Package: ggfacto (via r-universe)

September 2, 2024

Title Graphs for Correspondence Analysis

Version 0.3.1

Description Readable, complete and pretty graphs for correspondence analysis made with 'FactoMineR'. They can be rendered as interactive 'HTML' plots, showing useful informations at mouse hover. The interest is not mainly visual but statistical: it helps the reader to keep in mind the data contained in the cross-table or Burt table while reading the correspondence analysis, thus preventing over-interpretation. Most graphs are made with 'ggplot2', which means that you can use the + syntax to manually add as many graphical pieces you want, or change theme elements. 3D graphs are made with 'plotly'.

URL https://github.com/BriceNocenti/ggfacto

BugReports https://github.com/BriceNocenti/ggfacto/issues

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports FactoMineR (>= 2.0.0), ggiraph (>= 0.8.2), ggplot2 (>= 3.0.0), dplyr (>= 1.0.0), forcats (>= 0.5.0), purrr (>= 0.3.0), rlang (>= 0.4.0), stringr (>= 1.4.0), tibble (>= 3.0.0), tidyr (>= 1.0.0), tidyselect (>= 1.1.0), vctrs (>= 0.3.0), ggrepel (>= 0.9.0), gridExtra (>= 2.0), tabxplor (>= 1.0.3), withr (>= 2.0.0), magrittr (>= 1.5.0), data.table (>= 1.12.0), ggforce (>= 0.4.0)

Suggests widgetframe (>= 0.3.0), htmlwidgets (>= 1.4.0), finalfit (>= 1.0.0), kableExtra (>= 1.3.0), plotly (>= 4.1.0), stats (>= 4.0.0), grDevices (>= 4.0.0), stringi (>= 1.4.6), scales (>= 1.1.0)

Repository https://bricenocenti.r-universe.dev

RemoteUrl https://github.com/bricenocenti/ggfacto

RemoteRef HEAD

RemoteSha 98750898eb5ef6886d7e6987164dc850a03ec38a

Contents

---

benzecri_mrv	*Benzecri's modified rate of variance*

Description

Benzecri's modified rate of variance

Usage

```
benzecri_mrv(res.mca, fmt = FALSE)
```

Arguments

res.mca	The result of [MCA](#).
fmt	By default, the result is given as a numeric vector. Set to 'TRUE' to have a **tabxplor** link[tabxplor]{fmt} vector instead.

Value

A numeric vector (or fmt vector with 'fmt = TRUE').

Examples

```
data(tea, package = "FactoMineR")
res.mca <- MCA2(tea, active_vars = 1:18)
benzecri_mrv(res.mca)
```

---

ggca   *Readable and Interactive graph for simple correspondence analysis*

Description

A readable, complete and beautiful graph for simple correspondence analysis made with `FactoMineR::CA`. Interactive tooltips, appearing when hovering on points with mouse, allow to keep in mind all the content of the table while reading the graph. Since it is made in the spirit of ggplot2, it is possible to change theme or add another plot elements with +. Then, interactive tooltips won't appear until you pass the result through [ggi](ggi).

Usage

```
ggca(
  res.ca = res.ca,
  axes = c(1, 2),
  show_sup = FALSE,
  xlim,
  ylim,
  out_lims_move = FALSE,
  type = c("points", "text", "labels"),
  text_repel = FALSE,
  uppercase = "col",
  tooltips = c("row", "col"),
  rowtips_subtitle = "Row pct",
  coltips_subtitle = "Column pct",
  rowcolor_numbers = 0,
  colcolor_numbers = 0,
  cleannames = TRUE,
  filter = "",
  title,
  text_size = 3.5,
  dist_labels = c("auto", 0.12),
  right_margin = 0,
  size_scale_max = 8,
  use_theme = TRUE
)
```

Arguments

res.ca	An object created with `FactoMineR::CA`.
axes	The axes to print, as a numeric vector of length 2.

show_sup	When TRUE show supplementary rows and cols.
xlim, ylim	Horizontal and vertical axes limits, as double vectors of length 2.
out_lims_move	When TRUE, the points out of xlim or ylim are not removed, but moved at the edges of the graph.
type	Determines the way the two variables of the table are printed.

  - "points" : colored points with text legends
  - "text" : colored text
  - "labels" : colored labels

text_repel	When TRUE the graph is not interactive anymore, but the resulting image is better to print because points and labels don't overlaps. It uses ggrepel::geom_text_repel.
uppercase	Print "row" var or "col" var labels with uppercase.
tooltips	Choose the content of interactive tooltips at mouse hover : "col" for the table of columns percentages, "row" for line percentages, default to c("row", "col") for both.
rowtips_subtitle, coltips_subtitle	
	The subtitles used before the table in interactive tooltips.
rowcolor_numbers, colcolor_numbers	
	If row var or col var levels are prefixed with numbers(ex. : "1-"), the number of digits to use to create classes that will be used to add colors to points.
cleannames	Set to TRUE to clean levels names, by removing prefix numbers like "1-", and text in parentheses.
filter	Regex patterns to discard levels of row or col variables.
title	The title of the graph.
text_size	Size of text.
dist_labels	When type = "points", the distance of text and labels from points.
right_margin	A margin at the right, in cm. Useful to read tooltips over points placed at the right of the graph without formatting problems.
size_scale_max	Size of points.
use_theme	By default, a specific ggplot2 theme is used. Set to FALSE to customize your own theme.

Value

A ggplot object to be printed in the 'RStudio' Plots pane. Possibility to add other gg objects with +. Sending the result through ggi will draw the interactive graph in the Viewer pane using ggiraph.

Examples

```
# Make the correspondence analysis :

tabs <- table(forcats::gss_cat$race, forcats::gss_cat$marital)[-4,]
# tabs <- tabxplor::tab_plain(forcats::gss_cat, race, marital, df = TRUE)
res.ca <- FactoMineR::CA(tabs, graph = FALSE)
```

```
# Interactive plot :
graph.ca <- ggca(res.ca,
                 title = "Race by marital : correspondence analysis",
                 tooltips = c("row", "col"))
ggi(graph.ca) #to make the plot interactive

# Image plot :
ggca(res.ca,
     title = "Race by marical : correspondence analysis",
     text_repel = TRUE)
```

---

ggi   *Pass a MCA plot into a html interactive plot*

Description

Pass a MCA plot into a html interactive plot

Usage

```
ggi(
  plot = ggplot2::last_plot(),
  width = NULL,
  height = NULL,
  keep_ratio = TRUE,
  savewidget = FALSE,
  dir = NULL,
  name = "Plot",
  replace = FALSE,
  open = rlang::is_interactive(),
  iframe = NULL,
  pixel_width,
  ...
)
```

Arguments

plot	The plot, created with [ggmca](#) or [ggca.](#)
width	The width in centimeters. Default to printing device's size.
height	The height in centimeters. Default to printing device's size.
keep_ratio	By default, the height is forced based of the relative size of the MCA's axes. Set to FALSE to avoid this behavior.
savewidget	Should the html widget be saved on disk ?
dir	If saved as file, the directory in which to save the html widget. Default to temporary directory. Set global option "ggfacto.export_dir" with link[base:options](options) to change default directory.

name	The name of the file to save.
replace	Replace file ? By default, number added to find a new name.
open	Should the resulting file be opened at once ?
iframe	Create an html frame around the plot to ensure fixed dimensions. Useful when opening the plot in a web browser (but will produce a blank graph with **rmarkdown**). This is default behavior with savewidget = TRUE. Require package [widgetframe](#).
pixel_width	The width in pixels for [widgetframe](#).
...	Additional arguments to pass to [girafe](#) and [dsvg](#). fonts can be used to provide text fonts.

Value

An html plot.

---

ggmca   *Readable and Interactive graph for multiple correspondence analysis*

Description

A readable, complete and beautiful graph for multiple correspondence analysis made with FactoMineR::[MCA](#). Interactive tooltips, appearing when hovering near points with mouse, allow to keep in mind many important data (tables of active variables, and additional chosen variables) while reading the graph. Profiles of answers (from the graph of "individuals") are drawn in the back, and can be linked to FactoMineR::[HCPC](#) classes. Since it is made in the spirit of [ggplot2](#), it is possible to change theme or add another plot elements with +. Then, interactive tooltips won't appear until you pass the result through [ggi](#). Step-by-step functions : use [ggmca_data](#) to get the data frames with every parameter in a MCA printing, then modify, and pass to [ggmca_plot](#) to draw the graph.

Usage

```
ggmca(
  res.mca,
  dat,
  sup_vars,
  active_tables,
  tooltip_vars_1lv,
  tooltip_vars,
  axes = c(1, 2),
  axes_names = NULL,
  axes_reverse = NULL,
  type = c("text", "labels", "points", "numbers", "facets"),
  color_groups = "^.{0}",
  cah_color_groups = "^.+$",
  keep_levels,
  discard_levels,
```

```
    cleannames = TRUE,
    profiles = FALSE,
    profiles_tooltip_discard = "^Not |^No |^Pas |^Non ",
    cah,
    max_profiles = 5000,
    alpha_profiles = 0.7,
    color_profiles = TRUE,
    base_profiles_color = "#aaaaaa",
    text_repel = FALSE,
    title,
    actives_in_bold = NULL,
    sup_in_italic = FALSE,
    ellipses = NULL,
    xlim,
    ylim,
    out_lims_move = FALSE,
    shift_colors = 0,
    colornames_recode,
    scale_color_light = material_colors_light(),
    scale_color_dark = material_colors_dark(),
    text_size = 3.5,
    size_scale_max = 4,
    dist_labels = c("auto", 0.04),
    right_margin = 0,
    use_theme = TRUE,
    get_data = FALSE
)

ggmca_data(
    res.mca,
    dat,
    sup_vars,
    active_tables,
    tooltip_vars_1lv,
    tooltip_vars,
    color_groups = "^.{0}",
    cah_color_groups = "^.+$",
    keep_levels,
    discard_levels,
    cleannames = TRUE,
    profiles = FALSE,
    profiles_tooltip_discard = "^Pas |^Non |^Not |^No ",
    cah,
    max_profiles = 5000
)

ggmca_plot(
    data,
```

```
    axes = c(1, 2),
    axes_names = NULL,
    axes_reverse = NULL,
    type = c("text", "points", "labels", "active_vars_only", "numbers", "facets"),
    text_repel = FALSE,
    title,
    ellipses = NULL,
    actives_in_bold = NULL,
    sup_in_italic = FALSE,
    xlim,
    ylim,
    out_lims_move = FALSE,
    color_profiles = TRUE,
    base_profiles_color = "#aaaaaa",
    alpha_profiles = 0.7,
    shift_colors = 0,
    colornames_recode,
    scale_color_light = material_colors_light(),
    scale_color_dark = material_colors_dark(),
    text_size = 3.5,
    size_scale_max = 4,
    dist_labels = c("auto", 0.04),
    right_margin = 0,
    use_theme = TRUE,
    get_data = FALSE
)
```

Arguments

res.mca	An object created with FactoMineR::MCA.
dat	The data in which to find the supplementary variables, etc.
sup_vars	A character vectors of supplementary qualitative variables to print (they don't need to be passed in MCA before).
active_tables	Should colored crosstables be added in interactive tooltips ? 'active_tables = "sup"' crosses each 'sup_vars' with active variables. 'active_tables = "active"' crosses each active_variables with the other ones, giving results closely related with the burt table used to calculate multiple correspondance analysis. It may take time to calculate with many variables. 'active_tables = c("active", "sup")' do both. In tooltips, percentages are colored in blue when spread from mean is positive (over-representations), and in red when spread from mean is negative (under-representations), like in tab with 'color = "diff"'.
tooltip_vars_1lv	
	A character vectors of variables, whose first level (if character/factor) or weighted_mean (if numeric) will be added at the top of interactive tooltips.
tooltip_vars	A character vector of variables (character/factors), whose complete levels will be added at the bottom of interactive tooltips.
axes	The axes to print, as a numeric vector of length 2.

axes_names	Names of all the axes (not just the two selected ones), as a character vector.
axes_reverse	Possibility to reserve the coordinates of the axes by providing a numeric vector : '1' to invert left and right ; '2' to invert up and down ; '1:2' to invert both.
type	Determines the way sup_vars are printed.

- "text" : colored text
- "points" : colored points with text legends
- "labels" : colored labels
- "active_vars_only" : no sup_vars
- "numbers" : colored labels of prefix numbers, with small names
- "facets" : one graph of profiles of answer for each levels of the first sup_vars. A different color is used for each.

color_groups	By default, there is one color group for all the levels of each 'sup_vars'. It is possible to color 'sup_vars' with groups created upon their levels with [str_extract](#) and regexes. For exemple, 'color_groups = "^."' makes the groups upon the first character of each levels (uselful when their begin by numbers). color_groups = "^.{3}" upon the first three characters. color_groups = "NB.+$" takes anything between the '"NB"' and the end of levels names, etc.
cah_color_groups	
	Color groups for the 'cah' variable (HCPC clusters).
keep_levels	A character vector of variables levels to keep : others will be discarded.
discard_levels	A character vector of variables levels to discard.
cleannames	Set to TRUE to clean levels names, by removing prefix numbers like "1-", and text in parentheses.
profiles	When set to TRUE, profiles of answers are drawn in the back of the graph with light-grey points. When hovering with mouse in the interactive version (passed in [ggi](#)), the answers of individuals to active variables will appears. If cah is provided, to hover near one point will color all the points of the same [HCPC](#) class.
profiles_tooltip_discard	
	A regex pattern to remove useless levels among interactive tooltips for profiles of answers (ex. : levels expressing "no" answers).
cah	A HCPC clusters variable made with [HCPC](#) on 'res.mca', to link the answers-profiles points who share the same HCPC class (will be colored the same color and linked at mouse hover).
max_profiles	The maximum number of profiles points to print. Default to 5000.
alpha_profiles	The alpha (transparency, between 0 and 1) for profiles of answer.
color_profiles	By default, if cah is provided, profiles are colored based on cah levels (HCPC clusters). Set do FALSE to avoid this behaviour. You can also give a character vector with only some of the levels of the 'cah' variable .
base_profiles_color	
	The base color for answers profiles. Default to gray. Set to 'NULL' to discard profiles. With 'color_profiles', set to 'NULL' to discard the non-colored profiles.

text_repel	When TRUE the graph is not interactive anymore, but the resulting image is better to print because points and labels don't overlaps. It uses ggrepel::geom_text_repel.
title	The title of the graph.
actives_in_bold	
	Set to 'TRUE' to set active variables in bold font (and sup variables in plain).
sup_in_italic	Set to 'TRUE' to set sup variables in italics.
ellipses	Set to a number between 0 and 1 to draw a concentration ellipse for each level of the first sup_vars. 0.95 draw ellipses containing 95 individuals of each category. 0.5 draw median-ellipses, containing half the individuals of each category. Note that, if 'max_profiles' is provided, ellipses won't be made with all individuals.
xlim, ylim	Horizontal and vertical axes limits, as double vectors of length 2.
out_lims_move	When TRUE, the points out of xlim or ylim are not removed, but moved at the edges of the graph.
shift_colors	Change colors of the sup_vars points.
colornames_recode	
	A named character vector with fct_recode style to rename the levels of the color variable if needed (levels used for colors are printed in console message whenever the function is used).
scale_color_light	
	A scale color for sup vars points
scale_color_dark	
	A scale color for sup vars texts
text_size	Size of text.
size_scale_max	Size of points.
dist_labels	When type = points, the distance of labels from points.
right_margin	A margin at the right, in cm. Useful to read tooltips over points placed at the right of the graph without formatting problems.
use_theme	By default, a specific ggplot2 theme is used. Set to FALSE to customize your own theme.
get_data	Returns the data frame to create the plot instead of the plot itself.
data	A list of data frames made with ggmca_data.

Value

A ggplot object to be printed in the 'RStudio' Plots pane. Possibility to add other gg objects with +. Sending the result through ggi will draw the interactive graph in the Viewer pane using ggiraph.

A list containing the data frames to pass to ggmca_plot.

A ggplot object.

Functions

- ggmca_data(): get the data frames with all parameters to print a MCA graph
- ggmca_plot(): print MCA graph from data frames with parameters

Examples

```
data(tea, package = "FactoMineR")
res.mca <- MCA2(tea, active_vars = 1:18)

# Interactive graph for multiple correspondence analysis :
res.mca |>
  ggmca(tea, sup_vars = c("SPC"), ylim = c(NA, 1.2), text_repel = TRUE) |>
  ggi() #to make the graph interactive

# Interactive graph with access to all crosstables between active variables (burt table).
#  Spread from mean are colored and, usually, points near the middle will have less
#  colors, and points at the edges will have plenty. It may takes time to print, but
#  helps to interpret the MCA in close proximity with the underlying data.
res.mca |>
  ggmca(tea, ylim = c(NA, 1.2), active_tables = "active", text_repel = TRUE) |>
  ggi()

# Graph with colored HCPC clusters
cah <- FactoMineR::HCPC(res.mca, nb.clust = 6, graph = FALSE)
tea$clust <- cah$data.clust$clust
ggmca(res.mca, tea, cah = "clust", profiles = TRUE, text_repel = TRUE)

# Concentration ellipses for each levels of a supplementary variable :
ggmca(res.mca, tea, sup_vars = "SPC", ylim = c(NA, 1.2),
  ellipses = 0.5, text_repel = TRUE, profiles = TRUE)

# Graph of profiles of answer for each levels of a supplementary variable :
ggmca(res.mca, tea, sup_vars = "SPC", ylim = c(NA, 1.2),
  type = "facets", ellipses = 0.5, profiles = TRUE)
```

---

ggmca_3d   *Interactive 3D Plot for Multiple Correspondence Analyses (plotly::)*

Description

Interactive 3D Plot for Multiple Correspondence Analyses (plotly::)

Usage

```
ggmca_3d(
  res.mca,
  dat,
  cah,
  axes = 1:3,
  base_zoom = 1,
  remove_buttons = FALSE,
  cone_size = 0.15,
  view = "All",
```

```
    camera_view,
    aspectratio_from_eig = FALSE,
    title,
    ind_name.size = 10,
    max_point_size = 30,
    ...
)
```

Arguments

res.mca	An object created with FactoMineR::MCA.
dat	The data in which to find the cah variable, etc.
cah	A variable made with HCPC, to link the answers-profiles points who share the same HCPC class (will be colored the same color and linked at mouse hover).
axes	The axes to print, as a numeric vector of length 3.
base_zoom	The base level of zoom.
remove_buttons	Set to TRUE to remove buttons to change view.
cone_size	The size of the conic arrow at the end of each axe.
view	The starting point of view (in 3D) :

- "Plane 1-2" : Axes 1 and 2.
- "Plane 1-3" : Axes 1 and 3.
- "Plane 2-3" : Axes 2 and 3.
- "All" : A 3D perspective with Axes 1, 2, 3.

camera_view	Possibility to add a (replace 'view')
aspectratio_from_eig	
	Set to 'TRUE' to modify axes length based on eigenvalues.
title	The title of the graph.
ind_name.size	The size of the names of individuals.
max_point_size	The size of the biggest point.
...	Additional arguments to pass to ggmca.

Value

A plotly html interactive 3d (or 2d) graph.

Examples

```
data(tea, package = "FactoMineR")
res.mca <- MCA2(tea, active_vars = 1:18)
ggmca_3d(res.mca)

# 3D graph with colored HCPC clusters (cah)
res.mca_3axes <- MCA2(tea, active_vars = 1:18, ncp = 3)
cah <- FactoMineR::HCPC(res.mca_3axes, nb.clust = 6, graph = FALSE)
tea$clust <- cah$data.clust$clust
ggmca_3d(res.mca, dat = tea, cah = "clust")
```

---

ggmca_initial_dims	*Plot Initial Dimensions (Active Variables) of Multiple Correspondence Analysis*

Description

This function mostly have an educational value : it shows the initial dimensions of the Multiple Correspondence Analysis (active variables) in their initial reference frame. It shows the n dimensional space before the analysis is done. To see initial dimensions axes in the space built by the analysis (principal axes), use `ggmca_with_base_ref`.

Usage

```
ggmca_initial_dims(
  res.mca = res.mca,
  data,
  proj_just = c(1.5, 2),
  cleannames = TRUE,
  keep = NULL
)
```

Arguments

res.mca	An object created with FactoMineR::MCA.
data	The data in which to find the supplementary variables, etc.
proj_just	Horizontal justification of text of the coordinates on axes, as a character vector of length 2 (x and y).
cleannames	Set to TRUE to clean levels names, by removing prefix numbers like "1-", and text in parentheses.
keep	A character vector of the name of active variables to keep.

Value

A ggplot object to be printed in the 'RStudio' Plots pane. Possibility to add other gg objects with +. Sending the result through ggi will draw the interactive graph in the Viewer pane using ggiraph.

Examples

```
data(tea, package = "FactoMineR")
res.mca <- MCA2(tea, active_vars = 1:18)
ggmca_initial_dims(res.mca, data = tea)
```

---

ggmca_with_base_ref   *Plot Initial Dimensions (Active Variables) on a Multiple Correspon-*
                             dence Analyses

Description

This function mostly have an educational value : it shows the initial dimensions of the Multiple
Correspondence Analysis (active variables) in the space built by the analysis (principal axes). To
see initial dimensions in their initial reference frame, use `ggmca_initial_dims`.

Usage

```
ggmca_with_base_ref(res.mca = res.mca, axes = c(1, 2), keep = NULL)
```

Arguments

res.mca	An object created with FactoMineR::MCA.
axes	The axes to print, as a numeric vector of length 2.
keep	A character vector of the name of active variables to keep.

Value

A `ggplot` object to be printed in the 'RStudio' Plots pane. Possibility to add other gg objects with
+. Sending the result through `ggi` will draw the interactive graph in the Viewer pane using `ggiraph`.

Examples

```
data(tea, package = "FactoMineR")
res.mca <- MCA2(tea, active_vars = 1:18)
ggmca_with_base_ref(res.mca)

# It is more readable to select just a few active variables
lv2_vars <- dplyr::select(tea[1:18], where(~ nlevels(.) == 2)) |> names()
ggmca_with_base_ref(res.mca, keep = lv2_vars)

lv3_vars <- dplyr::select(tea[1:18], where(~ nlevels(.) == 3)) |> names()
ggmca_with_base_ref(res.mca, keep = lv3_vars)

lv4_vars <- dplyr::select(tea[1:18], where(~ nlevels(.) == 4)) |> names()
ggmca_with_base_ref(res.mca, keep = lv4_vars)

lv6_vars <- dplyr::select(tea[1:18], where(~ nlevels(.) == 6)) |> names()
ggmca_with_base_ref(res.mca, keep = lv6_vars)
```

---

ggpca_3d *Interactive 3D Plot for Principal Component Analyses (plotly::)*

Description

Interactive 3D Plot for Principal Component Analyses (plotly::)

Usage

```
ggpca_3d(
  res.pca,
  axes = c(1, 2, 3),
  princ_axes_print = -3:3,
  base_axe_n_breaks = 10,
  ind.size = 4,
  ind_name.size = 3,
  title,
  center = TRUE,
  var_names_on = "var",
  base_zoom = 1,
  remove_buttons = FALSE,
  cone_size = 0.33,
  view = "All",
  type = c("var", "ind", "main_plan", "projections"),
  camera_view,
  aspectratio_from_eig = FALSE,
  always_make_ind_tooltips = FALSE,
  var_color = "#4D4D4D",
  max_ind = 500,
  max_ind_seed
)
```

Arguments

res.pca	The result of `FactoMineR::PCA`.
axes	The axes to print, as a numeric vector of length 3 (or 2).
princ_axes_print	
	The breaks of the principal axes.
base_axe_n_breaks	
	The number of breaks in initial variables axes.
ind.size	The size of the points of individuals.
ind_name.size	The size of the names of individuals.
title	Plot title.
center	By default the plot is centered on the central point. Set to 'FALSE' to center on the origin of all variables (zero coordinates).

var_names_on	By default '"var"' the names of variables are drawn upon the initial axes. Set to '"cor"' to draw them upon correlation vectors instead.
base_zoom	The base level of zoom.
remove_buttons	Set to TRUE to remove buttons to change view.
cone_size	The size of the conic arrow at the end of each axe.
view	The starting point of view (in 3D) :

- "Plane 1-2" : Axes 1 and 2.
- "Plane 1-3" : Axes 1 and 3.
- "Plane 2-3" : Axes 2 and 3.
- "All" : A 3D perspective with Axes 1, 2, 3.

type	Which elements of the graph to print, among : #'

- "var" : initial variables axes, with breaks
- "cor" : normalized correlation vectors (length = 1)
- "cor_sphere" : a 3D sphere of standard deviation 1
- "ind" : points of individuals
- "ind_name" : names of individuals
- "main_plan" : the plan 1-2.
- "projections" : projections of mean point on initial variables
- "V" : vectors of the V transition matrix
- "vs" : vectors of the matrix of singular values

camera_view	Possibility to add a (replace 'view')
aspectratio_from_eig	
	Set to 'TRUE' to modify axes length based on eigenvalues.
always_make_ind_tooltips	
	Set to 'TRUE' to add interactive toolips for individuals.
var_color	The color of the initial variables/dimensions
max_ind	The maximun number of individuals to print.
max_ind_seed	The random seed used to sample individuals.

Value

A plotly html interactive 2d or 3d graph.

Examples

```
data(mtcars, package = "datasets")
mtcars <- mtcars[1:7] |> dplyr::rename(weight = wt)
res.pca <- FactoMineR::PCA(mtcars, graph = FALSE)

# Variables and individuals
ggpca_3d(res.pca)

# Circle of correlation 3D
ggpca_3d(res.pca, type = c("cor", "cor_sphere"),
```

```
        var_names_on = "cor", base_zoom = 0.6,
        princ_axes_print = -1:1, view = "All"
)
```

---

ggpca_cor_circle *Correlation Circle Plot for Principal Component Analysis*

Description

Correlation Circle Plot for Principal Component Analysis

Usage

```
ggpca_cor_circle(
  res.pca,
  axes = c(1, 2),
  proj = FALSE,
  interactive = TRUE,
  text_size = 3
)
```

Arguments

`res.pca`	The result of `FactoMineR::PCA`.
`axes`	The axes to print, as a numeric vector of length 2.
`proj`	Set to 'TRUE' to print projections of vectors over the two axes.
`interactive`	By default an html interactive plot is done. Set to 'FALSE' to get a normal `ggplot` graph.
`text_size`	Size of the texte.

Value

A `ggplot`.

Examples

```
data(mtcars, package = "datasets")
mtcars <- mtcars[1:7] |> dplyr::rename(weight = wt)
res.pca <- FactoMineR::PCA(mtcars, graph = FALSE)
ggpca_cor_circle(res.pca, interactive = FALSE)
```

---

ggsave2   *Save a plot as image*

Description

Save a plot as image

Usage

```
ggsave2(
  plot = ggplot2::last_plot(),
  dir = NULL,
  name = "Plot",
  xt = "png",
  dpi = 600,
  width = 21,
  height,
  scale = 1,
  replace = FALSE,
  open = rlang::is_interactive()
)
```

Arguments

plot	The plot, created with **ggplot2**.
dir	If saved as file, the directory in which to save the html widget. Default to temporary directory. Set global option "ggfacto.export_dir" with link[base:options]{options} to change default directory.
name	The name of the file to save.
xt	The extension name, when saving as image (interactive graph will always be .html).
dpi	The resolution.
width	The width in centimeters.
height	The height in centimeters. By default, width/1.41.
scale	Fixed ratio between horizontal and vertical axes.
replace	Replace file ? By default, number added to find a new name.
open	Should the resulting file be opened at once ?

Value

Creates a file, and opens it in 'RStudio' viewer, as a side effect.

HCPC_tab   *Multiple Tables for Hierarchical Clusters*

Description

Multiple Tables for Hierarchical Clusters

Usage

```
HCPC_tab(
  data,
  row_vars = character(),
  clust,
  wt,
  excl = character(),
  color = "diff",
  pct = "col",
  row_tot = "% of population",
  ...
)
```

Arguments

data	A data frame.
row_vars	<tidy-select> The row variables of the table, to cross with the clusters. Typically, actives variables of the MCA.
clust	In columns, the variable with the clusters, typically made with hierarchical clustering functions like HCPC (object 'res$data.clust$clust'). Can be either a symbol or a character vector of length 1 (for vars in 'data'), or an external variable (not in 'data') provided its length is equal to the number of rows of 'data'.
wt	The name of the weight variable. Leave empty for unweighted results.
excl	The name of the levels to exclude, as a character vector.
color	The type of colors to print, see tab.
pct	The type of percentages to print, see tab. Default to column percentages
row_tot	The name of the total line (frequencies of each cluster)
...	Additional arguments to pass to tab_many.

Value

A `tibble` of class `tab`, possibly with colored reading helpers.

Examples

```
data(tea, package = "FactoMineR")
res.mca_3axes <- MCA2(tea, active_vars = 1:18, ncp = 3)
cah <- FactoMineR::HCPC(res.mca_3axes, nb.clust = 6, graph = FALSE)
tea$clust <- cah$data.clust$clust
HCPC_tab(tea, row_vars = all_of(names(tea)[1:18]), clust = "clust") #|>
#tabxplor::tab_kable()
```

---

material_colors_dark   *Title Scale color dark for MCA.*

Description

Title Scale color dark for MCA.

Usage

```
material_colors_dark()
```

Value

A character vector of color codes, with color names.

Examples

```
material_colors_dark()
```

---

material_colors_light   *Title Scale color light for MCA.*

Description

Title Scale color light for MCA.

Usage

```
material_colors_light()
```

Value

A character vector of color codes, with color names.

Examples

```
material_colors_light()
```

MCA2   *Multiple Correspondence Analysis*

Description

A user-friendly wrapper around MCA, made to work better with **ggfacto** functions like ggmca. All variables can be selected by many different expressions, in the way of the 'tidyverse'. No supplementary vars are to be provided here, since they can be added afterward in ggmca.

Usage

```
MCA2(data, active_vars, wt, excl, ncp = 5, graph = FALSE, ...)
```

Arguments

data	The data frame.
active_vars	<tidy-select>
wt	<tidy-select>
excl	A character vector of regular expressions to exclude "junk" categories. Any level of an active variable with any of the detected patterns is not taken into account in the calculation of axes (which is called specific multiple correspondence analysis).
ncp	The number of axes to keep. Default to 5.
graph	By default no graph is made, since the result can be ploted with ggmca.
...	Additionnal arguments to pass to MCA.

Value

A 'res.mca' object, with all the data necessary to draw the MCA.

Examples

```
 data(tea, package = "FactoMineR")
res.mca <- MCA2(tea, active_vars = 1:18)

res.mca %>%
  ggmca(tea, sup_vars = c("SPC"), ylim = c(NA, 1.2), text_repel = TRUE) %>%
  ggi() #to make the graph interactive
```

---

mca_interpret   *Helper table to interpret multiple correspondence analysis*

Description

A table to help to interpret the meaning of axes in multiple correspondence analysis (MCA), based on Brigitte Le Roux, *Analyse geometrique des donnees multidimensionnelles*, Dunod, Paris, 2014 / Brigitte Le Roux and Henri Rouanet, *Geometric data analysis : from correspondence analysis to structured data analysis*, Kluwer, Boston, 2004. Only levels whose relative contribution to the variance of axis is superior to the mean contribution are kept. The spread between positive levels and negative levels of the same variable is calculated in percentages of the variance of the question/variable.

Usage

```
mca_interpret(
  res.mca = res.mca,
  axes = 1:min(res.mca$call$ncp, 5),
  type = c("html", "console")
)
```

Arguments

res.mca	An object created with FactoMineR::MCA,
axes	The axes to interpret, as an integer vector. Default to the first five axes.
type	By default, a html table is printed. Set to "console" to print in console or axes the numbers as a data.frame.

Value

An html table (or a `tibble`).

Examples

```
data(tea, package = "FactoMineR")
res.mca <- MCA2(tea, active_vars = 1:18)
mca_interpret(res.mca)
```

---

mean_sd_tab *Simple Mean and SD Summary*

Description

Simple Mean and SD Summary

Usage

```
mean_sd_tab(data, vars, wt)
```

Arguments

data	A data.frame.
vars	<tidy-select> The names of the numeric variables to compute means and sds with.
wt	The name of the weight variable, if needed.

Value

A data.frame.

Examples

```
mean_sd_tab(mtcars, 1:7)
```

PCA2 *Principal Component Analysis*

Description

A user-friendly wrapper around `PCA`, made to work better with **ggfacto** functions like `ggpca_cor_circle`. All variables can be selected by many different expressions, in the way of the 'tidyverse'. No supplementary vars are to be provided here, since they can be added afterward.

Usage

```
PCA2(
  data,
  active_vars,
  wt,
  col.w = NULL,
  ind_name,
  scale.unit = TRUE,
  ind.sup = NULL,
```

```
  ncp = 5,
  graph = FALSE,
  ...
)
```

Arguments

`data`	The data frame.
`active_vars`	<tidy-select> The names of the active variables.
`wt`	The name of the row weight variable
`col.w`	The weights of the columns, as a numeric vector of the same length than 'active_vars.'
`ind_name`	Possibly, a variable with the names of the individuals.
`scale.unit`	A boolean, if 'TRUE' (value set by default) then data are scaled to unit variance.
`ind.sup`	A vector indicating the indexes of the supplementary individuals.
`ncp`	Number of dimensions kept in the results (by default 5).
`graph`	A boolean, set to 'TRUE' to display the base graph.
`...`	Additional arguments to pass to `PCA`.

Value

A 'res.pca' object, with all the data necessary to draw the PCA.

Examples

```
active_vars <- c("mpg", "cyl", "hp", "drat", "qsec")
res.pca <- PCA2(mtcars, tidyselect::all_of(active_vars) )
```

---

pca_interpret	*Colored Table to Help Interpretation of Principal Component Analysis*

Description

Colored Table to Help Interpretation of Principal Component Analysis

Usage

```
pca_interpret(res.pca, axes = 1:3)
```

Arguments

`res.pca`	The result of `FactoMineR::PCA`.
`axes`	The axes to print, as a numeric vector.

Value

A tibble of class tabxplor

Examples

```
data(mtcars, package = "datasets")
mtcars <- mtcars[1:7] |> dplyr::rename(weight = wt)
res.pca <- FactoMineR::PCA(mtcars, graph = FALSE)
pca_interpret(res.pca)
```

---

pers_or_plot   *Modified odd ratios plot from 'finalfit'*

Description

Modified odd ratios plot from 'finalfit'

Usage

```
pers_or_plot(
  .data,
  dependent,
  explanatory,
  random_effect = NULL,
  factorlist = NULL,
  glmfit = NULL,
  confint_type = NULL,
  remove_ref = FALSE,
  break_scale = NULL,
  column_space = c(-0.5, 0, 0.2),
  dependent_label = NULL,
  prefix = "",
  suffix = ": OR (95% CI, p-value)",
  table_text_size = 5,
  title_text_size = 18,
  plot_opts = NULL,
  table_opts = NULL,
  return_df = FALSE,
  ...
)
```

Arguments

.data	Data frame.
dependent	Character vector of length 1: name of dependent variable (must have 2 levels).
explanatory	Character vector of any length: name(s) of explanatory variables.

random_effect	Character vector of length 1, name of random effect variable.
factorlist	Option to provide output directly from `summary_factorlist()`.
glmfit	Option to provide output directly from `glmmulti()` and `glmmixed()`.
confint_type	One of `c("profile", "default")` for GLM models or `c("default", "Wald", "profile", "boot")` for glmer models. Note `"default" == "Wald"`.
remove_ref	Logical. Remove reference level for factors.
break_scale	Manually specify x-axis breaks in format `c(0.1, 1, 10)`.
column_space	Adjust table column spacing.
dependent_label	
	Main label for plot.
prefix	Plots are titled by default with the dependent variable. This adds text before that label.
suffix	Plots are titled with the dependent variable. This adds text after that label.
table_text_size	
	Alter font size of table text.
title_text_size	
	Alter font size of title text.
plot_opts	A list of arguments to be appended to the ggplot call by `"+"`.
table_opts	A list of arguments to be appended to the ggplot table call by `"+"`.
return_df	To return the dataframe.
...	Other parameters.

Value

The odd ratios plot as a `ggplot2` object.

---

theme_facto   *A ggplot2 Theme for Geometrical Data Analysis*

Description

A ggplot2 Theme for Geometrical Data Analysis

Usage

```
theme_facto(
  res,
  axes = c(1, 2),
  legend.position = c("none", "left", "right", "bottom", "top"),
  no_color_scale = FALSE,
  size_scale_max = 8,
  xlim,
  ylim
)
```

Arguments

`res`	An object created with FactoMineR::`MCA`, `CA`, etc.
`axes`	The axes to print, as a numeric vector of length 2.
`legend.position`	
	One of `c("none", "left", "right", "bottom", "top")`.
`no_color_scale`	When TRUE, you can provide color_scale next without warning.
`size_scale_max`	Maximum size of the points.
`xlim`	Horizontal axe limits.
`ylim`	Vertical axe limits.

Value

A list of ggplot2 objects.

Index