Package 'wpa'

June 6, 2024

Type Package

Title Tools for Analysing and Visualising Viva Insights Data

Version 1.9.1

Description Opinionated functions that enable easier and faster analysis of Viva Insights 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 Viva Insights 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 Viva Insights 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/, https://microsoft.github.io/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 7.3.1

Suggests knitr, extrafont, lifecycle, fst, glue, flexdashboard, lmtest, sandwich, testthat (>= 3.0.0)

Language en-US

Config/testthat/edition 3

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 2024-06-06 13:20:02 UTC

Contents

`afterhours_dist` *Distribution of After-hours Collaboration Hours as a 100% stacked bar*

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: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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 After-hours Collaboration: `afterhours_fizz()`, `afterhours_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `external_rank()`

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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 After-hours Collaboration: afterhours_dist(), afterhours_line(), afterhours_rank(), afterhours_summary(), afterhours_trend(), external_rank()

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: • `"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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 After-hours Collaboration: afterhours_dist(), afterhours_fizz(), afterhours_rank(), afterhours_summary(), afterhours_trend(), external_rank()

Examples

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

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

---

afterhours_rank *Rank groups with high After-Hours Collaboration Hours*

Description

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

Usage

```
afterhours_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: • "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: • "plot" (default) • "table" 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 After-hours Collaboration: `afterhours_dist()`, `afterhours_fizz()`, `afterhours_line()`, `afterhours_summary()`, `afterhours_trend()`, `external_rank()`

`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

`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: • `"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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 After-hours Collaboration: afterhours_dist(), afterhours_fizz(), afterhours_line(), afterhours_rank(), afterhours_trend(), external_rank()

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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 After-hours Collaboration: afterhours_dist(), afterhours_fizz(), afterhours_line(), afterhours_rank(), afterhours_summary(), external_rank()

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 *Calculate Weight of Evidence (WOE) and Information Value (IV) between a single predictor and a single outcome variable.*

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` *Convert "CamelCase" to "Camel Case"*

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 *Generate a Capacity report in HTML*

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 Viva Insights,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, *excluding the file extension*. 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 *Check whether a data frame contains all the required variable*

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: • Standard Person Query • Ways of Working Assessment Query • Hourly Collaboration Query All person-level query have a `PersonId` column and a `Date` column.
`return`	String specifying what to return. This must be one of the following strings: • `"message"` (default) • `"text"` See `Value` for more information.
`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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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 Viva Insights, 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, *excluding the file extension*. 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` *Collaboration - Stacked Area Plot*

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: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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 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: • `"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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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

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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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 *Collaboration Ranking*

Description

This function scans a standard query output for groups with high levels of 'Weekly Digital Collaboration'. Returns a plot by default, with an option to return 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: • "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: • "plot" (default) • "table" 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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,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, *excluding the file extension*. 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 *Collaboration Summary*

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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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 *Add comma separator for thousands*

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` *Generate a Connectivity 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 **connectivity** metrics, including external/internal network size vs breadth (`Networking_outside_organization`, `Networking_outside_domain`).

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, *excluding the file extension*. 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 write.table().

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))),
  legend_lab = NULL,
  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: • "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 rgb2hex().
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.
legend_lab	String. Option to override legend title/label. Defaults to NULL, where the metric name will be populated instead.
rank	String specifying how to rank the bars. Valid inputs are:

- "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_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), 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

`data`	Plotting data as a data frame.
`group_var`	String containing name of variable for the group.
`bar_var`	String containing name of variable representing the value of the bars.
`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).
percent	Logical value to determine whether to show labels as percentage signs. Defaults to FALSE.
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 rgb2hex().
rounding	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_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), 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:

- `"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()`, `create_bar_asis()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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 Flexible: `create_bar()`, `create_bar_asis()`, `create_bubble()`, `create_density()`, `create_dist()`, `create_fizz()`, `create_hist()`, `create_inc()`, `create_line()`, `create_line_asis()`, `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` *Create a bubble plot with two selected Viva Insights metrics (General Purpose), with size representing the number of employees in the group.*

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()`, `create_bar_asis()`, `create_boxplot()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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 Flexible: `create_bar()`, `create_bar_asis()`, `create_boxplot()`, `create_density()`, `create_dist()`, `create_fizz()`, `create_hist()`, `create_inc()`, `create_line()`, `create_line_asis()`, `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_density *Create a density plot for any metric*

Description

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

Usage

```
create_density(
  data,
  metric,
  hrvar = "Organization",
  mingroup = 5,
  ncol = NULL,
  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.
ncol	Numeric value setting the number of columns on the plot. Defaults to NULL (automatic).
return	String specifying what to return. This must be one of the following strings: • "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 density plot 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), create_period_scatter(), create_rank(), create_sankey(), create_scatter(), create_stacked(), create_tracking(), create_trend(), period_change()

Examples

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

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

# Return plot but coerce plot to two columns
create_density(sq_data, metric = "Collaboration_hours", hrvar = "Organization", ncol = 2)

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

---

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: • "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)
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 cut_hour() 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), 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, percent = FALSE)
```

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.
percent	Logical value specifying whether to display numeric columns as percentages.

Value

Returns an HTML widget displaying rectangular data.

See Also

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

Examples

```
out_tb <- hrvar_count(sq_data, hrvar = "Organization", return = "table")
create_dt(out_tb)
```

---

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

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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_hist(), create_inc(), create_line(), create_line_asis(), 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,
  ncol = NULL,
  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 `ggplot2::geom_histogram()`. Defaults to 1.
`ncol`	Numeric value setting the number of columns on the plot. Defaults to `NULL` (automatic).
`return`	String specifying what to return. This must be one of the following strings: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_density()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `create_line_asis()`, `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 plot but coerce plot to two columns
create_hist(sq_data, metric = "Collaboration_hours", hrvar = "Organization", ncol = 2)

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

`create_inc` *Create an incidence analysis reflecting proportion of population scoring above or below a threshold for a metric*

Description

An incidence analysis is generated, with each value in the table reflecting the proportion of the population that is above or below a threshold for a specified metric. There is an option to only provide a single `hrvar` in which a bar plot is generated, or two `hrvar` values where an incidence table (heatmap) is generated.

Usage

```
create_inc(
  data,
  metric,
  hrvar,
  mingroup = 5,
  threshold,
  position,
  return = "plot"
)

create_incidence(
  data,
  metric,
  hrvar,
  mingroup = 5,
  threshold,
  position,
  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	Character vector of at most length 2 containing the name of the HR Variable by which to split metrics.
mingroup	Numeric value setting the privacy threshold / minimum group size. Defaults to 5.
threshold	Numeric value specifying the threshold.
position	String containing the below valid values: • "above": show incidence of those equal to or above the threshold • "below": show incidence of those equal to or below the threshold
return	String specifying what to return. This must be one of the following strings: • "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 heat map.
- "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_line(), create_line_asis(), create_period_scatter(), create_rank(), create_sankey(), create_scatter(), create_stacked(), create_tracking(), create_trend(), period_change()

Examples

```
# Only a single HR attribute
create_inc(
  data = sq_data,
  metric = "After_hours_collaboration_hours",
  hrvar = "Organization",
  threshold = 4,
  position = "above"
)

# Two HR attributes
create_inc(
  data = sq_data,
  metric = "Collaboration_hours",
  hrvar = c("LevelDesignation", "Organization"),
  threshold = 20,
  position = "below"
)
```

`create_ITSA` *Estimate an effect of intervention on every Viva Insights metric in input file by applying single-group Interrupted Time-Series Analysis (ITSA)*

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', 'portes', 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 Viva Insights 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: • `'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' and 'lmtest'. Please install these separately from CRAN prior to running the function.

As of May 2022, the 'portes' package was archived from CRAN. The dependency has since been removed and dependent functions `Ljungbox()` incorporated into the **wpa** package.

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 = "12/15/2019",
  before_end = "12/29/2019",
  after_start = "1/5/2020",
  after_end = "1/26/2020",
  ac_lags_max = 7,
  return = "table")

# Returns list of plots

plot_list <-
  create_ITSA(
    data = sq_data,
    before_start = "12/15/2019",
    before_end = "12/29/2019",
    after_start = "1/5/2020",
    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`, where p-value calculation does not happen altogether.
`return`	String specifying what to return. This must be one of the following strings: • `"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 > 10, 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,
  ncol = NULL,
  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.
ncol	Numeric value setting the number of columns on the plot. Defaults to NULL (automatic).
return	String specifying what to return. This must be one of the following strings: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), 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 but coerce plot to two columns
sq_data %>%
  create_line(
    metric = "Collaboration_hours",
    hrvar = "Organization",
    ncol = 2
    )

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

---

create_line_asis *Create a line chart without aggregation for any metric*

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 rgb2hex().

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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), create_line_asis(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), create_rank(), create_sankey(), create_scatter(), create_stacked(), create_tracking(), create_trend(), period_change()

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

Examples

```
# Return plot
create_period_scatter(sq_data,
                      hrvar = "LevelDesignation",
                      before_start = "2019-12-15",
```

```
                    before_end = "2019-12-29",
                    after_start = "2020-01-05",
                    after_end = "2020-01-26")

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

`create_rank` *Rank all groups across HR attributes on a selected Viva Insights metric*

Description

This function scans a standard Person query output for groups with high levels of a given Viva Insights Metric. Returns a plot by default, with an option to return 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: • `"plot"` (default) • `"table"` 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), create_line_asis(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), 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 advanced insights 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 *Create a sankey chart from a two-column count table*

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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), create_line_asis(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), 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` *Create a Scatter plot with two selected Viva Insights metrics (General Purpose)*

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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), create_line_asis(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), 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))),
  legend_lab = NULL,
  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	String. Option to override plot title.
plot_subtitle	String. Option to override plot subtitle.
legend_lab	String. Option to override legend title/label. Defaults to NULL, where the metric name will be populated instead.
rank	String specifying how to rank the bars. Valid inputs are: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), create_line_asis(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), 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

`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" percentage signs. Defaults to `FALSE`.
`plot_title`	An option to override plot title.
`plot_subtitle`	An option to override plot subtitle.
`percent`	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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `create_line_asis()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), 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 Viva Insights 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,
  palette = c("steelblue4", "aliceblue", "white", "mistyrose1", "tomato1"),
  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.

palette	Character vector containing colour codes, ranked from the lowest value to the highest value. This is passed directly to ggplot2::scale_fill_gradientn().
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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), create_line_asis(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(), create_period_scatter(), create_rank(), create_sankey(), create_scatter(), create_stacked(), create_tracking(), period_change()

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

Examples

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

# custom colours
create_trend(
  sq_data,
  metric = "Collaboration_hours",
  hrvar = "LevelDesignation",
  palette = c(
    "#FB6107",
    "#F3DE2C",
    "#7CB518",
    "#5C8001"
```

```
)
)
```

---

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

`metric`	A numeric variable representing hours.
`cuts`	A numeric vector of minimum length 3 to represent the cut points required. The minimum and maximum values provided in the vector are inclusive.
`unit`	String to specify the unit of the labels. Defaults to "hours".
`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.

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 advanced insights in Viva Insights. Note that this is largely interchangeable with a **Ways of Working Assessment** query, with the exception of some additional variables and the different variable names used for `Collaboration_hours` and `Instant_Message_hours`.

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.

See Also

Other Data: `em_data`, `g2g_data`, `mt_data`, `p2p_data_sim()`, `sq_data`

`email_dist` *Distribution of Email Hours as a 100% stacked bar*

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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), create_line_asis(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `create_line_asis()`, `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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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 *Email Time Trend - Line Chart*

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: • "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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `create_line_asis()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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 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` *Email Hours Ranking*

Description

This function scans a standard query output for groups with high levels of 'Weekly Email Collaboration'. Returns a plot by default, with an option to return 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 = "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: • "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: • "plot" (default) • "table" 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_summary(), email_trend(), external_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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` *Email Summary*

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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_trend(), external_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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(), external_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Emails: email_dist(), email_fizz(), email_line(), email_rank(), email_summary()

`em_data` *Sample Hourly Collaboration data*

Description

A sample dataset representing an Hourly Collaboration query. The data is grouped by week and contains columns for unscheduled calls, IMs sent, emails sent, and meetings. There are 24 columns per collaboration signal, representing each hour of the day.

Usage

```
em_data
```

Format

A data frame with 2000 rows and 105 variables:

PersonId

Date

Unscheduled_calls_23_24

Unscheduled_calls_22_23

Unscheduled_calls_21_22

Unscheduled_calls_20_21

Unscheduled_calls_19_20

Unscheduled_calls_18_19

Unscheduled_calls_17_18

Unscheduled_calls_16_17

Unscheduled_calls_15_16

Unscheduled_calls_14_15

Unscheduled_calls_13_14

Unscheduled_calls_12_13

Unscheduled_calls_11_12

Unscheduled_calls_10_11

Unscheduled_calls_09_10

Unscheduled_calls_08_09

Unscheduled_calls_07_08

Unscheduled_calls_06_07
Unscheduled_calls_05_06
Unscheduled_calls_04_05
Unscheduled_calls_03_04
Unscheduled_calls_02_03
Unscheduled_calls_01_02
Unscheduled_calls_00_01
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
Meetings_23_24
Meetings_22_23
Meetings_21_22
Meetings_20_21
Meetings_19_20
Meetings_18_19
Meetings_17_18
Meetings_16_17
Meetings_15_16
Meetings_14_15
Meetings_13_14
Meetings_12_13
Meetings_11_12
Meetings_10_11
Meetings_09_10
Meetings_08_09
Meetings_07_08
Meetings_06_07
Meetings_05_06

Meetings_04_05

Meetings_03_04

Meetings_02_03

Meetings_01_02

Meetings_00_01

LevelDesignation

Organization

TimeZone

IsActive

WorkingStartTimeSetInOutlook

WorkingEndTimeSetInOutlook

WorkingDaysSetInOutlook ...

Value

data frame.

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

x	Data frame or 'ggplot' object to be passed through.
method	Character string specifying the method of export. Valid inputs include: • "clipboard" (default if input is data frame) • "csv" • "png" (default if input is 'ggplot' object) • "svg" • "jpeg" • "pdf"
path	If exporting a file, enter the path and the desired file name, *excluding the file extension*. For example, "Analysis/SQ Overview".
timestamp	Logical vector specifying whether to include a timestamp in the file name. Defaults to TRUE.
width	Width of the plot
height	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_dist` *Distribution of External Collaboration Hours as a 100% stacked bar*

Description

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

Usage

```
external_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: • `"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)

Details

Uses the metric `External_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_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_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 External Collaboration: external_fizz(), external_line(), external_sum()

Examples

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

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

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

---

external_fizz *Distribution of External Collaboration Hours (Fizzy Drink plot)*

Description

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

Usage

```
external_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:

- `"plot"`
- `"table"`

See `Value` for more information.

Details

Uses the metric `Collaboration_hours_external`. 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_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_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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 External Collaboration: `external_dist()`, `external_line()`, `external_sum()`

Examples

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

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

`external_line` *External Collaboration Hours Time Trend - Line Chart*

Description

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

Usage

```
external_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: • `"plot"` • `"table"` See `Value` for more information.

Details

Uses the metric `Collaboration_hours_external`.

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_line()`, `afterhours_rank()`, `afterhours_summary()`, `afterhours_trend()`, `collaboration_area()`, `collaboration_dist()`, `collaboration_fizz()`, `collaboration_line()`, `collaboration_rank()`, `collaboration_sum()`, `collaboration_trend()`, `create_bar()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist(), external_fizz(), external_network_plot(), external_rank(), external_sum(), 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(), 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 External Collaboration: external_dist(), external_fizz(), external_sum()

Examples

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

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

---

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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_rank()`, `external_sum()`, `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()`, `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 Network: `g2g_data`, `internal_network_plot()`, `network_describe()`, `network_g2g()`, `network_p2p()`, `network_summary()`, `p2p_data_sim()`

Examples

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

---

external_rank *Rank groups with high External Collaboration Hours*

Description

This function scans a Standard Person Query for groups with high levels of External Collaboration. Returns a plot by default, with an option to return a table with all groups (across multiple HR attributes) ranked by hours of External Collaboration.

Usage

```
external_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: • "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: • "plot" (default) • "table" See Value for more information.

Details

Uses the metric Collaboration_hours_external. 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_rank(), afterhours_summary(), afterhours_trend(), collaboration_area(), collaboration_dist(), collaboration_fizz(), collaboration_line(), collaboration_rank(), collaboration_sum(), collaboration_trend(), create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_sum(), 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(), 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 After-hours Collaboration: afterhours_dist(), afterhours_fizz(), afterhours_line(), afterhours_rank(), afterhours_summary(), afterhours_trend()

`external_sum` *External Collaboration Summary*

Description

Provides an overview analysis of 'External Collaboration'. Returns a stacked bar plot of internal and external collaboration. Additional options available to return a summary table.

Usage

```
external_sum(
  data,
  hrvar = "Organization",
  mingroup = 5,
  stack_colours = c("#1d327e", "#1d7e6a"),
  return = "plot"
)

external_summary(
  data,
  hrvar = "Organization",
  mingroup = 5,
  stack_colours = c("#1d327e", "#1d7e6a"),
  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.

stack_colours	A character vector to specify the colour codes for the stacked bar charts.
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.

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_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `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()`, `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 External Collaboration: `external_dist()`, `external_fizz()`, `external_line()`

Examples

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

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

---

extract_date_range *Extract date period*

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 *Extract HR attribute variables*

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: • "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()`, `hrvar_count_all()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts()`, `identify_shifts_wp()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate()`, `subject_validate_report()`, `track_HR_change()`, `validation_report()`

Examples

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

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

---

flag_ch_ratio *Flag unusual high collaboration hours to after-hours collaboration hours ratio*

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: • `"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 diagnotic 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()`, `hrvar_count_all()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts()`, `identify_shifts_wp()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate()`, `subject_validate_report()`, `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: • "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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), track_HR_change(), validation_report()

Examples

```
flag_em_ratio(sq_data)
```

---

flag_extreme *Warn for extreme values by checking against a 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,
  mode = "above",
  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.
mode	String determining mode to use for identifying extreme values. • "above": checks whether value is great than the threshold (default) • "equal": checks whether value is equal to the threshold • "below": checks whether value is below the threshold
return	String specifying what to return. This must be one of the following strings: • "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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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)

# Check for values equal to threshold
```

```
flag_extreme(sq_data, "Email_hours", person = TRUE, mode = "equal", threshold = 0)

# Check for values below threshold
flag_extreme(sq_data, "Email_hours", person = TRUE, mode = "below", threshold = 5)
```

`flag_outlooktime` *Flag unusual outlook time settings for work day start and end time*

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()`, `hrvar_count_all()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts()`, `identify_shifts_wp()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate()`, `subject_validate_report()`, `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")
```

`flex_index` *Compute a Flexibility Index based on the Hourly Collaboration Query*

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",
  mode = "binary"
)
```

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:

- 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

`active_threshold` A numeric value specifying the minimum number of signals to be greater than in order to qualify as *active*. 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:

- `"plot"`
- `"data"`
- `"table"`

See `Value` for more information.

`plot_method` Character string for determining which plot to return.

- `"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

`mode` String specifying aggregation method for plot. Only applicable when `return = "plot"`. Valid options include:

- `"binary"`: convert hourly activity into binary blocks. In the plot, each block would display as solid.
- `"prop"`: calculate proportion of signals in each hour over total signals across 24 hours, then average across all work weeks. In the plot, each block would display as a heatmap.

Details

The **Flexibility Index** is a metric that has been developed to quantify and measure flexibility using behavioural data from Viva Insights. 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 Viva Insights.

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()`, `identify_shifts_wp()`, `plot_flex_index()`, `workpatterns_area()`, `workpatterns_classify()`, `workpatterns_classify_bw()`, `workpatterns_classify_pav()`, `workpatterns_hclust()`, `workpatterns_rank()`, `workpatterns_report()`

Examples

```
# Create a sample small dataset
orgs <- c("Customer Service", "Financial Planning", "Biz Dev")
em_data <- em_data[em_data$Organization %in% orgs, ]

# 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 representing a Group-to-Group Query. The grouping organizational attribute used here is `Organization`, where the variable have been prefixed with `TimeInvestors_` and `Collaborators_` to represent the direction of collaboration.

Usage

```
g2g_data
```

Format

A data frame with 1417 rows and 7 variables:

TimeInvestors_Organization

Collaborators_Organization

Date

Meetings

Meeting_hours

Email_hours

Collaboration_hours ...

Value

data frame.

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_p2p(), network_summary(), p2p_data_sim()

---

generate_report *Generate HTML report with list inputs*

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 *Generate HTML report based on existing RMarkdown documents*

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

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

Note

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

`GetResiduals` *Extract Residuals from ARIMA, VAR, or any Simulated Fitted Time Series Model*

Description

This utility function is useful to use in the portmanteau functions, BoxPierce, MahdiMcLeod, Hosking, LiMcLeod, LjungBox, and portest. GetResiduals() function takes a fitted time-series object with class "ar", "arima0", "Arima", ("ARIMA forecast ARIMA Arima"), "lm", ("glm" "lm"), "varest", or "list". and returns the residuals and the order from the fitted object.

This method and the bottom documentation is taken directly from the original 'portes' package.

Usage

```
GetResiduals(obj)
```

Arguments

`obj`	a fitted time-series model with class "ar", "arima0", "Arima", ("ARIMA forecast ARIMA Arima"), "lm", ("glm" "lm"), "varest", or "list".

Value

List of order of fitted time series model and residuals from this model.

Author(s)

Esam Mahdi and A.I. McLeod.

Examples

```
fit <- arima(Nile, c(1, 0, 1))
GetResiduals(fit)
```

`heat_colours` *Generate a vector of* `n` *contiguous colours, as a red-yellow-green palette.*

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

n	the number of colors (>= 1) to be in the palette.
alpha	an alpha-transparency level in the range of 0 to 1 (0 means transparent and 1 means opaque)
rev	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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 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(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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` *Create count of distinct fields and percentage of employees with missing values for all HR variables*

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 Viva Insights *Data 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(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), track_HR_change(), validation_report()

Examples

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

---

hrvar_trend *Track count of distinct people over time in a specified HR variable*

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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Data Validation: check_query(), extract_hr(), flag_ch_ratio(), flag_em_ratio(), flag_extreme(), flag_outlooktime(), hr_trend(), hrvar_count(), hrvar_count_all(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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 Data Validation: check_query(), extract_hr(), flag_ch_ratio(), flag_em_ratio(), flag_extreme(), flag_outlooktime(), hrvar_count(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), track_HR_change(), validation_report()

Examples

```
# Return plot
hr_trend(dv_data)

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

`identify_churn` *Identify employees who have churned from the dataset*

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: • `"message"` (default) • `"text"` • `"data"` See `Value` for more information.
`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, Viva Insights 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()`, `hrvar_count_all()`, `hrvar_trend()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_nkw()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts()`, `identify_shifts_wp()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate()`, `subject_validate_report()`, `track_HR_change()`, `validation_report()`

Examples

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

---

identify_datefreq *Identify date frequency based on a series of dates*

Description

[Experimental]

Takes a vector of dates and identify whether the frequency is 'daily', 'weekly', or 'monthly'. The primary use case for this function is to provide an accurate description of the query type used and for raising errors should a wrong date grouping be used in the data input.

Usage

```
identify_datefreq(x)
```

Arguments

`x`	Vector containing a series of dates.

Details

Date frequency detection works as follows:

- If at least three days of the week are present (e.g., Monday, Wednesday, Thursday) in the series, then the series is classified as 'daily'
- If the total number of months in the series is equal to the length, then the series is classified as 'monthly'
- If the total number of sundays in the series is equal to the length of the series, then the series is classified as 'weekly

Value

String describing the detected date frequency, i.e.:

- 'daily'
- 'weekly'
- 'monthly'

Limitations

One of the assumptions made behind the classification is that weeks are denoted with Sundays, hence the count of sundays to measure the number of weeks. In this case, weeks where a Sunday is missing would result in an 'unable to classify' error.

Another assumption made is that dates are evenly distributed, i.e. that the gap between dates are equal. If dates are unevenly distributed, e.g. only two days of the week are available for a given week, then the algorithm will fail to identify the frequency as 'daily'.

Examples

```
start_date <- as.Date("2022/06/26")
end_date <- as.Date("2022/11/27")

# Daily
day_seq <-
  seq.Date(
    from = start_date,
    to = end_date,
    by = "day"
  )

identify_datefreq(day_seq)

# Weekly
week_seq <-
  seq.Date(
    from = start_date,
    to = end_date,
    by = "week"
  )
```

```
identify_datefreq(week_seq)

# Monthly
month_seq <-
  seq.Date(
    from = start_date,
    to = end_date,
    by = "month"
  )
identify_datefreq(month_seq)
```

`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: • `"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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythresho identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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: • "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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_nkw(), identify_outlier(), identify_privacythresholc identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), track_HR_change(), validation_report()

---

identify_nkw *Identify Non-Knowledge workers in a Person Query using Collaboration Hours*

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 (>=), 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: • `"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()`, `hrvar_count_all()`, `hrvar_trend()`, `identify_churn()`, `identify_holidayweeks()`, `identify_inactiveweeks()`, `identify_outlier()`, `identify_privacythreshold()`, `identify_query()`, `identify_shifts()`, `identify_shifts_wp()`, `identify_tenure()`, `remove_outliers()`, `standardise_pq()`, `subject_validate()`, `subject_validate_report()`, `track_HR_change()`, `validation_report()`

`identify_outlier` *Identify metric outliers over a date interval*

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

data	A Standard Person Query dataset in the form of a data frame.
group_var	A string with the name of the grouping variable. Defaults to Date.
metric	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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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: • "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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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 *Identify the query type of the passed data frame*

Description

Pass an advanced insights query 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	An advanced insights query dataset in the form of a data frame. If the data is not identified as a valid 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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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 *Identify shifts based on outlook time settings for work day start and end time*

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: • `"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()`, `hrvar_count_all()`, `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()`, `subject_validate_report()`, `track_HR_change()`, `validation_report()`

Other Working Patterns: `flex_index()`, `identify_shifts_wp()`, `plot_flex_index()`, `workpatterns_area()`, `workpatterns_classify()`, `workpatterns_classify_bw()`, `workpatterns_classify_pav()`, `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` *Identify shifts based on binary activity*

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: • 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
`active_threshold`	A numeric value specifying the minimum number of signals to be greater than in order to qualify as *active*. 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: • "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 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(), hrvar_count_all(), 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(), subject_validate_report(), track_HR_change(), validation_report()

Other Working Patterns: flex_index(), identify_shifts(), plot_flex_index(), workpatterns_area(), workpatterns_classify(), workpatterns_classify_bw(), workpatterns_classify_pav(), 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 *Tenure calculation based on different input dates, returns data summary table or histogram*

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: • `"message"` • `"text"` • `"plot"` • `"data_cleaned"` • `"data_dirty"` • `"data"` See `Value` for more information.

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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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 *Read a Workplace Analytics query in '.csv' using and create a '.fst' file in the same directory for faster reading*

Description

Uses import_wpa() to read a Workplace Analytics query in '.csv' and convert this into the serialized '.csv' 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. "/files/standard query.csv". The converted FST file will be saved in the same directory with a different file extension.
...	Additional arguments to pass to import_wpa().

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` *Import a Workplace Analytics Query*

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. `"/files/standard query.csv"`
`standardise`	logical. If TRUE, `import_wpa()` runs `standardise_pq()` 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 `data.table::fread()`. See `data.table::fread()` 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), hr_trend(), hrvar_count(), hrvar_trend(), keymetrics_scan(), meeting_dist(), meeting_fizz(), meeting_line(), meeting_quality(), meeting_rank(), meeting_summary(), meeting_trend(), meetingtype_dist(), 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 Network: external_network_plot(), g2g_data, network_describe(), network_g2g(), network_p2p(), network_summary(), p2p_data_sim()

Examples

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

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

---

is_date_format *Identify whether string is a date format*

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 *Identify the WPA metrics that have the biggest change between two periods.*

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()`, `create_line_asis()`, `create_period_scatter()`, `create_trend()`, `period_change()`

Examples

```
# Returns a data frame
sq_data %>%
  IV_by_period(
    before_start = "2019-12-15",
    before_end = "2019-12-29",
    after_start = "2020-01-05",
    after_end = "2020-01-26"
  )
```

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 `Information::create_infotables()`, 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, *excluding the file extension*. 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 *Jitter metrics in a data frame*

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

`data`	Data frame containing a query.
`cols`	Character vector containing the metrics to jitter. When set to `NULL` (default), all numeric columns in the data frame are jittered.
`...`	Additional arguments to pass to `jitter()`.

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 *Run a summary of Key Metrics from the Standard Person Query data*

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",
  low = rgb2hex(7, 111, 161),
  mid = rgb2hex(241, 204, 158),
  high = rgb2hex(216, 24, 42),
  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".
low	String specifying colour code to use for low-value metrics. Arguments are passed directly to ggplot2::scale_fill_gradient2().
mid	String specifying colour code to use for mid-value metrics. Arguments are passed directly to ggplot2::scale_fill_gradient2().
high	String specifying colour code to use for high-value metrics. Arguments are passed directly to ggplot2::scale_fill_gradient2().
textsize	A numeric value specifying the text size to show in the plot.

Value

Returns a ggplot object by default, when '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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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()

Examples

```
# Heatmap plot is returned by default
keymetrics_scan(sq_data)

# Heatmap plot with custom colours
keymetrics_scan(sq_data, low = "purple", high = "yellow")

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

---

keymetrics_scan_asis *Run a summary of Key Metrics without aggregation*

Description

Return a heatmapped table directly from the aggregated / summarised data. Unlike `keymetrics_scan()` which performs a person-level aggregation, there is no calculation for `keymetrics_scan_asis()` and the values are rendered as they are passed into the function.

Usage

```
keymetrics_scan_asis(
  data,
  row_var,
  col_var,
  group_var = col_var,
  value_var = "value",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab = row_var,
```

```
  xlab = "Metrics",
  rounding = 1,
  low = rgb2hex(7, 111, 161),
  mid = rgb2hex(241, 204, 158),
  high = rgb2hex(216, 24, 42),
  textsize = 2
)
```

Arguments

data	data frame containing data to plot. It is recommended to provide data in a 'long' table format where one grouping column forms the rows, a second column forms the columns, and a third numeric columns forms the
row_var	String containing name of the grouping variable that will form the rows of the heatmapped table.
col_var	String containing name of the grouping variable that will form the columns of the heatmapped table.
group_var	String containing name of the grouping variable by which heatmapping would apply. Defaults to `col_var`.
value_var	String containing name of the value variable that will form the values of the heatmapped table. Defaults to `"value"`.
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).
rounding	Numeric value to specify number of digits to show in data labels
low	String specifying colour code to use for low-value metrics. Arguments are passed directly to `ggplot2::scale_fill_gradient2()`.
mid	String specifying colour code to use for mid-value metrics. Arguments are passed directly to `ggplot2::scale_fill_gradient2()`.
high	String specifying colour code to use for high-value metrics. Arguments are passed directly to `ggplot2::scale_fill_gradient2()`.
textsize	A numeric value specifying the text size to show in the plot.

Value

ggplot object for a heatmap table.

Examples

```
library(dplyr)

# Compute summary table
out_df <-
```

```
  sq_data %>%
  group_by(Organization) %>%
  summarise(
    across(
      .cols = c(
        Workweek_span,
        Collaboration_hours
        ),
      .fns = ~median(., na.rm = TRUE)
      ),
      .groups = "drop"
    ) %>%
tidyr::pivot_longer(
  cols = c("Workweek_span", "Collaboration_hours"),
  names_to = "metrics"
)

keymetrics_scan_asis(
  data = out_df,
  col_var = "metrics",
  row_var = "Organization"
)

# Show data the other way round
keymetrics_scan_asis(
  data = out_df,
  col_var = "Organization",
  row_var = "metrics",
  group_var = "metrics"
)
```

LjungBox *Ljung and Box Portmanteau Test*

Description

The Ljung-Box (1978) modified portmanteau test. In the multivariate time series, this test statistic is asymptotically equal to Hosking.

This method and the bottom documentation is taken directly from the original 'portes' package.

Usage

```
LjungBox(
  obj,
  lags = seq(5, 30, 5),
  order = 0,
  season = 1,
  squared.residuals = FALSE
)
```

Arguments

obj	a univariate or multivariate series with class "numeric", "matrix", "ts", or ("mts" "ts"). It can be also an object of fitted time-series model with class "ar", "arima0", "Arima", ("ARIMA forecast ARIMA Arima"), "lm", ("glm" "lm"), or "varest". obj may also an object with class "list" (see details and following examples).
lags	vector of lag auto-cross correlation coefficients used for Hosking test.
order	Default is zero for testing the randomness of a given sequence with class "numeric", "matrix", "ts", or ("mts" "ts"). In general order equals to the number of estimated parameters in the fitted model. If obj is an object with class "ar", "arima0", "Arima", "varest", ("ARIMA forecast ARIMA Arima"), or "list" then no need to enter the value of order as it will be automatically determined. For obj with other classes, the order is needed for degrees of freedom of asymptotic chi-square distribution.
season	seasonal periodicity for testing seasonality. Default is 1 for testing the non seasonality cases.
squared.residuals	if TRUE then apply the test on the squared values. This checks for Autoregressive Conditional Heteroscedastic, ARCH, effects. When squared.residuals = FALSE, then apply the test on the usual residuals.

Details

However the portmanteau test statistic can be applied directly on the output objects from the built in R functions ar(), ar.ols(), ar.burg(), ar.yw(), ar.mle(), arima(), arim0(), Arima(), auto.arima(), lm(), glm(), and VAR(), it works with output objects from any fitted model. In this case, users should write their own function to fit any model they want, where they may use the built in R functions FitAR(), garch(), garchFit(), fracdiff(), tar(), etc. The object obj represents the output of this function. This output must be a list with at least two outcomes: the fitted residual and the order of the fitted model (list(res = ..., order = ...)). See the following example with the function FitModel().

Note: In stats R, the function Box.test was built to compute the Box and Pierce (1970) and Ljung and Box (1978) test statistics only in the univariate case where we can not use more than one single lag value at a time. The functions BoxPierce and LjungBox are more accurate than Box.test function and can be used in the univariate or multivariate time series at vector of different lag values as well as they can be applied on an output object from a fitted model described in the description of the function BoxPierce.

Value

The Ljung and Box test statistic with the associated p-values for different lags based on the asymptotic chi-square distribution with k^2(lags-order) degrees of freedom.

Author(s)

Esam Mahdi and A.I. McLeod

References

Ljung, G.M. and Box, G.E.P (1978). "On a Measure of Lack of Fit in Time Series Models". Biometrika, 65, 297-303.

Examples

```
x <- rnorm(100)
LjungBox(x) # univariate test

x <- cbind(rnorm(100),rnorm(100))
LjungBox(x) # multivariate test
```

`map_IV` *Calculate Weight of Evidence (WOE) and Information Value (IV) between multiple predictors and a single outcome variable, returning a list of statistics.*

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 *Max-Min Scaling Function*

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 *Distribution of Meeting Types by number of Attendees and Duration*

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

`data`	Data frame. If a meeting query, must contain the variables `Attendee` and `DurationHours`.
`hrvar`	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.
`mingroup`	Numeric value setting the privacy threshold / minimum group size. Defaults to 5. Only applicable when using a Ways of Working Assessment query.
`return`	String specifying what to return. This must be one of the following strings: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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: • `"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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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(), meetingtype_dist_mt(), 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

data	Meeting Query data frame. Must contain the variables Attendee and DurationHours
return	String specifying what to return. This must be one of the following strings: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), 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(), meetingtype_dist_ca(), meetingtype_summary()

---

meetingtype_summary *Create a summary bar chart of the proportion of Meeting Hours spent in Long or Large Meetings*

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

data	Ways of Working Assessment query in the form of a data frame. Requires the following variables: • Bloated_meeting_hours • Lengthy_meeting_hours

- Workshop_meeting_hours
- 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:

- "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 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt()

`meeting_dist` *Distribution of Meeting Hours as a 100% stacked bar*

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: • `"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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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_extract(), meeting_fizz(), meeting_line(), meeting_quality(), meeting_rank(), meeting_skim(), meeting_summary(), meeting_tm_report(), meeting_trend(), meetingtype_dist(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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: • "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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(),

external_rank(), external_sum(), 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(), 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_line(), meeting_quality(), meeting_rank(), meeting_skim(), meeting_summary(), meeting_tm_report(), meeting_trend(), meetingtype_dist(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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 *Meeting Time Trend - Line Chart*

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: • "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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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_quality()`, `meeting_rank()`, `meeting_skim()`, `meeting_summary()`, `meeting_tm_report()`, `meeting_trend()`, `meetingtype_dist()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `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 *Run a meeting habits / meeting quality analysis*

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: • `"Low_quality_meeting_hours"` (default) • `"After_hours_meeting_hours"` • `"Conflicting_meeting_hours"` • `"Multitasking_meeting_hours"` • Any *meeting hour* variable that can be divided by `Meeting_hours` If the provided metric name is not found in the data, the function will use the first matched metric from the above list.
`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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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(), 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_rank(), meeting_skim(), meeting_summary(), meeting_tm_report(), meeting_trend(), meetingtype_dist(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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 *Meeting Hours Ranking*

Description

This function scans a standard query output for groups with high levels of Weekly Meeting Collaboration. Returns a plot by default, with an option to return 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 = "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: • "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: • "plot" (default) • "table" 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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_skim(), meeting_summary(), meeting_tm_report(), meeting_trend(), meetingtype_dist(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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

data	A standard person query data in the form of a data frame.
return	String specifying what to return. This must be one of the following strings: • "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()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `meetingtype_summary()`

Examples

```
meeting_skim(sq_data)
```

---

meeting_summary *Meeting Summary*

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: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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_tm_report()`, `meeting_trend()`, `meetingtype_dist()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `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 *Generate a Meeting Text Mining report in HTML*

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, *excluding the file extension*. For example, "meeting text mining report".
stopwords	A character vector OR 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), meetingtype_summary()

Other Text-mining: pairwise_count(), subject_validate(), subject_validate_report(), tm_clean(), tm_cooc(), tm_freq(), tm_wordcloud()

---

meeting_trend *Meeting Hours Time Trend*

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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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(), meetingtype_dist(), meetingtype_dist_ca(), meetingtype_dist_mt(), meetingtype_summary()

`mgrcoatt_dist` *Manager meeting coattendance distribution*

Description

Analyze degree of attendance between employes 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: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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.

See Also

Other Data: `dv_data`, `em_data`, `g2g_data`, `p2p_data_sim()`, `sq_data`

network_describe *Uncover HR attributes which best represent a population for a Person to Person query*

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 `network_p2p()`.
`hrvar`	Character vector of length 3 containing the HR attributes to be used. Defaults to `c("Organization", "LevelDesignation", "FunctionType")`.

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_p2p()`, `network_summary()`, `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(
    community = "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()
```

---

network_g2g *Create a network plot with the group-to-group query*

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

`data`	Data frame containing a G2G query.
`time_investor`	String containing the variable name for the Time Investor column.
`collaborator`	String containing the variable name for the Collaborator column.
`metric`	String containing the variable name for metric. Defaults to `Collaboration_hours`.
`algorithm`	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.
`node_colour`	String or named vector 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. • If a named vector is supplied, the names must match the values of the variable provided for the `time_investor` and `collaborator` columns. See example section for details.
`exc_threshold`	Numeric value between 0 and 1 specifying the exclusion threshold to apply. Defaults to 0.1, which means that the plot will only display collaboration above 10% of a node's total collaboration. This argument has no impact on `"data"` or `"table"` return.
`org_count`	Optional data frame to provide the size of each organization in the `collaborator` attribute. The data frame should contain only two columns: • 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: • `"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_p2p()`, `network_summary()`, `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 - custom-specific colours
# Get labels of orgs and assign random colours
org_str <- unique(g2g_data$TimeInvestors_Organization)

col_str <-
  sample(
    x = c("red", "green", "blue"),
    size = length(org_str),
    replace = TRUE
  )

# Create and supply a named vector to `node_colour`
names(col_str) <- org_str

g2g_data %>%
  network_g2g(node_colour = col_str)


# 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_p2p *Perform network analysis 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",
  return = "plot",
  centrality = NULL,
  community = NULL,
  weight = NULL,
  comm_args = NULL,
  layout = "mds",
  path = paste("p2p", NULL, sep = "_"),
  style = "igraph",
  bg_fill = "#FFFFFF",
  font_col = "grey20",
  legend_pos = "right",
  palette = "rainbow",
  node_alpha = 0.7,
  edge_alpha = 1,
  edge_col = "#777777",
  node_sizes = c(1, 20),
  seed = 1
)
```

Arguments

data	Data frame containing a person-to-person query.
hrvar	String containing the label for the HR attribute.
return	A different output is returned depending on the value passed to the return argument: • 'plot' (default) • 'plot-pdf' • 'sankey' • 'table' • 'data' • 'network'
centrality	string to determines which centrality measure is used to scale the size of the nodes. All centrality measures are automatically calculated when it is set to one of the below values, and reflected in the 'network' and 'data' outputs. Measures include: • betweenness • closeness • degree • eigenvector • pagerank When centrality is set to NULL, no centrality is calculated in the outputs and all the nodes would have the same size.
community	String determining which community detection algorithms to apply. Valid values include: • NULL (default): compute analysis or visuals without computing communities. • "louvain" • "leiden" • "edge_betweenness" • "fast_greedy" • "fluid_communities" • "infomap" • "label_prop" • "leading_eigen" • "optimal" • "spinglass" • "walk_trap" These values map to the community detection algorithms offered by igraph. For instance, "leiden" is based on igraph::cluster_leiden(). Please see the bottom of https://igraph.org/r/html/1.3.0/cluster_leiden.html on all applications and parameters of these algorithms. .

`weight` String to specify which column to use as weights for the network. To create a graph without weights, supply `NULL` to this argument.

`comm_args` list containing the arguments to be passed through to igraph's clustering algorithms. Arguments must be named. See examples section on how to supply arguments in a named list.

`layout` 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.

`style` String to specify which plotting style to use for the network plot. Valid values include:

- `"igraph"`
- `"ggraph"`

`bg_fill` String to specify background fill colour.

`font_col` String to specify font colour.

`legend_pos` String to specify position of legend. Defaults to `"right"`. See `ggplot2::theme()`. This is applicable for both the 'ggraph' and the fast plotting method. Valid inputs include:

- `"bottom"`
- `"top"`
- `"left"` -`"right"`

`palette` String specifying the function to generate 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.

`edge_col` String to specify edge link colour.

`node_sizes` Numeric vector of length two to specify the range of node sizes to rescale to, when `centrality` is set to a non-null value.

`seed` Seed for the random number generator passed to either `set.seed()` when the louvain or leiden community detection algorithm is used, to ensure consistency. Only applicable when `community` is set to one of the valid non-null values.

Value

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

- `'plot'`: return a network plot, interactively within R.
- `'plot-pdf'`: save a network plot as PDF. This option is recommended when the graph is large, which make take a long time to run if `return = 'plot'` is selected. Use this together with `path` to control the save location.

- 'sankey': return a sankey plot combining communities and HR attribute. This is only valid if a community detection method is selected at community.
- 'table': return a vertex summary table with counts in communities and HR attribute. When centrality is non-NULL, the average centrality values are calculated per group.
- 'data': return a vertex data file that matches vertices with communities and HR attributes.
- 'network': return 'igraph' object.

See Also

Other Network: external_network_plot(), g2g_data, internal_network_plot(), network_describe(), network_g2g(), network_summary(), p2p_data_sim()

Examples

```
p2p_df <- p2p_data_sim(dim = 1, size = 100)

# default - ggraph visual
network_p2p(data = p2p_df, style = "ggraph")

# return vertex table
network_p2p(data = p2p_df, return = "table")

# return vertex table with community detection
network_p2p(data = p2p_df, community = "leiden", return = "table")

# leiden - igraph style with custom resolution parameters
network_p2p(data = p2p_df, community = "leiden", comm_args = list("resolution" = 0.1))

# louvain - ggraph style, using custom palette
network_p2p(
  data = p2p_df,
  style = "ggraph",
  community = "louvain",
  palette = "heat_colors"
)

# leiden - return a sankey visual with custom resolution parameters
network_p2p(
  data = p2p_df,
  community = "leiden",
  return = "sankey",
  comm_args = list("resolution" = 0.1)
)

# using `fluid_communities` algorithm with custom parameters
network_p2p(
  data = p2p_df,
  community = "fluid_communities",
  comm_args = list("no.of.communities" = 5)
)
```

```
# Calculate centrality measures and leiden communities, return at node level
network_p2p(
  data = p2p_df,
  centrality = "betweenness",
  community = "leiden",
  return = "data"
) %>%
  dplyr::glimpse()
```

`network_summary` *Summarise node centrality statistics with an igraph object*

Description

Pass an igraph object to the function and obtain centrality statistics for each node in the object as a data frame. This function works as a wrapper of the centralization functions in 'igraph'.

Usage

```
network_summary(graph, hrvar = NULL, return = "table")
```

Arguments

`graph`	'igraph' object that can be returned from `network_g2g()` or `network_p2p()` when the `return` argument is set to `"network"`.
`hrvar`	String containing the name of the HR Variable by which to split metrics. Defaults to `NULL`.
`return`	String specifying what output to return. Valid inputs include: • `"table"` • `"network"` • `"plot"` See `Value` for more information.

Value

By default, a data frame containing centrality statistics. Available statistics include:

- betweenness: number of shortest paths going through a node.
- closeness: number of steps required to access every other node from a given node.
- degree: number of connections linked to a node.
- eigenvector: a measure of the influence a node has on a network.
- pagerank: calculates the PageRank for the specified vertices. Please refer to the igraph package documentation for the detailed technical definition.

When `"network"` is passed to `"return"`, an 'igraph' object is returned with additional node attributes containing centrality scores.

When `"plot"` is passed to `"return"`, a summary table is returned showing the average centrality scores by HR attribute. This is currently available if there is a valid HR attribute.

See Also

Other Network: external_network_plot(), g2g_data, internal_network_plot(), network_describe(), network_g2g(), network_p2p(), p2p_data_sim()

Examples

```
# Simulate a p2p network
p2p_data <- p2p_data_sim(size = 100)
g <- network_p2p(data = p2p_data, return = "network")

# Return summary table
network_summary(graph = g, return = "table")

# Return network with node centrality statistics
network_summary(graph = g, return = "network")

# Return summary plot
network_summary(graph = g, return = "plot", hrvar = "Organization")

# Simulate a g2g network and return table
g2 <- g2g_data %>% network_g2g(return = "network")
network_summary(graph = g2, return = "table")
```

---

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,
  dist_colours = c("#facebc", "#fcf0eb", "#b4d5dd", "#bfe5ee"),
  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.
dist_colours	A character vector of length four to specify colour codes for the stacked bars.
return	String specifying what to return. This must be one of the following strings: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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 *Distribution of Manager 1:1 Time (Fizzy Drink plot)*

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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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 *Frequency of Manager 1:1 Meetings as bar or 100% stacked bar chart*

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

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:

- "plot"
- "table"

mode — String specifying what method to use. This must be one of the following strings:

- "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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: • "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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `meetingtype_dist_ca()`, `meetingtype_dist_mt()`, `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 plot by default, with an option to return a table with a all of groups (across multiple HR attributes) ranked by manager 1:1 time.

Usage

```
one2one_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: • "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: • "plot" (default) • "table" 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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 *Manager 1:1 Time Trend*

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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), meetingtype_dist_ca(), meetingtype_dist_mt(), 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_p2p(), network_summary()

Examples

```
# Simulate a p2p dataset with 800 edges
p2p_data_sim(size = 200, nei = 4)
```

`pad2` *Create the two-digit zero-padded format*

Description

Create the two-digit zero-padded format

Usage

```
pad2(x)
```

Arguments

`x`	numeric value or vector with maximum two characters.

Value

Numeric value containing two-digit zero-padded values.

`pairwise_count` *Perform a pairwise count of words by id*

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(), subject_validate_report(), 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 *Plot the distribution of percentage change between periods of a Viva Insights metric by the number of employees.*

Description

This function also presents the p-value for the null hypothesis that the variable has not changed, using a Wilcox 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 Viva Insights query output.
compvar	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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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(), workloads_dist(), workloads_fizz(), workloads_line(), workloads_rank(), workloads_summary(), workloads_trend(), workpatterns_area(), workpatterns_rank()

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

Other Flexible: create_bar(), create_bar_asis(), create_boxplot(), create_bubble(), create_density(), create_dist(), create_fizz(), create_hist(), create_inc(), create_line(), create_line_asis(),

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 = "2019-12-29")


# Run plot with more specific arguments
period_change(sq_data,
              compvar = "Workweek_span",
              before_start = "2019-12-15",
              before_end = "2019-12-29",
              after_start = "2020-01-05",
              after_end = "2020-01-26",
              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: • "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,
  mode = "binary"
)
```

Arguments

data	Data frame. Direct data output from flex_index().
sig_label	Character string for identifying signal labels.
method	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.
start_hour	See flex_index().
end_hour	See flex_index().
mode	See flex_index().

Value

ggplot object. See method.

See Also

Other Working Patterns: flex_index(), identify_shifts(), identify_shifts_wp(), workpatterns_area(), workpatterns_classify(), workpatterns_classify_bw(), workpatterns_classify_pav(), 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_hourly_pat *Internal function for plotting the hourly activity patterns.*

Description

This is used within `plot_flex_index()` and `workpatterns_rank()`.

Usage

```
plot_hourly_pat(
  data,
  start_hour,
  end_hour,
  legend,
  legend_label,
  legend_text = "Observed activity",
  rows,
  title,
  subtitle,
  caption,
  ylab = paste("Top", rows, "activity patterns")
)
```

Arguments

`data`	Data frame containing three columns: • `patternRank` • `Hours` • `Freq`
`start_hour`	Numeric value to specify expected start hour.
`end_hour`	Numeric value to specify expected end hour.
`legend`	Data frame containing the columns: • `patternRank` • Any column to be used in the grey label box, supplied to `legend_label`
`legend_label`	String specifying column to display in the grey label box
`legend_text`	String to be used in the bottom legend label.
`rows`	Number of rows to show in plot.
`title`	String to specify plot title.
`subtitle`	String to specify plot subtitle.
`caption`	String to specify plot caption.
`ylab`	String to specify plot y-axis label.

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 *Calculate the p-value of the null hypothesis that two outcomes are from the same dataset*

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 *Read preamble*

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 *Remove outliers from a person query across time*

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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), standardise_pq(), subject_validate(), subject_validate_report(), 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 4403 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.

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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), subject_validate(), subject_validate_report(track_HR_change(), validation_report()

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

`subject_classify` *Create a new logical variable that classifies meetings by patterns in subject lines*

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 *Count top words in subject lines grouped by a custom attribute*

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: • "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:

- "plot"
- "table"
- "data"

See Value for more information.

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 character vector OR a single-column data frame labelled 'word' containing custom stopwords to remove.

... — Additional parameters to pass to tm_clean().

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 *Scan meeting subject and highlight items for review*

Description

This functions scans a meeting query and highlights meetings with subjects that include common exlusion 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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), 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 exlusion 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, *excluding the file extension*. 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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), 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 *Basic theme for 'wpa' visualisations*

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 *Clean subject line text prior to analysis*

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 character vector OR 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(), subject_validate_report(), 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 character vector OR 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: • "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()`, `subject_validate_report()`, `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 character vector OR 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: • "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(), subject_validate_report(), 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 character vector OR 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: • "plot" • "table" See Value for more information.
...	Additional parameters to be passed to ggwordcloud::geom_text_wordcloud()

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(), subject_validate_report(), tm_clean(), tm_cooc(), tm_freq()

Examples

```
tm_wordcloud(mt_data, keep = 30)

# Removing stopwords
tm_wordcloud(mt_data, keep = 30, stopwords = c("weekly", "update"))
```

`totals_bind` *Row-bind an identical data frame for computing grouped totals*

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

`data`	data frame
`target_col`	Character value of the column in which to impute `"Total"`. This is usually the intended grouping column.
`target_value`	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` *Fabricate a 'Total' HR variable*

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 *Reorder a value to the top of the summary table*

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

`data`	Summary table in the form of a data frame.
`target_col`	Character value of the column in which to reorder
`target_value`	Character value of the value in `target_col` 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` *Sankey chart of organizational movement between HR attributes and missing values (outside company move) (Data Overview)*

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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), 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 *Replace underscore with space*

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 *Generate a Data Validation report in HTML*

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, *excluding the file extension*.
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 hrvar_count_all().
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.

Since v1.6.2, the variable Call_hours is no longer a pre-requisite to run this report. A note is returned in-line instead of an error if the variable is not available.

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(), hrvar_count_all(), hrvar_trend(), identify_churn(), identify_holidayweeks(), identify_inactiveweeks(), identify_nkw(), identify_outlier(), identify_privacythreshold(), identify_query(), identify_shifts(), identify_shifts_wp(), identify_tenure(), remove_outliers(), standardise_pq(), subject_validate(), subject_validate_report(), track_HR_change()

---

wellbeing_report *Generate a Wellbeing Report in HTML*

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, *excluding the file extension*. 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 *Distribution of Work Week Span as a 100% stacked bar*

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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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_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: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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_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: • `"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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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_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` *Rank all groups across HR attributes for Work Week Span*

Description

This function scans a standard query output for groups with high levels of Work Week Span. Returns a plot by default, with an option to return 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: • "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: • "plot" (default) • "table" 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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_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` *Work Week Span Summary*

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: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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_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

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 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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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(), 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: • 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: • `"plot"` • `"table"` See `Value` for more information.
`values`	Character vector to specify whether to return percentages or absolute values in "data" and "plot". Valid values are: • `"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()`, `create_bar_asis()`, `create_boxplot()`, `create_bubble()`, `create_dist()`, `create_fizz()`, `create_inc()`, `create_line()`, `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_dist()`, `external_fizz()`, `external_line()`, `external_network_plot()`, `external_rank()`, `external_sum()`, `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()`, `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_rank()`

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

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

Examples

```
# Create a sample small dataset
orgs <- c("Customer Service", "Financial Planning", "Biz Dev")
em_data <- em_data[em_data$Organization %in% orgs, ]

# 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",
  exp_hours = NULL,
  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: • `"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.

`exp_hours` Numeric value representing the number of hours the population is expected to be active for throughout the workday. By default, this uses the difference between `end_hour` and `start_hour`. Only applicable with the 'bw' method.

`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 *active*. 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 eight archetypes:

- **0 Low Activity (< 3 hours on)**: fewer than 3 hours of active hours
- **1.1 Standard continuous (expected schedule)**: active hours equal to *expected hours*, with all activity confined within the expected start and end time
- **1.2 Standard continuous (shifted schedule)**: active hours equal to *expected hours*, with activity occurring beyond either the expected start or end time.
- **2.1 Standard flexible (expected schedule)**: active hours less than or equal to *expected hours*, with all activity confined within the expected start and end time
- **2.2 Standard flexible (shifted schedule)**: active hours less than or equal to *expected hours*, with activity occurring beyond either the expected start or end time.
- **3 Long flexible workday**: number of active hours exceed *expected hours*, with breaks occurring throughout
- **4 Long continuous workday**: number of active hours exceed *expected hours*, with activity happening in a continuous block (no breaks)
- **5 Always on (13h+)**: number of active hours greater than or equal to 13

Standard here denotes the behaviour of not exhibiting total number of active hours which exceed the expected total number of hours, as supplied by `exp_hours`. *Continuous* refers to the behaviour of *not* taking breaks, i.e. no inactive hours between the first and last active hours of the day, where *flexible* refers to the contrary.

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** - archetypes 1.1 and 1.2
- **Flexible** - archetypes 2.1 and 2.2
- **Long continuous** - archetype 4
- **Long flexible** - archetype 3
- **Always On** - archetype 5

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()`, `identify_shifts_wp()`, `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
# With custom expected hours
em_data %>%
  workpatterns_classify(
    method = "bw",
    return = "plot-area",
    exp_hours = 7
      )

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,
  exp_hours = NULL,
  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: • 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.
`exp_hours`	Numeric value representing the number of hours the population is expected to be active for throughout the workday. By default, this uses the difference between `end_hour` and `start_hour`.
`active_threshold`	A numeric value specifying the minimum number of signals to be greater than in order to qualify as *active*. Defaults to 0.
`return`	Character vector to specify what to return. Valid options include: • `"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(), identify_shifts_wp(), plot_flex_index(), workpatterns_area(), workpatterns_classify(), workpatterns_classify_pav(), 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: • "percent": percentage of signals divided by total signals (default) • "abs": absolute count of signals
signals	Character vector to specify which collaboration metrics to use:

- "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:

- `"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()`, `identify_shifts_wp()`, `plot_flex_index()`, `workpatterns_area()`, `workpatterns_classify()`, `workpatterns_classify_bw()`, `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: • `"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: • "percent": percentage of signals divided by total signals (default) • "abs": absolute count of signals
`signals`	Character vector to specify which collaboration metrics to use: • `"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"

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()`, `identify_shifts_wp()`, `plot_flex_index()`, `workpatterns_area()`, `workpatterns_classify()`, `workpatterns_classify_bw()`, `workpatterns_classify_pav()`, `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,
  mode = "binary",
  return = "plot"
)
```

Arguments

data	A data frame containing hourly collaboration data.
signals	Character vector to specify which collaboration metrics to use: • "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"
top	numeric value specifying how many top working patterns to display in plot, e.g. "10"
mode	string specifying aggregation method for plot. Valid options include: • "binary": convert hourly activity into binary blocks. In the plot, each block would display as solid. • "prop": calculate proportion of signals in each hour over total signals across 24 hours, then average across all work weeks. In the plot, each block would display as a heatmap.
return	String specifying what to return. This must be one of the following strings: • "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(), create_bar_asis(), create_boxplot(), create_bubble(), create_dist(), create_fizz(), create_inc(), create_line(), 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_dist(), external_fizz(), external_line(), external_network_plot(), external_rank(), external_sum(), 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(), 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()

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

Examples

```
# Plot by default
workpatterns_rank(
  data = em_data,
  signals = c(
    "email",
    "IM",
    "unscheduled_calls",
    "meetings"
  )
  )

# Plot with prop / heatmap mode
workpatterns_rank(
  data = em_data,
  mode = "prop"
)
```

---

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",
  exp_hours = NULL,
  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 workpatterns_classify().
start_hour	See workpatterns_classify().
end_hour	See workpatterns_classify().
exp_hours	See workpatterns_classify().
path	Pass the file path and the desired file name, *excluding the file extension*. 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(), identify_shifts_wp(), plot_flex_index(), workpatterns_area(), workpatterns_classify(), workpatterns_classify_bw(), workpatterns_classify_pav(), 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()

---

wrap_text *Wrap text based on character threshold*

Description

Wrap text in visualizations according to a preset character threshold. The next space in the string is replaced with \n, which will render as next line in plots and messages.

Usage

```
wrap_text(x, threshold = 15)
```

Arguments

x	String to wrap text
threshold	Numeric, defaults to 15. Number of character units by which the next space would be replaced with \n to move text to next line.

Examples

```
wrapped <- wrap_text(
  "The total entropy of an isolated system can never decrease."
  )
message(wrapped)
```

Index