvar = "LevelDesignation",
  mingroup = 5,
  path = "connectivity report",
  timestamp = TRUE
)
```

Arguments

| | |
|-----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . For example, "connectivity report". |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

See Also

Other Reports: [IV_report\(\)](#), [capacity_report\(\)](#), [coaching_report\(\)](#), [collaboration_report\(\)](#), [generate_report\(\)](#), [meeting_tm_report\(\)](#), [read_preamble\(\)](#), [subject_validate_report\(\)](#), [validation_report\(\)](#), [workpatterns_report\(\)](#)

copy_df

Copy a data frame to clipboard for pasting in Excel

Description

This is a pipe-optimised function, that feeds into `wpa::export()`, but can be used as a stand-alone function.

Based on the original function from <https://github.com/martinctc/surveytoolbox>.

Usage

```
copy_df(x, row.names = FALSE, col.names = TRUE, quietly = FALSE, ...)
```

Arguments

| | |
|-----------|---|
| x | Data frame to be passed through. Cannot contain list-columns or nested data frames. |
| row.names | A logical vector for specifying whether to allow row names. Defaults to FALSE. |
| col.names | A logical vector for specifying whether to allow column names. Defaults to FALSE. |
| quietly | Set this to TRUE to not print data frame on console |
| ... | Additional arguments for <code>write.table()</code> . |

Value

Copies a data frame to the clipboard with no return value.

See Also

Other Import and Export: [create_dt\(\)](#), [export\(\)](#), [import_to_fst\(\)](#), [import_wpa\(\)](#), [standardise_pq\(\)](#)

 create_bar

Mean Bar Plot for any metric

Description

Provides an overview analysis of a selected metric by calculating a mean per metric. Returns a bar plot showing the average of a selected metric by default. Additional options available to return a summary table.

Usage

```
create_bar(
  data,
  metric,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  bar_colour = "default",
  na.rm = FALSE,
  percent = FALSE,
  plot_title = us_to_space(metric),
  plot_subtitle = paste("Average by", tolower(camel_clean(hrvar))),
  rank = "descending",
  xlim = NULL,
  text_just = 0.5,
  text_colour = "#FFFFFF"
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: |

| | |
|---------------|---|
| | <ul style="list-style-type: none"> • "plot" • "table" |
| | See Value for more information. |
| bar_colour | String to specify colour to use for bars. In-built accepted values include "default" (default), "alert" (red), and "darkblue". Otherwise, hex codes are also accepted. You can also supply RGB values via <code>rgb2hex()</code> . |
| na.rm | A logical value indicating whether NA should be stripped before the computation proceeds. Defaults to FALSE. |
| percent | Logical value to determine whether to show labels as percentage signs. Defaults to FALSE. |
| plot_title | An option to override plot title. |
| plot_subtitle | An option to override plot subtitle. |
| rank | String specifying how to rank the bars. Valid inputs are: <ul style="list-style-type: none"> • "descending" - ranked highest to lowest from top to bottom (default). • "ascending" - ranked lowest to highest from top to bottom. • NULL - uses the original levels of the HR attribute. |
| xlim | An option to set max value in x axis. |
| text_just | [Experimental] A numeric value controlling for the horizontal position of the text labels. Defaults to 0.5. |
| text_colour | [Experimental] String to specify colour to use for the text labels. Defaults to "#FFFFFF". |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_hist()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `period_change()`

Examples

```
# Return a ggplot bar chart
create_bar(sq_data, metric = "Collaboration_hours", hrvar = "LevelDesignation")

# Change bar colour
create_bar(sq_data,
           metric = "After_hours_collaboration_hours",
           bar_colour = "alert")

# Custom data label positions and formatting
sq_data %>%
  create_bar(
    metric = "Meetings",
    text_just = 1.1,
    text_colour = "black",
    xlim = 20)

# Return a summary table
create_bar(sq_data,
           metric = "Collaboration_hours",
           hrvar = "LevelDesignation",
           return = "table")
```

create_bar_asis

Create a bar chart without aggregation for any metric

Description

This function creates a bar chart directly from the aggregated / summarised data. Unlike `create_bar()` which performs a person-level aggregation, there is no calculation for `create_bar_asis()` and the values are rendered as they are passed into the function.

Usage

```
create_bar_asis(
  data,
  group_var,
  bar_var,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab = group_var,
  xlab = bar_var,
```

```

    percent = FALSE,
    bar_colour = "default",
    rounding = 1
  )

```

Arguments

| | |
|-------------------------|--|
| <code>data</code> | Plotting data as a data frame. |
| <code>group_var</code> | String containing name of variable for the group. |
| <code>bar_var</code> | String containing name of variable representing the value of the bars. |
| <code>title</code> | Title of the plot. |
| <code>subtitle</code> | Subtitle of the plot. |
| <code>caption</code> | Caption of the plot. |
| <code>ylab</code> | Y-axis label for the plot (group axis) |
| <code>xlab</code> | X-axis label of the plot (bar axis). |
| <code>percent</code> | Logical value to determine whether to show labels as percentage signs. Defaults to FALSE. |
| <code>bar_colour</code> | String to specify colour to use for bars. In-built accepted values include "default" (default), "alert" (red), and "darkblue". Otherwise, hex codes are also accepted. You can also supply RGB values via <code>rgb2hex()</code> . |
| <code>rounding</code> | Numeric value to specify number of digits to show in data labels |

Value

'ggplot' object. A horizontal bar plot.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```

# Creating a custom bar plot without mean aggregation
library(dplyr)

sq_data %>%
  group_by(Organization) %>%
  summarise(across(.cols = Meeting_hours,
                  .fns = ~sum(., na.rm = TRUE))) %>%
  create_bar_asis(group_var = "Organization",
                 bar_var = "Meeting_hours",
                 title = "Total Meeting Hours over period",
                 subtitle = "By Organization",
                 caption = extract_date_range(sq_data, return = "text"),
                 bar_colour = "darkblue",
                 rounding = 0)

library(dplyr)

# Summarise Non-person-average median `Emails_sent`
med_df <-
  sq_data %>%
  group_by(Organization) %>%
  summarise(Emails_sent_median = median(Emails_sent))

med_df %>%
  create_bar_asis(
    group_var = "Organization",
    bar_var = "Emails_sent_median",
    title = "Median Emails Sent by Organization",
    subtitle = "Person Averaging Not Applied",
    bar_colour = "darkblue",
    caption = extract_date_range(sq_data, return = "text")
  )

```

create_boxplot

Box Plot for any metric

Description

Analyzes a selected metric and returns a box plot by default. Additional options available to return a table with distribution elements.

Usage

```

create_boxplot(
  data,
  metric,

```

```

    hrvar = "Organization",
    mingroup = 5,
    return = "plot"
)

```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Details

This is a general purpose function that powers all the functions in the package that produce box plots.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A box plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```
# Create a fizzy plot for Work Week Span by Level Designation
create_boxplot(sq_data,
               metric = "Workweek_span",
               hrvar = "LevelDesignation",
               return = "plot")

# Create a summary statistics table for Work Week Span by Organization
create_boxplot(sq_data,
               metric = "Workweek_span",
               hrvar = "Organization",
               return = "table")

# Create a fizzy plot for Collaboration Hours by Level Designation
create_boxplot(sq_data,
               metric = "Collaboration_hours",
               hrvar = "LevelDesignation",
               return = "plot")
```

| | |
|---------------|---|
| create_bubble | <i>Create a bubble plot with two selected Workplace Analytics metrics (General Purpose), with size representing the number of employees in the group.</i> |
|---------------|---|

Description

Returns a bubble plot of two selected metrics, using size to map the number of employees.

Usage

```
create_bubble(
  data,
  metric_x,
  metric_y,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  bubble_size = c(1, 10)
)
```

Arguments

| | |
|-------------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric_x | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| metric_y | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: - "plot" - "table" |
| bubble_size | A numeric vector of length two to specify the size range of the bubbles |

Details

This is a general purpose function that powers all the functions in the package that produce bubble plots.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bubble plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```

create_bubble(sq_data,
  "Internal_network_size",
  "External_network_size",
  "Organization")

create_bubble(sq_data,
  "Generated_workload_call_hours",
  "Generated_workload_email_hours",
  "Organization", mingroup = 100, return = "plot")

```

create_dist

Horizontal 100 percent stacked bar plot for any metric

Description

Provides an analysis of the distribution of a selected metric. Returns a stacked bar plot by default. Additional options available to return a table with distribution elements.

Usage

```

create_dist(
  data,
  metric,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  cut = c(15, 20, 25),
  dist_colours = c("#facebc", "#fcf0eb", "#b4d5dd", "#bfe5ee"),
  unit = "hours",
  lbound = 0,
  ubound = 100,
  sort_by = NULL,
  labels = NULL
)

```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | String containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |

| | |
|--------------|--|
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |
| cut | A numeric vector of length three to specify the breaks for the distribution, e.g. <code>c(10, 15, 20)</code> |
| dist_colours | A character vector of length four to specify colour codes for the stacked bars. |
| unit | String to specify what unit to use. This defaults to "hours" but can accept any custom string. See <code>cut_hour()</code> for more details. |
| lbound | Numeric. Specifies the lower bound (inclusive) value for the minimum label. Defaults to 0. |
| ubound | Numeric. Specifies the upper bound (inclusive) value for the maximum label. Defaults to 100. |
| sort_by | String to specify the bucket label to sort by. Defaults to NULL (no sorting). |
| labels | Character vector to override labels for the created categorical variables. Must be a named vector - see examples. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```

# Return plot
create_dist(sq_data, metric = "Collaboration_hours", hrvar = "Organization")

# Return summary table
create_dist(sq_data, metric = "Collaboration_hours", hrvar = "Organization", return = "table")

# Use custom labels by providing a label vector
eh_labels <- c(
  "Fewer than fifteen" = "< 15 hours",
  "Between fifteen and twenty" = "15 - 20 hours",
  "Between twenty and twenty-five" = "20 - 25 hours",
  "More than twenty-five" = "25+ hours"
)

sq_data %>%
  create_dist(metric = "Email_hours",
             labels = eh_labels, return = "plot")

# Sort by a category
sq_data %>%
  create_dist(metric = "Collaboration_hours",
             sort_by = "25+ hours")

```

create_dt

Create interactive tables in HTML with 'download' buttons.

Description

See <https://martinctc.github.io/blog/vignette-downloadable-tables-in-rmarkdown-with-the-dt-package/> for more.

Usage

```
create_dt(x, rounding = 1, freeze = 2)
```

Arguments

| | |
|----------|---|
| x | Data frame to be passed through. |
| rounding | Numeric vector to specify the number of decimal points to display |
| freeze | Number of columns from the left to 'freeze'. Defaults to 2, which includes the row number column. |

Value

Returns an HTML widget displaying rectangular data.

See Also

Other Import and Export: [copy_df\(\)](#), [export\(\)](#), [import_to_fst\(\)](#), [import_wpa\(\)](#), [standardise_pq\(\)](#)

`create_fizz`*Fizzy Drink / Jittered Scatter Plot for any metric*

Description

Analyzes a selected metric and returns a 'fizzy' scatter plot by default. Additional options available to return a table with distribution elements.

Usage

```
create_fizz(  
  data,  
  metric,  
  hrvar = "Organization",  
  mingroup = 5,  
  return = "plot"  
)
```

Arguments

| | |
|-----------------------|--|
| <code>data</code> | A Standard Person Query dataset in the form of a data frame. |
| <code>metric</code> | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| <code>hrvar</code> | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| <code>mingroup</code> | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| <code>return</code> | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "table" See Value for more information. |

Details

This is a general purpose function that powers all the functions in the package that produce 'fizzy drink' / jittered scatter plots.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A jittered scatter plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Flexible: `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_hist()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `period_change()`

Examples

```
# Create a fizzy plot for Work Week Span by Level Designation
create_fizz(sq_data, metric = "Workweek_span", hrvar = "LevelDesignation", return = "plot")

# Create a summary statistics table for Work Week Span by Organization
create_fizz(sq_data, metric = "Workweek_span", hrvar = "Organization", return = "table")

# Create a fizzy plot for Collaboration Hours by Level Designation
create_fizz(sq_data, metric = "Collaboration_hours", hrvar = "LevelDesignation", return = "plot")
```

create_hist

Create a histogram plot for any metric

Description

Provides an analysis of the distribution of a selected metric. Returns a faceted histogram by default. Additional options available to return the underlying frequency table.

Usage

```
create_hist(
  data,
  metric,
  hrvar = "Organization",
  mingroup = 5,
  binwidth = 1,
```

```
    return = "plot"
  )
```

Arguments

| | |
|----------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | String containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| binwidth | Numeric value for setting binwidth argument within <code>ggplot2::geom_histogram()</code> . Defaults to 1. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" • "data" • "frequency" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A faceted histogram for the metric.
- "table": data frame. A summary table for the metric.
- "data": data frame. Data with calculated person averages.
- "frequency": list of data frames. Each data frame contains the frequencies used in each panel of the plotted histogram.

See Also

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```
# Return plot for whole organization
create_hist(sq_data, metric = "Collaboration_hours", hrvar = NULL)

# Return plot
create_hist(sq_data, metric = "Collaboration_hours", hrvar = "Organization")

# Return summary table
```



```
create_hist(sq_data,
            metric = "Collaboration_hours",
            hrvar = "Organization",
            return = "table")
```

| | |
|-------------|--|
| create_ITSA | <i>Estimate an effect of intervention on every WPA metric in input file by applying single-group Interrupted Time-Series Analysis (ITSA)</i> |
|-------------|--|

Description

`r lifecycle::badge('experimental')`

This function implements ITSA method described in the paper 'Conducting interrupted time-series analysis for single- and multiple-group comparisons', Ariel Linden, The Stata Journal (2015), 15, Number 2, pp. 480-500

This function further requires the installation of 'sandwich', 'portex', and 'lmtest' in order to work. These packages can be installed from CRAN using `install.packages()`.

Usage

```
create_ITSA(
  data,
  before_start = min(as.Date(data$Date, "%m/%d/%Y")),
  before_end,
  after_start,
  after_end = max(as.Date(data$Date, "%m/%d/%Y")),
  ac_lags_max = 7,
  return = "table"
)
```

Arguments

| | |
|--------------|---|
| data | Person Query as a dataframe including date column named Date. This function assumes the data format is MM/DD/YYYY as is standard in a Workplace Analytics query output. |
| before_start | Start date of 'before' time period in MM/DD/YYYY format as character type. Before time period is the period before the intervention (e.g. training program, re-org, shift to remote work) occurs and bounded by before_start and before_end parameters. Longer period increases likelihood of achieving more statistically significant results. Defaults to earliest date in dataset. |
| before_end | End date of 'before' time period in MM/DD/YYYY format as character type. |
| after_start | Start date of 'after' time period in MM/DD/YYYY format as character type. After time period is the period after the intervention occurs and bounded by after_start and after_end parameters. Longer period increases likelihood of achieving more statistically significant results. Defaults to date after before_end. |

| | |
|-------------|---|
| after_end | End date of 'after' time period in MM/DD/YYYY format as character type. Defaults to latest date in dataset. |
| ac_lags_max | maximum lag for autocorrelation test. Default is 7 |
| return | String specifying what output to return. Defaults to "table". Valid return options include: <ul style="list-style-type: none">• 'plot': return a list of plots.• 'table': return data.frame with estimated models' coefficients and their corresponding p-values You should look for significant p-values in beta_2 to indicate an immediate treatment effect, and/or in beta_3 to indicate a treatment effect over time |

Details

This function uses the additional package dependencies 'sandwich', 'lmtest', and 'portes'. Please install these separately from CRAN prior to running the function.

Author(s)

Aleksey Ashikhmin alashi@microsoft.com

See Also

Other Flexible Input: [period_change\(\)](#)

Examples

```
# Returns summary table

create_ITSA(
  data = sq_data,
  before_start = "11/03/2019",
  before_end = "12/15/2019",
  after_start = "12/29/2019",
  after_end = "1/26/2020",
  ac_lags_max = 7,
  return = "table")

# Returns list of plots

plot_list <-
  create_ITSA(
    data = sq_data,
    before_start = "11/03/2019",
    before_end = "12/15/2019",
    after_start = "12/29/2019",
    after_end = "1/26/2020",
    ac_lags_max = 7,
    return = 'plot')
```

```
# Extract a plot as an example
plot_list$Workweek_span
```

create_IV

Calculate Information Value for a selected outcome variable

Description

Specify an outcome variable and return IV outputs. All numeric variables in the dataset are used as predictor variables.

Usage

```
create_IV(
  data,
  predictors = NULL,
  outcome,
  bins = 5,
  siglevel = 0.05,
  exc_sig = FALSE,
  return = "plot"
)
```

Arguments

| | |
|------------|--|
| data | A Person Query dataset in the form of a data frame. |
| predictors | A character vector specifying the columns to be used as predictors. Defaults to NULL, where all numeric vectors in the data will be used as predictors. |
| outcome | A string specifying a binary variable, i.e. can only contain the values 1 or 0. |
| bins | Number of bins to use, defaults to 5. |
| siglevel | Significance level to use in comparing populations for the outcomes, defaults to 0.05 |
| exc_sig | Logical value determining whether to exclude values where the p-value lies below what is set at siglevel. Defaults to FALSE. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "summary" • "list" • "plot-WOE" • "IV" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bar plot showing the IV value of the top (maximum 12) variables.
- "summary": data frame. A summary table for the metric.
- "list": list. A list of outputs for all the input variables.
- "plot-WOE": A list of 'ggplot' objects that show the WOE for each predictor used in the model.
- "IV" returns a list object which mirrors the return in `Information::create_infotables()`.

See Also

Other Variable Association: [IV_by_period\(\)](#), [IV_report\(\)](#), [plot_WOE\(\)](#)

Other Information Value: [IV_by_period\(\)](#), [IV_report\(\)](#), [plot_WOE\(\)](#)

Examples

```
# Return a summary table of IV
sq_data %>%
  dplyr::mutate(X = ifelse(Workweek_span > 40, 1, 0)) %>%
  create_IV(outcome = "X",
            predictors = c("Email_hours",
                          "Meeting_hours",
                          "Instant_Message_hours"),
            return = "plot")

# Return summary
sq_data %>%
  dplyr::mutate(X = ifelse(Collaboration_hours > 2, 1, 0)) %>%
  create_IV(outcome = "X",
            predictors = c("Email_hours", "Meeting_hours"),
            return = "summary")
```

create_line

Time Trend - Line Chart for any metric

Description

Provides a week by week view of a selected metric, visualised as line charts. By default returns a line chart for the defined metric, with a separate panel per value in the HR attribute. Additional options available to return a summary table.

Usage

```
create_line(
  data,
  metric,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot"
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Details

This is a general purpose function that powers all the functions in the package that produce faceted line plots.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A faceted line plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#),

```
meetingtype_dist(), meetingtype_summary(), mgrcoatt_dist(), mgrrel_matrix(), one2one_dist(),
one2one_fizz(), one2one_freq(), one2one_line(), one2one_rank(), one2one_sum(), one2one_trend(),
period_change(), workloads_dist(), workloads_fizz(), workloads_line(), workloads_rank(),
workloads_summary(), workloads_trend(), workpatterns_area(), workpatterns_rank()
```

```
Other Flexible: create_bar_asis(), create_bar(), create_boxplot(), create_bubble(), create_dist(),
create_fizz(), create_hist(), create_line_asis(), create_period_scatter(), create_rank(),
create_sankey(), create_scatter(), create_stacked(), create_tracking(), create_trend(),
period_change()
```

```
Other Time-series: IV_by_period(), create_line_asis(), create_period_scatter(), create_trend(),
period_change()
```

Examples

```
# Return plot of Email Hours
sq_data %>% create_line(metric = "Email_hours", return = "plot")

# Return plot of Collaboration Hours
sq_data %>% create_line(metric = "Collaboration_hours", return = "plot")

# Return plot of Work week span and cut by `LevelDesignation`
sq_data %>% create_line(metric = "Workweek_span", hrvar = "LevelDesignation")
```

| | |
|------------------|---|
| create_line_asis | <i>Create a line chart without aggregation for any metric</i> |
|------------------|---|

Description

This function creates a line chart directly from the aggregated / summarised data. Unlike `create_line()` which performs a person-level aggregation, there is no calculation for `create_line_asis()` and the values are rendered as they are passed into the function. The only requirement is that a `date_var` is provided for the x-axis.

Usage

```
create_line_asis(
  data,
  date_var = "Date",
  metric,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab = date_var,
  xlab = metric,
  line_colour = rgb2hex(0, 120, 212)
)
```

Arguments

| | |
|-------------|---|
| data | Plotting data as a data frame. |
| date_var | String containing name of variable for the horizontal axis. |
| metric | String containing name of variable representing the line. |
| title | Title of the plot. |
| subtitle | Subtitle of the plot. |
| caption | Caption of the plot. |
| ylab | Y-axis label for the plot (group axis) |
| xlab | X-axis label of the plot (bar axis). |
| line_colour | String to specify colour to use for the line. Hex codes are accepted. You can also supply RGB values via <code>rgb2hex()</code> . |

Value

Returns a 'ggplot' object representing a line plot.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Other Time-series: [IV_by_period\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```
library(dplyr)

# Median `Emails_sent` grouped by `Date`
# Without Person Averaging
med_df <-
```

```

sq_data %>%
  group_by(Date) %>%
  summarise(Emails_sent_median = median(Emails_sent))

med_df %>%
  create_line_asis(
    date_var = "Date",
    metric = "Emails_sent_median",
    title = "Median Emails Sent",
    subtitle = "Person Averaging Not Applied",
    caption = extract_date_range(sq_data, return = "text")
  )

```

create_period_scatter *Period comparison scatter plot for any two metrics*

Description

Returns two side-by-side scatter plots representing two selected metrics, using colour to map an HR attribute and size to represent number of employees. Returns a faceted scatter plot by default, with additional options to return a summary table.

Usage

```

create_period_scatter(
  data,
  hrvar = "Organization",
  metric_x = "Multitasking_meeting_hours",
  metric_y = "Meeting_hours",
  before_start = min(as.Date(data$Date, "%m/%d/%Y")),
  before_end,
  after_start = as.Date(before_end) + 1,
  after_end = max(as.Date(data$Date, "%m/%d/%Y")),
  before_label = "Period 1",
  after_label = "Period 2",
  mingroup = 5,
  return = "plot"
)

```

Arguments

| | |
|----------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | HR Variable by which to split metrics. Accepts a character vector, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| metric_x | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| metric_y | Character string containing the name of the metric, e.g. "Collaboration_hours" |

| | |
|--------------|--|
| before_start | Start date of "before" time period in YYYY-MM-DD |
| before_end | End date of "before" time period in YYYY-MM-DD |
| after_start | Start date of "after" time period in YYYY-MM-DD |
| after_end | End date of "after" time period in YYYY-MM-DD |
| before_label | String to specify a label for the "before" period. Defaults to "Period 1". |
| after_label | String to specify a label for the "after" period. Defaults to "Period 2". |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |

Details

This is a general purpose function that powers all the functions in the package that produce faceted scatter plots.

Value

Returns a 'ggplot' object showing two scatter plots side by side representing the two periods.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Other Time-series: [IV_by_period\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```
# Return plot
create_period_scatter(sq_data,
                     hrvar = "LevelDesignation",
```

```

before_start = "2019-11-03",
before_end = "2019-12-31",
after_start = "2020-01-01",
after_end = "2020-01-26")

# Return a summary table
create_period_scatter(sq_data, before_end = "2019-12-31", return = "table")

```

| | |
|-------------|--|
| create_rank | <i>Rank all groups across HR attributes on a selected Workplace Analytics metric</i> |
|-------------|--|

Description

This function scans a standard Person query output for groups with high levels of a given Workplace Analytics Metric. Returns a table with all groups (across multiple HR attributes) ranked by the specified metric.

Usage

```

create_rank(
  data,
  metric,
  hrvar = extract_hr(data, exclude_constants = TRUE),
  mingroup = 5,
  return = "table",
  mode = "simple",
  plot_mode = 1
)

```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> "table" (default) "plot" See Value for more information. |
| mode | String to specify calculation mode. Must be either: |

- "simple"
 - "combine"
- plot_mode Numeric vector to determine which plot mode to return. Must be either 1 or 2, and is only used when return = "plot".
- 1: Top and bottom five groups across the data population are highlighted
 - 2: Top and bottom groups *per* organizational attribute are highlighted

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bubble plot where the x-axis represents the metric, the y-axis represents the HR attributes, and the size of the bubbles represent the size of the organizations. Note that there is no plot output if mode is set to "combine".
- "table": data frame. A summary table for the metric.

Author(s)

Carlos Morales Torrado carlos.morales@microsoft.com

Martin Chan martin.chan@microsoft.com

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```
sq_data_small <- dplyr::slice_sample(sq_data, prop = 0.1)

# Plot mode 1 - show top and bottom five groups
create_rank(
```

```

    data = sq_data_small,
    hrvar = c("FunctionType", "LevelDesignation"),
    metric = "Emails_sent",
    return = "plot",
    plot_mode = 1
  )

# Plot mode 2 - show top and bottom groups per HR variable
create_rank(
  data = sq_data_small,
  hrvar = c("FunctionType", "LevelDesignation"),
  metric = "Emails_sent",
  return = "plot",
  plot_mode = 2
)

# Return a table
create_rank(
  data = sq_data_small,
  metric = "Emails_sent",
  return = "table"
)

# Return a table - combination mode
create_rank(
  data = sq_data_small,
  metric = "Emails_sent",
  mode = "combine",
  return = "table"
)

```

create_rank_combine *Create combination pairs of HR variables and run 'create_rank()'*

Description

Create pairwise combinations of HR variables and compute an average of a specified Workplace Analytics metric.

Usage

```
create_rank_combine(data, hrvar = extract_hr(data), metric, mingroup = 5)
```

Arguments

data A Standard Person Query dataset in the form of a data frame.

| | |
|----------|--|
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| metric | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |

Details

This function is called when the mode argument in create_rank() is specified as "combine".

Value

Data frame containing the following variables:

- hrvar: placeholder column that denotes the output as "Combined".
- group: pairwise combinations of HR attributes with the HR attribute in square brackets followed by the value of the HR attribute.
- Name of the metric (as passed to metric)
- n

Examples

```
# Use a small sample for faster runtime
sq_data_small <- dplyr::slice_sample(sq_data, prop = 0.1)

create_rank_combine(
  data = sq_data_small,
  metric = "Email_hours"
)
```

| | |
|---------------|--|
| create_sankey | <i>Create a sankey chart from a two-column count table</i> |
|---------------|--|

Description

Create a 'networkD3' style sankey chart based on a long count table with two variables. The input data should have three columns, where each row is a unique group:

1. Variable 1
2. Variable 2
3. Count

Usage

```
create_sankey(data, var1, var2, count = "n")
```

Arguments

| | |
|-------|--|
| data | Data frame of the long count table. |
| var1 | String containing the name of the variable to be shown on the left. |
| var2 | String containing the name of the variable to be shown on the right. |
| count | String containing the name of the count variable. |

Value

A 'sankeyNetwork' and 'htmlwidget' object containing a two-tier sankey plot. The output can be saved locally with `htmlwidgets::saveWidget()`.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```
sq_data %>%
  dplyr::count(Organization, FunctionType) %>%
  create_sankey(var1 = "Organization", var2 = "FunctionType")
```

| | |
|----------------|--|
| create_scatter | <i>Create a Scatter plot with two selected Workplace Analytics metrics (General Purpose)</i> |
|----------------|--|

Description

Returns a scatter plot of two selected metrics, using colour to map an HR attribute. Returns a scatter plot by default, with additional options to return a summary table.

Usage

```
create_scatter(  
  data,  
  metric_x,  
  metric_y,  
  hrvar = "Organization",  
  mingroup = 5,  
  return = "plot"  
)
```

Arguments

| | |
|----------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric_x | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| metric_y | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |

Details

This is a general purpose function that powers all the functions in the package that produce scatter plots.

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Flexible: `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_hist()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `period_change()`

Examples

```
create_scatter(sq_data,
  "Internal_network_size",
  "External_network_size",
  "Organization")

create_scatter(sq_data,
  "Generated_workload_call_hours",
  "Generated_workload_email_hours",
  "Organization", mingroup = 100, return = "plot")
```

create_stacked

Horizontal stacked bar plot for any metric

Description

Creates a sum total calculation using selected metrics, where the typical use case is to create different definitions of collaboration hours. Returns a stacked bar plot by default. Additional options available to return a summary table.

Usage

```
create_stacked(
  data,
  hrvar = "Organization",
```



```

metrics = c("Meeting_hours", "Email_hours"),
mingroup = 5,
return = "plot",
stack_colours = c("#1d627e", "#34b1e2", "#b4d5dd", "#adc0cb"),
percent = FALSE,
plot_title = "Collaboration Hours",
plot_subtitle = paste("Average by", tolower(camel_clean(hrvar))),
rank = "descending",
xlim = NULL,
text_just = 0.5,
text_colour = "#FFFFFF"
)

```

Arguments

| | |
|---------------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| metrics | A character vector to specify variables to be used in calculating the "Total" value, e.g. c("Meeting_hours", "Email_hours"). The order of the variable names supplied determine the order in which they appear on the stacked plot. |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |
| stack_colours | A character vector to specify the colour codes for the stacked bar charts. |
| percent | Logical value to determine whether to show labels as percentage signs. Defaults to FALSE. |
| plot_title | An option to override plot title. |
| plot_subtitle | An option to override plot subtitle. |
| rank | String specifying how to rank the bars. Valid inputs are: <ul style="list-style-type: none"> • "descending" - ranked highest to lowest from top to bottom (default). • "ascending" - ranked lowest to highest from top to bottom. • NULL - uses the original levels of the HR attribute. |
| xlim | An option to set max value in x axis. |
| text_just | [Experimental] A numeric value controlling for the horizontal position of the text labels. Defaults to 0.5. |
| text_colour | [Experimental] String to specify colour to use for the text labels. Defaults to "#FFFFFF". |

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Flexible: `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_hist()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_tracking()`, `create_trend()`, `period_change()`

Examples

```
sq_data %>%
  create_stacked(hrvar = "LevelDesignation",
                metrics = c("Meeting_hours", "Email_hours"),
                return = "plot")
```

```
sq_data %>%
  create_stacked(hrvar = "FunctionType",
                metrics = c("Meeting_hours",
                           "Email_hours",
                           "Call_hours",
                           "Instant_Message_hours"),
                return = "plot",
                rank = "ascending")
```

```
sq_data %>%
  create_stacked(hrvar = "FunctionType",
                metrics = c("Meeting_hours",
                           "Email_hours",
                           "Call_hours",
                           "Instant_Message_hours"),
                return = "table")
```

create_tracking

Create a line chart that tracks metrics over time with a 4-week rolling average

Description**[Experimental]**

Create a two-series line chart that visualizes a set of metric over time for the selected population, with one of the series being a four-week rolling average.

Usage

```
create_tracking(
  data,
  metric,
  plot_title = us_to_space(metric),
  plot_subtitle = "Measure over time",
  percent = FALSE
)
```

Arguments

| | |
|----------------------------|---|
| <code>data</code> | A Standard Person Query dataset in the form of a data frame. |
| <code>metric</code> | Character string containing the name of the metric, e.g. "Collaboration_hours" percentage signs. Defaults to FALSE. |
| <code>plot_title</code> | An option to override plot title. |
| <code>plot_subtitle</code> | An option to override plot subtitle. |
| <code>percent</code> | Logical value to determine whether to show labels as percentage signs. Defaults to FALSE. |

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```
sq_data %>%
  create_tracking(
    metric = "Collaboration_hours",
    percent = FALSE
  )
```

 create_trend

Heat mapped horizontal bar plot over time for any metric

Description

Provides a week by week view of a selected Workplace Analytics metric. By default returns a week by week heatmap bar plot, highlighting the points in time with most activity. Additional options available to return a summary table.

Usage

```
create_trend(
  data,
  metric,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  legend_title = "Hours"
)
```

Arguments

| | |
|--------------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | Character string containing the name of the metric, e.g. "Collaboration_hours" |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |
| legend_title | String to be used as the title of the legend. Defaults to "Hours". |

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Flexible: `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_hist()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `period_change()`

Other Time-series: `IV_by_period()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `period_change()`

Examples

```
create_trend(sq_data, metric = "Collaboration_hours", hrvar = "LevelDesignation")
```

cut_hour

Convert a numeric variable for hours into categorical

Description

Supply a numeric variable, e.g. `Collaboration_hours`, and return a character vector.

Usage

```
cut_hour(metric, cuts, unit = "hours", lbound = 0, ubound = 100)
```

Arguments

| | |
|---------------------|---|
| <code>metric</code> | A numeric variable representing hours. |
| <code>cuts</code> | A numeric vector of minimum length 3 to represent the cut points required. The minimum and maximum values provided in the vector are inclusive. |
| <code>unit</code> | String to specify the unit of the labels. Defaults to "hours". |
| <code>lbound</code> | Numeric. Specifies the lower bound (inclusive) value for the minimum label. Defaults to 0. |

ubound Numeric. Specifies the upper bound (inclusive) value for the maximum label. Defaults to 100.

Details

This is used within `create_dist()` for numeric to categorical conversion.

Value

Character vector representing a converted categorical variable, appended with the label of the unit. See examples for more information.

See Also

Other Support: `camel_clean()`, `check_inputs()`, `combine_signals()`, `extract_date_range()`, `extract_hr()`, `heat_colours()`, `is_date_format()`, `maxmin()`, `p_test()`, `pairwise_count()`, `plot_WOE()`, `read_preamble()`, `rgb2hex()`, `totals_bind()`, `totals_col()`, `totals_reorder()`, `tstamp()`, `us_to_space()`, `wrap()`

Examples

```
# Direct use
cut_hour(1:30, cuts = c(15, 20, 25))

# Use on a query
cut_hour(sq_data$Collaboration_hours, cuts = c(10, 15, 20))
```

dv_data

Sample Standard Person Query dataset for Data Validation

Description

A dataset generated from a Standard Person Query from Workplace Analytics.

Usage

```
dv_data
```

Format

A data frame with 897 rows and 69 variables:

PersonId

Date

Workweek_span

Meetings_with_skip_level

Meeting_hours_with_skip_level

Generated_workload_email_hours
Generated_workload_email_recipients
Generated_workload_instant_messages_hours
Generated_workload_instant_messages_recipients
Generated_workload_call_hours
Generated_workload_call_participants
Generated_workload_calls_organized
External_network_size
Internal_network_size
Networking_outside_company
Networking_outside_organization
After_hours_meeting_hours
Open_1_hour_block
Open_2_hour_blocks
Total_focus_hours
Low_quality_meeting_hours
Total_emails_sent_during_meeting
Meetings
Meeting_hours
Conflicting_meeting_hours
Multitasking_meeting_hours
Redundant_meeting_hours_lower_level_
Redundant_meeting_hours_organizational_
Time_in_self_organized_meetings
Meeting_hours_during_working_hours
Generated_workload_meeting_attendees
Generated_workload_meeting_hours
Generated_workload_meetings_organized
Manager_coaching_hours_1_on_1
Meetings_with_manager
Meeting_hours_with_manager
Meetings_with_manager_1_on_1
Meeting_hours_with_manager_1_on_1
After_hours_email_hours
Emails_sent
Email_hours
Working_hours_email_hours

After_hours_instant_messages
Instant_messages_sent
Instant_Message_hours
Working_hours_instant_messages
After_hours_collaboration_hours
Collaboration_hours
Collaboration_hours_external
Working_hours_collaboration_hours
After_hours_in_calls
Total_calls
Call_hours
Working_hours_in_calls
Domain
FunctionType
LevelDesignation
Layer
Region
Organization
zId
attainment
TimeZone
HourlyRate
IsInternal
IsActive
HireDate
WorkingStartTimeSetInOutlook
WorkingEndTimeSetInOutlook ...

Value

data frame.

Source

<https://workplaceanalytics-demo.office.com/en-us/Home>

See Also

Other Data: [em_data](#), [g2g_data](#), [mt_data](#), [p2p_data_sim\(\)](#), [sq_data](#)

| | |
|------------|--|
| email_dist | <i>Distribution of Email Hours as a 100% stacked bar</i> |
|------------|--|

Description

Analyze Email Hours distribution. Returns a stacked bar plot by default. Additional options available to return a table with distribution elements.

Usage

```
email_dist(  
  data,  
  hrvar = "Organization",  
  mingroup = 5,  
  return = "plot",  
  cut = c(5, 10, 15)  
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "table" See Value for more information. |
| cut | A numeric vector of length three to specify the breaks for the distribution, e.g. c(10, 15, 20) |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Emails: `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`

Examples

```
# Return plot
email_dist(sq_data, hrvar = "Organization")

# Return summary table
email_dist(sq_data, hrvar = "Organization", return = "table")

# Return result with a custom specified breaks
email_dist(sq_data, hrvar = "LevelDesignation", cut = c(4, 7, 9))
```

email_fizz

Distribution of Email Hours (Fizzy Drink plot)

Description

Analyze weekly email hours distribution, and returns a 'fizzy' scatter plot by default. Additional options available to return a table with distribution elements.

Usage

```
email_fizz(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

`data` A Standard Person Query dataset in the form of a data frame.

`hrvar` String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes).

| | |
|----------|--|
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A jittered scatter plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Emails: [email_dist\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#)

Examples

```
# Return plot
email_fizz(sq_data, hrvar = "Organization", return = "plot")

# Return summary table
email_fizz(sq_data, hrvar = "Organization", return = "table")
```

| | |
|------------|--------------------------------------|
| email_line | <i>Email Time Trend - Line Chart</i> |
|------------|--------------------------------------|

Description

Provides a week by week view of email time, visualised as line charts. By default returns a line chart for email hours, with a separate panel per value in the HR attribute. Additional options available to return a summary table.

Usage

```
email_line(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A faceted line plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#),

one2one_fizz(), one2one_freq(), one2one_line(), one2one_rank(), one2one_sum(), one2one_trend(),
 period_change(), workloads_dist(), workloads_fizz(), workloads_line(), workloads_rank(),
 workloads_summary(), workloads_trend(), workpatterns_area(), workpatterns_rank()
 Other Emails: email_dist(), email_fizz(), email_rank(), email_summary(), email_trend()

Examples

```
# Return a line plot
email_line(sq_data, hrvar = "LevelDesignation")

# Return summary table
email_line(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|------------|----------------------------|
| email_rank | <i>Email Hours Ranking</i> |
|------------|----------------------------|

Description

This function scans a standard query output for groups with high levels of 'Weekly Email Collaboration'. Returns a table with a all of groups (across multiple HR attributes) ranked by hours of digital collaboration.

Usage

```
email_rank(
  data,
  hrvar = extract_hr(data),
  mingroup = 5,
  mode = "simple",
  plot_mode = 1,
  return = "table"
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| mode | String to specify calculation mode. Must be either: <ul style="list-style-type: none"> • "simple" • "combine" |

| | |
|-----------|---|
| plot_mode | Numeric vector to determine which plot mode to return. Must be either 1 or 2, and is only used when return = "plot". <ul style="list-style-type: none"> • 1: Top and bottom five groups across the data population are highlighted • 2: Top and bottom groups <i>per</i> organizational attribute are highlighted |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "table" (default) • "plot" See Value for more information. |

Details

Uses the metric Email_hours. See create_rank() for applying the same analysis to a different metric.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bubble plot where the x-axis represents the metric, the y-axis represents the HR attributes, and the size of the bubbles represent the size of the organizations. Note that there is no plot output if mode is set to "combine".
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Emails: [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#)

Examples

```
# Return rank table
email_rank(
  data = sq_data,
  return = "table"
)
```

```
# Return plot
email_rank(
  data = sq_data,
  return = "plot"
)
```

| | |
|---------------|----------------------|
| email_summary | <i>Email Summary</i> |
|---------------|----------------------|

Description

Provides an overview analysis of weekly email hours. Returns a bar plot showing average weekly email hours by default. Additional options available to return a summary table.

Usage

```
email_summary(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

```
email_sum(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Emails: `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_trend()`

Examples

```
# Return a ggplot bar chart
email_summary(sq_data, hrvar = "LevelDesignation")

# Return a summary table
email_summary(sq_data, hrvar = "LevelDesignation", return = "table")
```

email_trend

Email Hours Time Trend

Description

Provides a week by week view of email time. By default returns a week by week heatmap, highlighting the points in time with most activity. Additional options available to return a summary table.

Usage

```
email_trend(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |

return Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table".

Details

Uses the metric Email_hours.

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Emails: [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#)

em_data

Sample Hourly Collaboration data with Email and IMs

Description

A sample dataset generated from an Hourly Collaboration query from the Workplace Analytics demo tenant. The data is grouped by week and contains columns for both IMs sent and Emails sent.

Usage

em_data

Format

A data frame with 41567 rows and 56 variables:

PersonId

Date

IMs_sent_23_24

IMs_sent_22_23

IMs_sent_21_22

IMs_sent_20_21

IMs_sent_19_20

IMs_sent_18_19

IMs_sent_17_18

IMs_sent_16_17

IMs_sent_15_16

IMs_sent_14_15

IMs_sent_13_14

IMs_sent_12_13

IMs_sent_11_12

IMs_sent_10_11

IMs_sent_09_10

IMs_sent_08_09

IMs_sent_07_08

IMs_sent_06_07

IMs_sent_05_06

IMs_sent_04_05

IMs_sent_03_04

IMs_sent_02_03

IMs_sent_01_02

IMs_sent_00_01

Emails_sent_23_24

Emails_sent_22_23

Emails_sent_21_22

Emails_sent_20_21

Emails_sent_19_20

Emails_sent_18_19

Emails_sent_17_18

Emails_sent_16_17

Emails_sent_15_16

Emails_sent_14_15
Emails_sent_13_14
Emails_sent_12_13
Emails_sent_11_12
Emails_sent_10_11
Emails_sent_09_10
Emails_sent_08_09
Emails_sent_07_08
Emails_sent_06_07
Emails_sent_05_06
Emails_sent_04_05
Emails_sent_03_04
Emails_sent_02_03
Emails_sent_01_02
Emails_sent_00_01
Domain
LevelDesignation
Layer
Organization
IsInternal
IsActive ...

Value

data frame.

Source

<https://workplaceanalytics-demo.office.com/en-us/Home>

See Also

Other Data: [dv_data](#), [g2g_data](#), [mt_data](#), [p2p_data_sim\(\)](#), [sq_data](#)

 export

Export 'wpa' outputs to CSV, clipboard, or save as images

Description

A general use function to export 'wpa' outputs to CSV, clipboard, or save as images. By default, `export()` copies a data frame to the clipboard. If the input is a 'ggplot' object, the default behaviour is to export a PNG.

Usage

```
export(
  x,
  method = "clipboard",
  path = "wpa export",
  timestamp = TRUE,
  width = 12,
  height = 9
)
```

Arguments

| | |
|------------------------|--|
| <code>x</code> | Data frame or 'ggplot' object to be passed through. |
| <code>method</code> | Character string specifying the method of export. Valid inputs include: <ul style="list-style-type: none"> • "clipboard" (default if input is data frame) • "csv" • "png" (default if input is 'ggplot' object) • "svg" • "jpeg" • "pdf" |
| <code>path</code> | If exporting a file, enter the path and the desired file name, <i>excluding the file extension</i> . For example, "Analysis/SQ Overview". |
| <code>timestamp</code> | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |
| <code>width</code> | Width of the plot |
| <code>height</code> | Height of the plot |

Value

A different output is returned depending on the value passed to the `method` argument:

- "clipboard": no return - data frame is saved to clipboard.
- "csv": CSV file containing data frame is saved to specified path.
- "png": PNG file containing 'ggplot' object is saved to specified path.

- "svg": SVG file containing 'ggplot' object is saved to specified path.
- "jpeg": JPEG file containing 'ggplot' object is saved to specified path.
- "pdf": PDF file containing 'ggplot' object is saved to specified path.

Author(s)

Martin Chan martin.chan@microsoft.com

See Also

Other Import and Export: [copy_df\(\)](#), [create_dt\(\)](#), [import_to_fst\(\)](#), [import_wpa\(\)](#), [standardise_pq\(\)](#)

external_network_plot *Plot External Network Breadth and Size as a scatter plot*

Description

Plot the external network metrics for a HR variable as a scatter plot, showing 'External Network Breadth' as the vertical axis and 'External Network Size' as the horizontal axis.

Usage

```
external_network_plot(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  bubble_size = c(1, 8)
)
```

Arguments

| | |
|-------------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: - "plot" - "table" |
| bubble_size | A numeric vector of length two to specify the size range of the bubbles |

Details

Uses the metrics `External_network_size` and `Networking_outside_company`.

Value

'ggplot' object showing a bubble plot with external network size as the x-axis and external network breadth as the y-axis. The size of the bubbles represent the number of unique employees in each group.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Network: [g2g_data](#), [internal_network_plot\(\)](#), [network_describe\(\)](#), [network_g2g\(\)](#), [network_leiden\(\)](#), [network_louvain\(\)](#), [network_p2p\(\)](#), [p2p_data_sim\(\)](#)

Examples

```
# Return plot
sq_data %>% external_network_plot(return = "plot")
```

| | |
|--------------------|----------------------------|
| extract_date_range | <i>Extract date period</i> |
|--------------------|----------------------------|

Description

Return a data frame with the start and end date of the query data by default. There are options to return a descriptive string, which is used in the caption of plots in this package.

Usage

```
extract_date_range(data, return = "table")
```

Arguments

| | |
|--------|--|
| data | Data frame containing a query to pass through. The data frame must contain a Date column. Accepts a Person query or a Meeting query. |
| return | String specifying what output to return. Returns a table by default ("table"), but allows returning a descriptive string ("text"). |

Value

A different output is returned depending on the value passed to the return argument:

- "table": data frame. A summary table containing the start and end date for the dataset.
- "text": string. Contains a descriptive string on the start and end date for the dataset.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

| | |
|------------|---------------------------------------|
| extract_hr | <i>Extract HR attribute variables</i> |
|------------|---------------------------------------|

Description

This function uses a combination of variable class, number of unique values, and regular expression matching to extract HR / organisational attributes from a data frame.

Usage

```
extract_hr(data, max_unique = 50, exclude_constants = TRUE, return = "names")
```

Arguments

| | |
|-------------------|--|
| data | A data frame to be passed through. |
| max_unique | A numeric value representing the maximum number of unique values to accept for an HR attribute. Defaults to 50. |
| exclude_constants | Logical value to specify whether single-value HR attributes are to be excluded. Defaults to TRUE. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "names" • "vars" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "names": character vector identifying all the names of HR variables present in the data.
- "vars": data frame containing all the columns of HR variables present in the data.

See Also

Other Support: `camel_clean()`, `check_inputs()`, `combine_signals()`, `cut_hour()`, `extract_date_range()`, `heat_colours()`, `is_date_format()`, `maxmin()`, `p_test()`, `pairwise_count()`, `plot_WOE()`, `read_preamble()`, `rgb2hex()`, `totals_bind()`, `totals_col()`, `totals_reorder()`, `tstamp()`, `us_to_space()`, `wrap()`

Other Data Validation: `check_query()`, `flag_ch_ratio()`, `flag_em_ratio()`, `flag_extreme()`, `flag_outlooktime()`, `hr_trend()`, `hrvar_count_all()`, `hrvar_count()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts_wp()`, `identify_shifts()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate_report()`, `subject_validate()`, `track_HR_change()`, `validation_report()`

Examples

```
sq_data %>% extract_hr(return = "names")
```

```
sq_data %>% extract_hr(return = "vars")
```

| | |
|---------------|---|
| flag_ch_ratio | <i>Flag unusual high collaboration hours to after-hours collaboration hours ratio</i> |
|---------------|---|

Description

This function flags persons who have an unusual ratio of collaboration hours to after-hours collaboration hours. Returns a character string by default.

Usage

```
flag_ch_ratio(data, threshold = c(1, 30), return = "message")
```

Arguments

| | |
|-----------|--|
| data | A data frame containing a Person Query. |
| threshold | Numeric value specifying the threshold for flagging. Defaults to 30. |
| return | String to specify what to return. Options include: <ul style="list-style-type: none"> • "message" • "text" • "data" |

Value

A different output is returned depending on the value passed to the return argument:

- "message": message in the console containing diagnostic summary
- "text": string containing diagnostic summary
- "data": data frame. Person-level data with flags on unusually high or low ratios

Metrics used

The metric Collaboration_hours is used in the calculations. Please ensure that your query contains a metric with the exact same name.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
flag_ch_ratio(sq_data)
```

```
data.frame(PersonId = c("Alice", "Bob"),
           Collaboration_hours = c(30, 0.5),
           After_hours_collaboration_hours = c(0.5, 30)) %>%
  flag_ch_ratio()
```

flag_em_ratio

Flag Persons with unusually high Email Hours to Emails Sent ratio

Description

This function flags persons who have an unusual ratio of email hours to emails sent. If the ratio between Email Hours and Emails Sent is greater than the threshold, then observations tied to a PersonId is flagged as unusual.

Usage

```
flag_em_ratio(data, threshold = 1, return = "text")
```

Arguments

| | |
|-----------|---|
| data | A data frame containing a Person Query. |
| threshold | Numeric value specifying the threshold for flagging. Defaults to 1. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "text" • "data" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "text": string. A diagnostic message.
- "data": data frame. Person-level data with those flagged with unusual ratios.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
flag_em_ratio(sq_data)
```

flag_extreme

Warn if a certain metric exceeds an arbitrary threshold

Description

This is used as part of data validation to check if there are extreme values in the dataset.

Usage

```
flag_extreme(data, metric, person = TRUE, threshold, return = "message")
```

Arguments

| | |
|-----------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | A character string specifying the metric to test. |
| person | A logical value to specify whether to calculate person-averages. Defaults to TRUE (person-averages calculated). |
| threshold | Numeric value specifying the threshold for flagging. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "text" • "message" • "table" |

See Value for more information.

Value

A different output is returned depending on the value passed to the return argument:

- "text": string. A diagnostic message.
- "message": message on console. A diagnostic message.
- "table": data frame. A person-level table with PersonId and the extreme values of the selected metric.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
# The threshold values are intentionally set low to trigger messages.
flag_extreme(sq_data, "Email_hours", threshold = 15)

# Return a summary table
flag_extreme(sq_data, "Email_hours", threshold = 15, return = "table")

# Person-week level
flag_extreme(sq_data, "Email_hours", person = FALSE, threshold = 15)
```

| | |
|------------------|---|
| flag_outlooktime | <i>Flag unusual outlook time settings for work day start and end time</i> |
|------------------|---|

Description

This function flags unusual outlook calendar settings for start and end time of work day.

Usage

```
flag_outlooktime(data, threshold = c(4, 15), return = "message")
```

Arguments

| | |
|-----------|---|
| data | A data frame containing a Person Query. |
| threshold | A numeric vector of length two, specifying the hour threshold for flagging. Defaults to c(4, 15). |
| return | String specifying what to return. This must be one of the following strings: |

- "text" (default)
- "message"
- "data"

Value

A different output is returned depending on the value passed to the return argument:

- "text": string. A diagnostic message.
- "message": message on console. A diagnostic message.
- "data": data frame. Data where flag is present.

See Value for more information.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
# Demo with `dv_data`
flag_outlooktime(dv_data)

# Example where Outlook Start and End times are imputed
spq_df <- sq_data

spq_df$WorkingStartTimeSetInOutlook <- "6:30"

spq_df$WorkingEndTimeSetInOutlook <- "23:30"

# Return a message
flag_outlooktime(spq_df, threshold = c(5, 13))

# Return data
flag_outlooktime(spq_df, threshold = c(5, 13), return = "data")
```

Description

[Experimental]

Pass an Hourly Collaboration query and compute a Flexibility Index for the entire population. The Flexibility Index is a quantitative measure of the freedom for employees to work at a time of their choice.

Usage

```
flex_index(
  data,
  hrvar = NULL,
  signals = c("email", "IM"),
  active_threshold = 0,
  start_hour = "0900",
  end_hour = "1700",
  return = "plot",
  plot_method = "common"
)
```

Arguments

| | |
|------------------|--|
| data | Hourly Collaboration query to be passed through as data frame. |
| hrvar | A string specifying the HR attribute to cut the data by. Defaults to NULL. This only affects the function when "table" is returned. |
| signals | Character vector to specify which collaboration metrics to use: You may use "email" for emails only, "IM" for Teams messages only, or a combination of the two c("email", "IM") (default). |
| active_threshold | A numeric value specifying the minimum number of signals to be greater than in order to qualify as <i>active</i> . Defaults to 0. |
| start_hour | A character vector specifying starting hours, e.g. "0900" |
| end_hour | A character vector specifying end hours, e.g. "1700" |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "data" • "table" See Value for more information. |
| plot_method | Character string for determining which plot to return. <ul style="list-style-type: none"> • "sample" plots a sample of ten working pattern • "common" plots the ten most common working patterns • "time" plots the Flexibility Index for the group over time |

Details

The **Flexibility Index** is a metric that has been developed to quantify and measure flexibility using behavioural data from Workplace Analytics. Flexibility here refers to the freedom of employees to adopt a working arrangement of their own choice, and more specifically refers to **time** flexibility (*whenever I want*) as opposed to **geographical** flexibility (*wherever I want*).

The **Flexibility Index** is a score between 0 and 1, and is calculated based on three component measures:

- **ChangeHours**: this represents the freedom to define work start and end time. Teams that embrace flexibility allow members to start and end their workday at different times.
- **TakeBreaks**: this represents the freedom define one's own schedule. In teams that embrace flexibility, some members will choose to organize / split their day in different ways (e.g. take a long lunch-break, disconnect in the afternoon and reconnect in the evening, etc.).
- **ControlHours**: this represents the freedom to switch off. Members who choose alternative arrangements should be able to maintain a workload that is broadly equivalent to those that follow standard arrangements.

The **Flexibility Index** returns with one single score for each person-week, plus the **three** sub-component binary variables (**TakeBreaks**, **ChangeHours**, **ControlHours**). At the person-week level, each score can only have the values 0, 0.33, 0.66, and 1. The Flexibility Index should only be interpreted as a **group** of person-weeks, e.g. the average Flexibility Index of a team of 6 over time, where the possible values would range from 0 to 1.

Value

A different output is returned depending on the value passed to the return argument:

- `"plot"`: 'ggplot' object. A random of ten working patterns are displayed, with diagnostic data and the Flexibility Index shown on the plot.
- `"data"`: data frame. The original input data appended with the Flexibility Index and the component scores. Can be used with `plot_flex_index()` to recreate visuals found in `flex_index()`.
- `"table"`: data frame. A summary table for the metric.

Context

The central feature of flexible working arrangements is that it is the employee rather the employer who chooses the working arrangement. *Observed flexibility* serves as a proxy to assess whether a flexible working arrangement are in place. The Flexibility Index is an attempt to create such a proxy for quantifying and measuring flexibility, using behavioural data from Workplace Analytics.

Recurring disconnection time

The key component of **TakeBreaks** in the Flexibility Index is best interpreted as 'recurring disconnection time'. This denotes an hourly block where there is consistently no activity occurring throughout the week. Note that this applies a stricter criterion compared to the common definition of a break, which is simply a time interval where no active work is being done, and thus the more specific terminology 'recurring disconnection time' is preferred.

Returning the raw data

The raw data containing the computed Flexibility Index can be returned with the following:

```
em_data %>%
  flex_index(return = "data")
```

See Also

Other Working Patterns: [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [plot_flex_index\(\)](#), [workpatterns_area\(\)](#), [workpatterns_classify_bw\(\)](#), [workpatterns_classify_pav\(\)](#), [workpatterns_classify\(\)](#), [workpatterns_hclust\(\)](#), [workpatterns_rank\(\)](#), [workpatterns_report\(\)](#)

Examples

```
# Examples of how to test the plotting options individually
# Sample of 10 work patterns
em_data %>%
  flex_index(return = "plot", plot_method = "sample")

# 10 most common work patterns
em_data %>%
  flex_index(return = "plot", plot_method = "common")

# Plot Flexibility Index over time
em_data %>%
  flex_index(return = "plot", plot_method = "time")

# Return a summary table with the computed Flexibility Index
em_data %>%
  flex_index(hrvar = "Organization", return = "table")
```

g2g_data

Sample Group-to-Group dataset

Description

A demo dataset generated from a Group-to-Group Query from Workplace Analytics.

Usage

```
g2g_data
```

Format

A data frame with 3517 rows and 7 variables:

TimeInvestors_Organization

Collaborators_Organization

Date
Meetings
Meeting_hours
Email_hours
Collaboration_hours ...

Value

data frame.

Source

<https://workplaceanalytics-demo.office.com/en-us/Home>

See Also

Other Data: [dv_data](#), [em_data](#), [mt_data](#), [p2p_data_sim\(\)](#), [sq_data](#)

Other Network: [external_network_plot\(\)](#), [internal_network_plot\(\)](#), [network_describe\(\)](#), [network_g2g\(\)](#), [network_leiden\(\)](#), [network_louvain\(\)](#), [network_p2p\(\)](#), [p2p_data_sim\(\)](#)

| | |
|-----------------|--|
| generate_report | <i>Generate HTML report with list inputs</i> |
|-----------------|--|

Description

This is a support function using a list-pmap workflow to create a HTML document, using RMarkdown as the engine.

Usage

```
generate_report(  
  title = "My minimal HTML generator",  
  filename = "minimal_html",  
  outputs = output_list,  
  titles,  
  subheaders,  
  echos,  
  levels,  
  theme = "united",  
  preamble = ""  
)
```


Arguments

| | |
|------------|--|
| title | Character string to specify the title of the chunk. |
| filename | File name to be used in the exported HTML. |
| outputs | A list of outputs to be added to the HTML report. Note that outputs, titles, echos, and levels must have the same length |
| titles | A list/vector of character strings to specify the title of the chunks. |
| subheaders | A list/vector of character strings to specify the subheaders for each chunk. |
| echos | A list/vector of logical values to specify whether to display code. |
| levels | A list/vector of numeric value to specify the header level of the chunk. |
| theme | Character vector to specify theme to be used for the report. E.g. "united", "default". |
| preamble | A preamble to appear at the beginning of the report, passed as a text string. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

Creating a custom report

Below is an example on how to set up a custom report.

The first step is to define the content that will go into a report and assign the outputs to a list.

```
# Step 1: Define Content
output_list <-
  list(sq_data %>% workloads_summary(return = "plot"),
       sq_data %>% workloads_summary(return = "table")) %>%
  purrr::map_if(is.data.frame, create_dt)
```

The next step is to add a list of titles for each of the objects on the list:

```
# Step 2: Add Corresponding Titles
title_list <- c("Workloads Summary - Plot", "Workloads Summary - Table")
n_title <- length(title_list)
```

The final step is to run `generate_report()`. This can all be wrapped within a function such that the function can be used to generate a HTML report.

```
# Step 3: Generate Report
generate_report(title = "My First Report",
               filename = "My First Report",
               outputs = output_list,
               titles = title_list,
               subheaders = rep("", n_title),
               echos = rep(FALSE, n_title)
```

Author(s)

Martin Chan martin.chan@microsoft.com

See Also

Other Reports: [IV_report\(\)](#), [capacity_report\(\)](#), [coaching_report\(\)](#), [collaboration_report\(\)](#), [connectivity_report\(\)](#), [meeting_tm_report\(\)](#), [read_preamble\(\)](#), [subject_validate_report\(\)](#), [validation_report\(\)](#), [workpatterns_report\(\)](#)

| | |
|------------------|---|
| generate_report2 | <i>Generate HTML report based on existing RMarkdown documents</i> |
|------------------|---|

Description

This is a support function that accepts parameters and creates a HTML document based on an RMarkdown template. This is an alternative to `generate_report()` which instead creates an RMarkdown document from scratch using individual code chunks.

Usage

```
generate_report2(
  output_format = rmarkdown::html_document(toc = TRUE, toc_depth = 6, theme = "cosmo"),
  output_file = "report.html",
  output_dir = getwd(),
  report_title = "Report",
  rmd_dir = system.file("rmd_template/minimal.rmd", package = "wpa"),
  ...
)
```

Arguments

| | |
|----------------------------|--|
| <code>output_format</code> | output format in <code>rmarkdown::render()</code> . Default is <code>rmarkdown::html_document(toc = TRUE, toc_depth = 6, theme = "cosmo")</code> . |
| <code>output_file</code> | output file name in <code>rmarkdown::render()</code> . Default is <code>"report.html"</code> . |
| <code>output_dir</code> | output directory for report in <code>rmarkdown::render()</code> . Default is user's current directory. |
| <code>report_title</code> | report title. Default is <code>"Report"</code> . |
| <code>rmd_dir</code> | string specifying the path to the directory containing the RMarkdown template files. |
| <code>...</code> | other arguments to be passed to <code>params</code> . For instance, pass <code>hrvar</code> if the RMarkdown document requires a <code>'hrvar'</code> parameter. |

Note

The implementation of this function was inspired by the 'DataExplorer' package by boxuancui, with credits due to the original author.

| | |
|--------------|--|
| heat_colours | <i>Generate a vector of n contiguous colours, as a red-yellow-green palette.</i> |
|--------------|--|

Description

Takes a numeric value `n` and returns a character vector of colour HEX codes corresponding to the heat map palette.

Usage

```
heat_colours(n, alpha, rev = FALSE)
```

```
heat_colors(n, alpha, rev = FALSE)
```

Arguments

| | |
|--------------------|---|
| <code>n</code> | the number of colors (≥ 1) to be in the palette. |
| <code>alpha</code> | an alpha-transparency level in the range of 0 to 1 (0 means transparent and 1 means opaque) |
| <code>rev</code> | logical indicating whether the ordering of the colors should be reversed. |

Value

A character vector containing the HEX codes and the same length as `n` is returned.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
barplot(rep(10, 50), col = heat_colours(n = 50), border = NA)
```

```
barplot(rep(10, 50), col = heat_colours(n = 50, alpha = 0.5, rev = TRUE),  
border = NA)
```

hrvar_count

*Create a count of distinct people in a specified HR variable***Description**

This function enables you to create a count of the distinct people by the specified HR attribute. The default behaviour is to return a bar chart as typically seen in 'Analysis Scope'.

Usage

```
hrvar_count(data, hrvar = "Organization", return = "plot")
```

```
analysis_scope(data, hrvar = "Organization", return = "plot")
```

Arguments

| | |
|--------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation". If a vector with more than one value is provided, the HR attributes are automatically concatenated. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object containing a bar plot.
- "table": data frame containing a count table.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#),

```
period_change(),workloads_dist(),workloads_fizz(),workloads_line(),workloads_rank(),
workloads_summary(),workloads_trend(),workpatterns_area(),workpatterns_rank()
```

```
Other Data Validation: check_query(), extract_hr(), flag_ch_ratio(), flag_em_ratio(),
flag_extreme(), flag_outlooktime(), hr_trend(), hrvar_count_all(), hrvar_trend(), identify_churn(),
identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(),
identify_privacythreshold(), identify_query(), identify_shifts_wp(), identify_shifts(),
identify_tenure(), remove_outliers(), standardise_pq(), subject_validate_report(),
subject_validate(), track_HR_change(), validation_report()
```

Examples

```
# Return a bar plot
hrvar_count(sq_data, hrvar = "LevelDesignation")

# Return a summary table
hrvar_count(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|-----------------|---|
| hrvar_count_all | <i>Create count of distinct fields and percentage of employees with missing values for all HR variables</i> |
|-----------------|---|

Description

[Experimental]

This function enables you to create a summary table to validate organizational data. This table will provide a summary of the data found in the Workplace Analytics Sources page. This function will return a summary table with the count of distinct fields per HR attribute and the percentage of employees with missing values for that attribute. See `hrvar_count()` function for more detail on the specific HR attribute of interest.

Usage

```
hrvar_count_all(
  data,
  n_var = 50,
  return = "message",
  threshold = 100,
  maxna = 20
)
```

Arguments

| | |
|--------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| n_var | number of HR variables to include in report as rows. Default is set to 50 HR variables. |
| return | String to specify what to return |

| | |
|-----------|--|
| threshold | The max number of unique values allowed for any attribute. Default is 100. |
| maxna | The max percentage of NAs allowable for any column. Default is 20. |

Value

Returns an error message by default, where 'text' is passed in return.

- 'table': data frame. A summary table listing the number of distinct fields and percentage of missing values for the specified number of HR attributes will be returned.
- 'message': outputs a message indicating which values are beyond the specified thresholds.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
# Return a summary table of all HR attributes
hrvar_count_all(sq_data, return = "table")
```

| | |
|-------------|--|
| hrvar_trend | <i>Track count of distinct people over time in a specified HR variable</i> |
|-------------|--|

Description

This function provides a week by week view of the count of the distinct people by the specified HR attribute. The default behaviour is to return a week by week heatmap bar plot.

Usage

```
hrvar_trend(data, hrvar = "Organization", return = "plot")
```

Arguments

| | |
|--------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation". If a vector with more than one value is provided, the HR attributes are automatically concatenated. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object containing a bar plot.
- "table": data frame containing a count table.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Data Validation: `check_query()`, `extract_hr()`, `flag_ch_ratio()`, `flag_em_ratio()`, `flag_extreme()`, `flag_outlooktime()`, `hr_trend()`, `hrvar_count_all()`, `hrvar_count()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts_wp()`, `identify_shifts()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate_report()`, `subject_validate()`, `track_HR_change()`, `validation_report()`

Examples

```
# Return a bar plot
hrvar_trend(sq_data, hrvar = "LevelDesignation")

# Return a summary table
hrvar_trend(sq_data, hrvar = "LevelDesignation", return = "table")
```

hr_trend

Employee count over time

Description

Returns a line chart showing the change in employee count over time. Part of a data validation process to check for unusual license growth / declines over time.

Usage

```
hr_trend(data, return = "plot")
```

Arguments

| | |
|--------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. A line plot showing employee count over time.
- "table": data frame containing a summary table.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
# Return plot
hr_trend(dv_data)

# Return summary table
hr_trend(dv_data, return = "table")
```

| | |
|----------------|---|
| identify_churn | <i>Identify employees who have churned from the dataset</i> |
|----------------|---|

Description

This function identifies and counts the number of employees who have churned from the dataset by measuring whether an employee who is present in the first n ($n1$) weeks of the data is present in the last n ($n2$) weeks of the data.

Usage

```
identify_churn(data, n1 = 6, n2 = 6, return = "message", flip = FALSE)
```

Arguments

| | |
|--------|---|
| data | A Person Query as a data frame. Must contain a PersonId. |
| n1 | A numeric value specifying the number of weeks at the beginning of the period that defines the measured employee set. Defaults to 6. |
| n2 | A numeric value specifying the number of weeks at the end of the period to calculate whether employees have churned from the data. Defaults to 6. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "message" (default) • "text" • "data" <p>See Value for more information.</p> |
| flip | Logical, defaults to FALSE. This determines whether to reverse the logic of identifying the non-overlapping set. If set to TRUE, this effectively identifies new-joiners, or those who were not present in the first n weeks of the data but were present in the final n weeks. |

Details

An additional use case of this function is the ability to identify "new-joiners" by using the argument `flip`.

If an employee is present in the first n weeks of the data but not present in the last n weeks of the data, the function considers the employee as churned. As the measurement period is defined by the number of weeks from the start and the end of the passed data frame, you may consider filtering the dates accordingly before running this function.

Another assumption that is in place is that any employee whose PersonId is not available in the data has churned. Note that there may be other reasons why an employee's PersonId may not be present, e.g. maternity/paternity leave, Workplace Analytics license has been removed, shift to a low-collaboration role (to the extent that he/she becomes inactive).

Value

A different output is returned depending on the value passed to the return argument:

- "message": Message on console. A diagnostic message.
- "text": String. A diagnostic message.
- "data": Character vector containing the the PersonId of employees who have been identified as churned.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
sq_data %>% identify_churn(n1 = 3, n2 = 3, return = "message")
```

identify_holidayweeks *Identify Holiday Weeks based on outliers*

Description

This function scans a standard query output for weeks where collaboration hours is far outside the mean. Returns a list of weeks that appear to be holiday weeks and optionally an edited dataframe with outliers removed. By default, missing values are excluded.

As best practice, run this function prior to any analysis to remove atypical collaboration weeks from your dataset.

Usage

```
identify_holidayweeks(data, sd = 1, return = "message")
```

Arguments

| | |
|--------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| sd | The standard deviation below the mean for collaboration hours that should define an outlier week. Enter a positive number. Default is 1 standard deviation. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "message" (default) • "data" |

- "data_cleaned"
- "data_dirty"
- "plot"

See Value for more information.

Value

A different output is returned depending on the value passed to the return argument:

- "message": message on console. a message is printed identifying holiday weeks.
- "data": data frame. A dataset with outlier weeks flagged in a new column is returned as a dataframe.
- "data_cleaned": data frame. A dataset with outlier weeks removed is returned.
- "data_dirty": data frame. A dataset with only outlier weeks is returned.
- "plot": ggplot object. A line plot of Collaboration Hours with holiday weeks highlighted.

Metrics used

The metric `Collaboration_hours` is used in the calculations. Please ensure that your query contains a metric with the exact same name.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
# Return a message by default
identify_holidayweeks(sq_data)

# Return plot
identify_holidayweeks(sq_data, return = "plot")
```

 identify_inactiveweeks

Identify Inactive Weeks

Description

This function scans a standard query output for weeks where collaboration hours is far outside the mean for any individual person in the dataset. Returns a list of weeks that appear to be inactive weeks and optionally an edited dataframe with outliers removed.

As best practice, run this function prior to any analysis to remove atypical collaboration weeks from your dataset.

Usage

```
identify_inactiveweeks(data, sd = 2, return = "text")
```

Arguments

| | |
|--------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| sd | The standard deviation below the mean for collaboration hours that should define an outlier week. Enter a positive number. Default is 1 standard deviation. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "text" • "data_cleaned" • "data_dirty" See Value for more information. |

Value

Returns an error message by default, where 'text' is returned. When 'data_cleaned' is passed, a dataset with outlier weeks removed is returned as a dataframe. When 'data_dirty' is passed, a dataset with outlier weeks is returned as a dataframe.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

| | |
|--------------|---|
| identify_nkw | <i>Identify Non-Knowledge workers in a Person Query using Collaboration Hours</i> |
|--------------|---|

Description

This function scans a standard query output to identify employees with consistently low collaboration signals. Returns the % of non-knowledge workers identified by Organization, and optionally an edited data frame with non-knowledge workers removed, or the full data frame with the kw/nkw flag added.

Usage

```
identify_nkw(data, collab_threshold = 5, return = "data_summary")
```

Arguments

| | |
|------------------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| collab_threshold | Positive numeric value representing the collaboration hours threshold that should be exceeded as an average for the entire analysis period for the employee to be categorized as a knowledge worker ("kw"). Default is set to 5 collaboration hours. Any versions after v1.4.3, this uses a "greater than or equal to" logic (\geq), in which case persons with exactly 5 collaboration hours will pass. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "text" • "data_with_flag" • "data_clean" • "data_summary" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "text": string. Returns a diagnostic message.
- "data_with_flag": data frame. Original input data with an additional column containing the kw/nkw flag.
- "data_clean": data frame. Data frame with non-knowledge workers excluded.
- "data_summary": data frame. A summary table by organization listing the number and % of non-knowledge workers.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

| | |
|------------------|--|
| identify_outlier | <i>Identify metric outliers over a date interval</i> |
|------------------|--|

Description

This function takes in a selected metric and uses z-score (number of standard deviations) to identify outliers across time. There are applications in this for identifying weeks with abnormally low collaboration activity, e.g. holidays. Time as a grouping variable can be overridden with the `group_var` argument.

Usage

```
identify_outlier(data, group_var = "Date", metric = "Collaboration_hours")
```

Arguments

| | |
|------------------------|--|
| <code>data</code> | A Standard Person Query dataset in the form of a data frame. |
| <code>group_var</code> | A string with the name of the grouping variable. Defaults to Date. |
| <code>metric</code> | Character string containing the name of the metric, e.g. "Collaboration_hours" |

Value

Returns a data frame with Date (if grouping variable is not set), the metric, and the corresponding z-score.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
identify_outlier(sq_data, metric = "Collaboration_hours")
```

```
identify_privacythreshold
```

Identify groups under privacy threshold

Description

This function scans a standard query output for groups with of employees under the privacy threshold. The method consists in reviewing each individual HR attribute, and count the distinct people within each group.

Usage

```
identify_privacythreshold(  
  data,  
  hrvar = extract_hr(data),  
  mingroup = 5,  
  return = "table"  
)
```

Arguments

| | |
|----------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | A list of HR Variables to consider in the scan. Defaults to all HR attributes identified. |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "table"• "text" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "table": data frame. A summary table of groups that fall below the privacy threshold.
- "text": string. A diagnostic message.

Returns a ggplot object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
# Return a summary table
dv_data %>% identify_privacythreshold(return = "table")

# Return a diagnostic message
dv_data %>% identify_privacythreshold(return = "text")
```

| | |
|----------------|---|
| identify_query | <i>Identify the query type of the passed data frame</i> |
|----------------|---|

Description

Pass a Workplace Analytics dataset and return the identified query type as a string. This function uses variable name string matching to 'guess' the query type of the data frame.

Usage

```
identify_query(data, threshold = 2)
```

Arguments

| | |
|-----------|---|
| data | A Workplace Analytics dataset in the form of a data frame. If the data is not identified as a Workplace Analytics dataset, the function will return an error. |
| threshold | Debugging use only. Increase to raise the 'strictness' of the guessing algorithm. Defaults to 2. |

Value

String. A diagnostic message is returned.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
identify_query(sq_data) # Standard query
identify_query(mt_data) # Meeting query
identify_query(em_data) # Hourly collaboration query
## Not run:
identify_query(iris) # Will return an error
identify_query(mtcars) # Will return an error

## End(Not run)
```

| | |
|-----------------|---|
| identify_shifts | <i>Identify shifts based on outlook time settings for work day start and end time</i> |
|-----------------|---|

Description

This function uses outlook calendar settings for start and end time of work day to identify work shifts. The relevant variables are WorkingStartTimeSetInOutlook and WorkingEndTimeSetInOutlook.

Usage

```
identify_shifts(data, return = "plot")
```

Arguments

| | |
|--------|--|
| data | A data frame containing data from the Hourly Collaboration query. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "table"• "data" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. A bar plot for the weekly count of shifts.
- "table": data frame. A summary table for the count of shifts.
- "data": data frame. Input data appended with the Shifts columns.

See Also

Other Data Validation: `check_query()`, `extract_hr()`, `flag_ch_ratio()`, `flag_em_ratio()`, `flag_extreme()`, `flag_outlooktime()`, `hr_trend()`, `hrvar_count_all()`, `hrvar_count()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts_wp()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate_report()`, `subject_validate()`, `track_HR_change()`, `validation_report()`

Other Working Patterns: `flex_index()`, `identify_shifts_wp()`, `plot_flex_index()`, `workpatterns_area()`, `workpatterns_classify_bw()`, `workpatterns_classify_pav()`, `workpatterns_classify()`, `workpatterns_hclust()`, `workpatterns_rank()`, `workpatterns_report()`

Examples

```
# Return plot
dv_data %>% identify_shifts()

# Return summary table
dv_data %>% identify_shifts(return = "table")
```

| | |
|--------------------|---|
| identify_shifts_wp | <i>Identify shifts based on binary activity</i> |
|--------------------|---|

Description

This function uses the Hourly Collaboration query and computes binary activity to identify the 'behavioural' work shift. This is a distinct method to `identify_shifts()`, which instead uses outlook calendar settings for start and end time of work day to identify work shifts. The two methods can be compared to gauge the accuracy of existing Outlook settings.

Usage

```
identify_shifts_wp(
  data,
  signals = c("email", "IM"),
  active_threshold = 1,
  start_hour = 9,
  end_hour = 17,
  percent = FALSE,
  n = 10,
  return = "plot"
)
```

Arguments

| | |
|------------------|--|
| data | A data frame containing data from the Hourly Collaboration query. |
| signals | Character vector to specify which collaboration metrics to use: <ul style="list-style-type: none"> • a combination of signals, such as <code>c("email", "IM")</code> (default) • "email" for emails only • "IM" for Teams messages only • "unscheduled_calls" for Unscheduled Calls only • "meetings" for Meetings only |
| active_threshold | A numeric value specifying the minimum number of signals to be greater than in order to qualify as <i>active</i> . Defaults to 0. |
| start_hour | A character vector specifying starting hours, e.g. "0900". Note that this currently only supports hourly increments. If the official hours specifying checking in and 9 AM and checking out at 5 PM, then "0900" should be supplied here. |
| end_hour | A character vector specifying starting hours, e.g. "1700". Note that this currently only supports hourly increments. If the official hours specifying checking in and 9 AM and checking out at 5 PM, then "1700" should be supplied here. |
| percent | Logical value to determine whether to show labels as percentage signs. Defaults to FALSE. |
| n | Numeric value specifying number of shifts to show. Defaults to 10. This parameter is only used when return is set to "plot", |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" • "data" <p>See Value for more information.</p> |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. A bar plot for the weekly count of shifts.
- "table": data frame. A summary table for the count of shifts.
- "data": data frame. Input data appended with the following columns:
 - Start
 - End
 - DaySpan
 - Shifts

See Also

Other Data Validation: `check_query()`, `extract_hr()`, `flag_ch_ratio()`, `flag_em_ratio()`, `flag_extreme()`, `flag_outlooktime()`, `hr_trend()`, `hrvar_count_all()`, `hrvar_count()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate_report()`, `subject_validate()`, `track_HR_change()`, `validation_report()`

Other Working Patterns: `flex_index()`, `identify_shifts()`, `plot_flex_index()`, `workpatterns_area()`, `workpatterns_classify_bw()`, `workpatterns_classify_pav()`, `workpatterns_classify()`, `workpatterns_hclust()`, `workpatterns_rank()`, `workpatterns_report()`

Examples

```
# Return plot
em_data %>% identify_shifts_wp()

# Return plot - showing percentages
em_data %>% identify_shifts_wp(percent = TRUE)

# Return table
em_data %>% identify_shifts_wp(return = "table")
```

| | |
|-----------------|---|
| identify_tenure | <i>Tenure calculation based on different input dates, returns data summary table or histogram</i> |
|-----------------|---|

Description

This function calculates employee tenure based on different input dates. `identify_tenure` uses the latest Date available if user selects "Date", but also have flexibility to select a specific date, e.g. "1/1/2020".

Usage

```
identify_tenure(
  data,
  end_date = "Date",
  beg_date = "HireDate",
  maxten = 40,
  return = "message"
)
```

Arguments

| | |
|----------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| end_date | A string specifying the name of the date variable representing the latest date. Defaults to "Date". |
| beg_date | A string specifying the name of the date variable representing the hire date. Defaults to "HireDate". |
| maxten | A numeric value representing the maximum tenure. If the tenure exceeds this threshold, it would be accounted for in the flag message. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "message" • "text" • "plot" • "data_cleaned" • "data_dirty" • "data" <p>See Value for more information.</p> |

Value

A different output is returned depending on the value passed to the return argument:

- "message": message on console with a diagnostic message.
- "text": string containing a diagnostic message.
- "plot": 'ggplot' object. A line plot showing tenure.
- "data_cleaned": data frame filtered only by rows with tenure values lying within the threshold.
- "data_dirty": data frame filtered only by rows with tenure values lying outside the threshold.
- "data": data frame with the PersonId and a calculated variable called TenureYear is returned.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Examples

```
library(dplyr)
# Add HireDate to sq_data
sq_data2 <-
  sq_data %>%
```

```
mutate(HireDate = as.Date("1/1/2015", format = "%m/%d/%Y"))
identify_tenure(sq_data2)
```

| | |
|---------------|---|
| import_to_fst | <i>Read a Workplace Analytics query in '.csv' using and create a '.fst' file in the same directory for faster reading</i> |
|---------------|---|

Description

Uses `import_wpa()` to read a Workplace Analytics query in '.csv' and convert this into the serialized '.fst' format which is much faster to read. The 'fst' package must be installed, or an error message is returned.

Usage

```
import_to_fst(path, ...)
```

Arguments

| | |
|------|---|
| path | String containing the path to the Workplace Analytics query to be imported. The input file must be a CSV file, and the file extension must be explicitly entered, e.g. <code>"/files/standard query.csv"</code> . The converted FST file will be saved in the same directory with a different file extension. |
| ... | Additional arguments to pass to <code>import_wpa()</code> . |

Details

The `fst` package provides a way to serialize data frames in R which makes loading data much faster than CSV. `import_to_fst()` converts a CSV file into a FST file in the specified directory.

Once this FST file is created, it can be read into R using `fst::read_fst()`. Since `import_to_fst()` only does conversion but not loading, it should normally only be run once at the beginning of each piece of analysis, and `fst::read_fst()` should take over the job of data loading at the start of your analysis script.

Internally, `import_to_fst()` uses `import_wpa()`, and additional arguments to `import_wpa()` can be passed with `...`

Value

There is no return value. A file with '.fst' extension is written to the same directory where the '.csv' file is read in.

See Also

Other Import and Export: `copy_df()`, `create_dt()`, `export()`, `import_wpa()`, `standardise_pq()`

| | |
|------------|---|
| import_wpa | <i>Import a Workplace Analytics Query</i> |
|------------|---|

Description

Import a Workplace Analytics Query from a local CSV File, with variable classifications optimised for other 'wpa' functions.

Usage

```
import_wpa(x, standardise = FALSE, encoding = "UTF-8")
```

Arguments

| | |
|-------------|---|
| x | String containing the path to the Workplace Analytics query to be imported. The input file must be a CSV file, and the file extension must be explicitly entered, e.g. <code>"/files/standard query.csv"</code> |
| standardise | logical. If TRUE, <code>import_wpa()</code> runs <code>standardise_pq()</code> to make a Collaboration Assessment query's columns name standard and consistent with a Standard Person Query. Note that this will have no effect if the query being imported is not a Ways of Working Assessment query. Defaults as FALSE. |
| encoding | String to specify encoding to be used within <code>data.table::fread()</code> . See <code>data.table::fread()</code> documentation for more information. Defaults to 'UTF-8'. |

Details

`import_wpa()` uses `data.table::fread()` to import CSV files for speed, and by default `stringsAsFactors` is set to FALSE. A data frame is returned by the function (not a `data.table`).

Value

A tibble is returned.

See Also

Other Import and Export: [copy_df\(\)](#), [create_dt\(\)](#), [export\(\)](#), [import_to_fst\(\)](#), [standardise_pq\(\)](#)

internal_network_plot *Plot Internal Network Breadth and Size as a scatter plot*

Description

Plot the internal network metrics for a HR variable as a scatter plot, showing Internal Network Breadth as the vertical axis and Internal Network Size as the horizontal axis.

Usage

```
internal_network_plot(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  bubble_size = c(1, 8)
)
```

Arguments

| | |
|-------------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: - "plot" - "table" |
| bubble_size | A numeric vector of length two to specify the size range of the bubbles |

Details

Uses the metrics `Internal_network_size` and `Networking_outside_organization`.

Value

'ggplot' object showing a bubble plot with internal network size as the x-axis and internal network breadth as the y-axis. The size of the bubbles represent the number of unique employees in each group.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#),

[create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#),
[create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#),
[email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [keymetrics_scan\(\)](#),
[meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#),
[meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#),
[meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#),
[one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#),
[workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#),
[workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Network: [external_network_plot\(\)](#), [g2g_data](#), [network_describe\(\)](#), [network_g2g\(\)](#),
[network_leiden\(\)](#), [network_louvain\(\)](#), [network_p2p\(\)](#), [p2p_data_sim\(\)](#)

Examples

```
# Return plot
sq_data %>% internal_network_plot(return = "plot")

# Return summary table
sq_data %>% internal_network_plot(return = "table")
```

| | |
|----------------|---|
| is_date_format | <i>Identify whether string is a date format</i> |
|----------------|---|

Description

This function uses regular expression to determine whether a string is of the format "mdy", separated by "-", "/", or ".", returning a logical vector.

Usage

```
is_date_format(string)
```

Arguments

string Character string to test whether is a date format.

Value

logical value indicating whether the string is a date format.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#),
[extract_hr\(\)](#), [heat_colours\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#),
[rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
is_date_format("1/5/2020")
```

| | |
|--------------|---|
| IV_by_period | <i>Identify the WPA metrics that have the biggest change between two periods.</i> |
|--------------|---|

Description**[Experimental]**

This function uses the Information Value algorithm to predict which Workplace Analytics metrics are most explained by the change in dates.

Usage

```
IV_by_period(
  data,
  before_start = min(as.Date(data$Date, "%m/%d/%Y")),
  before_end,
  after_start = as.Date(before_end) + 1,
  after_end = max(as.Date(data$Date, "%m/%d/%Y")),
  mybins = 10,
  return = "table"
)
```

Arguments

| | |
|--------------|--|
| data | Person Query as a dataframe including date column named "Date" This function assumes the data format is MM/DD/YYYY as is standard in a Workplace Analytics query output. |
| before_start | Start date of "before" time period in YYYY-MM-DD. Defaults to earliest date in dataset. |
| before_end | End date of "before" time period in YYYY-MM-DD |
| after_start | Start date of "after" time period in YYYY-MM-DD. Defaults to day after before_end. |
| after_end | End date of "after" time period in YYYY-MM-DD. Defaults to latest date in dataset. |
| mybins | Number of bins to cut the data into for Information Value analysis. Defaults to 10. |
| return | String specifying what to return. The current only valid option is "table". |

Value

data frame containing all the variables and the corresponding Information Value.

Author(s)

Mark Powers mark.powers@microsoft.com

See Also

Other Variable Association: [IV_report\(\)](#), [create_IV\(\)](#), [plot_WOE\(\)](#)

Other Information Value: [IV_report\(\)](#), [create_IV\(\)](#), [plot_WOE\(\)](#)

Other Time-series: [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_trend\(\)](#), [period_change\(\)](#)

Examples

```
# Returns a data frame
sq_data %>%
  IV_by_period(
    before_start = "2019-11-03",
    before_end = "2019-11-09",
    after_start = "2019-11-10",
    after_end = "2019-11-16"
  )
```

IV_report

Generate a Information Value HTML Report

Description

The function generates an interactive HTML report using Standard Person Query data as an input. The report contains a full Information Value analysis, a data exploration technique that helps determine which columns in a data set have predictive power or influence on the value of a specified dependent variable.

Usage

```
IV_report(
  data,
  predictors = NULL,
  outcome,
  bins = 5,
  max_var = 9,
  path = "IV report",
  timestamp = TRUE
)
```

Arguments

| | |
|------------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| predictors | A character vector specifying the columns to be used as predictors. Defaults to NULL, where all numeric vectors in the data will be used as predictors. |
| outcome | A string specifying a binary variable, i.e. can only contain the values 1 or 0. |
| bins | Number of bins to use in <code>Information::create_infotables()</code> , defaults to 10. |
| max_var | Numeric value to represent the maximum number of variables to show on plots. |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . For example, "IV report". |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

Creating a report

Below is an example on how to run the report.

```
library(dplyr)

sq_data %>%
  mutate(CH_binary = ifelse(Collaboration_hours > 12, 1, 0)) %>% # Simulate binary variable
  IV_report(outcome = "CH_binary",
            predictors = c("Email_hours", "Workweek_span"))
```

See Also

Other Reports: [capacity_report\(\)](#), [coaching_report\(\)](#), [collaboration_report\(\)](#), [connectivity_report\(\)](#), [generate_report\(\)](#), [meeting_tm_report\(\)](#), [read_preamble\(\)](#), [subject_validate_report\(\)](#), [validation_report\(\)](#), [workpatterns_report\(\)](#)

Other Variable Association: [IV_by_period\(\)](#), [create_IV\(\)](#), [plot_WOE\(\)](#)

Other Information Value: [IV_by_period\(\)](#), [create_IV\(\)](#), [plot_WOE\(\)](#)

| | |
|----------------|---------------------------------------|
| jitter_metrics | <i>Jitter metrics in a data frame</i> |
|----------------|---------------------------------------|

Description

Convenience wrapper around `jitter()` to add a layer of anonymity to a query. This can be used in combination with `anonymise()` to produce a demo dataset from real data.

Usage

```
jitter_metrics(data, cols = NULL, ...)
```

Arguments

| | |
|-------------------|---|
| <code>data</code> | Data frame containing a query. |
| <code>cols</code> | Character vector containing the metrics to jitter. When set to <code>NULL</code> (default), all numeric columns in the data frame are jittered. |
| <code>...</code> | Additional arguments to pass to <code>jitter()</code> . |

See Also

`anonymise`

Examples

```
jittered <- jitter_metrics(sq_data, cols = "Collaboration_hours")
head(
  data.frame(
    original = sq_data$Collaboration_hours,
    jittered = jittered$Collaboration_hours
  )
)
```

| | |
|-----------------|---|
| keymetrics_scan | <i>Run a summary of Key Metrics from the Standard Person Query data</i> |
|-----------------|---|

Description

Returns a heatmapped table by default, with options to return a table.

Usage

```
keymetrics_scan(
  data,
  hrvar = "Organization",
  mingroup = 5,
  metrics = c("Workweek_span", "Collaboration_hours",
    "After_hours_collaboration_hours", "Meetings", "Meeting_hours",
    "After_hours_meeting_hours", "Low_quality_meeting_hours",
    "Meeting_hours_with_manager_1_on_1", "Meeting_hours_with_manager", "Emails_sent",
    "Email_hours", "After_hours_email_hours", "Generated_workload_email_hours",
    "Total_focus_hours", "Internal_network_size", "Networking_outside_organization",
    "External_network_size", "Networking_outside_company"),
  return = "plot",
  textsize = 2
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| metrics | A character vector containing the variable names to calculate averages of. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |
| textsize | A numeric value specifying the text size to show in the plot. |

Value

Returns a ggplot object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#),

```
one2one_freq(), one2one_line(), one2one_rank(), one2one_sum(), one2one_trend(), period_change(),
workloads_dist(), workloads_fizz(), workloads_line(), workloads_rank(), workloads_summary(),
workloads_trend(), workpatterns_area(), workpatterns_rank()
```

Examples

```
# Heatmap plot is returned by default
keymetrics_scan(sq_data)

# Return summary table
keymetrics_scan(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|--------|---|
| map_IV | <i>Calculate Weight of Evidence (WOE) and Information Value (IV) between multiple predictors and a single outcome variable, returning a list of statistics.</i> |
|--------|---|

Description

This is a wrapper around `calculate_IV()` to loop through multiple predictors and calculate their Weight of Evidence (WOE) and Information Value (IV) with respect to an outcome variable.

Usage

```
map_IV(data, predictors = NULL, outcome, bins = 10)
```

Arguments

| | |
|------------|--|
| data | Data frame containing the data. |
| predictors | Character vector containing the names of the predictor variables. If NULL (default) is supplied, all numeric variables in the data will be used. |
| outcome | String containing the name of the outcome variable. |
| bins | Numeric value representing the number of bins to use. Defaults to 10. |

Details

The approach used mirrors the one used in `Information::create_infotables()`.

Value

A list of data frames is returned as an output. The first layer of the list contains `Tables` and `Summary`:

- `Tables` is a list of data frames containing the WOE and cumulative sum IV for each predictor.
- `Summary` is a single data frame containing the IV for all predictors.

| | |
|--------|---------------------------------|
| maxmin | <i>Max-Min Scaling Function</i> |
|--------|---------------------------------|

Description

This function allows you to scale vectors or an entire data frame using the max-min scaling method. A numeric vector is always returned.

Usage

```
maxmin(x)
```

Arguments

`x` Pass a vector or the required columns of a data frame through this argument.

Details

This is used within `keymetrics_scan()` to enable row-wise heatmapping. Originally implemented in <https://github.com/martinctc/surveytoolbox>.

Value

Returns a numeric vector with the input rescaled.

See Also

Other Support: `camel_clean()`, `check_inputs()`, `combine_signals()`, `cut_hour()`, `extract_date_range()`, `extract_hr()`, `heat_colours()`, `is_date_format()`, `p_test()`, `pairwise_count()`, `plot_WOE()`, `read_preamble()`, `rgb2hex()`, `totals_bind()`, `totals_col()`, `totals_reorder()`, `tstamp()`, `us_to_space()`, `wrap()`

Examples

```
numbers <- c(15, 40, 10, 2)
maxmin(numbers)
```

| | |
|------------------|--|
| meetingtype_dist | <i>Distribution of Meeting Types by number of Attendees and Duration</i> |
|------------------|--|

Description

Calculate the hour distribution of internal meeting types. This is a wrapper around `meetingtype_dist_mt()` and `meetingtype_dist_ca()`, depending on whether a Meeting Query or a Ways of Working Assessment Query is passed as an input.

Usage

```
meetingtype_dist(data, hrvar = NULL, mingroup = 5, return = "plot")
```

Arguments

| | |
|-----------------------|---|
| <code>data</code> | Data frame. If a meeting query, must contain the variables <code>Attendee</code> and <code>DurationHours</code> . |
| <code>hrvar</code> | Character string to specify the HR attribute to split the data by. Note that this is only applicable if a Ways of Working Assessment query is passed to the function. If a Meeting Query is passed instead, this argument is ignored. |
| <code>mingroup</code> | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. Only applicable when using a Ways of Working Assessment query. |
| <code>return</code> | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the `return` argument:

- "plot": ggplot object. A matrix of meeting types with duration and the number of attendees. If using a Ways of Working Assessment query with `meetingtype_dist_ca()` and an HR attribute with more than one unique value is passed to `hrvar`, a stacked bar plot is returned.
- "table": data frame. A summary table.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#),

```
meeting_rank(), meeting_summary(), meeting_trend(), meetingtype_dist_ca(), meetingtype_dist_mt(),
meetingtype_summary(), mgrcoatt_dist(), mgrrel_matrix(), one2one_dist(), one2one_fizz(),
one2one_freq(), one2one_line(), one2one_rank(), one2one_sum(), one2one_trend(), period_change(),
workloads_dist(), workloads_fizz(), workloads_line(), workloads_rank(), workloads_summary(),
workloads_trend(), workpatterns_area(), workpatterns_rank()
```

```
Other Meetings: meeting_dist(), meeting_extract(), meeting_fizz(), meeting_line(), meeting_quality(),
meeting_rank(), meeting_skim(), meeting_summary(), meeting_tm_report(), meeting_trend(),
meetingtype_dist_ca(), meetingtype_dist_mt(), meetingtype_summary()
```

Examples

```
# Implementation using Standard Meeting Query
meetingtype_dist(mt_data)
```

meetingtype_dist_ca *Meeting Type Distribution (Ways of Working Assessment Query)*

Description

Calculate the hour distribution of internal meeting types, using a Ways of Working Assessment Query with core Workplace Analytics variables as an input.

Usage

```
meetingtype_dist_ca(data, hrvar = NULL, mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | Meeting Query data frame. Must contain the variables Attendee and DurationHours |
| hrvar | Character string to specify the HR attribute to split the data by. |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. A matrix of meeting types with duration and the number of attendees. If using a Ways of Working Assessment query with meetingtype_dist_ca() and an HR attribute with more than one unique value is passed to hrvar, a stacked bar plot is returned.
- "table": data frame. A summary table.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Meetings: `meeting_dist()`, `meeting_extract()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_skim()`, `meeting_summary()`, `meeting_tm_report()`, `meeting_trend()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`

`meetingtype_dist_mt` *Meeting Type Distribution (Meeting Query)*

Description

Calculate the hour distribution of internal meeting types, using a Meeting Query with core Workplace Analytics variables as an input.

Usage

```
meetingtype_dist_mt(data, return = "plot")
```

Arguments

| | |
|---------------------|--|
| <code>data</code> | Meeting Query data frame. Must contain the variables <code>Attendee</code> and <code>DurationHours</code> |
| <code>return</code> | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the `return` argument:

- "plot": ggplot object. A matrix of meeting types with duration and the number of attendees. If using a Ways of Working Assessment query with `meetingtype_dist_ca()` and an HR attribute with more than one unique value is passed to `hrvar`, a stacked bar plot is returned.
- "table": data frame. A summary table.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Meetings: `meeting_dist()`, `meeting_extract()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_skim()`, `meeting_summary()`, `meeting_tm_report()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist()`, `meetingtype_summary()`

| | |
|----------------------------------|--|
| <code>meetingtype_summary</code> | <i>Create a summary bar chart of the proportion of Meeting Hours spent in Long or Large Meetings</i> |
|----------------------------------|--|

Description

This function creates a bar chart showing the percentage of meeting hours which are spent in long or large meetings.

Usage

```
meetingtype_summary(  
  data,  
  hrvar = "Organization",  
  mingroup = 5,  
  return = "plot"  
)
```

```
meetingtype_sum(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|-------------------|---|
| <code>data</code> | Ways of Working Assessment query in the form of a data frame. Requires the following variables: <ul style="list-style-type: none"> • <code>Bloated_meeting_hours</code> • <code>Lengthy_meeting_hours</code> • <code>Workshop_meeting_hours</code> |
|-------------------|---|

| | |
|----------|---|
| | <ul style="list-style-type: none"> • All_hands_meeting_hours • Status_update_meeting_hours • Decision_making_meeting_hours • One_on_one_meeting_hours |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" <p>See Value for more information.</p> |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. A horizontal bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Meetings: [meeting_dist\(\)](#), [meeting_extract\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_skim\(\)](#), [meeting_summary\(\)](#), [meeting_tm_report\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#)

| | |
|--------------|--|
| meeting_dist | <i>Distribution of Meeting Hours as a 100% stacked bar</i> |
|--------------|--|

Description

Analyze Meeting Hours distribution. Returns a stacked bar plot by default. Additional options available to return a table with distribution elements.

Usage

```
meeting_dist(  
  data,  
  hrvar = "Organization",  
  mingroup = 5,  
  return = "plot",  
  cut = c(5, 10, 15)  
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "table" See Value for more information. |
| cut | A numeric vector of length three to specify the breaks for the distribution, e.g. c(10, 15, 20) |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Meetings: [meeting_extract\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_skim\(\)](#), [meeting_summary\(\)](#), [meeting_tm_report\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#)

Examples

```
# Return plot
meeting_dist(sq_data, hrvar = "Organization")

# Return summary table
meeting_dist(sq_data, hrvar = "Organization", return = "table")

# Return result with a custom specified breaks
meeting_dist(sq_data, hrvar = "LevelDesignation", cut = c(4, 7, 9))
```

```
meeting_extract
```

Extract top low-engagement meetings from the Meeting Query

Description

Pass a Standard Meeting Query and extract the top low engagement meetings.

Usage

```
meeting_extract(
  data,
  recurring_only = TRUE,
  top_n = 30,
  fte_month = 180,
  fte_week = 40,
  return = "table"
)
```

Arguments

| | |
|----------------|--|
| data | Data frame containing a Standard Meeting Query to pass through. |
| recurring_only | Logical value indicating whether to only filter by recurring meetings. |
| top_n | Numeric value for the top number of results to return in the output. |
| fte_month | Numeric value for the assumed number of employee hours per month for conversion calculations. Defaults to 180. |
| fte_week | Numeric value for the assumed number of employee hours per week for conversion calculations. Defaults to 180. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "table" • "data" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "table": data frame. A summary table containing the top n low engagement meetings
- "data": data frame. Contains the full computed metrics related to the top n low engagement meetings

See Also

Other Meetings: [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_skim\(\)](#), [meeting_summary\(\)](#), [meeting_tm_report\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#)

Examples

```
meeting_extract(mt_data,
                recurring_only = FALSE,
                top_n = 10,
                return = "table")
```

meeting_fizz

Distribution of Meeting Hours (Fizzy Drink plot)

Description

Analyze weekly meeting hours distribution, and returns a 'fizzy' scatter plot by default. Additional options available to return a table with distribution elements.

Usage

```
meeting_fizz(data, hrvar = "Organization", mingroup = 5, return = "plot")
```


Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Details

Uses the metric Meeting_hours.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A jittered scatter plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Meetings: [meeting_dist\(\)](#), [meeting_extract\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_skim\(\)](#), [meeting_summary\(\)](#), [meeting_tm_report\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#)

Examples

```
# Return plot
meeting_fizz(sq_data, hrvar = "Organization", return = "plot")

# Return summary table
meeting_fizz(sq_data, hrvar = "Organization", return = "table")
```

| | |
|--------------|--|
| meeting_line | <i>Meeting Time Trend - Line Chart</i> |
|--------------|--|

Description

Provides a week by week view of meeting time, visualised as line charts. By default returns a line chart for meeting hours, with a separate panel per value in the HR attribute. Additional options available to return a summary table.

Usage

```
meeting_line(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A faceted line plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Meetings: `meeting_dist()`, `meeting_extract()`, `meeting_fizz()`, `meeting_quality()`, `meeting_rank()`, `meeting_skim()`, `meeting_summary()`, `meeting_tm_report()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`

Examples

```
# Return a line plot
meeting_line(sq_data, hrvar = "LevelDesignation")

# Return summary table
meeting_line(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|-----------------|--|
| meeting_quality | <i>Run a meeting habits / meeting quality analysis</i> |
|-----------------|--|

Description

Return an analysis of Meeting Quality with a bubble plot, using a Standard Person Query as an input.

Usage

```
meeting_quality(
  data,
  hrvar = "Organization",
  metric_x = "Low_quality_meeting_hours",
  mingroup = 5,
  return = "plot"
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| metric_x | String specifying which variable to show in the x-axis when returning a plot. Must be one of the following: <ul style="list-style-type: none"> • "Low_quality_meeting_hours" (default) • "After_hours_meeting_hours" • "Conflicting_meeting_hours" • "Multitasking_meeting_hours" • Any <i>meeting hour</i> variable that can be divided by Meeting_hours |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: - "plot" - "table" |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bubble plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Meetings: [meeting_dist\(\)](#), [meeting_extract\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_rank\(\)](#), [meeting_skim\(\)](#), [meeting_summary\(\)](#), [meeting_tm_report\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#)

Examples

```
# Return plot
meeting_quality(sq_data, return = "plot")

# Return plot - showing multi-tasking %
meeting_quality(sq_data,
  metric_x = "Multitasking_meeting_hours",
  return = "plot")

# Return summary table
meeting_quality(sq_data, return = "table")
```

| | |
|--------------|------------------------------|
| meeting_rank | <i>Meeting Hours Ranking</i> |
|--------------|------------------------------|

Description

This function scans a standard query output for groups with high levels of Weekly Meeting Collaboration. Returns a table with a all of groups (across multiple HR attributes) ranked by hours of digital collaboration.

Usage

```
meeting_rank(
  data,
  hrvar = extract_hr(data),
  mingroup = 5,
  mode = "simple",
  plot_mode = 1,
  return = "table"
)
```

Arguments

| | |
|-----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| mode | String to specify calculation mode. Must be either: <ul style="list-style-type: none"> • "simple" • "combine" |
| plot_mode | Numeric vector to determine which plot mode to return. Must be either 1 or 2, and is only used when return = "plot". |

- 1: Top and bottom five groups across the data population are highlighted
- 2: Top and bottom groups *per* organizational attribute are highlighted

return String specifying what to return. This must be one of the following strings:

- "table" (default)
- "plot"

See Value for more information.

Details

Uses the metric Meeting_hours. See create_rank() for applying the same analysis to a different metric.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bubble plot where the x-axis represents the metric, the y-axis represents the HR attributes, and the size of the bubbles represent the size of the organizations. Note that there is no plot output if mode is set to "combine".
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Meetings: [meeting_dist\(\)](#), [meeting_extract\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_skim\(\)](#), [meeting_summary\(\)](#), [meeting_tm_report\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#)

Examples

```
# Return rank table
meeting_rank(
  data = sq_data,
  return = "table"
)
```

```
# Return plot
meeting_rank(
  data = sq_data,
  return = "plot"
)
```

`meeting_skim`*Produce a skim summary of meeting hours*

Description

This function returns a skim summary in the console when provided a standard query in the input.

Usage

```
meeting_skim(data, return = "message")
```

Arguments

| | |
|---------------------|---|
| <code>data</code> | A standard person query data in the form of a data frame. |
| <code>return</code> | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "message"• "text"• "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "message": message in console.
- "text": string.
- "table": data frame.

See Also

Other Meetings: [meeting_dist\(\)](#), [meeting_extract\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_tm_report\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#)

Examples

```
meeting_skim(sq_data)
```

| | |
|-----------------|------------------------|
| meeting_summary | <i>Meeting Summary</i> |
|-----------------|------------------------|

Description

Provides an overview analysis of weekly meeting hours. Returns a bar plot showing average weekly meeting hours by default. Additional options available to return a summary table.

Usage

```
meeting_summary(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

```
meeting_sum(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#),


```
meetingtype_summary(), mgrcoatt_dist(), mgrrel_matrix(), one2one_dist(), one2one_fizz(),
one2one_freq(), one2one_line(), one2one_rank(), one2one_sum(), one2one_trend(), period_change(),
workloads_dist(), workloads_fizz(), workloads_line(), workloads_rank(), workloads_summary(),
workloads_trend(), workpatterns_area(), workpatterns_rank()
```

```
Other Meetings: meeting_dist(), meeting_extract(), meeting_fizz(), meeting_line(), meeting_quality(),
meeting_rank(), meeting_skim(), meeting_tm_report(), meeting_trend(), meetingtype_dist_ca(),
meetingtype_dist_mt(), meetingtype_dist(), meetingtype_summary()
```

Examples

```
# Return a ggplot bar chart
meeting_summary(sq_data, hrvar = "LevelDesignation")

# Return a summary table
meeting_summary(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|-------------------|--|
| meeting_tm_report | <i>Generate a Meeting Text Mining report in HTML</i> |
|-------------------|--|

Description

Create a text mining report in HTML based on Meeting Subject Lines

Usage

```
meeting_tm_report(
  data,
  path = "meeting text mining report",
  stopwords = NULL,
  timestamp = TRUE,
  keep = 100,
  seed = 100
)
```

Arguments

| | |
|-----------|--|
| data | A Meeting Query dataset in the form of a data frame. |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . For example, "meeting text mining report". |
| stopwords | A single-column data frame labelled 'word' containing custom stopwords to remove. |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |
| keep | A numeric vector specifying maximum number of words to keep. |
| seed | A numeric vector to set seed for random generation. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

See Also

Other Reports: `IV_report()`, `capacity_report()`, `coaching_report()`, `collaboration_report()`, `connectivity_report()`, `generate_report()`, `read_preamble()`, `subject_validate_report()`, `validation_report()`, `workpatterns_report()`

Other Meetings: `meeting_dist()`, `meeting_extract()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_skim()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`

Other Text-mining: `pairwise_count()`, `subject_validate_report()`, `subject_validate()`, `tm_clean()`, `tm_cooc()`, `tm_freq()`, `tm_wordcloud()`

| | |
|---------------|---------------------------------|
| meeting_trend | <i>Meeting Hours Time Trend</i> |
|---------------|---------------------------------|

Description

Provides a week by week view of meeting time. By default returns a week by week heatmap, highlighting the points in time with most activity. Additional options available to return a summary table.

Usage

```
meeting_trend(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |

Details

Uses the metric Meeting_hours.

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Meetings: [meeting_dist\(\)](#), [meeting_extract\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_skim\(\)](#), [meeting_summary\(\)](#), [meeting_tm_report\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#)

mgrcoatt_dist

Manager meeting coattendance distribution

Description

Analyze degree of attendance between employees and their managers. Returns a stacked bar plot of different buckets of coattendance. Additional options available to return a table with distribution elements.

Usage

```
mgrcoatt_dist(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> "plot" "table" |

See Value for more information.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. A stacked bar plot showing the distribution of manager co-attendance time.
- "table": data frame. A summary table for manager co-attendance time.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Managerial Relations: [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#)

Examples

```
# Return plot
mgrcoatt_dist(sq_data, hrvar = "Organization", return = "plot")

# Return summary table
mgrcoatt_dist(sq_data, hrvar = "Organization", return = "table")
```

mgrrel_matrix

Manager Relationship 2x2 Matrix

Description

Generate the Manager-Relationship 2x2 matrix, returning a 'ggplot' object by default. Additional options available to return a "wide" or "long" summary table.

Usage

```
mgrrel_matrix(
  data,
  hrvar = NULL,
  mingroup = 5,
  return = "plot",
  plot_colors = c("#fe7f4f", "#b4d5dd", "#facebc", "#fcf0eb"),
  threshold = 15
)
```

Arguments

| | |
|-------------|--|
| data | Standard Person Query data to pass through. Accepts a data frame. |
| hrvar | HR Variable by which to split metrics. Accepts a character vector, e.g. "Organization". Defaults to NULL. |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" • "data" See Value for more information. |
| plot_colors | Pass a character vector of length 4 containing HEX codes to specify colors to use in plotting. |
| threshold | Specify a numeric value to determine threshold (in minutes) for 1:1 manager hours. Defaults to 15. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. When NULL is passed to hrvar, a two-by-two grid where the size of the grid represents total percentage of employees is returned. Otherwise, a horizontal stacked bar plot is returned.
- "table": data frame. A summary table is returned.
- "data": data frame. A long table grouped at the PersonId level with the following columns:
 - PersonId
 - HR variable supplied to hrvar
 - CoattendanceRate
 - Meeting_hours_with_manager_1_on_1
 - mgr1on1
 - Type

Author(s)

Lucas Hogner lucas.hogner@microsoft.com

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Managerial Relations: `mgrcoatt_dist()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`

Examples

```
# Return matrix
mgrrel_matrix(sq_data)

# Return stacked bar plot
mgrrel_matrix(sq_data, hrvar = "Organization")

## Visualize coaching style types
# Ensure dplyr is loaded
library(dplyr)

# Extract PersonId and Coaching Type
match_df <-
  sq_data %>%
  mgrrel_matrix(return = "data") %>%
  select(PersonId, Type)

# Join and visualize baseline
sq_data %>%
  left_join(match_df, by = "PersonId") %>%
  keymetrics_scan(hrvar = "Type",
                 return = "plot")
```

mt_data

Sample Meeting Query dataset

Description

A dataset generated from a Meeting Query from Workplace Analytics.

Usage

mt_data

Format

A data frame with 2001 rows and 30 variables:

- MeetingId**
- StartDate**
- StartTimeUTC**
- EndDate**
- EndTimeUTC**
- Attendee_meeting_hours**
- Attendees**
- Organizer_Domain**
- Organizer_FunctionType**
- Organizer_LevelDesignation**
- Organizer_Layer**
- Organizer_Region**
- Organizer_Organization**
- Organizer_zId**
- Organizer_attainment**
- Organizer_TimeZone**
- Organizer_HourlyRate**
- Organizer_IsInternal**
- Organizer_PersonId**
- IsCancelled**
- DurationHours**
- IsRecurring**
- Subject**
- TotalAccept**
- TotalNoResponse**
- TotalDecline**
- TotalNoEmailsDuringMeeting**
- TotalNoDoubleBooked**
- TotalNoAttendees**
- MeetingResources**
- Attendees_with_conflicting_meetings**
- Invitees**

Emails_sent_during_meetings
Attendees_multitasking
Redundant_attendees
Total_meeting_cost
Total_redundant_hours ...

Value

data frame.

Source

<https://workplaceanalytics-demo.office.com/en-us/Home>

See Also

Other Data: [dv_data](#), [em_data](#), [g2g_data](#), [p2p_data_sim\(\)](#), [sq_data](#)

| | |
|------------------|---|
| network_describe | <i>Uncover HR attributes which best represent a population for a Person to Person query</i> |
|------------------|---|

Description

[Experimental]

Returns a data frame that gives a percentage of the group combinations that best represent the population provided. Uses a person to person query. This is used internally within `network_p2p()`.

Usage

```
network_describe(
  data,
  hrvar = c("Organization", "LevelDesignation", "FunctionType")
)
```

Arguments

| | |
|-------|--|
| data | Data frame containing a vertex table output from <code>network_p2p()</code> . |
| hrvar | Character vector of length 3 containing the HR attributes to be used. Defaults to <code>c("Organization", "LevelDesignation", "FunctionType")</code> . |

Value

data frame. A summary table giving the percentage of group combinations that best represent the provided data.

Author(s)

Tannaz Sattari Tabrizi Tannaz.Sattari@microsoft.com

See Also

Other Network: [external_network_plot\(\)](#), [g2g_data](#), [internal_network_plot\(\)](#), [network_g2g\(\)](#), [network_leiden\(\)](#), [network_louvain\(\)](#), [network_p2p\(\)](#), [p2p_data_sim\(\)](#)

Examples

```
# Simulate a P2P edge list
sim_data <- p2p_data_sim()

# Perform Louvain Community Detection and return vertices
lc_df <-
  sim_data %>%
  network_p2p(
    display = "louvain",
    return = "data"
  )

# Join org data from input edge list
joined_df <-
  lc_df %>%
  dplyr::left_join(
    sim_data %>%
    dplyr::select(TieOrigin_PersonId,
                  TieOrigin_Organization,
                  TieOrigin_LevelDesignation,
                  TieOrigin_City),
    by = c("name" = "TieOrigin_PersonId"))

# Describe cluster 2
joined_df %>%
  # dplyr::filter(cluster == "2") %>%
  network_describe(
    hrvar = c(
      "Organization",
      "LevelDesignation",
      "City"
    )
  ) %>%
  dplyr::glimpse()
```

Description

Pass a data frame containing a group-to-group query and return a network plot. Automatically handles "Collaborators_within_group" and "Other_collaborators" within query data.

Usage

```
network_g2g(
  data,
  time_investor = NULL,
  collaborator = NULL,
  metric = "Collaboration_hours",
  algorithm = "fr",
  node_colour = "lightblue",
  exc_threshold = 0.1,
  org_count = NULL,
  subtitle = "Collaboration Across Organizations",
  return = "plot"
)

g2g_network(
  data,
  time_investor = NULL,
  collaborator = NULL,
  metric = "Collaboration_hours",
  algorithm = "fr",
  node_colour = "lightblue",
  exc_threshold = 0.1,
  org_count = NULL,
  subtitle = "Collaboration Across Organizations",
  return = "plot"
)
```

Arguments

| | |
|----------------------------|---|
| <code>data</code> | Data frame containing a G2G query. |
| <code>time_investor</code> | String containing the variable name for the Time Investor column. |
| <code>collaborator</code> | String containing the variable name for the Collaborator column. |
| <code>metric</code> | String containing the variable name for metric. Defaults to <code>Collaboration_hours</code> . |
| <code>algorithm</code> | String to specify the node placement algorithm to be used. Defaults to "fr" for the force-directed algorithm of Fruchterman and Reingold. See https://rdrr.io/cran/ggraph/man/layout_tbl_graph_igraph.html for a full list of options. |
| <code>node_colour</code> | String to specify the colour to be used for displaying nodes. Defaults to "lightblue". If "vary" is supplied, a different colour is shown for each node at random. |
| <code>exc_threshold</code> | Exclusion threshold to apply. |
| <code>org_count</code> | Optional data frame to provide the size of each organization in the collaborator attribute. The data frame should contain only two columns: |

| | |
|----------|---|
| | <ul style="list-style-type: none"> • Name of the collaborator attribute excluding any prefixes, e.g. "Organization". Must be of character or factor type. • "n". Must be of numeric type. Defaults to NULL, where node sizes will be fixed. |
| subtitle | String to override default plot subtitle. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" • "network" • "data" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A group-to-group network plot.
- "table": data frame. An interactive matrix of the network.
- "network": 'igraph' object used for creating the network plot.
- "data": data frame. A long table of the underlying data.

See Also

Other Network: [external_network_plot\(\)](#), [g2g_data](#), [internal_network_plot\(\)](#), [network_describe\(\)](#), [network_leiden\(\)](#), [network_louvain\(\)](#), [network_p2p\(\)](#), [p2p_data_sim\(\)](#)

Examples

```
# Return a network plot
g2g_data %>% network_g2g()

# Return a network plot - Meeting hours and 5% threshold
g2g_data %>%
  network_g2g(time_investor = "TimeInvestors_Organization",
              collaborator = "Collaborators_Organization",
              metric = "Meeting_hours",
              exc_threshold = 0.05)

# Return a network plot with circle layout
# Vary node colours and add org sizes
org_tb <- hrvar_count(
  sq_data,
  hrvar = "Organization",
  return = "table"
)

g2g_data %>%
  network_g2g(algorithm = "circle",
              node_colour = "vary",
```

```

        org_count = org_tb)

# Return an interaction matrix
# Minimum arguments specified
g2g_data %>%
  network_g2g(return = "table")

```

| | |
|----------------|---|
| network_leiden | <i>Implement the Leiden community detection on a Person to Person network query</i> |
|----------------|---|

Description

[Experimental]

Take a P2P network query and implement the Leiden community detection method. To run this function, you will require all the pre-requisites of the 'leiden' package installed, which includes Python and 'reticulate'.

Usage

```

network_leiden(
  data,
  hrvar = "Organization",
  bg_fill = "#000000",
  font_col = "#FFFFFF",
  algorithm = "mds",
  path = "network_p2p_leiden",
  node_alpha = 0.8,
  res = 0.5,
  seed = 1,
  desc_hrvar = c("Organization", "LevelDesignation", "FunctionType"),
  return = "plot-leiden",
  size_threshold = 5000
)

```

Arguments

| | |
|-----------|--|
| data | Data frame containing a person-to-person query. |
| hrvar | String containing the label for the HR attribute. |
| bg_fill | String to specify background fill colour. |
| font_col | String to specify font and link colour. |
| algorithm | String to specify the node placement algorithm to be used. Defaults to "mds" for the deterministic multi-dimensional scaling of nodes. See https://rdrr.io/cran/ggraph/man/layout_tbl_graph_igraph.html for a full list of options. |

| | |
|----------------|---|
| path | File path for saving the PDF output. Defaults to a timestamped path based on current parameters. |
| node_alpha | A numeric value between 0 and 1 to specify the transparency of the nodes. Defaults to 0.7. |
| res | Resolution parameter to be passed to <code>leiden::leiden()</code> . Defaults to 0.5. |
| seed | Seed for the random number generator passed to <code>leiden::leiden()</code> to ensure consistency. Only applicable when <code>display</code> is set to "leiden". |
| desc_hrvar | Character vector of length 3 containing the HR attributes to use when returning the "describe" output. See <code>network_describe()</code> . |
| return | String specifying what output to return. Defaults to "plot-leiden". Valid return options include: <ul style="list-style-type: none"> • 'plot-leiden': return a network plot coloured by leiden communities, saving a PDF to path. • 'plot-hrvar': return a network plot coloured by HR attribute, saving a PDF to path. • 'plot-sankey': return a sankey plot combining communities and HR attribute. • 'table': return a vertex summary table with counts in communities and HR attribute. • 'data': return a vertex data file that matches vertices with communities and HR attributes. • 'describe': return a list of data frames which describe each of the identified communities. The first data frame is a summary table of all the communities. • 'network': return 'igraph' object. |
| size_threshold | Numeric value representing the maximum number of edges before <code>network_leiden()</code> switches to use a more efficient, but less elegant plotting method (native <code>igraph</code>). Defaults to 5000. Set as 0 to coerce to a fast plotting method every time, and Inf to always use the default plotting method (with 'ggraph'). |

Value

See return.

Simulating and running Leiden Community Detection

Below is an example on how to simulate a network and run the function.

```
# Simulate a small person-to-person dataset
p2p_data <- p2p_data_sim(size = 50)

# Return leiden, console, plot
p2p_data %>%
  network_leiden(path = NULL,
                 return = "plot")
...

```

See Also

Other Network: [external_network_plot\(\)](#), [g2g_data](#), [internal_network_plot\(\)](#), [network_describe\(\)](#), [network_g2g\(\)](#), [network_louvain\(\)](#), [network_p2p\(\)](#), [p2p_data_sim\(\)](#)

| | |
|-----------------|--|
| network_louvain | <i>Implement the Louvain community detection on a Person to Person network query</i> |
|-----------------|--|

Description**[Experimental]**

Take a P2P network query and implement the Louvain community detection method. The 'igraph' implementation of the Louvain method is used.

Usage

```
network_louvain(
  data,
  hrvar = "Organization",
  bg_fill = "#000000",
  font_col = "#FFFFFF",
  node_alpha = 0.8,
  algorithm = "mds",
  path = "network_p2p_louvain",
  desc_hrvar = c("Organization", "LevelDesignation", "FunctionType"),
  return = "plot-louvain",
  size_threshold = 5000
)
```

Arguments

| | |
|------------|--|
| data | Data frame containing a person-to-person query. |
| hrvar | String containing the label for the HR attribute. |
| bg_fill | String to specify background fill colour. |
| font_col | String to specify font and link colour. |
| node_alpha | A numeric value between 0 and 1 to specify the transparency of the nodes. Defaults to 0.7. |
| algorithm | String to specify the node placement algorithm to be used. Defaults to "mds" for the deterministic multi-dimensional scaling of nodes. See https://rdr.io/cran/ggraph/man/layout_tbl_graph_igraph.html for a full list of options. |
| path | File path for saving the PDF output. Defaults to a timestamped path based on current parameters. |
| desc_hrvar | Character vector of length 3 containing the HR attributes to use when returning the "describe" output. See network_describe() . |

- return** String specifying what output to return. Defaults to "plot-louvain". Valid return options include:
- 'plot-louvain': return a network plot coloured by Louvain communities, saving a PDF to path.
 - 'plot-hrvar': return a network plot coloured by HR attribute, saving a PDF to path.
 - 'plot-sankey': return a sankey plot combining communities and HR attribute.
 - 'table': return a vertex summary table with counts in communities and HR attribute.
 - 'data': return a vertex data file that matches vertices with communities and HR attributes.
 - 'describe': return a list of data frames which describe each of the identified communities. The first data frame is a summary table of all the communities.
 - 'network': return 'igraph' object.
- size_threshold** Numeric value representing the maximum number of edges before `network_leiden()` switches to use a more efficient, but less elegant plotting method (native igraph). Defaults to 5000. Set as 0 to coerce to a fast plotting method every time, and Inf to always use the default plotting method (with 'ggraph').

Value

See return.

See Also

Other Network: [external_network_plot\(\)](#), [g2g_data](#), [internal_network_plot\(\)](#), [network_describe\(\)](#), [network_g2g\(\)](#), [network_leiden\(\)](#), [network_p2p\(\)](#), [p2p_data_sim\(\)](#)

Examples

```
# Simulate a small person-to-person dataset
p2p_data <- p2p_data_sim(size = 50)

# Return louvain, console, plot
p2p_data %>%
  network_louvain(path = NULL,
                 return = "plot")
```

`network_p2p`*Create a network plot with the person-to-person query*

Description

[Experimental]

Analyse a person-to-person (P2P) network query, with multiple visualisation and analysis output options. Pass a data frame containing a person-to-person query and return a network visualization. Options are available for community detection using either the Louvain or the Leiden algorithms.

Usage

```
network_p2p(  
  data,  
  hrvar = "Organization",  
  display = "hrvar",  
  return = "plot",  
  path = paste0("network_p2p_", display),  
  desc_hrvar = c("Organization", "LevelDesignation", "FunctionType"),  
  bg_fill = "#FFFFFF",  
  font_col = "grey20",  
  legend_pos = "bottom",  
  palette = "rainbow",  
  node_alpha = 0.7,  
  edge_alpha = 1,  
  res = 0.5,  
  seed = 1,  
  algorithm = "mds",  
  size_threshold = 5000,  
  weight = "StrongTieScore"  
)
```

Arguments

| | |
|----------------------|--|
| <code>data</code> | Data frame containing a person-to-person query. |
| <code>hrvar</code> | String containing the label for the HR attribute. |
| <code>display</code> | String determining what output to return. Valid values include: <ul style="list-style-type: none">• <code>"hrvar"</code> (default): compute analysis or visuals without computing communities.• <code>"louvain"</code>: compute analysis or visuals with community detection, using the Louvain algorithm.• <code>"leiden"</code>: compute analysis or visuals with community detection, using the Leiden algorithm. This requires all the pre-requisites of the leiden package installed, which includes Python and reticulate. |

| | |
|----------------|---|
| return | String specifying what output to return. This must be one of the following strings: <ul style="list-style-type: none"> • 'plot' (default) • 'sankey' • 'table' • 'data' • 'describe' • 'network' See Value for more information. |
| path | File path for saving the PDF output. Defaults to a timestamped path based on current parameters. |
| desc_hrvar | Character vector of length 3 containing the HR attributes to use when returning the "describe" output. See <code>network_describe()</code> . |
| bg_fill | String to specify background fill colour. |
| font_col | String to specify font and link colour. |
| legend_pos | String to specify position of legend. Defaults to "bottom". See <code>ggplot2::theme()</code> . This is applicable for both the 'ggraph' and the fast plotting method. Valid inputs include: <ul style="list-style-type: none"> • "bottom" • "top" • "left" - "right" |
| palette | Function for generating a colour palette with a single argument n. Uses "rainbow" by default. |
| node_alpha | A numeric value between 0 and 1 to specify the transparency of the nodes. Defaults to 0.7. |
| edge_alpha | A numeric value between 0 and 1 to specify the transparency of the edges (only for 'ggraph' mode). Defaults to 1. |
| res | Resolution parameter to be passed to <code>leiden::leiden()</code> . Defaults to 0.5. |
| seed | Seed for the random number generator passed to <code>leiden::leiden()</code> to ensure consistency. Only applicable when <code>display</code> is set to "leiden". |
| algorithm | String to specify the node placement algorithm to be used. Defaults to "mds" for the deterministic multi-dimensional scaling of nodes. See https://rdr.io/cran/ggraph/man/layout_tbl_graph_igraph.html for a full list of options. |
| size_threshold | Numeric value representing the maximum number of edges before <code>network_leiden()</code> switches to use a more efficient, but less elegant plotting method (native <code>igraph</code>). Defaults to 5000. Set as 0 to coerce to a fast plotting method every time, and Inf to always use the default plotting method (with 'ggraph'). |
| weight | String to specify which column to use as weights for the network. Defaults to "StrongTieScore. To create a graph without weights, supply NULL to this argument. |

Value

A different output is returned depending on the value passed to the return argument:

- 'plot': return a network plot.
- 'sankey': return a sankey plot combining communities and HR attribute. This is only valid if a community detection method is selected at display.
- 'table': return a vertex summary table with counts in communities and HR attribute.
- 'data': return a vertex data file that matches vertices with communities and HR attributes.
- 'describe': return a list of data frames which describe each of the identified communities. The first data frame is a summary table of all the communities. This is only valid if a community detection method is selected at display.
- 'network': return 'igraph' object.

Running Leiden communities

Running Leiden communities requires python dependencies installed. You can run the following:

```
# Return a network plot to console, coloured by Leiden communities
p2p_data %>%
  network_p2p(display = "leiden",
              path = NULL,
              return = "plot")
```

When installing the 'leiden' package, you may be required to install the Python libraries 'python-igraph' and 'leidenalg'. You can install them with:

```
reticulate::py_install("python-igraph")
reticulate::py_install("leidenalg")
```

See Also

Other Network: [external_network_plot\(\)](#), [g2g_data](#), [internal_network_plot\(\)](#), [network_describe\(\)](#), [network_g2g\(\)](#), [network_leiden\(\)](#), [network_louvain\(\)](#), [p2p_data_sim\(\)](#)

Examples

```
# Simulate a small person-to-person dataset
p2p_data <- p2p_data_sim(size = 50)

# Return a network plot to console, coloured by hrvar
p2p_data %>%
  network_p2p(display = "hrvar",
              path = NULL,
              return = "plot")

# Return a network plot to console, coloured by Louvain communities
p2p_data %>%
  network_p2p(display = "louvain",
              path = NULL,
```

```

return = "plot")

# Return a network plot to console
# Coloured by Leiden communities
# Using Fruchterman-Reingold force-directed layout algorithm
# Force the use of fast plotting method
p2p_data %>%
  network_p2p(display = "hrvar",
              path = NULL,
              return = "plot",
              algorithm = "lgl",
              size_threshold = 0)

# Return a data frame matching HR variable and communities to nodes
# Using Louvain communities
p2p_data %>%
  network_p2p(display = "louvain",
              return = "data",
              algorithm = "fr")

```

one2one_dist

Distribution of Manager 1:1 Time as a 100% stacked bar

Description

Analyze Manager 1:1 Time distribution. Returns a stacked bar plot of different buckets of 1:1 time. Additional options available to return a table with distribution elements.

Usage

```

one2one_dist(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  cut = c(5, 15, 30)
)

```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |

| | |
|--------|--|
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |
| cut | A numeric vector of length three to specify the breaks for the distribution, e.g. <code>c(10, 15, 20)</code> |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Managerial Relations: `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`

Examples

```
# Return plot
one2one_dist(sq_data, hrvar = "Organization", return = "plot")

# Return summary table
one2one_dist(sq_data, hrvar = "Organization", return = "table")
```

| | |
|--------------|--|
| one2one_fizz | <i>Distribution of Manager 1:1 Time (Fizzy Drink plot)</i> |
|--------------|--|

Description

Analyze weekly Manager 1:1 Time distribution, and returns a 'fizzy' scatter plot by default. Additional options available to return a table with distribution elements.

Usage

```
one2one_fizz(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A jittered scatter plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#),

`workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`,
`workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Managerial Relations: `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_freq()`,
`one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`

Examples

```
# Return plot
one2one_fizz(sq_data, hrvar = "Organization", return = "plot")

# Return a summary table
one2one_fizz(sq_data, hrvar = "Organization", return = "table")
```

| | |
|--------------|---|
| one2one_freq | <i>Frequency of Manager 1:1 Meetings as bar or 100% stacked bar chart</i> |
|--------------|---|

Description

[Experimental]

This function calculates the average number of weeks (cadence) between of 1:1 meetings between an employee and their manager. Returns a distribution plot for typical cadence of 1:1 meetings. Additional options available to return a bar plot, tables, or a data frame with a cadence of 1 on 1 meetings metric.

Usage

```
one2one_freq(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  mode = "dist",
  sort_by = "Quarterly or less\n(>10 weeks)"
)
```

Arguments

| | |
|-----------------------|--|
| <code>data</code> | A Standard Person Query dataset in the form of a data frame. |
| <code>hrvar</code> | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| <code>mingroup</code> | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| <code>return</code> | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> "plot" |

| | |
|---------|--|
| | <ul style="list-style-type: none"> • "table" |
| mode | String specifying what method to use. This must be one of the following strings: <ul style="list-style-type: none"> • "dist" • "sum" |
| sort_by | String to specify the bucket label to sort by. Defaults to NULL (no sorting). |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked bar plot for the metric.
- "table": data frame. A summary table for the metric.

Distribution view

For this view, there are four categories of cadence:

- Weekly (once per week)
- Twice monthly or more (up to 3 weeks)
- Monthly (3 - 6 weeks)
- Every two months (6 - 10 weeks)
- Quarterly or less (> 10 weeks)

In the occasion there are zero 1:1 meetings with managers, this is included into the last category, i.e. 'Quarterly or less'. Note that when mode is set to "sum", these rows are simply excluded from the calculation.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Managerial Relations: [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#)

Examples

```
# Return plot, mode dist
one2one_freq(sq_data,
             hrvar = "Organization",
             return = "plot",
             mode = "dist")

# Return plot, mode sum
one2one_freq(sq_data,
             hrvar = "Organization",
             return = "plot",
             mode = "sum")

# Return summary table
one2one_freq(sq_data, hrvar = "Organization", return = "table")
```

one2one_line

Manager 1:1 Time Trend - Line Chart

Description

Provides a week by week view of 1:1 time with managers, visualised as line charts. By default returns a line chart for 1:1 meeting hours, with a separate panel per value in the HR attribute. Additional options available to return a summary table.

Usage

```
one2one_line(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Details

Uses the metric Meeting_hours_with_manager_1_on_1.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A faceted line plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Managerial Relations: [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#)

Examples

```
# Return a line plot
one2one_line(sq_data, hrvar = "LevelDesignation")

# Return summary table
one2one_line(sq_data, hrvar = "LevelDesignation", return = "table")
```

one2one_rank

Manager 1:1 Time Ranking

Description

This function scans a standard query output for groups with high levels of 'Manager 1:1 Time'. Returns a table with a all of groups (across multiple HR attributes) ranked by hours of digital collaboration.

Usage

```
one2one_rank(
  data,
  hrvar = extract_hr(data),
  mingroup = 5,
  mode = "simple",
  plot_mode = 1,
  return = "table"
)
```

Arguments

| | |
|-----------|---|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| mode | String to specify calculation mode. Must be either: <ul style="list-style-type: none"> • "simple" • "combine" |
| plot_mode | Numeric vector to determine which plot mode to return. Must be either 1 or 2, and is only used when return = "plot". <ul style="list-style-type: none"> • 1: Top and bottom five groups across the data population are highlighted • 2: Top and bottom groups <i>per</i> organizational attribute are highlighted |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "table" (default) • "plot" See Value for more information. |

Details

Uses the metric Meeting_hours_with_manager_1_on_1. See create_rank() for applying the same analysis to a different metric.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bubble plot where the x-axis represents the metric, the y-axis represents the HR attributes, and the size of the bubbles represent the size of the organizations. Note that there is no plot output if mode is set to "combine".
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Managerial Relations: `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_sum()`, `one2one_trend()`

Examples

```
# Return rank table
one2one_rank(
  data = sq_data,
  return = "table"
)

# Return plot
one2one_rank(
  data = sq_data,
  return = "plot"
)
```

 one2one_sum

Manager 1:1 Time Summary

Description

Provides an overview analysis of Manager 1:1 Time. Returns a bar plot showing average weekly minutes of Manager 1:1 Time by default. Additional options available to return a summary table.

Usage

```
one2one_sum(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

```
one2one_summary(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Managerial Relations: [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_trend\(\)](#)

Examples

```
# Return a ggplot bar chart
one2one_sum(sq_data, hrvar = "LevelDesignation")

# Return a summary table
one2one_sum(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|---------------|-------------------------------|
| one2one_trend | <i>Manager 1:1 Time Trend</i> |
|---------------|-------------------------------|

Description

Provides a week by week view of scheduled manager 1:1 Time. By default returns a week by week heatmap, highlighting the points in time with most activity. Additional options available to return a summary table.

Usage

```
one2one_trend(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |

Details

Uses the metric Meeting_hours_with_manager_1_on_1.

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [period_change\(\)](#),

[workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#),
[workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Managerial Relations: [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#),
[one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#)

p2p_data_sim

Simulate a person-to-person query using a Watts-Strogatz model

Description

Generate an person-to-person query / edgelist based on the graph according to the Watts-Strogatz small-world network model. Organizational data fields are also simulated for Organization, LevelDesignation, and City.

Usage

```
p2p_data_sim(dim = 1, size = 300, nei = 5, p = 0.05)
```

Arguments

| | |
|------|--|
| dim | Integer constant, the dimension of the starting lattice. |
| size | Integer constant, the size of the lattice along each dimension. |
| nei | Integer constant, the neighborhood within which the vertices of the lattice will be connected. |
| p | Real constant between zero and one, the rewiring probability. |

Details

This is a wrapper around `igraph::watts.strogatz.game()`. See `igraph` documentation for details on methodology. Loop edges and multiple edges are disabled. Size of the network can be changing the arguments `size` and `nei`.

Value

data frame with the same column structure as a person-to-person flexible query. This has an edgelist structure and can be used directly as an input to `network_p2p()`.

See Also

Other Data: [dv_data](#), [em_data](#), [g2g_data](#), [mt_data](#), [sq_data](#)

Other Network: [external_network_plot\(\)](#), [g2g_data](#), [internal_network_plot\(\)](#), [network_describe\(\)](#),
[network_g2g\(\)](#), [network_leiden\(\)](#), [network_louvain\(\)](#), [network_p2p\(\)](#)

Examples

```
# Simulate a p2p dataset with 800 edges
p2p_data_sim(size = 200, nei = 4)
```

| | |
|----------------|--|
| pairwise_count | <i>Perform a pairwise count of words by id</i> |
|----------------|--|

Description

This is a 'data.table' implementation that mimics the output of pairwise_count() from 'widyr' to reduce package dependency. This is used internally within tm_cooc().

Usage

```
pairwise_count(data, id = "line", word = "word")
```

Arguments

| | |
|------|--|
| data | Data frame output from tm_clean(). |
| id | String to represent the id variable. Defaults to "line". |
| word | String to represent the word variable. Defaults to "word". |

Value

data frame with the following columns representing a pairwise count:

- "item1"
- "item2"
- "n"

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Other Text-mining: [meeting_tm_report\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [tm_clean\(\)](#), [tm_cooc\(\)](#), [tm_freq\(\)](#), [tm_wordcloud\(\)](#)

Examples

```
td <- data.frame(line = c(1, 1, 2, 2),
                 word = c("work", "meeting", "catch", "up"))

pairwise_count(td, id = "line", word = "word")
```

| | |
|---------------|---|
| period_change | <i>Plot the distribution of percentage change between periods of a WpA metric by the number of employees.</i> |
|---------------|---|

Description

This function also presents the p-value for the null hypothesis that the variable has not changed, using a Wilcoxon signed-rank test.

Usage

```
period_change(
  data,
  compvar,
  before_start = min(as.Date(data$Date, "%m/%d/%Y")),
  before_end,
  after_start = as.Date(before_end) + 1,
  after_end = max(as.Date(data$Date, "%m/%d/%Y")),
  return = "count"
)
```

Arguments

| | |
|--------------|--|
| data | Person Query as a dataframe including date column named "Date" This function assumes the data format is MM/DD/YYYY as is standard in a Workplace Analytics query output. |
| compvar | WpA comparison variable to compare person change before and after For example, "Collaboration_hours" |
| before_start | Start date of "before" time period in YYYY-MM-DD |
| before_end | End date of "before" time period in YYYY-MM-DD |
| after_start | Start date of "after" time period in YYYY-MM-DD |
| after_end | End date of "after" time period in YYYY-MM-DD |
| return | Character vector specifying whether to return plot as Count or Percentage of Employees. Valid inputs include: <ul style="list-style-type: none"> • "count" (default) • "percentage" • "table" |

Value

ggplot object showing a bar plot (histogram) of change for two time intervals.

Author(s)

Mark Powers mark.powers@microsoft.com

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Time-series: [IV_by_period\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_trend\(\)](#)

Other Flexible: [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_hist\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#)

Other Flexible Input: [create_ITSA\(\)](#)

Examples

```
# Run plot
period_change(sq_data, compvar = "Workweek_span", before_end = "2019-11-16")
```

```
# Run plot with more specific arguments
period_change(sq_data,
  compvar = "Workweek_span",
  before_start = "2019-11-03",
  before_end = "2019-11-16",
  after_start = "2019-12-03",
  after_end = "2019-12-16",
  return = "percentage")
```

personas_hclust

Create hierarchical clusters of selected metrics using a Person query

Description**[Questioning]**

Apply hierarchical clustering to selected metrics. Person averages are computed prior to clustering. The hierarchical clustering uses cosine distance and the ward.D method of agglomeration.

Usage

```
personas_hclust(data, metrics, k = 4, return = "plot")
```

Arguments

| | |
|---------|---|
| data | A data frame containing PersonId and selected metrics for clustering. |
| metrics | Character vector containing names of metrics to use for clustering. See examples section. |
| k | Numeric vector to specify the k number of clusters to cut by. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "data"• "table"• "hclust" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A heatmap plot comparing the key metric averages of the clusters as per `keymetrics_scan()`.
- "data": data frame. Raw data with clusters appended
- "table": data frame. Summary table for identified clusters
- "hclust": 'hclust' object. hierarchical model generated by the function.

Author(s)

Ainize Cidoncha ainize.cidoncha@microsoft.com

See Also

Other Clustering: [workpatterns_classify\(\)](#), [workpatterns_hclust\(\)](#)

Examples

```
# Return plot
personas_hclust(sq_data,
                metrics = c("Collaboration_hours", "Workweek_span"),
                k = 4)

# Return summary table
personas_hclust(sq_data,
                metrics = c("Collaboration_hours", "Workweek_span"),
                k = 4,
                return = "table")
```

```
# Return data with clusters appended
personas_hclust(sq_data,
               metrics = c("Collaboration_hours", "Workweek_span"),
               k = 4,
               return = "data")
```

plot_flex_index *Plot a Sample of Working Patterns using Flexibility Index output*

Description

This is a helper function for plotting visualizations for the Flexibility Index using the data output from `flex_index()`. This is used within `flex_index()` itself as an internal function.

Usage

```
plot_flex_index(
  data,
  sig_label = "Signals_sent_",
  method = "sample",
  start_hour = 9,
  end_hour = 17
)
```

Arguments

| | |
|-------------------------|--|
| <code>data</code> | Data frame. Direct data output from <code>flex_index()</code> . |
| <code>sig_label</code> | Character string for identifying signal labels. |
| <code>method</code> | Character string for determining which plot to return. Options include "sample", "common", and "time". "sample" plots a sample of ten working patterns; "common" plots the ten most common working patterns; "time" plots the Flexibility Index for the group over time. |
| <code>start_hour</code> | See <code>flex_index()</code> . |
| <code>end_hour</code> | See <code>flex_index()</code> . |

Value

ggplot object. See method.

See Also

Other Working Patterns: [flex_index\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [workpatterns_area\(\)](#), [workpatterns_classify_bw\(\)](#), [workpatterns_classify_pav\(\)](#), [workpatterns_classify\(\)](#), [workpatterns_hclust\(\)](#), [workpatterns_rank\(\)](#), [workpatterns_report\(\)](#)

Examples

```
# Pre-calculate Flexibility Index
fi_output <- flex_index(em_data, return = "data")

# Examples of how to test the plotting options individually
# Sample of 10 work patterns
plot_flex_index(fi_output, method = "sample")

# 10 most common work patterns
plot_flex_index(fi_output, method = "common")

# Plot Flexibility Index over time
plot_flex_index(fi_output, method = "time")
```

plot_WOE

Plot WOE graphs with an IV object

Description

Internal function within `create_IV()` that plots WOE graphs using an IV object. Can also be used for plotting individual WOE graphs.

Usage

```
plot_WOE(IV, predictor)
```

Arguments

| | |
|-----------|---|
| IV | IV object created with 'Information'. |
| predictor | String with the name of the predictor variable. |

Value

'ggplot' object. Bar plot with 'WOE' as the y-axis and bins of the predictor variable as the horizontal axis.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Other Variable Association: [IV_by_period\(\)](#), [IV_report\(\)](#), [create_IV\(\)](#)

Other Information Value: [IV_by_period\(\)](#), [IV_report\(\)](#), [create_IV\(\)](#)

| | |
|--------|---|
| p_test | <i>Calculate the p-value of the null hypothesis that two outcomes are from the same dataset</i> |
|--------|---|

Description

Specify an outcome variable and return p-test outputs. All numeric variables in the dataset are used as predictor variables.

Usage

```
p_test(data, outcome, behavior, paired = FALSE)
```

Arguments

| | |
|----------|--|
| data | A Person Query dataset in the form of a data frame. |
| outcome | A string specifying the name of a binary variable, i.e. can only contain the values 1 or 0. Used to group the two distributions. |
| behavior | A character vector specifying the column to be used as the behavior to test. |
| paired | Specify whether the dataset is paired or not. Defaults to TRUE. |

Details

This function is a wrapper around `wilcox.test()` from 'stats'.

Value

Returns a numeric value representing the p-value outcome of the test.

Author(s)

Mark Powers mark.powers@microsoft.com

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
# Simulate a binary variable X
# Returns a single p-value
library(dplyr)
sq_data %>%
  mutate(X = ifelse(Email_hours > 6, 1, 0)) %>%
  p_test(outcome = "X", behavior = "External_network_size")
```

| | |
|---------------|----------------------|
| read_preamble | <i>Read preamble</i> |
|---------------|----------------------|

Description

Read in a preamble to be used within each individual reporting function. Reads from the Markdown file installed with the package.

Usage

```
read_preamble(path)
```

Arguments

| | |
|------|--|
| path | Text string containing the path for the appropriate Markdown file. |
|------|--|

Value

String containing the text read in from the specified Markdown file.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Other Reports: [IV_report\(\)](#), [capacity_report\(\)](#), [coaching_report\(\)](#), [collaboration_report\(\)](#), [connectivity_report\(\)](#), [generate_report\(\)](#), [meeting_tm_report\(\)](#), [subject_validate_report\(\)](#), [validation_report\(\)](#), [workpatterns_report\(\)](#)

| | |
|-----------------|--|
| remove_outliers | <i>Remove outliers from a person query across time</i> |
|-----------------|--|

Description

[Experimental]

This function takes in a selected metric and uses z-score (number of standard deviations) to identify and remove outlier weeks for individuals across time. There are applications in this for removing weeks with abnormally low collaboration activity, e.g. holidays. Retains metrics with $z > -2$.

Function is based on [identify_outlier\(\)](#), but implements a more elaborate approach as the outliers are identified and removed **with respect to each individual**, as opposed to the group. Note that [remove_outliers\(\)](#) has a longer runtime compared to [identify_outlier\(\)](#).

Usage

```
remove_outliers(data, metric = "Collaboration_hours")
```

Arguments

| | |
|--------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| metric | Character string containing the name of the metric, e.g. "Collaboration_hours" |

Details

For mature functions to remove common outliers, please see the following:

- `identify_holidayweeks()`
- `identify_nkw()`
- `identify_inactiveweeks`

Value

Returns a new data frame, "cleaned_data" with all metrics, having removed the person-weeks that are below 2 standard deviations of each individual's collaboration activity.

Author(s)

Mark Powers mark.powers@microsoft.com

See Also

Other Data Validation: `check_query()`, `extract_hr()`, `flag_ch_ratio()`, `flag_em_ratio()`, `flag_extreme()`, `flag_outlooktime()`, `hr_trend()`, `hrvar_count_all()`, `hrvar_count()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts_wp()`, `identify_shifts()`, `identify_tenure()`, `standardise_pq()`, `subject_validate_report()`, `subject_validate()`, `track_HR_change()`, `validation_report()`

rgb2hex

Convert rgb to HEX code

Description

Convert rgb to HEX code

Usage

```
rgb2hex(r, g, b)
```

Arguments

| | |
|---------|--|
| r, g, b | Values that correspond to the three RGB parameters |
|---------|--|

Value

Returns a string containing a HEX code.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

 sq_data

Sample Standard Person Query dataset

Description

A dataset generated from a Standard Person Query from Workplace Analytics.

Usage

sq_data

Format

A data frame with 13442 rows and 66 variables:

PersonId

Date

Workweek_span Time between the person's first sent email or meeting attended and the last email or meeting for each day of the work week.

Meetings_with_skip_level

Meeting_hours_with_skip_level

Generated_workload_email_hours

Generated_workload_email_recipients

Generated_workload_instant_messages_hours

Generated_workload_instant_messages_recipients

Generated_workload_call_hours

Generated_workload_call_participants

Generated_workload_calls_organized

External_network_size

Internal_network_size

Networking_outside_company

Networking_outside_organization

After_hours_meeting_hours
Open_1_hour_block
Open_2_hour_blocks
Total_focus_hours
Low_quality_meeting_hours
Total_emails_sent_during_meeting
Meetings
Meeting_hours
Conflicting_meeting_hours
Multitasking_meeting_hours
Redundant_meeting_hours_lower_level_
Redundant_meeting_hours_organizational_
Time_in_self_organized_meetings
Meeting_hours_during_working_hours
Generated_workload_meeting_attendees
Generated_workload_meeting_hours
Generated_workload_meetings_organized
Manager_coaching_hours_1_on_1
Meetings_with_manager
Meeting_hours_with_manager
Meetings_with_manager_1_on_1
Meeting_hours_with_manager_1_on_1
After_hours_email_hours
Emails_sent
Email_hours Number of hours the person spent sending and receiving emails.
Working_hours_email_hours
After_hours_instant_messages
Instant_messages_sent
Instant_Message_hours
Working_hours_instant_messages
After_hours_collaboration_hours
Collaboration_hours
Collaboration_hours_external
Working_hours_collaboration_hours
After_hours_in_calls
Total_calls
Call_hours

Working_hours_in_calls
Domain
FunctionType
LevelDesignation
Layer
Region
Organization
zId
attainment
TimeZone
HourlyRate
IsInternal
IsActive ...

Value

data frame.

Source

<https://workplaceanalytics-demo.office.com/en-us/Home>

See Also

Other Data: [dv_data](#), [em_data](#), [g2g_data](#), [mt_data](#), [p2p_data_sim\(\)](#)

standardise_pq

Standardise variable names to a Standard Person Query

Description

This function standardises the variable names to a Standard Person Query, where the standard use case is to pass a Ways of Working Assessment Query to the function.

Usage

```
standardise_pq(data)
```

```
standardize_pq(data)
```

Arguments

data A Ways of Working Assessment query to pass through as a data frame.

Details

The following standardisation steps are taken:

- Collaboration_hrs -> Collaboration_hours
- Instant_message_hours -> Instant_Message_hours

Value

data frame containing the formatted query passed to the function.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Other Import and Export: [copy_df\(\)](#), [create_dt\(\)](#), [export\(\)](#), [import_to_fst\(\)](#), [import_wpa\(\)](#)

| | |
|------------------|--|
| subject_classify | <i>Create a new logical variable that classifies meetings by patterns in subject lines</i> |
|------------------|--|

Description

Take a meeting query with subject lines and create a new TRUE/FALSE column which classifies meetings by a provided set of patterns in the subject lines.

Usage

```
subject_classify(  
  data,  
  var_name = "class",  
  keywords = NULL,  
  pattern = NULL,  
  ignore_case = FALSE,  
  return = "data"  
)
```

Arguments

| | |
|----------|---|
| data | A Meeting Query dataset in the form of a data frame. |
| var_name | String containing the name of the new column to be created. |
| keywords | Character vector containing the keywords to match. |

| | |
|-------------|---|
| pattern | String to use for regular expression matching instead of keywords. When both keywords and pattern are supplied, pattern takes priority and is used instead. |
| ignore_case | Logical value to determine whether to ignore case when performing pattern matching. |
| return | String specifying what output to return. |

Examples

```
class_df <-
  mt_data %>%
  subject_classify(
    var_name = "IsSales",
    keywords = c("sales", "marketing")
  )

class_df %>% dplyr::count(IsSales)

# Return a table directly
mt_data %>% subject_classify(pattern = "annual", return = "table")
```

| | |
|--------------|---|
| subject_scan | <i>Count top words in subject lines grouped by a custom attribute</i> |
|--------------|---|

Description

[Experimental]

This function generates a matrix of the top occurring words in meetings, grouped by a specified attribute such as organisational attribute, day of the week, or hours of the day.

Usage

```
subject_scan(
  data,
  hrvar,
  mode = NULL,
  top_n = 10,
  token = "words",
  return = "plot",
  weight = NULL,
  stopwords = NULL,
  ...
)

tm_scan(
  data,
  hrvar,
```

```

mode = NULL,
top_n = 10,
token = "words",
return = "plot",
weight = NULL,
stopwords = NULL,
...
)

```

Arguments

| | |
|-----------|---|
| data | A Meeting Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Note that the prefix 'Organizer_' or equivalent will be required. |
| mode | String specifying what variable to use for grouping subject words. Valid values include: <ul style="list-style-type: none"> • "hours" • "days" • NULL (defaults to hrvar) When the value passed to mode is not NULL, the value passed to hrvar will be discarded and instead be over-written by setting specified in mode. |
| top_n | Numeric value specifying the top number of words to show. |
| token | A character vector accepting either "words" or "ngrams", determining type of tokenisation to return. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" • "data" <p>See Value for more information.</p> |
| weight | String specifying the column name of a numeric variable for weighting data, such as "Invitees". The column must contain positive integers. Defaults to NULL, where no weighting is applied. |
| stopwords | A single-column data frame labelled 'word' containing custom stopwords to remove. |
| ... | Additional parameters to pass to <code>tm_clean()</code> . |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A heatmapped grid.
- "table": data frame. A summary table for the metric.
- "data": data frame.

Examples

```

# return a heatmap table for words
mt_data %>% subject_scan(hrvar = "Organizer_Organization")

# return a heatmap table for ngrams
mt_data %>%
  subject_scan(
    hrvar = "Organizer_Organization",
    token = "ngrams",
    n = 2)

# return raw table format
mt_data %>% subject_scan(hrvar = "Organizer_Organization", return = "table")

# grouped by hours
mt_data %>% subject_scan(mode = "hours")

# grouped by days
mt_data %>% subject_scan(mode = "days")

```

| | |
|------------------|--|
| subject_validate | <i>Scan meeting subject and highlight items for review</i> |
|------------------|--|

Description

This functions scans a meeting query and highlights meetings with subjects that include common exclusion terms. It is intended to be used by an analyst to validate raw data before conducting additional analysis. Returns a summary in the console by default. Additional option to return the underlying data with a flag of items for review.

Usage

```
subject_validate(data, return = "text")
```

Arguments

| | |
|--------|---|
| data | A meeting query in the form of a data frame. |
| return | A string specifying what to return. Returns a message in the console by default, where 'text' is passed in return. When 'table' is passed, a summary table with common terms found is printed. When 'data' is passed, a the original data with an additional flag column is returned as a data frame. |

Value

Returns a message in the console by default, where 'text' is passed in return. When 'table' is passed, a summary table with common terms found is printed. When 'data' is passed, a the original data with an additional flag column is returned as a data frame.

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [track_HR_change\(\)](#), [validation_report\(\)](#)

Other Text-mining: [meeting_tm_report\(\)](#), [pairwise_count\(\)](#), [subject_validate_report\(\)](#), [tm_clean\(\)](#), [tm_cooc\(\)](#), [tm_freq\(\)](#), [tm_wordcloud\(\)](#)

subject_validate_report

Generate Meeting Text Mining report in HTML for Common Exclusion Terms

Description

This functions creates a text mining report in HTML based on Meeting Subject Lines for data validation. It scans a meeting query and highlights meetings with subjects that include common exclusion terms. It is intended to be used by an analyst to validate raw data before conducting additional analysis. Returns a HTML report by default.

Usage

```
subject_validate_report(
  data,
  path = "Subject Lines Validation Report",
  timestamp = TRUE,
  keep = 100,
  seed = 100
)
```

Arguments

| | |
|-----------|--|
| data | A Meeting Query dataset in the form of a data frame. |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . For example, "meeting text mining report". |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |
| keep | A numeric vector specifying maximum number of words to keep. |
| seed | A numeric vector to set seed for random generation. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

See Also

Other Data Validation: `check_query()`, `extract_hr()`, `flag_ch_ratio()`, `flag_em_ratio()`, `flag_extreme()`, `flag_outlooktime()`, `hr_trend()`, `hrvar_count_all()`, `hrvar_count()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts_wp()`, `identify_shifts()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate()`, `track_HR_change()`, `validation_report()`

Other Text-mining: `meeting_tm_report()`, `pairwise_count()`, `subject_validate()`, `tm_clean()`, `tm_cooc()`, `tm_freq()`, `tm_wordcloud()`

Other Reports: `IV_report()`, `capacity_report()`, `coaching_report()`, `collaboration_report()`, `connectivity_report()`, `generate_report()`, `meeting_tm_report()`, `read_preamble()`, `validation_report()`, `workpatterns_report()`

 theme_wpa

Main theme for 'wpa' visualisations

Description

A theme function applied to 'ggplot' visualisations in 'wpa'. Install and load 'extrafont' to use custom fonts for plotting.

Usage

```
theme_wpa(font_size = 12, font_family = "Segoe UI")
```

Arguments

| | |
|-------------|---|
| font_size | Numeric value that prescribes the base font size for the plot. The text elements are defined relatively to this base font size. Defaults to 12. |
| font_family | Character value specifying the font family to be used in the plot. The default value is "Segoe UI". To ensure you can use this font, install and load extrafont prior to plotting. There is an initialisation process that is described by: https://stackoverflow.com/questions/34522732/changing-fonts-in-ggplot2 |

Value

Returns a ggplot object with the applied theme.

See Also

Other Themes: `theme_wpa_basic()`

| | |
|-----------------|---|
| theme_wpa_basic | <i>Basic theme for 'wpa' visualisations</i> |
|-----------------|---|

Description

A theme function applied to 'ggplot' visualisations in 'wpa'. Based on theme_wpa() but has no font requirements.

Usage

```
theme_wpa_basic(font_size = 12)
```

Arguments

| | |
|-----------|---|
| font_size | Numeric value that prescribes the base font size for the plot. The text elements are defined relatively to this base font size. Defaults to 12. |
|-----------|---|

Value

Returns a ggplot object with the applied theme.

See Also

Other Themes: [theme_wpa\(\)](#)

| | |
|----------|--|
| tm_clean | <i>Clean subject line text prior to analysis</i> |
|----------|--|

Description

This function processes the Subject column in a Meeting Query by applying tokenisation using tidytext::unnest_tokens(), and removing any stopwords supplied in a data frame (using the argument stopwords). This is a sub-function that feeds into tm_freq(), tm_cooc(), and tm_wordcloud(). The default is to return a data frame with tokenised counts of words or ngrams.

Usage

```
tm_clean(data, token = "words", stopwords = NULL, ...)
```

Arguments

| | |
|-----------|--|
| data | A Meeting Query dataset in the form of a data frame. |
| token | A character vector accepting either "words" or "ngrams", determining type of tokenisation to return. |
| stopwords | A single-column data frame labelled 'word' containing custom stopwords to remove. |
| ... | Additional parameters to pass to tidytext::unnest_tokens(). |

Value

data frame with two columns:

- line
- word

See Also

Other Text-mining: [meeting_tm_report\(\)](#), [pairwise_count\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [tm_cooc\(\)](#), [tm_freq\(\)](#), [tm_wordcloud\(\)](#)

Examples

```
# words
tm_clean(mt_data)

# ngrams
tm_clean(mt_data, token = "ngrams")
```

tm_cooc

Analyse word co-occurrence in subject lines and return a network plot

Description

This function generates a word co-occurrence network plot, with options to return a table. This function is used within `meeting_tm_report()`.

Usage

```
tm_cooc(data, stopwords = NULL, seed = 100, return = "plot", lmult = 0.05)
```

Arguments

| | |
|-----------|--|
| data | A Meeting Query dataset in the form of a data frame. |
| stopwords | A single-column data frame labelled 'word' containing custom stopwords to remove. |
| seed | A numeric vector to set seed for random generation. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |
| lmult | A multiplier to adjust the line width in the output plot. Defaults to 0.05. |

Details

This function uses `tm_clean()` as the underlying data wrangling function. There is an option to remove stopwords by passing a data frame into the `stopwords` argument.

Value

A different output is returned depending on the value passed to the `return` argument:

- "plot": 'ggplot' and 'ggraph' object. A network plot.
- "table": data frame. A summary table.

Author(s)

Carlos Morales carlos.morales@microsoft.com

See Also

Other Text-mining: [meeting_tm_report\(\)](#), [pairwise_count\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [tm_clean\(\)](#), [tm_freq\(\)](#), [tm_wordcloud\(\)](#)

Examples

```
# Demo using a subset of `mt_data`  
mt_data %>%  
  dplyr::slice(1:20) %>%  
  tm_cooc(lmult = 0.01)
```

tm_freq

Perform a Word or Ngram Frequency Analysis and return a Circular Bar Plot

Description

Generate a circular bar plot with frequency of words / ngrams. This function is used within `meeting_tm_report()`.

Usage

```
tm_freq(data, token = "words", stopwords = NULL, keep = 100, return = "plot")
```

Arguments

| | |
|-----------|--|
| data | A Meeting Query dataset in the form of a data frame. |
| token | A character vector accepting either "words" or "ngram", determining type of tokenisation to return. |
| stopwords | A single-column data frame labelled 'word' containing custom stopwords to remove. |
| keep | A numeric vector specifying maximum number of words to keep. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Details

This function uses `tm_clean()` as the underlying data wrangling function. There is an option to remove stopwords by passing a data frame into the `stopwords` argument.

Value

A different output is returned depending on the value passed to the `return` argument:

- "plot": 'ggplot' object. A circular bar plot.
- "table": data frame. A summary table.

See Also

Other Text-mining: [meeting_tm_report\(\)](#), [pairwise_count\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [tm_clean\(\)](#), [tm_cooc\(\)](#), [tm_wordcloud\(\)](#)

Examples

```
tm_freq(mt_data, token = "words")
tm_freq(mt_data, token = "ngrams")
```

tm_wordcloud

Generate a wordcloud with meeting subject lines

Description

Generate a wordcloud with the meeting query. This is a sub-function that feeds into `meeting_tm_report()`.

Usage

```
tm_wordcloud(
  data,
  stopwords = NULL,
  seed = 100,
  keep = 100,
  return = "plot",
  ...
)
```

Arguments

| | |
|-----------|--|
| data | A Meeting Query dataset in the form of a data frame. |
| stopwords | A single-column data frame labelled 'word' containing custom stopwords to remove. |
| seed | A numeric vector to set seed for random generation. |
| keep | A numeric vector specifying maximum number of words to keep. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |
| ... | Additional parameters to be passed to <code>ggwordcloud::geom_text_wordcloud()</code> |

Details

Uses the 'ggwordcloud' package for the underlying implementation, thus returning a 'ggplot' object. Additional layers can be added onto the plot using a ggplot + syntax. The recommendation is not to return over 100 words in a word cloud.

This function uses `tm_clean()` as the underlying data wrangling function. There is an option to remove stopwords by passing a data frame into the `stopwords` argument.

Value

A different output is returned depending on the value passed to the `return` argument:

- "plot": 'ggplot' object containing a word cloud.
- "table": data frame returning the data used to generate the word cloud.

See Also

Other Text-mining: [meeting_tm_report\(\)](#), [pairwise_count\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [tm_clean\(\)](#), [tm_cooc\(\)](#), [tm_freq\(\)](#)

Examples

```
tm_wordcloud(mt_data, keep = 30)
```

| | |
|-------------|--|
| totals_bind | <i>Row-bind an identical data frame for computing grouped totals</i> |
|-------------|--|

Description

Row-bind an identical data frame and impute a specific column with the `target_value`, which defaults as "Total". The purpose of this is to enable to creation of summary tables with a calculated "Total" row. See example below on usage.

Usage

```
totals_bind(data, target_col, target_value = "Total")
```

Arguments

| | |
|---------------------------|---|
| <code>data</code> | data frame |
| <code>target_col</code> | Character value of the column in which to impute "Total". This is usually the intended grouping column. |
| <code>target_value</code> | Character value to impute in the new data frame to row-bind. Defaults to "Total". |

Value

data frame with twice the number of rows of the input data frame, where half of those rows will have the `target_col` column imputed with the value from `target_value`.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
sq_data %>%
  totals_bind(target_col = "LevelDesignation", target_value = "Total") %>%
  collab_sum(hrvar = "LevelDesignation", return = "table")
```

| | |
|------------|--|
| totals_col | <i>Fabricate a 'Total' HR variable</i> |
|------------|--|

Description

Create a 'Total' column of character type comprising exactly of one unique value. This is a convenience function for returning a no-HR attribute view when NULL is supplied to the hrvar argument in functions.

Usage

```
totals_col(data, total_value = "Total")
```

Arguments

| | |
|-------------|---|
| data | data frame |
| total_value | Character value defining the name and the value of the "Total" column. Defaults to "Total". An error is returned if an existing variable has the same name as the supplied value. |

Value

data frame containing an additional 'Total' column on top of the input data frame.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
# Create a visual without HR attribute breaks
sq_data %>%
  totals_col() %>%
  collab_fizz(hrvar = "Total")
```

| | |
|----------------|--|
| totals_reorder | <i>Reorder a value to the top of the summary table</i> |
|----------------|--|

Description

For a given data frame, reorder a row to the first row of that data frame through matching a *value* of a *variable*. The intended usage of this function is to be used for reordering the "Total" row, and *not* with "flat" data. This can be used in conjunction with `totals_bind()`, which is used to create a "Total" row in the data.

Usage

```
totals_reorder(data, target_col, target_value = "Total")
```

Arguments

| | |
|---------------------------|--|
| <code>data</code> | Summary table in the form of a data frame. |
| <code>target_col</code> | Character value of the column in which to reorder |
| <code>target_value</code> | Character value of the value in <code>target_col</code> to match |

Value

data frame with the 'Total' row reordered to the bottom.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

Examples

```
sq_data %>%
  totals_bind(target_col = "LevelDesignation",
             target_value = "Total") %>%
  collab_sum(hrvar = "LevelDesignation",
            return = "table") %>%
  totals_reorder(target_col = "group", target_value = "Total")
```

| | |
|-----------------|--|
| track_HR_change | <i>Sankey chart of organizational movement between HR attributes and missing values (outside company move) (Data Overview)</i> |
|-----------------|--|

Description

Creates a list of everyone at a specified start date and a specified end date then aggregates up people who have moved between organizations between this to points of time and visualizes the move through a sankey chart.

Through this chart you can see:

- The HR attribute/orgs that have the highest move out
- The HR attribute/orgs that have the highest move in
- The number of people that do not have that HR attribute or if they are no longer in the system

Usage

```
track_HR_change(
  data,
  start_date = min(data$Date),
  end_date = max(data$Date),
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  NA_replacement = "Out of Company"
)
```

Arguments

| | |
|----------------|---|
| data | A Person Query dataset in the form of a data frame. |
| start_date | A start date to compare changes. See end_date. |
| end_date | An end date to compare changes. See start_date. |
| hrvar | HR Variable by which to compare changes between, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table". |
| NA_replacement | Character replacement for NA defaults to "out of company" |

Value

Returns a 'NetworkD3' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

Author(s)

Tannaz Sattari Tabrizi Tannaz.Sattari@microsoft.com

See Also

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [validation_report\(\)](#)

Examples

```
dv_data %>% track_HR_change()
```

tstamp

Generate a time stamp

Description

This function generates a time stamp of the format 'yymmdd_hhmmss'. This is a support function and is not intended for direct use.

Usage

```
tstamp()
```

Value

String containing the timestamp in the format 'yymmdd_hhmmss'.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [us_to_space\(\)](#), [wrap\(\)](#)

| | |
|-------------|--------------------------------------|
| us_to_space | <i>Replace underscore with space</i> |
|-------------|--------------------------------------|

Description

Convenience function to convert underscores to space

Usage

```
us_to_space(x)
```

Arguments

x String to replace all occurrences of _ with a single space

Value

Character vector containing the modified string.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [wrap\(\)](#)

Examples

```
us_to_space("Meeting_hours_with_manager_1_on_1")
```

| | |
|-------------------|--|
| validation_report | <i>Generate a Data Validation report in HTML</i> |
|-------------------|--|

Description

The function generates an interactive HTML report using Standard Person Query data as an input. The report contains checks on Workplace Analytics query outputs to provide diagnostic information for the Analyst prior to analysis.

An additional Standard Meeting Query can be provided to perform meeting subject line related checks. This is optional and the validation report can be run without it.

Usage

```
validation_report(
  data,
  meeting_data = NULL,
  hrvar = "Organization",
  path = "validation report",
  hrvar_threshold = 150,
  timestamp = TRUE
)
```

Arguments

| | |
|-----------------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| meeting_data | An optional Meeting Query dataset in the form of a data frame. |
| hrvar | HR Variable by which to split metrics, defaults to "Organization" but accepts any character vector, e.g. "Organization" |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . |
| hrvar_threshold | Numeric value determining the maximum number of unique values to be allowed to qualify as a HR variable. This is passed directly to the threshold argument within <code>hrvar_count_all()</code> . |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |

Details

For your input to `data` or `meeting_data`, please use the function `wpa::import_wpa()` to import your csv query files into R. This function will standardize format and prepare the data as input for this report.

If you are passing a Ways of Working Assessment query instead of a Standard Person query to the `data` argument, please also use `standardise_pq()` to make the variable names consistent with a Standard Person Query.

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

Checking functions within `validation_report()`

- `check_query()`
- `flag_ch_ratio()`
- `hrvar_count_all()`
- `identify_privacythreshold()`
- `identify_nkw()`
- `identify_holidayweeks()`

- [subject_validate\(\)](#)
- [identify_tenure\(\)](#)
- [flag_outlooktime\(\)](#)
- [identify_shifts\(\)](#)
- [track_HR_change\(\)](#)

You can browse each individual function for details on calculations.

Creating a report

Below is an example on how to run the report.

```
validation_report(dv_data,
                  meeting_data = mt_data,
                  hrvar = "Organization")
```

See Also

Other Reports: [IV_report\(\)](#), [capacity_report\(\)](#), [coaching_report\(\)](#), [collaboration_report\(\)](#), [connectivity_report\(\)](#), [generate_report\(\)](#), [meeting_tm_report\(\)](#), [read_preamble\(\)](#), [subject_validate_report\(\)](#), [workpatterns_report\(\)](#)

Other Data Validation: [check_query\(\)](#), [extract_hr\(\)](#), [flag_ch_ratio\(\)](#), [flag_em_ratio\(\)](#), [flag_extreme\(\)](#), [flag_outlooktime\(\)](#), [hr_trend\(\)](#), [hrvar_count_all\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [identify_churn\(\)](#), [identify_holidayweeks\(\)](#), [identify_inactiveweeks\(\)](#), [identify_nkw\(\)](#), [identify_outlier\(\)](#), [identify_privacythreshold\(\)](#), [identify_query\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [identify_tenure\(\)](#), [remove_outliers\(\)](#), [standardise_pq\(\)](#), [subject_validate_report\(\)](#), [subject_validate\(\)](#), [track_HR_change\(\)](#)

| | |
|------------------|--|
| wellbeing_report | <i>Generate a Wellbeing Report in HTML</i> |
|------------------|--|

Description

[Experimental]

Generate a static HTML report on wellbeing, taking a custom Wellbeing Query and an Hourly Collaboration query as inputs. See Required metrics section for more details on the required inputs for the Wellbeing Query. Note that this function is currently still in experimental/development stage and may experience changes in the near term.

Usage

```
wellbeing_report(
  wbq,
  hcq,
  hrvar = "Organization",
  mingroup = 5,
```

```

start_hour = "0900",
end_hour = "1700",
path = "wellbeing_report"
)

```

Arguments

| | |
|------------|--|
| wbq | Data frame. A custom Wellbeing Query dataset based on the Person Query. If certain metrics are missing from the Wellbeing / Person Query, the relevant visual will show up with an indicative message. |
| hcq | Data frame. An Hourly Collaboration Query dataset. |
| hrvar | String specifying HR attribute to cut by archetypes. Defaults to Organization. |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| start_hour | A character vector specifying starting hours, e.g. "0900". Note that this currently only supports hourly increments. If the official hours specifying checking in and 9 AM and checking out at 5 PM, then "0900" should be supplied here. |
| end_hour | A character vector specifying starting hours, e.g. "1700". Note that this currently only supports hourly increments. If the official hours specifying checking in and 9 AM and checking out at 5 PM, then "1700" should be supplied here. |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . Defaults to "wellbeing_report". |

Required metrics

A full list of the required metrics are as follows:

- Urgent_meeting_hours
- IMs_sent_other_level
- IMs_sent_same_level
- Emails_sent_other_level
- Emails_sent_same_level
- Emails_sent
- IMs_sent
- Meeting_hours_intimate_group
- Meeting_hours_1on1
- Urgent_email_hours
- Unscheduled_call_hours
- Meeting_hours
- Instant_Message_hours
- Email_hours
- Total_focus_hours
- Weekend_IMs_sent

- Weekend_emails_sent
- After_hours_collaboration_hours
- After_hours_meeting_hours
- After_hours_instant_messages
- After_hours_in_unscheduled_calls
- After_hours_email_hours
- Collaboration_hours
- Workweek_span

| | |
|----------------|---|
| workloads_dist | <i>Distribution of Work Week Span as a 100% stacked bar</i> |
|----------------|---|

Description

Analyze Work Week Span distribution. Returns a stacked bar plot by default. Additional options available to return a table with distribution elements.

Usage

```
workloads_dist(
  data,
  hrvar = "Organization",
  mingroup = 5,
  return = "plot",
  cut = c(15, 30, 45)
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |
| cut | A numeric vector of length three to specify the breaks for the distribution, e.g. c(10, 15, 20) |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A stacked bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Workweek Span: [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#)

Examples

```
# Return plot
workloads_dist(sq_data, hrvar = "Organization", return = "plot")

# Return a summary table
workloads_dist(sq_data, hrvar = "Organization", return = "table")
```

workloads_fizz

Distribution of Work Week Span (Fizzy Drink plot)

Description

Analyze Work Week Span distribution, and returns a 'fizzy' scatter plot by default. Additional options available to return a table with distribution elements.

Usage

```
workloads_fizz(data, hrvar = "Organization", mingroup = 5, return = "plot")
```


Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A jittered scatter plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Workweek Span: [workloads_dist\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#)

Examples

```
# Return plot
workloads_fizz(sq_data, hrvar = "Organization", return = "plot")

# Return summary table
workloads_fizz(sq_data, hrvar = "Organization", return = "table")
```

workloads_line

*Workloads Time Trend - Line Chart***Description**

Provides a week by week view of 'Work Week Span', visualised as line charts. By default returns a line chart for collaboration hours, with a separate panel per value in the HR attribute. Additional options available to return a summary table.

Usage

```
workloads_line(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A faceted line plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#),

[one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Workweek Span: [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#)

Examples

```
# Return a line plot
workloads_line(sq_data, hrvar = "LevelDesignation")

# Return summary table
workloads_line(sq_data, hrvar = "LevelDesignation", return = "table")
```

| | |
|----------------|--|
| workloads_rank | <i>Rank all groups across HR attributes for Work Week Span</i> |
|----------------|--|

Description

This function scans a standard query output for groups with high levels of Work Week Span. Returns a table with a all of groups (across multiple HR attributes) ranked by work week span.

Usage

```
workloads_rank(
  data,
  hrvar = extract_hr(data),
  mingroup = 5,
  mode = "simple",
  plot_mode = 1,
  return = "table"
)
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| mode | String to specify calculation mode. Must be either: <ul style="list-style-type: none"> • "simple" • "combine" |

| | |
|-----------|---|
| plot_mode | Numeric vector to determine which plot mode to return. Must be either 1 or 2, and is only used when return = "plot". <ul style="list-style-type: none"> • 1: Top and bottom five groups across the data population are highlighted • 2: Top and bottom groups <i>per</i> organizational attribute are highlighted |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "table" (default) • "plot" See Value for more information. |

Details

Uses the metric `Workweek_span`. See `create_rank()` for applying the same analysis to a different metric.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bubble plot where the x-axis represents the metric, the y-axis represents the HR attributes, and the size of the bubbles represent the size of the organizations. Note that there is no plot output if mode is set to "combine".
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Workweek Span: `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_summary()`, `workloads_trend()`

Examples

```
# Return rank table
workloads_rank(
  data = sq_data,
  return = "table"
```

```
)  
  
# Return plot  
workloads_rank(  
  data = sq_data,  
  return = "plot"  
)
```

| | |
|-------------------|-------------------------------|
| workloads_summary | <i>Work Week Span Summary</i> |
|-------------------|-------------------------------|

Description

Provides an overview analysis of 'Work Week Span'. Returns a bar plot showing average weekly utilization hours by default. Additional options available to return a summary table.

Usage

```
workloads_summary(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

```
workloads_sum(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|----------|--|
| data | A Standard Person Query dataset in the form of a data frame. |
| hrvar | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": 'ggplot' object. A bar plot for the metric.
- "table": data frame. A summary table for the metric.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_trend()`, `workpatterns_area()`, `workpatterns_rank()`

Other Workweek Span: `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_trend()`

Examples

```
# Return a ggplot bar chart
workloads_summary(sq_data, hrvar = "LevelDesignation")

# Return a summary table
workloads_summary(sq_data, hrvar = "LevelDesignation", return = "table")
```

workloads_trend

Work Week Span Time Trend

Description

Provides a week by week view of Work Week Span. By default returns a week by week heatmap, highlighting the points in time with most activity. Additional options available to return a summary table.

Usage

```
workloads_trend(data, hrvar = "Organization", mingroup = 5, return = "plot")
```

Arguments

| | |
|-----------------------|--|
| <code>data</code> | A Standard Person Query dataset in the form of a data frame. |
| <code>hrvar</code> | String containing the name of the HR Variable by which to split metrics. Defaults to "Organization". To run the analysis on the total instead of splitting by an HR attribute, supply NULL (without quotes). |
| <code>mingroup</code> | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |

return Character vector specifying what to return, defaults to "plot". Valid inputs are "plot" and "table".

Details

Uses the metric `Workweek_span`.

Value

Returns a 'ggplot' object by default, where 'plot' is passed in return. When 'table' is passed, a summary table is returned as a data frame.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workpatterns_area\(\)](#), [workpatterns_rank\(\)](#)

Other Workweek Span: [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#)

Examples

```
# Run plot
workloads_trend(sq_data)

# Run table
workloads_trend(sq_data, hrvar = "LevelDesignation", return = "table")
```

workpatterns_area *Create an area plot of emails and IMs by hour of the day*

Description

Uses the Hourly Collaboration query to produce an area plot of Emails sent and IMs sent attended by hour of the day.

Usage

```
workpatterns_area(
  data,
  hrvar = "Organization",
  mingroup = 5,
  signals = c("email", "IM"),
  return = "plot",
  values = "percent",
  start_hour = "0900",
  end_hour = "1700"
)
```

Arguments

| | |
|------------|---|
| data | A data frame containing data from the Hourly Collaboration query. |
| hrvar | HR Variable by which to split metrics. Accepts a character vector, defaults to "Organization" but accepts any character vector, e.g. "LevelDesignation" |
| mingroup | Numeric value setting the privacy threshold / minimum group size, defaults to 5. |
| signals | Character vector to specify which collaboration metrics to use: <ul style="list-style-type: none"> • a combination of signals, such as c("email", "IM") (default) • "email" for emails only • "IM" for Teams messages only • "unscheduled_calls" for Unscheduled Calls only • "meetings" for Meetings only |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "table" <p>See Value for more information.</p> |
| values | Character vector to specify whether to return percentages or absolute values in "data" and "plot". Valid values are: <ul style="list-style-type: none"> • "percent": percentage of signals divided by total signals (default) • "abs": absolute count of signals |
| start_hour | A character vector specifying starting hours, e.g. "0900" |
| end_hour | A character vector specifying starting hours, e.g. "1700" |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. An overlapping area plot (default).
- "table": data frame. A summary table.

See Also

Other Visualization: [afterhours_dist\(\)](#), [afterhours_fizz\(\)](#), [afterhours_line\(\)](#), [afterhours_rank\(\)](#), [afterhours_summary\(\)](#), [afterhours_trend\(\)](#), [collaboration_area\(\)](#), [collaboration_dist\(\)](#), [collaboration_fizz\(\)](#), [collaboration_line\(\)](#), [collaboration_rank\(\)](#), [collaboration_sum\(\)](#), [collaboration_trend\(\)](#), [create_bar_asis\(\)](#), [create_bar\(\)](#), [create_boxplot\(\)](#), [create_bubble\(\)](#), [create_dist\(\)](#), [create_fizz\(\)](#), [create_line_asis\(\)](#), [create_line\(\)](#), [create_period_scatter\(\)](#), [create_rank\(\)](#), [create_sankey\(\)](#), [create_scatter\(\)](#), [create_stacked\(\)](#), [create_tracking\(\)](#), [create_trend\(\)](#), [email_dist\(\)](#), [email_fizz\(\)](#), [email_line\(\)](#), [email_rank\(\)](#), [email_summary\(\)](#), [email_trend\(\)](#), [external_network_plot\(\)](#), [hr_trend\(\)](#), [hrvar_count\(\)](#), [hrvar_trend\(\)](#), [internal_network_plot\(\)](#), [keymetrics_scan\(\)](#), [meeting_dist\(\)](#), [meeting_fizz\(\)](#), [meeting_line\(\)](#), [meeting_quality\(\)](#), [meeting_rank\(\)](#), [meeting_summary\(\)](#), [meeting_trend\(\)](#), [meetingtype_dist_ca\(\)](#), [meetingtype_dist_mt\(\)](#), [meetingtype_dist\(\)](#), [meetingtype_summary\(\)](#), [mgrcoatt_dist\(\)](#), [mgrrel_matrix\(\)](#), [one2one_dist\(\)](#), [one2one_fizz\(\)](#), [one2one_freq\(\)](#), [one2one_line\(\)](#), [one2one_rank\(\)](#), [one2one_sum\(\)](#), [one2one_trend\(\)](#), [period_change\(\)](#), [workloads_dist\(\)](#), [workloads_fizz\(\)](#), [workloads_line\(\)](#), [workloads_rank\(\)](#), [workloads_summary\(\)](#), [workloads_trend\(\)](#), [workpatterns_rank\(\)](#)

Other Working Patterns: [flex_index\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [plot_flex_index\(\)](#), [workpatterns_classify_bw\(\)](#), [workpatterns_classify_pav\(\)](#), [workpatterns_classify\(\)](#), [workpatterns_hclust\(\)](#), [workpatterns_rank\(\)](#), [workpatterns_report\(\)](#)

Other Working Patterns: [flex_index\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [plot_flex_index\(\)](#), [workpatterns_classify_bw\(\)](#), [workpatterns_classify_pav\(\)](#), [workpatterns_classify\(\)](#), [workpatterns_hclust\(\)](#), [workpatterns_rank\(\)](#), [workpatterns_report\(\)](#)

Examples

```
# Return visualization of percentage distribution
workpatterns_area(em_data, return = "plot", values = "percent")

# Return visualization of absolute values
workpatterns_area(em_data, return = "plot", values = "abs")

# Return summary table
workpatterns_area(em_data, return = "table")
```

`workpatterns_classify` *Classify working pattern personas using a rule based algorithm*

Description**[Experimental]**

Apply a rule based algorithm to emails or instant messages sent by hour of day. Uses a binary week-based ('bw') method by default, with options to use the the person-average volume-based ('pav') method.

Usage

```
workpatterns_classify(
  data,
  hrvar = "Organization",
  values = "percent",
  signals = c("email", "IM"),
  start_hour = "0900",
  end_hour = "1700",
  mingroup = 5,
  active_threshold = 0,
  method = "bw",
  return = "plot"
)
```

Arguments

| | |
|------------------|--|
| data | A data frame containing data from the Hourly Collaboration query. |
| hrvar | A string specifying the HR attribute to cut the data by. Defaults to NULL. This only affects the function when "table" is returned, and is only applicable for method = "bw". |
| values | Only valid if using pav method. Character vector to specify whether to return percentages or absolute values in "data" and "plot". Valid values are "percent" (default) and "abs". |
| signals | Character vector to specify which collaboration metrics to use: <ul style="list-style-type: none"> • "email" (default) for emails only • "IM" for Teams messages only • "unscheduled_calls" for Unscheduled Calls only • "meetings" for Meetings only • or a combination of signals, such as c("email", "IM") |
| start_hour | A character vector specifying starting hours, e.g. "0900". Note that this currently only supports hourly increments. If the official hours specifying checking in and 9 AM and checking out at 5 PM, then "0900" should be supplied here. |
| end_hour | A character vector specifying starting hours, e.g. "1700". Note that this currently only supports hourly increments. If the official hours specifying checking in and 9 AM and checking out at 5 PM, then "1700" should be supplied here. |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| active_threshold | A numeric value specifying the minimum number of signals to be greater than in order to qualify as <i>active</i> . Defaults to 0. Only applicable for the binary-week method. |
| method | String to pass through specifying which method to use for classification. By default, a binary week-based (bw) method is used, with options to use the the person-average volume-based (pav) method. |
| return | String specifying what to return. This must be one of the following strings: |

- "plot"
- "data"
- "table"
- "plot-area"
- "plot-hrvar" (only for bw method)
- "plot-dist" (only for bw method)

See Value for more information.

Details

The working patterns archetypes are a set of segments created based on the aggregated hourly activity of employees. A motivation of creating these archetypes is to capture the diversity in working patterns, where for instance employees may choose to take multiple or extended breaks throughout the day, or choose to start or end earlier/later than their standard working hours. Two methods have been developed to capture the different working patterns.

This function is a wrapper around `workpatterns_classify_bw()` and `workpatterns_classify_pav()`, and calls each function depending on what is supplied to the `method` argument. Both methods implement a rule-based classification of either **person-weeks** or **persons** that pull apart different working patterns.

See individual sections below for details on the two different implementations.

Value

Character vector to specify what to return. Valid options include:

- "plot": ggplot object. With the bw method, this returns a grid showing the distribution of archetypes by 'breaks' and number of active hours (default). With the pav method, this returns a faceted bar plot which shows the percentage of signals sent in each hour, with each facet representing an archetype.
- "data": data frame. The raw data with the classified archetypes.
- "table": data frame. A summary table of the archetypes.
- "plot-area": ggplot object. With the bw method, this returns an area plot of the percentages of archetypes shown over time. With the pav method, this returns an area chart which shows the percentage of signals sent in each hour, with each line representing an archetype.
- "plot-hrvar": ggplot object. A bar plot showing the count of archetypes, faceted by the supplied HR attribute. This is only available for the bw method.
- "plot-dist": returns a heatmap plot of signal distribution by hour and archetypes. This is only available for the bw method.

Binary Week method

This method classifies each **person-week** into one of the seven archetypes:

- **0 < 3 hours on**: fewer than 3 hours of active hours
- **1 Standard with breaks workday**: active for fewer than *expected hours*, with no activity outside working hours

- **2 Standard continuous workday:** number of active hours equal *expected hours*, with no activity outside working hours
- **3 Standard flexible workday:** number of active hours are less than or equal to *expected hours*, with some activity outside working hours
- **4 Long flexible workday:** number of active hours exceed *expected hours*, with breaks occurring throughout
- **5 Long continuous workday:** number of active hours exceed *expected hours*, with activity happening in a continuous block (no breaks)
- **6 Always on (13h+):** number of active hours greater than or equal to 13

This is the recommended method over pav for several reasons:

1. bw ignores *volume effects*, where activity volume can still bias the results towards the 'standard working hours'.
2. It captures the intuition that each individual can have 'light' and 'heavy' weeks with respect to workload.

The notion of 'breaks' in the 'binary-week' method is best understood as 'recurring disconnection time'. This denotes an hourly block where there is consistently no activity occurring throughout the week. Note that this applies a stricter criterion compared to the common definition of a break, which is simply a time interval where no active work is being done, and thus the more specific terminology 'recurring disconnection time' is preferred.

In the standard plot output, the archetypes have been abbreviated to show the following:

- **Low Activity** - archetype 0
- **Standard** - archetype 2
- **Flexible** - archetypes 1 and 3
- **Long continuous** - archetype 5
- **Long flexible** - archetype 4
- **Always On** - archetype 6

Person Average method

This method classifies each **person** (based on unique PersonId) into one of the six archetypes:

- **Absent:** Fewer than 10 signals over the week.
- **Extended Hours - Morning:** 15%+ of collaboration before start hours and less than 70% within standard hours, and less than 15% of collaboration after end hours
- **Extended Hours - Evening:** Less than 15% of collaboration before start hours and less than 70% within standard hours, and 15%+ of collaboration after end hours
- **Overnight workers:** less than 30% of collaboration happens within standard hours
- **Standard Hours:** over 70% of collaboration within standard hours
- **Always On:** over 15% of collaboration happens before starting hour and end hour (both conditions must satisfy) and less than 70% of collaboration within standard hours

Flexibility Index

The Working Patterns archetypes as calculated using the binary-week method shares many similarities with the Flexibility Index (see `flex_index()`):

- Both are computed directly from the Hourly Collaboration Flexible Query.
- Both apply the same binary conversion of activity on the signals from the Hourly Collaboration Flexible Query.

Author(s)

Ainize Cidoncha ainize.cidoncha@microsoft.com

Carlos Morales Torrado carlos.morales@microsoft.com

Martin Chan martin.chan@microsoft.com

See Also

Other Clustering: [personas_hclust\(\)](#), [workpatterns_hclust\(\)](#)

Other Working Patterns: [flex_index\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [plot_flex_index\(\)](#), [workpatterns_area\(\)](#), [workpatterns_classify_bw\(\)](#), [workpatterns_classify_pav\(\)](#), [workpatterns_hclust\(\)](#), [workpatterns_rank\(\)](#), [workpatterns_report\(\)](#)

Examples

```
# Returns a plot by default
em_data %>% workpatterns_classify(method = "bw")

# Return an area plot
em_data %>% workpatterns_classify(method = "bw", return = "plot-area")

em_data %>% workpatterns_classify(method = "bw", return = "table")

em_data %>% workpatterns_classify(method = "pav")

em_data %>% workpatterns_classify(method = "pav", return = "plot-area")
```

workpatterns_classify_bw

Classify working pattern week archetypes using a rule-based algorithm, using the binary week-based ('bw') method.

Description**[Experimental]**

Apply a rule based algorithm to emails sent by hour of day, using the binary week-based ('bw') method.

Usage

```
workpatterns_classify_bw(
  data,
  hrvar = NULL,
  signals = c("email", "IM"),
  start_hour = "0900",
  end_hour = "1700",
  mingroup = 5,
  active_threshold = 0,
  return = "plot"
)
```

Arguments

| | |
|------------------|---|
| data | A data frame containing email by hours data. |
| hrvar | A string specifying the HR attribute to cut the data by. Defaults to NULL. This only affects the function when "table" is returned. |
| signals | Character vector to specify which collaboration metrics to use: <ul style="list-style-type: none"> • a combination of signals, such as c("email", "IM") (default) • "email" for emails only • "IM" for Teams messages only • "unscheduled_calls" for Unscheduled Calls only • "meetings" for Meetings only |
| start_hour | A character vector specifying starting hours, e.g. "0900". Note that this currently only supports hourly increments. If the official hours specifying checking in and 9 AM and checking out at 5 PM, then "0900" should be supplied here. |
| end_hour | A character vector specifying starting hours, e.g. "1700". Note that this currently only supports hourly increments. If the official hours specifying checking in and 9 AM and checking out at 5 PM, then "1700" should be supplied here. |
| mingroup | Numeric value setting the privacy threshold / minimum group size. Defaults to 5. |
| active_threshold | A numeric value specifying the minimum number of signals to be greater than in order to qualify as <i>active</i> . Defaults to 0. |
| return | Character vector to specify what to return. Valid options include: <ul style="list-style-type: none"> • "plot": returns a grid showing the distribution of archetypes by 'breaks' and number of active hours (default) • "plot-dist": returns a heatmap plot of signal distribution by hour and archetypes |

- "data": returns the raw data with the classified archetypes
- "table": returns a summary table of the archetypes
- "plot-area": returns an area plot of the percentages of archetypes shown over time
- "plot-hrvar": returns a bar plot showing the count of archetypes, faceted by the supplied HR attribute.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": returns a summary grid plot of the classified archetypes (default). A 'ggplot' object.
- "data": returns a data frame of the raw data with the classified archetypes
- "table": returns a data frame of summary table of the archetypes
- "plot-area": returns an area plot of the percentages of archetypes shown over time. A 'ggplot' object.
- "plot-hrvar": returns a bar plot showing the count of archetypes, faceted by the supplied HR attribute. A 'ggplot' object.

Author(s)

Ainize Cidoncha ainize.cidoncha@microsoft.com

See Also

Other Working Patterns: [flex_index\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [plot_flex_index\(\)](#), [workpatterns_area\(\)](#), [workpatterns_classify_pav\(\)](#), [workpatterns_classify\(\)](#), [workpatterns_hclust\(\)](#), [workpatterns_rank\(\)](#), [workpatterns_report\(\)](#)

workpatterns_classify_pav

Classify working pattern personas using a rule based algorithm, using the person-average volume-based ('pav') method.

Description

[Experimental]

Apply a rule based algorithm to emails or instant messages sent by hour of day. This uses a person-average volume-based ('pav') method.

Usage

```
workpatterns_classify_pav(
  data,
  values = "percent",
  signals = c("email", "IM"),
  start_hour = "0900",
  end_hour = "1700",
  return = "plot"
)
```

Arguments

| | |
|------------|---|
| data | A data frame containing data from the Hourly Collaboration query. |
| values | Character vector to specify whether to return percentages or absolute values in "data" and "plot". Valid values are: <ul style="list-style-type: none"> • "percent": percentage of signals divided by total signals (default) • "abs": absolute count of signals |
| signals | Character vector to specify which collaboration metrics to use: <ul style="list-style-type: none"> • "email" (default) for emails only • "IM" for Teams messages only, • "unscheduled_calls" for Unscheduled Calls only • "meetings" for Meetings only • or a combination of signals, such as c("email", "IM") |
| start_hour | A character vector specifying starting hours, e.g. "0900" |
| end_hour | A character vector specifying starting hours, e.g. "1700" |
| return | Character vector to specify what to return. Valid options include: <ul style="list-style-type: none"> • "plot": returns a bar plot of signal distribution by hour and archetypes (default) • "data": returns the raw data with the classified archetypes • "table": returns a summary table of the archetypes • "plot-area": returns an overlapping area plot |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": returns a bar plot of signal distribution by hour and archetypes (default). A 'ggplot' object.
- "data": returns a data frame of the raw data with the classified archetypes.
- "table": returns a data frame of a summary table of the archetypes.
- "plot-area": returns an overlapping area plot. A 'ggplot' object.

Author(s)

Ainize Cidoncha ainize.cidoncha@microsoft.com

See Also

Other Working Patterns: [flex_index\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [plot_flex_index\(\)](#), [workpatterns_area\(\)](#), [workpatterns_classify_bw\(\)](#), [workpatterns_classify\(\)](#), [workpatterns_hclust\(\)](#), [workpatterns_rank\(\)](#), [workpatterns_report\(\)](#)

workpatterns_hclust *Create a hierarchical clustering of email or IMs by hour of day*

Description**[Experimental]**

Apply hierarchical clustering to emails sent by hour of day. The hierarchical clustering uses cosine distance and the ward.D method of agglomeration.

Usage

```
workpatterns_hclust(
  data,
  k = 4,
  return = "plot",
  values = "percent",
  signals = "email",
  start_hour = "0900",
  end_hour = "1700"
)
```

Arguments

| | |
|---------|--|
| data | A data frame containing data from the Hourly Collaboration query. |
| k | Numeric vector to specify the k number of clusters to cut by. |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none"> • "plot" • "data" • "table" • "plot-area" • "hclust" • "dist" See Value for more information. |
| values | Character vector to specify whether to return percentages or absolute values in "data" and "plot". Valid values are: <ul style="list-style-type: none"> • "percent": percentage of signals divided by total signals (default) • "abs": absolute count of signals |
| signals | Character vector to specify which collaboration metrics to use: |

| | |
|------------|---|
| | <ul style="list-style-type: none"> • "email" (default) for emails only • "IM" for Teams messages only • "unscheduled_calls" for Unscheduled Calls only • "meetings" for Meetings only • or a combination of signals, such as <code>c("email", "IM")</code> |
| start_hour | A character vector specifying starting hours, e.g. "0900" |
| end_hour | A character vector specifying starting hours, e.g. "1700" |

Details

The hierarchical clustering is applied on the person-average volume-based (pav) level. In other words, the clustering is applied on a dataset where the collaboration hours are averaged by person and calculated as % of total daily collaboration.

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object of a bar plot (default)
- "data": data frame containing raw data with the clusters
- "table": data frame containing a summary table. Percentages of signals are shown, e.g. x% of signals are sent by y hour of the day.
- "plot-area": ggplot object. An overlapping area plot
- "hclust": hclust object for the hierarchical model
- "dist": distance matrix used to build the clustering model

See Also

Other Clustering: [personas_hclust\(\)](#), [workpatterns_classify\(\)](#)

Other Working Patterns: [flex_index\(\)](#), [identify_shifts_wp\(\)](#), [identify_shifts\(\)](#), [plot_flex_index\(\)](#), [workpatterns_area\(\)](#), [workpatterns_classify_bw\(\)](#), [workpatterns_classify_pav\(\)](#), [workpatterns_classify\(\)](#), [workpatterns_rank\(\)](#), [workpatterns_report\(\)](#)

Examples

```
# Run clusters, returning plot
workpatterns_hclust(em_data, k = 5, return = "plot")

# Run clusters, return raw data
workpatterns_hclust(em_data, k = 4, return = "data") %>% head()

# Run clusters for instant messages only, return hclust object
workpatterns_hclust(em_data, k = 4, return = "hclust", signals = c("IM"))
```

workpatterns_rank *Create a rank table of working patterns*

Description

Takes in an Hourly Collaboration query and returns a count table of working patterns, ranked from the most common to the least.

Usage

```
workpatterns_rank(  
  data,  
  signals = c("email", "IM"),  
  start_hour = "0900",  
  end_hour = "1700",  
  top = 10,  
  return = "plot"  
)
```

Arguments

| | |
|------------|--|
| data | A data frame containing hourly collaboration data. |
| signals | Character vector to specify which collaboration metrics to use: You may use "email" (default) for emails only, "IM" for Teams messages only, or a combination of the two c("email", "IM"). |
| start_hour | A character vector specifying starting hours, e.g. "0900" |
| end_hour | A character vector specifying starting hours, e.g. "1700" |
| top | number specifying how many top working patterns to display in plot, e.g. "10" |
| return | String specifying what to return. This must be one of the following strings: <ul style="list-style-type: none">• "plot"• "table" See Value for more information. |

Value

A different output is returned depending on the value passed to the return argument:

- "plot": ggplot object. A plot with the y-axis showing the top ten working patterns and the x-axis representing each hour of the day.
- "table": data frame. A summary table for the top working patterns.

See Also

Other Visualization: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar_asis()`, `create_bar()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_line_asis()`, `create_line()`, `create_period_scatter()`, `create_rank()`, `create_sankey()`, `create_scatter()`, `create_stacked()`, `create_tracking()`, `create_trend()`, `email_dist()`, `email_fizz()`, `email_line()`, `email_rank()`, `email_summary()`, `email_trend()`, `external_network_plot()`, `hr_trend()`, `hrvar_count()`, `hrvar_trend()`, `internal_network_plot()`, `keymetrics_scan()`, `meeting_dist()`, `meeting_fizz()`, `meeting_line()`, `meeting_quality()`, `meeting_rank()`, `meeting_summary()`, `meeting_trend()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_dist()`, `meetingtype_summary()`, `mgrcoatt_dist()`, `mgrrel_matrix()`, `one2one_dist()`, `one2one_fizz()`, `one2one_freq()`, `one2one_line()`, `one2one_rank()`, `one2one_sum()`, `one2one_trend()`, `period_change()`, `workloads_dist()`, `workloads_fizz()`, `workloads_line()`, `workloads_rank()`, `workloads_summary()`, `workloads_trend()`, `workpatterns_area()`

Other Working Patterns: `flex_index()`, `identify_shifts_wp()`, `identify_shifts()`, `plot_flex_index()`, `workpatterns_area()`, `workpatterns_classify_bw()`, `workpatterns_classify_pav()`, `workpatterns_classify()`, `workpatterns_hclust()`, `workpatterns_report()`

Examples

```
workpatterns_rank(em_data)
```

```
workpatterns_report    Generate a report on working patterns in HTML
```

Description**[Experimental]**

This function takes a Hourly Collaboration query and generates a HTML report on working patterns archetypes. Archetypes are created using the binary-week method.

Usage

```
workpatterns_report(
  data,
  hrvar = "Organization",
  signals = c("email", "IM"),
  start_hour = "0900",
  end_hour = "1700",
  path = "workpatterns report",
  timestamp = TRUE
)
```

Arguments

| | |
|------------|--|
| data | A Hourly Collaboration Query dataset in the form of a data frame. |
| hrvar | String specifying HR attribute to cut by archetypes. Defaults to Organization. |
| signals | See <code>workpatterns_classify()</code> . |
| start_hour | See <code>workpatterns_classify()</code> . |
| end_hour | See <code>workpatterns_classify()</code> . |
| path | Pass the file path and the desired file name, <i>excluding the file extension</i> . For example, "scope report". |
| timestamp | Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE. |

Value

An HTML report with the same file name as specified in the arguments is generated in the working directory. No outputs are directly returned by the function.

See Also

Other Reports: `IV_report()`, `capacity_report()`, `coaching_report()`, `collaboration_report()`, `connectivity_report()`, `generate_report()`, `meeting_tm_report()`, `read_preamble()`, `subject_validate_report()`, `validation_report()`

Other Working Patterns: `flex_index()`, `identify_shifts_wp()`, `identify_shifts()`, `plot_flex_index()`, `workpatterns_area()`, `workpatterns_classify_bw()`, `workpatterns_classify_pav()`, `workpatterns_classify()`, `workpatterns_hclust()`, `workpatterns_rank()`

 wrap

Add a character at the start and end of a character string

Description

This function adds a character at the start and end of a character string, where the default behaviour is to add a double quote.

Usage

```
wrap(string, wrapper = "\"")
```

Arguments

| | |
|---------|---------------------------------------|
| string | Character string to be wrapped around |
| wrapper | Character to wrap around string |

Value

Character vector containing the modified string.

See Also

Other Support: [camel_clean\(\)](#), [check_inputs\(\)](#), [combine_signals\(\)](#), [cut_hour\(\)](#), [extract_date_range\(\)](#), [extract_hr\(\)](#), [heat_colours\(\)](#), [is_date_format\(\)](#), [maxmin\(\)](#), [p_test\(\)](#), [pairwise_count\(\)](#), [plot_WOE\(\)](#), [read_preamble\(\)](#), [rgb2hex\(\)](#), [totals_bind\(\)](#), [totals_col\(\)](#), [totals_reorder\(\)](#), [tstamp\(\)](#), [us_to_space\(\)](#)

Index

- * **After-hours Collaboration**
 - afterhours_dist, 5
 - afterhours_fizz, 7
 - afterhours_line, 8
 - afterhours_rank, 10
 - afterhours_summary, 11
 - afterhours_trend, 12
- * **Clustering**
 - personas_hclust, 177
 - workpatterns_classify, 217
 - workpatterns_hclust, 225
- * **Collaboration**
 - collaboration_area, 20
 - collaboration_dist, 21
 - collaboration_fizz, 23
 - collaboration_line, 25
 - collaboration_rank, 26
 - collaboration_sum, 29
 - collaboration_trend, 31
- * **Data Validation**
 - check_query, 18
 - extract_hr, 87
 - flag_ch_ratio, 88
 - flag_em_ratio, 89
 - flag_extreme, 90
 - flag_outlooktime, 91
 - hr_trend, 103
 - hrvar_count, 100
 - hrvar_count_all, 101
 - hrvar_trend, 102
 - identify_churn, 105
 - identify_holidayweeks, 106
 - identify_inactiveweeks, 108
 - identify_nkw, 109
 - identify_outlier, 110
 - identify_privacythreshold, 111
 - identify_query, 112
 - identify_shifts, 113
 - identify_shifts_wp, 114
 - identify_tenure, 116
 - remove_outliers, 182
 - standardise_pq, 186
 - subject_validate, 190
 - subject_validate_report, 191
 - track_HR_change, 201
 - validation_report, 203
- * **Data**
 - dv_data, 70
 - em_data, 81
 - g2g_data, 95
 - mt_data, 150
 - p2p_data_sim, 174
 - sq_data, 184
- * **Emails**
 - email_dist, 73
 - email_fizz, 74
 - email_line, 76
 - email_rank, 77
 - email_summary, 79
 - email_trend, 80
- * **Flexible Input**
 - create_ITSA, 49
 - period_change, 176
- * **Flexible**
 - create_bar, 35
 - create_bar_asis, 37
 - create_boxplot, 39
 - create_bubble, 41
 - create_dist, 43
 - create_fizz, 46
 - create_hist, 47
 - create_line, 52
 - create_line_asis, 54
 - create_period_scatter, 56
 - create_rank, 58
 - create_sankey, 61
 - create_scatter, 63
 - create_stacked, 64

- create_tracking, 66
- create_trend, 68
- period_change, 176
- * **Import and Export**
 - copy_df, 34
 - create_dt, 45
 - export, 84
 - import_to_fst, 118
 - import_wpa, 119
 - standardise_pq, 186
- * **Information Value**
 - create_IV, 51
 - IV_by_period, 122
 - IV_report, 123
 - plot_WOE, 180
- * **Interrupted Time-Series Analysis**
 - create_ITSA, 49
- * **Managerial Relations**
 - mgrcoatt_dist, 147
 - mgrrel_matrix, 148
 - one2one_dist, 163
 - one2one_fizz, 165
 - one2one_freq, 166
 - one2one_line, 168
 - one2one_rank, 169
 - one2one_sum, 171
 - one2one_trend, 173
- * **Meetings**
 - meeting_dist, 134
 - meeting_extract, 135
 - meeting_fizz, 136
 - meeting_line, 138
 - meeting_quality, 139
 - meeting_rank, 141
 - meeting_skim, 143
 - meeting_summary, 144
 - meeting_tm_report, 145
 - meeting_trend, 146
 - meetingtype_dist, 129
 - meetingtype_dist_ca, 130
 - meetingtype_dist_mt, 131
 - meetingtype_summary, 132
- * **Network**
 - external_network_plot, 85
 - g2g_data, 95
 - internal_network_plot, 120
 - network_describe, 152
 - network_g2g, 153
 - network_leiden, 156
 - network_louvain, 158
 - network_p2p, 160
 - p2p_data_sim, 174
- * **Reports**
 - capacity_report, 16
 - coaching_report, 19
 - collaboration_report, 28
 - connectivity_report, 33
 - generate_report, 96
 - IV_report, 123
 - meeting_tm_report, 145
 - read_preamble, 182
 - subject_validate_report, 191
 - validation_report, 203
 - workpatterns_report, 228
- * **Support**
 - camel_clean, 15
 - check_inputs, 17
 - combine_signals, 32
 - cut_hour, 69
 - extract_date_range, 86
 - extract_hr, 87
 - heat_colours, 99
 - is_date_format, 121
 - maxmin, 128
 - p_test, 181
 - pairwise_count, 175
 - plot_WOE, 180
 - read_preamble, 182
 - rgb2hex, 183
 - totals_bind, 198
 - totals_col, 199
 - totals_reorder, 200
 - tstamp, 202
 - us_to_space, 203
 - wrap, 229
- * **Text-mining**
 - meeting_tm_report, 145
 - pairwise_count, 175
 - subject_validate, 190
 - subject_validate_report, 191
 - tm_clean, 193
 - tm_cooc, 194
 - tm_freq, 195
 - tm_wordcloud, 196
- * **Themes**
 - theme_wpa, 192

- theme_wpa_basic, 193
- * **Time-series**
 - create_line, 52
 - create_line_asis, 54
 - create_period_scatter, 56
 - create_trend, 68
 - IV_by_period, 122
 - period_change, 176
- * **Variable Association**
 - create_IV, 51
 - IV_by_period, 122
 - IV_report, 123
 - plot_WOE, 180
- * **Visualization**
 - afterhours_dist, 5
 - afterhours_fizz, 7
 - afterhours_line, 8
 - afterhours_rank, 10
 - afterhours_summary, 11
 - afterhours_trend, 12
 - collaboration_area, 20
 - collaboration_dist, 21
 - collaboration_fizz, 23
 - collaboration_line, 25
 - collaboration_rank, 26
 - collaboration_sum, 29
 - collaboration_trend, 31
 - create_bar, 35
 - create_bar_asis, 37
 - create_boxplot, 39
 - create_bubble, 41
 - create_dist, 43
 - create_fizz, 46
 - create_line, 52
 - create_line_asis, 54
 - create_period_scatter, 56
 - create_rank, 58
 - create_sankey, 61
 - create_scatter, 63
 - create_stacked, 64
 - create_tracking, 66
 - create_trend, 68
 - email_dist, 73
 - email_fizz, 74
 - email_line, 76
 - email_rank, 77
 - email_summary, 79
 - email_trend, 80
 - external_network_plot, 85
 - hr_trend, 103
 - hrvar_count, 100
 - hrvar_trend, 102
 - internal_network_plot, 120
 - keymetrics_scan, 125
 - meeting_dist, 134
 - meeting_fizz, 136
 - meeting_line, 138
 - meeting_quality, 139
 - meeting_rank, 141
 - meeting_summary, 144
 - meeting_trend, 146
 - meetingtype_dist, 129
 - meetingtype_dist_ca, 130
 - meetingtype_dist_mt, 131
 - meetingtype_summary, 132
 - mgrcoatt_dist, 147
 - mgrrel_matrix, 148
 - one2one_dist, 163
 - one2one_fizz, 165
 - one2one_freq, 166
 - one2one_line, 168
 - one2one_rank, 169
 - one2one_sum, 171
 - one2one_trend, 173
 - period_change, 176
 - workloads_dist, 207
 - workloads_fizz, 208
 - workloads_line, 210
 - workloads_rank, 211
 - workloads_summary, 213
 - workloads_trend, 214
 - workpatterns_area, 215
 - workpatterns_rank, 227
- * **Working Patterns**
 - flex_index, 92
 - identify_shifts, 113
 - identify_shifts_wp, 114
 - plot_flex_index, 179
 - workpatterns_area, 215
 - workpatterns_classify, 217
 - workpatterns_classify_bw, 221
 - workpatterns_classify_pav, 223
 - workpatterns_hclust, 225
 - workpatterns_rank, 227
 - workpatterns_report, 228
- * **Workweek Span**

- workloads_dist, 207
- workloads_fizz, 208
- workloads_line, 210
- workloads_rank, 211
- workloads_summary, 213
- workloads_trend, 214
- * datasets**
 - dv_data, 70
 - em_data, 81
 - g2g_data, 95
 - mt_data, 150
 - sq_data, 184
- * max-min**
 - maxmin, 128
- afterhours_dist, 5, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 31, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- afterhours_fizz, 6, 7, 7, 9, 11–13, 21, 23, 24, 26, 28, 30, 31, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- afterhours_line, 6–8, 8, 11–13, 21, 23, 24, 26, 28, 30, 31, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- afterhours_rank, 6–9, 10, 12, 13, 21, 23, 24, 26, 28, 30, 31, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- afterhours_sum (afterhours_summary), 11
- afterhours_summary, 6–9, 11, 11, 13, 21, 23, 24, 26, 28, 30, 31, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- afterhours_trend, 6–9, 11, 12, 12, 21, 23, 24, 26, 28, 30, 31, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- analysis_scope (hrvar_count), 100
- anonymise, 14
- anonymize (anonymise), 14
- calculate_IV, 15
- camel_clean, 15, 17, 32, 70, 87, 88, 99, 121, 128, 175, 180–182, 184, 198–200, 202, 203, 230
- capacity_report, 16, 20, 29, 34, 98, 124, 146, 182, 192, 205, 229
- check_inputs, 16, 17, 32, 70, 87, 88, 99, 121, 128, 175, 180–182, 184, 198–200, 202, 203, 230
- check_query, 18, 88–92, 101–104, 106–108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
- coaching_report, 17, 19, 29, 34, 98, 124, 146, 182, 192, 205, 229
- collab_area (collaboration_area), 20
- collab_dist (collaboration_dist), 21
- collab_fizz (collaboration_fizz), 23
- collab_line (collaboration_line), 25
- collab_rank (collaboration_rank), 26
- collab_sum (collaboration_sum), 29
- collab_summary (collaboration_sum), 29
- collaboration_area, 6, 8, 9, 11–13, 20, 23, 24, 26, 28, 30–32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- collaboration_dist, 6, 8, 9, 11–13, 21, 21,

- 24, 26, 28, 30–32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- collaboration_fizz, 6, 8, 9, 11–13, 21, 23, 23, 26, 28, 30–32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- collaboration_line, 6, 8, 9, 11–13, 21, 23, 24, 25, 28, 30–32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- collaboration_rank, 6, 8, 9, 11–13, 21, 23, 24, 26, 26, 30–32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- collaboration_report, 17, 20, 28, 34, 98, 124, 146, 182, 192, 205, 229
- collaboration_sum, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 29, 31, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- collaboration_summary
(collaboration_sum), 29
- collaboration_trend, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 31, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- combine_signals, 16, 17, 32, 70, 87, 88, 99, 121, 128, 175, 180–182, 184, 198–200, 202, 203, 230
- comma, 33
- connectivity_report, 17, 20, 29, 33, 98, 124, 146, 182, 192, 205, 229
- copy_df, 34, 45, 85, 118, 119, 187
- create_bar, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 31, 35, 38, 40–42, 44, 47, 48, 53–55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- create_bar_asis, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 31, 36, 37, 37, 40–42, 44, 47, 48, 53–55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- create_boxplot, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 31, 36–38, 39, 42, 44, 47, 48, 53–55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- create_bubble, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 31, 36–38, 40, 41, 41, 44, 47, 48, 53–55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- create_dist, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 31, 36–38, 40–42, 43, 47, 48, 53–55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104,

- 120, 126, 129, 131–133, 135, 137,
 139, 140, 142, 144, 147, 148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 create_dt, 35, 45, 85, 118, 119, 187
 create_fizz, 6, 8, 9, 11–13, 21, 23, 24, 26,
 28, 30, 31, 36–38, 40–42, 44, 46, 48,
 53–55, 57, 59, 62, 64, 66, 67, 69,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 120, 126, 129, 131–133, 135, 137,
 139, 140, 142, 144, 147, 148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 create_hist, 37, 38, 41, 42, 44, 47, 47, 54,
 55, 57, 59, 62, 64, 66, 67, 69, 177
 create_ITSA, 49, 177
 create_IV, 51, 123, 124, 180
 create_line, 6, 8, 9, 11–13, 21, 23, 24, 26,
 28, 30, 31, 36–38, 40–42, 44, 47, 48,
 52, 55, 57, 59, 62, 64, 66, 67, 69,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 120, 123, 126, 129, 131–133, 135,
 137, 139, 140, 142, 144, 147, 148,
 150, 164, 165, 167, 169, 171–173,
 177, 208–210, 212, 214, 215, 217,
 228
 create_line(), 9
 create_line_asis, 6, 8, 9, 11–13, 21, 23, 24,
 26, 28, 30, 31, 36–38, 40–42, 44, 47,
 48, 53, 54, 54, 57, 59, 62, 64, 66, 67,
 69, 74–76, 78, 80, 81, 86, 100, 103,
 104, 120, 123, 126, 129, 131–133,
 135, 137, 139, 140, 142, 144, 147,
 148, 150, 164, 165, 167, 169,
 171–173, 177, 208–210, 212, 214,
 215, 217, 228
 create_period_scatter, 6, 8, 9, 11–13, 21,
 23, 24, 26, 28, 30, 31, 36–38, 40–42,
 44, 47, 48, 53–55, 56, 59, 62, 64, 66,
 67, 69, 74–76, 78, 80, 81, 86, 100,
 103, 104, 120, 123, 126, 129,
 131–133, 135, 137, 139, 140, 142,
 144, 147, 148, 150, 164, 165, 167,
 169, 171–173, 177, 208–210, 212,
 214, 215, 217, 228
 create_rank, 6, 8, 9, 11–13, 21, 23, 24, 26,
 28, 30, 31, 36–38, 40–42, 44, 47, 48,
 53–55, 57, 58, 62, 64, 66, 67, 69,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 121, 126, 129, 131–133, 135, 137,
 139, 140, 142, 144, 147, 148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 create_rank_combine, 60
 create_sankey, 6, 8, 9, 11–13, 21, 23, 24, 26,
 28, 30, 31, 36–38, 40–42, 44, 47, 48,
 53–55, 57, 59, 61, 64, 66, 67, 69,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 121, 126, 129, 131–133, 135, 137,
 139, 140, 142, 144, 147, 148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 create_scatter, 6, 8, 9, 11–13, 21, 23, 24,
 26, 28, 30, 31, 36–38, 40–42, 44, 47,
 48, 53–55, 57, 59, 62, 63, 66, 67, 69,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 121, 126, 129, 131–133, 135, 137,
 139, 140, 142, 144, 147, 148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 create_stacked, 6, 8, 9, 11–13, 21, 23, 24,
 26, 28, 30, 31, 36–38, 40–42, 44, 47,
 48, 53–55, 57, 59, 62, 64, 64, 67, 69,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 121, 126, 129, 131–133, 135, 137,
 139, 140, 142, 144, 147, 148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 create_tracking, 6, 8, 9, 11–13, 21, 23, 24,
 26, 28, 30, 31, 36–38, 40–42, 44, 47,
 48, 53–55, 57, 59, 62, 64, 66, 66, 69,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 121, 126, 129, 131–133, 135, 137,
 139, 140, 142, 144, 147, 148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 create_trend, 6, 8, 9, 11–13, 21, 23, 24, 26,
 28, 30, 31, 36–38, 40–42, 44, 47, 48,
 53–55, 57, 59, 62, 64, 66, 67, 68,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 121, 123, 126, 129, 131–133, 135,
 137, 139, 140, 142, 144, 147, 148,
 150, 164, 165, 167, 169, 171–173,
 177, 208–210, 212, 214, 215, 217,
 228
 cut_hour, 16, 17, 32, 69, 87, 88, 99, 121, 128,

- 175, 180–182, 184, 198–200, 202, 203, 230
- dv_data, 70, 83, 96, 152, 174, 186
- em_data, 72, 81, 96, 152, 174, 186
- email_dist, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 73, 75–78, 80, 81, 86, 100, 103, 104, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- email_fizz, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74, 74, 76–78, 80, 81, 86, 100, 103, 104, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- email_line, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74, 75, 76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- email_rank, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–77, 77, 80, 81, 86, 100, 103, 104, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- email_sum (email_summary), 79
- email_summary, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–78, 79, 81, 86, 100, 103, 104, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- email_trend, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–78, 80, 80, 86, 100, 103, 104, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- export, 35, 45, 84, 118, 119, 187
- external_network_plot, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 85, 96, 100, 103, 104, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 153, 155, 158, 159, 162, 164, 165, 167, 169, 171–174, 177, 208–210, 212, 214, 215, 217, 228
- extract_date_range, 16, 17, 32, 70, 86, 88, 99, 121, 128, 175, 180–182, 184, 198–200, 202, 203, 230
- extract_hr, 16, 17, 19, 32, 70, 87, 87, 89–92, 99, 101–104, 106–108, 110, 112, 114, 116, 117, 121, 128, 175, 180–184, 187, 191, 192, 198–200, 202, 203, 205, 230
- flag_ch_ratio, 19, 88, 88, 90–92, 101–104, 106–108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
- flag_em_ratio, 19, 88, 89, 89, 91, 92, 101–104, 106–108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
- flag_extreme, 19, 88–90, 90, 92, 101–104, 106–108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
- flag_outlooktime, 19, 88–91, 91, 101–104, 106–108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
- flex_index, 92, 114, 116, 179, 217, 221, 223, 225, 226, 228, 229
- g2g_data, 72, 83, 86, 95, 121, 152, 153, 155, 158, 159, 162, 174, 186
- g2g_network (network_g2g), 153
- generate_report, 17, 20, 29, 34, 96, 124, 146, 182, 192, 205, 229
- generate_report2, 98
- heat_colors (heat_colours), 99

- heat_colours*, 16, 17, 32, 70, 87, 88, 99, 121, 128, 175, 180–182, 184, 198–200, 202, 203, 230
hr_trend, 6, 8, 9, 11–13, 19, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 88–92, 100–103, 103, 106–108, 110, 112, 114, 116, 117, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 183, 187, 191, 192, 202, 205, 208–210, 212, 214, 215, 217, 228
hrvar_count, 6, 8, 9, 11–13, 19, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 88–92, 100, 102–104, 106–108, 110, 112, 114, 116, 117, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 183, 187, 191, 192, 202, 205, 208–210, 212, 214, 215, 217, 228
hrvar_count_all, 19, 88–92, 101, 101, 103, 104, 106–108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
hrvar_trend, 6, 8, 9, 11–13, 19, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 88–92, 100–102, 102, 104, 106–108, 110, 112, 114, 116, 117, 121, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 183, 187, 191, 192, 202, 205, 208–210, 212, 214, 215, 217, 228
identify_churn, 19, 88–92, 101–104, 105, 107, 108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
identify_holidayweeks, 19, 88–92, 101–104, 106, 106, 108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
identify_inactiveweeks, 19, 88–92, 101–104, 106, 107, 108, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
identify_nkw, 19, 88–92, 101–104, 106–108, 109, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
identify_outlier, 19, 88–92, 101–104, 106–108, 110, 110, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
identify_privacythreshold, 19, 88–92, 101–104, 106–108, 110, 111, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
identify_query, 19, 88–92, 101–104, 106–108, 110, 112, 112, 114, 116, 117, 183, 187, 191, 192, 202, 205
identify_shifts, 19, 88–92, 95, 101–104, 106–108, 110, 112, 113, 116, 117, 179, 183, 187, 191, 192, 202, 205, 217, 221, 223, 225, 226, 228, 229
identify_shifts_wp, 19, 88–92, 95, 101–104, 106–108, 110, 112, 114, 114, 117, 179, 183, 187, 191, 192, 202, 205, 217, 221, 223, 225, 226, 228, 229
identify_tenure, 19, 88–92, 101–104, 106–108, 110, 112, 114, 116, 116, 183, 187, 191, 192, 202, 205
import_to_fst, 35, 45, 85, 118, 119, 187
import_wpa, 35, 45, 85, 118, 119, 187
internal_network_plot, 6, 8, 9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 96, 100, 103, 104, 120, 126, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 153, 155, 158, 159, 162, 164, 165, 167, 169, 171–174, 177, 208–210, 212, 214, 215, 217, 228
is_date_format, 16, 17, 32, 70, 87, 88, 99, 121, 128, 175, 180–182, 184, 198–200, 202, 203, 230
IV_by_period, 52, 54, 55, 57, 69, 122, 124, 177, 180
IV_report, 17, 20, 29, 34, 52, 98, 123, 123, 146, 180, 182, 192, 205, 229
jitter_metrics, 125
keymetrics_scan, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53,

- 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 125, 129, 131–133, 135, 137, 139, 140, 142, 144, 147, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- map_IV, 127
- maxmin, 16, 17, 32, 70, 87, 88, 99, 121, 128, 175, 180–182, 184, 198–200, 202, 203, 230
- meeting_dist, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 129–133, 134, 136, 137, 139, 140, 142–148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meeting_extract, 130–133, 135, 135, 137, 139, 140, 142, 143, 145–147
- meeting_fizz, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 129–133, 135, 136, 136, 139, 140, 142–148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meeting_line, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 129–133, 135–137, 138, 140, 142–148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meeting_quality, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 129–133, 135–137, 139, 139, 142–148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meeting_rank, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 130–133, 135–137, 139, 140, 141, 143–148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meeting_skim, 130–133, 135–137, 139, 140, 142, 143, 145–147
- meeting_sum(meeting_summary), 144
- meeting_summary, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 130–133, 135–137, 139, 140, 142, 143, 144, 146–148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meeting_tm_report, 17, 20, 29, 34, 98, 124, 130–133, 135–137, 139, 140, 142, 143, 145, 145, 147, 175, 182, 191, 192, 194–197, 205, 229
- meeting_trend, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 130–133, 135–137, 139, 140, 142–146, 146, 148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meetingtype_dist, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 54, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 129, 131–133, 135–137, 139, 140, 142–148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meetingtype_dist_ca, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 130, 130, 132, 133, 135–137, 139, 140, 142–148, 150, 164, 165, 167, 169, 171–173, 177, 208–210, 212, 214, 215, 217, 228
- meetingtype_dist_mt, 7–9, 11–13, 21, 23, 24, 26, 28, 30, 32, 36, 38, 40, 42, 44, 47, 53, 55, 57, 59, 62, 64, 66, 67, 69, 74–76, 78, 80, 81, 86, 100, 103, 104, 121, 126, 130, 131, 131, 133,

- 135–137, 139, 140, 142–148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 meetingtype_sum (meetingtype_summary),
 132
 meetingtype_summary, 7–9, 11–13, 21, 23,
 24, 26, 28, 30, 32, 36, 38, 40, 42, 44,
 47, 54, 55, 57, 59, 62, 64, 66, 67, 69,
 74–76, 78, 80, 81, 86, 100, 103, 104,
 121, 126, 130–132, 132, 135–137,
 139, 140, 142, 143, 145–148, 150,
 164, 165, 167, 169, 171–173, 177,
 208–210, 212, 214, 215, 217, 228
 mgrcoatt_dist, 7–9, 11–13, 21, 23, 24, 26,
 28, 30, 32, 36, 38, 40, 42, 44, 47, 54,
 55, 57, 59, 62, 64, 66, 67, 69, 74–76,
 78, 80, 81, 86, 100, 103, 104, 121,
 126, 130–133, 135, 137, 139, 140,
 142, 145, 147, 147, 150, 164–167,
 169, 171–174, 177, 208–210, 212,
 214, 215, 217, 228
 mgrrel_matrix, 7–9, 11–13, 21, 23, 24, 26,
 28, 30, 32, 36, 38, 40, 42, 44, 47, 54,
 55, 57, 59, 62, 64, 66, 67, 69, 74–76,
 78, 80, 81, 86, 100, 103, 104, 121,
 126, 130–133, 135, 137, 139, 140,
 142, 145, 147, 148, 148, 164–167,
 169, 171–174, 177, 208–210, 212,
 214, 215, 217, 228
 mt_data, 72, 83, 96, 150, 174, 186

 network_describe, 86, 96, 121, 152, 155,
 158, 159, 162, 174
 network_g2g, 86, 96, 121, 153, 153, 158, 159,
 162, 174
 network_leiden, 86, 96, 121, 153, 155, 156,
 159, 162, 174
 network_louvain, 86, 96, 121, 153, 155, 158,
 158, 162, 174
 network_p2p, 86, 96, 121, 153, 155, 158, 159,
 160, 174

 one2one_dist, 7–9, 11–13, 21, 23, 24, 26, 28,
 30, 32, 36, 38, 40, 42, 44, 47, 54, 55,
 57, 59, 62, 64, 66, 67, 69, 74–76, 78,
 80, 81, 86, 100, 103, 104, 121, 126,
 130–133, 135, 137, 139, 140, 142,
 145, 147, 148, 150, 163, 165–167,
 169, 171–174, 177, 208–210, 212,
 214, 215, 217, 228
 one2one_fizz, 7–9, 11–13, 21, 23, 24, 26, 28,
 30, 32, 36, 38, 40, 42, 44, 47, 54, 55,
 57, 59, 62, 64, 66, 67, 69, 74, 75, 77,
 78, 80, 81, 86, 100, 103, 104, 121,
 126, 130–133, 135, 137, 139, 140,
 142, 145, 147, 148, 150, 164, 165,
 167, 169, 171–174, 177, 208, 209,
 211, 212, 214, 215, 217, 228
 one2one_freq, 7–9, 11–13, 21, 23, 24, 26, 28,
 30, 32, 36, 38, 40, 42, 44, 47, 54, 55,
 57, 59, 62, 64, 66, 67, 69, 74, 75, 77,
 78, 80, 81, 86, 100, 103, 104, 121,
 127, 130–133, 135, 137, 139, 140,
 142, 145, 147, 148, 150, 164–166,
 166, 169, 171–174, 177, 208, 209,
 211, 212, 214, 215, 217, 228
 one2one_line, 7–9, 11–13, 21, 23, 24, 26, 28,
 30, 32, 36, 38, 40, 42, 44, 47, 54, 55,
 57, 59, 62, 64, 66, 67, 69, 74, 75, 77,
 78, 80, 81, 86, 100, 103, 104, 121,
 127, 130–133, 135, 137, 139, 140,
 142, 145, 147, 148, 150, 164–167,
 168, 171–174, 177, 208, 209, 211,
 212, 214, 215, 217, 228
 one2one_rank, 7–9, 11–13, 21, 23, 24, 26, 28,
 30, 32, 36, 38, 40, 42, 44, 47, 54, 55,
 57, 59, 62, 64, 66, 67, 69, 74, 75, 77,
 78, 80, 81, 86, 100, 103, 104, 121,
 127, 130–133, 135, 137, 139, 140,
 142, 145, 147, 148, 150, 164–167,
 169, 169, 172–174, 177, 208, 209,
 211, 212, 214, 215, 217, 228
 one2one_sum, 7–9, 11–13, 21, 23, 24, 26, 28,
 30, 32, 36, 38, 40, 42, 44, 47, 54, 55,
 57, 59, 62, 64, 66, 67, 69, 74, 75, 77,
 78, 80, 81, 86, 100, 103, 104, 121,
 127, 130–133, 135, 137, 139, 140,
 142, 145, 147, 148, 150, 164–167,
 169, 171, 171, 173, 174, 177, 208,
 209, 211, 212, 214, 215, 217, 228
 one2one_summary (one2one_sum), 171
 one2one_trend, 7–9, 11–13, 21, 23, 24, 26,
 28, 30, 32, 36, 38, 40, 42, 44, 47, 54,
 55, 57, 59, 62, 64, 66, 67, 69, 74, 75,
 77, 78, 80, 81, 86, 100, 103, 104,
 121, 127, 130–133, 135, 137, 139,

- 140, 142, 145, 147, 148, 150,
164–167, 169, 171, 172, 173, 177,
208, 209, 211, 212, 214, 215, 217,
228
- p2p_data_sim, 72, 83, 86, 96, 121, 152, 153,
155, 158, 159, 162, 174, 186
- p_test, 16, 17, 32, 70, 87, 88, 99, 121, 128,
175, 180, 181, 182, 184, 198–200,
202, 203, 230
- pairwise_count, 16, 17, 32, 70, 87, 88, 99,
121, 128, 146, 175, 180–182, 184,
191, 192, 194–200, 202, 203, 230
- period_change, 7–9, 11–13, 21, 23, 24, 26,
28, 30, 32, 36–38, 40–42, 44, 47, 48,
50, 54, 55, 57, 59, 62, 64, 66, 67, 69,
74, 75, 77, 78, 80, 81, 86, 101, 103,
104, 121, 123, 127, 130–133, 135,
137, 139, 140, 142, 145, 147, 148,
150, 164, 165, 167, 169, 171–173,
176, 208, 209, 211, 212, 214, 215,
217, 228
- personas_hclust, 177, 221, 226
- plot_flex_index, 95, 114, 116, 179, 217,
221, 223, 225, 226, 228, 229
- plot_WOE, 16, 17, 32, 52, 70, 87, 88, 99, 121,
123, 124, 128, 175, 180, 181, 182,
184, 198–200, 202, 203, 230
- read_preamble, 16, 17, 20, 29, 32, 34, 70, 87,
88, 98, 99, 121, 124, 128, 146, 175,
180, 181, 182, 184, 192, 198–200,
202, 203, 205, 229, 230
- remove_outliers, 19, 88–92, 101–104,
106–108, 110, 112, 114, 116, 117,
182, 187, 191, 192, 202, 205
- rgb2hex, 16, 17, 32, 70, 87, 88, 99, 121, 128,
175, 180–182, 183, 198–200, 202,
203, 230
- sq_data, 72, 83, 96, 152, 174, 184
- standardise_pq, 19, 35, 45, 85, 88–92,
101–104, 106–108, 110, 112, 114,
116–119, 183, 186, 191, 192, 202,
205
- standardize_pq (standardise_pq), 186
- subject_classify, 187
- subject_scan, 188
- subject_validate, 19, 88–92, 101–104,
106–108, 110, 112, 114, 116, 117,
146, 175, 183, 187, 190, 192,
194–197, 202, 205
- subject_validate_report, 17, 19, 20, 29,
34, 88–92, 98, 101–104, 106–108,
110, 112, 114, 116, 117, 124, 146,
175, 182, 183, 187, 191, 191,
194–197, 202, 205, 229
- theme_wpa, 192, 193
- theme_wpa_basic, 192, 193
- tm_clean, 146, 175, 191, 192, 193, 195–197
- tm_cooc, 146, 175, 191, 192, 194, 194, 196,
197
- tm_freq, 146, 175, 191, 192, 194, 195, 195,
197
- tm_scan (subject_scan), 188
- tm_wordcloud, 146, 175, 191, 192, 194–196,
196
- totals_bind, 16, 17, 32, 70, 87, 88, 99, 121,
128, 175, 180–182, 184, 198, 199,
200, 202, 203, 230
- totals_col, 16, 17, 32, 70, 87, 88, 99, 121,
128, 175, 180–182, 184, 198, 199,
200, 202, 203, 230
- totals_reorder, 16, 17, 32, 70, 87, 88, 99,
121, 128, 175, 180–182, 184, 198,
199, 200, 202, 203, 230
- track_HR_change, 19, 88–92, 101–104,
106–108, 110, 112, 114, 116, 117,
183, 187, 191, 192, 201, 205
- tstamp, 16, 17, 32, 70, 87, 88, 99, 121, 128,
175, 180–182, 184, 198–200, 202,
203, 230
- us_to_space, 16, 17, 32, 70, 87, 88, 99, 121,
128, 175, 180–182, 184, 198–200,
202, 203, 230
- validation_report, 17, 19, 20, 29, 34,
88–92, 98, 101–104, 106–108, 110,
112, 114, 116, 117, 124, 146, 182,
183, 187, 191, 192, 202, 203, 229
- wellbeing_report, 205
- workloads_dist, 7–9, 11–13, 21, 23, 24, 26,
28, 30, 32, 36, 38, 40, 42, 44, 47, 54,
55, 57, 59, 62, 64, 66, 67, 69, 74, 75,

- 77, 78, 80, 81, 86, 101, 103, 104,
121, 127, 130–133, 135, 137, 139,
140, 142, 145, 147, 148, 150, 164,
166, 167, 169, 171, 172, 174, 177,
207, 209, 211, 212, 214, 215, 217,
228
- workloads_fizz, 7–9, 11–13, 21, 23, 24, 26,
28, 30, 32, 36, 38, 40, 42, 44, 47, 54,
55, 57, 59, 62, 64, 66, 67, 69, 74, 75,
77, 78, 80, 81, 86, 101, 103, 104,
121, 127, 130–133, 135, 137, 139,
140, 142, 145, 147, 148, 150, 164,
166, 167, 169, 171, 172, 174, 177,
208, 209, 211, 212, 214, 215, 217,
228
- workloads_line, 7–9, 11–13, 21, 23, 24, 26,
28, 30, 32, 36, 38, 40, 42, 44, 47, 54,
55, 57, 59, 62, 64, 66, 67, 69, 74, 75,
77, 78, 80, 81, 86, 101, 103, 104,
121, 127, 130–133, 135, 137, 139,
140, 142, 145, 147, 148, 150, 164,
166, 167, 169, 171, 172, 174, 177,
208, 209, 210, 212, 214, 215, 217,
228
- workloads_rank, 7–9, 11–13, 21, 23, 24, 26,
28, 30, 32, 36, 38, 40, 42, 44, 47, 54,
55, 57, 59, 62, 64, 66, 67, 69, 74, 75,
77, 78, 80, 81, 86, 101, 103, 104,
121, 127, 130–133, 135, 137, 139,
140, 142, 145, 147, 148, 150, 164,
166, 167, 169, 171, 172, 174, 177,
208, 209, 211, 211, 214, 215, 217,
228
- workloads_sum (workloads_summary), 213
- workloads_summary, 7–9, 11–13, 21, 23, 24,
26, 28, 30, 32, 36, 38, 40, 42, 44, 47,
54, 55, 57, 59, 62, 64, 66, 67, 69, 74,
75, 77, 78, 80, 81, 86, 101, 103, 104,
121, 127, 130–133, 135, 137, 139,
140, 142, 145, 147, 148, 150, 164,
166, 167, 169, 171, 172, 174, 177,
208, 209, 211, 212, 213, 215, 217,
228
- workloads_trend, 7–9, 11–13, 21, 23, 24, 26,
28, 30, 32, 36, 38, 40, 42, 44, 47, 54,
55, 57, 59, 62, 64, 66, 67, 69, 74, 75,
77, 78, 80, 81, 86, 101, 103, 104,
121, 127, 130–133, 135, 137, 139,
140, 142, 145, 147, 148, 150, 164,
166, 167, 169, 171, 172, 174, 177,
208, 209, 211, 212, 214, 214, 217,
228
- workpatterns_area, 7–9, 11–13, 21, 23, 24,
26, 28, 30, 32, 36, 38, 40, 42, 44, 47,
54, 55, 57, 59, 62, 64, 66, 67, 69, 74,
75, 77, 78, 80, 81, 86, 95, 101, 103,
104, 114, 116, 121, 127, 130–133,
135, 137, 139, 140, 142, 145, 147,
148, 150, 164, 166, 167, 169, 171,
172, 174, 177, 179, 208, 209, 211,
212, 214, 215, 215, 221, 223, 225,
226, 228, 229
- workpatterns_classify, 95, 114, 116, 178,
179, 217, 217, 223, 225, 226, 228,
229
- workpatterns_classify_bw, 95, 114, 116,
179, 217, 221, 221, 225, 226, 228,
229
- workpatterns_classify_pav, 95, 114, 116,
179, 217, 221, 223, 223, 226, 228,
229
- workpatterns_hclust, 95, 114, 116, 178,
179, 217, 221, 223, 225, 225, 228,
229
- workpatterns_rank, 7–9, 11–13, 21, 23, 24,
26, 28, 30, 32, 36, 38, 40, 42, 44, 47,
54, 55, 57, 59, 62, 64, 66, 67, 69, 74,
75, 77, 78, 80, 81, 86, 95, 101, 103,
104, 114, 116, 121, 127, 130–133,
135, 137, 139, 140, 142, 145, 147,
148, 150, 164, 166, 167, 169, 171,
172, 174, 177, 179, 208, 209, 211,
212, 214, 215, 217, 221, 223, 225,
226, 227, 229
- workpatterns_report, 17, 20, 29, 34, 95, 98,
114, 116, 124, 146, 179, 182, 192,
205, 217, 221, 223, 225, 226, 228,
228
- wrap, 16, 17, 32, 70, 87, 88, 99, 121, 128, 175,
180–182, 184, 198–200, 202, 203,
229