

Package ‘fplot’

July 22, 2025

Type Package

Title Automatic Distribution Graphs Using Formulas

Version 1.1.0

Imports stats, graphics, utils, Formula, Rcpp, grDevices, dreamerr(>= 1.1.0)

Suggests knitr, rmarkdown, fixest, pdftools

LinkingTo Rcpp

Depends R(>= 3.5.0), data.table

Description Easy way to plot regular/weighted/conditional distributions by using formulas. The core of the package concerns distribution plots which are automatic: the many options are tailored to the data at hand to offer the nicest and most meaningful graphs possible -- with no/minimum user input. Further provide functions to plot conditional trends and box plots. See <<https://lrberge.github.io/fplot/>> for more information.

License GPL-3

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation yes

Author Laurent Berge [aut, cre]

Maintainer Laurent Berge <laurent.berge@u-bordeaux.fr>

Repository CRAN

Date/Publication 2023-08-24 16:20:06 UTC

Contents

fplot-package	2
export_graph_start	2
fit.off	5
pdf_fit	6

plot_box	9
plot_distr	12
plot_lines	17
png_fit	19
setFplot_dict	22
setFplot_distr	23
setFplot_page	26
us_pub_econ	28
Index	29

fplot-package	<i>Aggregate/conditional graphs and automatic layout using formulas</i>
---------------	---

Description

fplot provides automatic plotting of common graphs (distributions, lines, bar plots and boxplots). The syntax uses formulas, allowing aggregate/conditional/weighted graphs with minimum efforts. The many arguments are automatically adjusted to the data in order to provide the nicest and most meaningful graphs.

Details

The core functions is `plot_distr` to draw distributions. Two other graphical functions are provided for convenience: `plot_lines` to represent the (usually temporal) evolution of some variables, and `plot_box` to easily represent conditional boxplots.

It also integrates tools to easily export graphs: `pdf_fit` and `png_fit`. In these functions, instead of providing the size of the graphics, you instead give the point size that the text should have in the final document—because an exported graph usually ends up in a document. You can set the size of your document with the function `setFplot_page`. If you use the function `fit.off` to close the connection, you will also see how the export looks like in the Viewer pane.

Author(s)

Maintainer: Laurent Berge <laurent.berge@u-bordeaux.fr>

export_graph_start	<i>Graph export with guaranteed text size</i>
--------------------	---

Description

This function facilitates graph exportation by taking into account the final destination of the graph (typically a document) and allowing the user to use point size, an intuitive unit in written documents, as the graph scaler. Once located in the final document, the text of the graph at the default size will be at the defined point size.

Usage

```

export_graph_start(
  file,
  pt = 10,
  width = 1,
  height,
  w2h = 1.75,
  h2w,
  sideways = FALSE,
  res = 300,
  type = NULL,
  ...
)

export_graph_end()

```

Arguments

file	Character scalar. The name of the file in which to save the graph. If the argument type is NULL, the type of file is deduced from the extension. If your file extension is different from your file type, you need to use the argument type.
pt	The size of the text, in pt, once the figure is inserted in your final document. The default is 10. This means that all text appearing in the plot with cex = 1 will appear with 10pt-sized fonts in your document.
width	The width of the graph, expressed in percentage of the width of the body-text of the document in which it will be inserted. Default is 1, which means that the graph will take 100% of the text width. It can also be equal to a character of the type "100%" or "80%". Alternatively, the following units are valid. Relative sizes: "pw" (page width), "tw" (text width), "ph" (page height), "th" (text height). Absolute sizes: "in", "cm", and "px".
height	Numeric between 0 and 1 or character scalar. The height of the graph, expressed in percentage of the height of the body-text of the document in which it will be inserted. Default is missing, and the height is determined by the other argument w2h. This argument should range between 0 and 1. It can also be equal to a character of the type "100%" or "80%". Alternatively, the following units are valid. Relative sizes: "pw" (page width), "tw" (text width), "ph" (page height), "th" (text height). Absolute sizes: "in", "cm", and "px".
w2h	Numeric scalar. Used to determine the height of the figure based on the width. By default it is equal to 1.75 which means that the graph will be 1.75 larger than tall. Note that when argument sideways = TRUE, the default for the height becomes 90%.
h2w	Numeric scalar, default is missing. Used to determine the aspect ratio of the figure.
sideways	Logical, defaults to FALSE. If the figure will be placed in landscape in the final document, then sideways should be equal to TRUE. If TRUE, then the argument width now refers to the height of the text, and the argument height to its width.

res	Numeric, the resolution in ppi. Default is 300.
type	Character scalar, default is NULL. The type of file to be created. If NULL, the default, then the type of file is deduced from the extension.
...	Other arguments to be passed to bmp , png , jpeg , or tiff . For example: antialias, bg, etc.

Details

To export a ggplot2 graph, remember that you need to **print** it!

```
library(ggplot2)
data = data.frame(x = c(1, 2, 3, 4, 5), y = c(2, 4, 6, 8, 10))

# NOT GOOD
export_graph_start("test.pdf")
ggplot(data, aes(x, y)) +
  geom_point(color = "#54BF98") +
  geom_line(color = "#d34661")
export_graph_end()

# GOOD
my_graph = ggplot(data, aes(x, y)) +
  geom_point(color = "#54BF98") +
  geom_line(color = "#d34661")

export_graph_start("test.pdf")
print(my_graph)
export_graph_end()
```

When the function `export_graph_end()` is called, the resulting exported graph is displayed in the Viewer. The viewer function is found with `getOption("viewer")` and should work on RStudio and VSCode (with the R extension).

Value

These functions do not return anything in R. `export_graph_start` creates a file linked to the R graphics engine, in which subsequent plots are saved. `export_graph_end` closes the connection and the file.

Functions

- `export_graph_end()`: Ends the connection to the current export and creates the file.

Setting the page size

You can set the page size with the function [setFplot_page](#), which defines the size of the page and its margins to deduce the size of the body of the text in which the figures will be inserted. By default the page is considered to be US-letter with *normal* margins (not too big nor thin).

It is important to set the page size appropriately to have a final plotting-text size guaranteed once the figure is inserted in the document.

Author(s)

Laurent Berge

See Also

The tool to set the page size and the exporting defaults: [setFplot_page](#). Exporting functions [pdf_fit](#), [png_fit](#), [jpeg_fit](#).

The functions [export_graph_start\(\)](#) and [export_graph_end\(\)](#) provide similar features.

Examples

```
tmpFile = file.path(tempdir(), "png_examples.pdf")

# we start the exportation
export_graph_start(tmpFile, pt = 8)

plot(1, 1, type = "n", ann = FALSE)
text(1, 1, "This text will be displayed in 8pt.")

# the line below closes the connection and displays the
# graph in the viewer pane if appropriate
export_graph_end()
```

fit.off

Closes the current plotting device and shows the result in the viewer

Description

This function is deprecated: Please use the functions [export_graph_start\(\)](#) and [export_graph_end\(\)](#) instead.

Usage

```
fit.off()
```

Details

To be used in combination with [pdf_fit](#) or [png_fit](#) when exporting images. It performs exactly the same thing as `dev.off()` but additionally shows the resulting graph in the viewer pane provided you're using RStudio.

To view the results of PDF exports, the function `pdf_convert` from package `pdftools` is used to convert the PDF files into images – so you need to have installed `pdftools` to make it work.

In PDFs, only the first page will be viewed.

Value

This function does not return anything in R. It closes the connection between the R graphics engine and a file that has been defined via one of the functions: `pdf_fit``png_fit`

Author(s)

Laurent Berge

See Also

The tool to set the page size and the exporting defaults: `setFplot_page`. Exporting functions `pdf_fit`, `png_fit`, `jpeg_fit`.

The functions `export_graph_start()` and `export_graph_end()` provide similar features.

Examples

```
# Exportation example
# The functions pdf_fit, png_fit, etc, guarantee the right
# point size of the texts present in the graph.
# But you must give the exact size the graph will take in your final document.
# => first use the function setFplot_page, default is:
# setFplot_page(page = "us", margins = "normal")
# By default the graph takes 100% of the text width

data(us_pub_econ)

tmpFile = file.path(tempdir(), "DISTR -- institutions.png")

png_fit(tmpFile)
plot_distr(~institution, us_pub_econ)
fit.off()

# What's the consequence of increasing the point size of the text?
png_fit(tmpFile, pt = 15)
plot_distr(~institution, us_pub_econ)
fit.off()
```

pdf_fit

PDF export with guaranteed text size

Description

(*This function is deprecated: Please use the functions `export_graph_start()` and `export_graph_end()` instead.*) This function is an alternative to `pdf`, it makes it easy to export figures of appropriate size that should end up in a document. Instead of providing the height and width of the figure, you provide the fraction of the text-width the figure should take, and the target font-size at which the plotting text should be rendered. The size of the plotting text, once the figure is in the final document, is guaranteed.

Usage

```
pdf_fit(
  file,
  pt = 10,
  width = 1,
  height,
  w2h = 1.75,
  h2w,
  sideways = FALSE,
  ...
)
```

Arguments

<code>file</code>	The name of the file to which export the figure.
<code>pt</code>	The size of the text, in pt, once the figure is inserted in your final document. The default is 10. This means that all text appearing in the plot with <code>cex = 1</code> will appear with 10pt-sized fonts in your document.
<code>width</code>	The width of the graph, expressed in percentage of the width of the body-text of the document in which it will be inserted. Default is 1, which means that the graph will take 100% of the text width. It can also be equal to a character of the type "100%" or "80%". Alternatively, the following units are valid. Relative sizes: "pw" (page width), "tw" (text width), "ph" (page height), "th" (text height). Absolute sizes: "in", "cm", and "px".
<code>height</code>	Numeric between 0 and 1 or character scalar. The height of the graph, expressed in percentage of the height of the body-text of the document in which it will be inserted. Default is missing, and the height is determined by the other argument <code>w2h</code> . This argument should range between 0 and 1. It can also be equal to a character of the type "100%" or "80%". Alternatively, the following units are valid. Relative sizes: "pw" (page width), "tw" (text width), "ph" (page height), "th" (text height). Absolute sizes: "in", "cm", and "px".
<code>w2h</code>	Numeric scalar. Used to determine the height of the figure based on the width. By default it is equal to 1.75 which means that the graph will be 1.75 larger than tall. Note that when argument <code>sideways = TRUE</code> , the default for the height becomes 90%.
<code>h2w</code>	Numeric scalar, default is missing. Used to determine the aspect ratio of the figure.
<code>sideways</code>	Logical, defaults to FALSE. If the figure will be placed in landscape in the final document, then <code>sideways</code> should be equal to TRUE. If TRUE, then the argument <code>width</code> now refers to the height of the text, and the argument <code>height</code> to its width.
<code>...</code>	Other arguments to be passed to pdf .

Details

If you use `fit.off` instead of `dev.off` to close the graph, the resulting graph will be displayed in the viewer pane. So you don't have to open the document to see how it looks.

To export a ggplot2 graph, remember that you need to **print** it!

```
library(ggplot2)
data = data.frame(x = c(1, 2, 3, 4, 5), y = c(2, 4, 6, 8, 10))

# NOT GOOD
pdf_fit("test.pdf")
ggplot(data, aes(x, y)) +
  geom_point(color = "#54BF98") +
  geom_line(color = "#d34661")
fit.off()

# GOOD
my_graph = ggplot(data, aes(x, y)) +
  geom_point(color = "#54BF98") +
  geom_line(color = "#d34661")

pdf_fit("test.pdf")
print(my_graph)
fit.off()
```

Value

This function does not return anything. It connects the output of the R graphics engine to a file.

Setting the page size

You can set the page size with the function [setFplot_page](#), which defines the size of the page and its margins to deduce the size of the body of the text in which the figures will be inserted. By default the page is considered to be US-letter with *normal* margins (not too big nor thin).

It is important to set the page size appropriately to have a final plotting-text size guaranteed once the figure is inserted in the document.

Author(s)

Laurent Berge

See Also

To set the geometry and the defaults: [setFplot_page](#). To close the graph and display it on the viewer pane: [fit.off](#).

Examples

```
# This function creates figures made to be inserted
# in a Latex document (US-letter with "normal" margins)
# By default, the figures should take 100% of the
# text width. If so, the size of the text in the figures
# will be exact.

# You need pdftools and knitr to display PDFs in the viewer pane with fit.off
```



```

if(require(pdftools) && require(knitr)){

  tmpFile = file.path(tempdir(), "pdf_examples.pdf")

  pdf_fit(tmpFile, pt = 8)
  plot(1, 1, type = "n", ann = FALSE)
  text(1, 1, "This text will be displayed in 8pt.")
  fit.off()

  pdf_fit(tmpFile, pt = 12)
  plot(1, 1, type = "n", ann = FALSE)
  text(1, 1, "This text will be displayed in 12pt.")
  fit.off()

  pdf_fit(tmpFile, pt = 12, sideways = TRUE)
  plot(1, 1, type = "n", ann = FALSE)
  text(1, 1, "This text will be displayed in 12pt if in sideways.")
  fit.off()

  # If we reduce the end plot width but keep font size constant
  # this will lead to a very big font as compared to the plot
  pdf_fit(tmpFile, pt = 8, width = "50%")
  plot(1, 1, type = "n", ann = FALSE)
  text(1, 1, "This text will be displayed in 8pt\nif in 50% of the text width.")
  fit.off()
}

```

plot_box

Boxplots with possibly moderators

Description

This function allows to draw a boxplot, with possibly separating different moderators.

Usage

```

plot_box(
  fml,
  data,
  case,
  moderator,
  inCol,
  outCol = "black",
  density = -1,

```

```

lty = 1,
pch = 18,
addLegend = TRUE,
legend_options = list(),
lwd = 2,
outlier,
dict = NULL,
dict_case,
dict_moderator,
order_case,
order_moderator,
addMean,
mean.col = "darkred",
mean.pch = 18,
mean.cex = 2,
mod.title = TRUE,
labels.tilted,
trunc = 20,
trunc.method = "auto",
line.max,
...
)

```

Arguments

fml	A numeric vector or a formula of the type: <code>vars ~ moderator_1 moderator_2</code> . Note that if a formula is provided then the argument 'data' must be provided. You can plot several variables, if you don't want a moderator, use 1 instead: e.g. <code>plot_box(Petal.Width + Petal.Length ~ 1, iris)</code> . You can plot all numeric variables from a data set using <code>"."</code> : <code>plot_box(. ~ 1, iris)</code> .
data	A data.frame/data.table containing the relevant information.
case	When argument fml is a vector, this argument can receive a vector of cases.
moderator	When argument fml is a vector, this argument can receive a vector of moderators.
inCol	A vector of colors that will be used for within the boxes.
outCol	The color of the outer box. Default is black.
density	The density of lines within the boxes. By default it is equal to -1, which means the boxes are filled with color.
lty	The type of lines for the border of the boxes. Default is 1 (solid line).
pch	The patch of the outliers. Default is 18.
addLegend	Default is TRUE. Should a legend be added at the top of the graph if there is more than one moderator?
legend_options	A list. Other options to be passed to legend which concerns the legend for the moderator.
lwd	The width of the lines making the boxes. Default is 2.
outlier	Default is TRUE. Should the outliers be displayed?

dict	A dictionary to rename the variables names in the axes and legend. Should be a named vector. By default it s the value of <code>getFplot_dict()</code> , which you can set with the function <code>setFplot_dict</code> .
dict_case	A named character vector. If provided, it changes the values of the variable 'case' to the ones contained in the vector <code>dict_case</code> . Example: I want to change my variable named "a" to "Australia" and "b" to "Brazil", then I used <code>dict=c(a="Australia",b="Brazil")</code> .
dict_moderator	A named character vector. If provided, it changes the values of the variable 'moderator' to the ones contained in the vector <code>dict_moderator</code> . Example: I want to change my variable named "a" to "Australia" and "b" to "Brazil", then I used <code>dict=c(a="Australia",b="Brazil")</code> .
order_case	Character vector. This element is used if the user wants the 'case' values to be ordered in a certain way. This should be a regular expression (see regex help for more info). There can be more than one regular expression. The variables satisfying the first regular expression will be placed first, then the order follows the sequence of regular expressions.
order_moderator	Character vector. This element is used if the user wants the 'moderator' values to be ordered in a certain way. This should be a regular expression (see regex help for more info). There can be more than one regular expression. The variables satisfying the first regular expression will be placed first, then the order follows the sequence of regular expressions.
addMean	Whether to add the average for each boxplot. Default is true.
mean.col	The color of the mean. Default is darkred.
mean.pch	The patch of the mean, default is 18.
mean.cex	The cex of the mean, default is 2.
mod.title	Character scalar. The title of the legend in case there is a moderator. You can set it to TRUE (the default) to display the moderator name. To display no title, set it to NULL or FALSE.
labels.tilted	Whether there should be tilted labels. Default is FALSE except when the data is split by moderators (see <code>mod.method</code>).
trunc	If the main variable is a character, its values are truncated to <code>trunc</code> characters. Default is 20. You can set the truncation method with the argument <code>trunc.method</code> .
trunc.method	If the elements of the x-axis need to be truncated, this is the truncation method. It can be "auto", "right" or "mid".
line.max	Option for the x-axis, how far should the labels go. Default is 1 for normal labels, 2 for tilted labels.
...	Other parameters to be passed to <code>plot</code> .

Value

Invisibly returns the coordinates of the x-axis.

Author(s)

Laurent Berge

Examples

```
# Simple iris boxplot
plot(1:10)

# All numeric variables
plot_box(. ~ 1, iris)

# All numeric variable / splitting by species
plot_box(. ~ Species, iris)

# idem but with renaming
plot_box(. ~ Species, iris, dict = c(Species="Iris species",
                                     setosa="SETOSA", Petal.Width="Width (Petal)"))

# Now using two moderators
base = iris
base$period = sample(1:4, 150, TRUE)

plot_box(Petal.Length ~ period | Species, base)
```

plot_distr*Plot distributions, possibly conditional*

Description

This function plots distributions of items (a bit like an histogram) which can be easily conditioned over.

Usage

```
plot_distr(
  fml,
  data,
  moderator,
  weight,
  sorted,
  log,
  nbins,
  bin.size,
  legend_options = list(),
```

```

    top,
    yaxis.show = TRUE,
    yaxis.num,
    col,
    border = "black",
    mod.method,
    within,
    total,
    mod.select,
    mod.NA = FALSE,
    at_5,
    labels.tilted,
    other,
    cumul = FALSE,
    plot = TRUE,
    sep,
    centered = TRUE,
    weight.fun,
    int.categorical,
    dict = NULL,
    mod.title = TRUE,
    labels.angle,
    cex.axis,
    trunc = 20,
    trunc.method = "auto",
    ...
)

```

Arguments

fml	A formula or a vector. If a formula, it must be of the type: <code>weights ~ var moderator</code> . If there are no moderator nor weights, you can use directly a vector, or use a one-sided formula <code>fml = ~var</code> . You can use multiple variables as weights, if so, you cannot use moderators at the same time. See examples.
data	A data.frame: data set containing the variables in the formula.
moderator	Optional, only if argument fml is a vector. A vector of moderators.
weight	Optional, only if argument fml is a vector. A vector of (positive) weights.
sorted	Logical: should the first elements displayed be the most frequent? By default this is the case except for numeric values put to log or to integers.
log	Logical, only used when the data is numeric. If TRUE, then the data is put to logarithm beforehand. By default numeric values are put to log if the log variation exceeds 3.
nbins	Maximum number of items displayed. The default depends on the number of moderator cases. When there is no moderator, the default is 15, augmented to 20 if there are less than 20 cases.
bin.size	Only used for numeric values. If provided, it creates bins of observations of size bin.size. It creates bins by default for numeric non-integer data.

legend_options	A list. Other options to be passed to legend which concerns the legend for the moderator.
top	What to display on the top of the bars. Can be equal to "frac" (for shares), "nb" or "none". The default depends on the type of the plot. To disable it you can also set it to FALSE or the empty string.
yaxis.show	Whether the y-axis should be displayed, default is TRUE.
yaxis.num	Whether the y-axis should display regular numbers instead of frequencies in percentage points. By default it shows numbers only when the data is weighted with a different function than the sum. For conditionnal distributions, a numeric y-axis can be displayed only when <code>mod.method = "sideTotal"</code> , <code>mod.method = "splitTotal"</code> or <code>mod.method = "stack"</code> , since for the within distributions it does not make sense (because the data is rescaled for each moderator).
col	A vector of colors, default is close to paired. You can also use "set1" or "paired".
border	Outer color of the bars. Defaults is "black". Use NA to remove the borders.
mod.method	A character scalar: either i) "split", the default for categorical data, ii) "side", the default for data in logarithmic form or numeric data, or iii) "stack". This is only used when there is more than one moderator. If "split": there is one separate histogram for each moderator case. If "side": moderators are represented side by side for each value of the variable. If "stack": the bars of the moderators are stacked onto each other, the bar heights representing the distribution in the total population. You can use the other arguments within and total to say whether the distributions should be within each moderator or over the total distribution.
within	Logical, default is missing. Whether the distributions should be scaled to reflect the distribution within each moderator value. By default it is TRUE if <code>mod.method</code> is different from "stack".
total	Logical, default is missing. Whether the distributions should be scaled to reflect the total distribution (and not the distribution within each moderator value). By default it is TRUE only if <code>mod.method="stack"</code> .
mod.select	Which moderators to select. By default the top 3 moderators in terms of frequency (or in terms of weight value if there's a weight) are displayed. If provided, it must be a vector of moderator values whose length cannot be greater than 5. Alternatively, you can put an integer between 1 and 5. This argument also accepts regular expressions.
mod.NA	Logical, default is FALSE. If TRUE, and if the moderator contains NA values, all NA values from the moderator will be treated as a regular case: allows to display the distribution for missing values.
at_5	Equal to FALSE, "roman" or "line". When plotting categorical variables, adds a small Roman number under every 5 bars (<code>at_5 = "roman"</code>), or draws a thick axis line every 5 bars (<code>at_5 = "line"</code>). Helps to get the rank of the bars. The default depends on the type of data – Not implemented when there is a moderator.
labels.tilted	Whether there should be tilted labels. Default is FALSE except when the data is split by moderators (see <code>mod.method</code>).
other	Logical. Should there be a last column counting for the observations not displayed? Default is TRUE except when the data is split.

cumul	Logical, default is FALSE. If TRUE, then the cumulative distribution is plotted.
plot	Logical, default is TRUE. If FALSE nothing is plotted, only the data is returned.
sep	Positive number. The separation space between the bars. The scale depends on the type of graph.
centered	Logical, default is TRUE. For numeric data only and when sorted=FALSE, whether the histogram should be centered on the mode.
weight.fun	A function, by default it is sum. Aggregate function to be applied to the weight with respect to variable and the moderator. See examples.
int.categorical	Logical. Whether integers should be treated as categorical variables. By default they are treated as categorical only when their range is small (i.e. smaller than 1000).
dict	A dictionary to rename the variables names in the axes and legend. Should be a named vector. By default it s the value of getFplot_dict(), which you can set with the function setFplot_dict .
mod.title	Character scalar. The title of the legend in case there is a moderator. You can set it to TRUE (the default) to display the moderator name. To display no title, set it to NULL or FALSE.
labels.angle	Only if the labels of the x-axis are tilted. The angle of the tilt.
cex.axis	Cex value to be passed to biased labels. By defaults, it finds automatically the right value.
trunc	If the main variable is a character, its values are truncated to trunc characters. Default is 20. You can set the truncation method with the argument trunc.method.
trunc.method	If the elements of the x-axis need to be truncated, this is the truncation method. It can be "auto", "right" or "mid".
...	Other elements to be passed to plot.

Details

Most default values can be modified with the function [setFplot_distr](#).

Value

This function returns *invisibly* the output data.table containing the processed data used for plotting. With the argument plot = FALSE, only the data is returned.

Author(s)

Laurent Berge

See Also

To plot temporal evolutions: [plot_lines](#). For boxplot: [plot_box](#). To export graphs: [pdf_fit](#), [png_fit](#), [fit.off](#).

Examples

```

# Data on publications from U.S. institutions
data(us_pub_econ)

# 0) Let's set a dictionary for a better display of variables
setFplot_dict(c(institution = "U.S. Institution", jnl_top_25p = "Top 25% Pub.",
                jnl_top_5p = "Top 5% Pub.", Frequency = "Publications"))

# 1) Let's plot the distribution of publications by institutions:
plot_distr(~institution, us_pub_econ)

# When there is only the variable, you can use a vector instead:
plot_distr(us_pub_econ$institution)

# 2) Now the production of institution weighted by journal quality
plot_distr(jnl_top_5p ~ institution, us_pub_econ)

# You can plot several variables:
plot_distr(1 + jnl_top_25p + jnl_top_5p ~ institution, us_pub_econ)

# 3) Let's plot the journal distribution for the top 3 institutions

# We can get the data from the previous graph
graph_data = plot_distr(jnl_top_5p ~ institution, us_pub_econ, plot = FALSE)
# And then select the top universities
top3_instit = graph_data$x[1:3]
top5_instit = graph_data$x[1:5] # we'll use it later

# Now the distribution of journals
plot_distr(~ journal | institution, us_pub_econ[institution %in% top3_instit])
# Alternatively, you can use the argument mod.select:
plot_distr(~ journal | institution, us_pub_econ, mod.select = top3_instit)

# 3') Same graph as before with "other" column, 5 institutions
plot_distr(~ journal | institution, us_pub_econ,
           mod.select = top5_instit, other = TRUE)

#
# Example with continuous data
#

# regular histogram
plot_distr(iris$Sepal.Length)

# now splitting by species:
plot_distr(~ Sepal.Length | Species, iris)

# idem but the three distr. are separated:
plot_distr(~ Sepal.Length | Species, iris, mod.method = "split")

# Now the three are stacked
plot_distr(~ Sepal.Length | Species, iris, mod.method = "stack")

```

plot_lines

Display means conditionnally on some other values

Description

The typical use of this function is to represents trends of average along some categorical variable.

Usage

```
plot_lines(
  fml,
  data,
  time,
  moderator,
  mod.select,
  mod.NA = TRUE,
  smoothing_window = 0,
  fun,
  col = "set1",
  lty = 1,
  pch = c(19, 17, 15, 8, 5, 4, 3, 1),
  legend_options = list(),
  pt.cex = 2,
  lwd = 2,
  dict = NULL,
  mod.title = TRUE,
  ...
)
```

Arguments

fml	A formula of the type variable ~ time moderator. Note that the moderator is optional. Can also be a vector representing the elements of the variable. If a formula is provided, then you must add the argument 'data'. You can use multiple variables. If so, you cannot use a moderator at the same time.
data	Data frame containing the variables of the formula. Used only if the argument 'fml' is a formula.
time	Only if argument 'fml' is a vector. It should be the vector of 'time' identifiers to average over.
moderator	Only if argument 'fml' is a vector. It should be a vector of conditional values to average over. This is an optional parameter.

mod.select	Which moderators to select. By default the top 5 moderators in terms of frequency (or in terms of the value of fun in case of identical frequencies) are displayed. If provided, it must be a vector of moderator values whose length cannot be greater than 10. Alternatively, you can put an integer between 1 and 10.
mod.NA	Logical, default is FALSE. If TRUE, and if the moderator contains NA values, all NA values from the moderator will be treated as a regular case: allows to display the distribution for missing values.
smoothing_window	Default is 0. The number of time periods to average over. Note that if it is provided the new value for each period is the average of the current period and the smoothing_window time periods before and after.
fun	Function to apply when aggregating the values on the time variable. Default is mean.
col	The colors. Either a vector or a keyword (“Set1” or “paired”). By default those are the “Set1” colors colorBrewer. This argument is used only if there is a moderator.
lty	The line types, in the case there are more than one moderator. By default it is equal to 1 (ie no difference between moderators).
pch	The form types of the points, in the case there are more than one moderator. By default it is equal to \8code(19, 17, 15, 8, 5, 4, 3, 1).
legend_options	A list containing additional parameters for the function legend – only concerns the moderator. Note that you can set the additionnal arguments trunc and trunc.method which relates to the number of characters to show and the truncation method. By default the algorithm truncates automatically when needed.
pt.cex	Default to 2. The cex of the points.
lwd	Default to 2. The width of the lines.
dict	A dictionary to rename the variables names in the axes and legend. Should be a named vector. By default it s the value of getFplot_dict(), which you can set with the function setFplot_dict .
mod.title	Character scalar. The title of the legend in case there is a moderator. You can set it to TRUE (the default) to display the moderator name. To display no title, set it to NULL or FALSE.
...	Other arguments to be passed to the function plot.

Value

This function returns *invisibly* the output data.table containing the processed data used for plotting.

Author(s)

Laurent Berge

Examples

```
data(airquality)

plot_lines(Ozone ~ Day, airquality)

plot_lines(Ozone ~ Day | Month, airquality)

plot_lines(Ozone ~ Month | cut(Day, 8), airquality)
```

png_fit

PNG export with guaranteed text size

Description

(This function is deprecated: Please use the functions [export_graph_start\(\)](#) and [export_graph_end\(\)](#) instead.) This is an alternative to [png](#) and others. It makes it easy to export figures that should end up in documents. Instead of providing the height and width of the figure, you provide the fraction of the text-width the figure should take, and the target font-size at which the plotting text should be rendered. The size of the plotting text, once the figure is in the final document, is guaranteed.

Usage

```
png_fit(
  file,
  pt = 10,
  width = 1,
  height,
  w2h = 1.75,
  h2w,
  sideways = FALSE,
  res = 300,
  ...
)

tiff_fit(
  file,
  pt = 10,
  width = 1,
  height,
  w2h = 1.75,
  h2w,
  sideways = FALSE,
  res = 300,
  ...
)
```

```

jpeg_fit(
  file,
  pt = 10,
  width = 1,
  height,
  w2h = 1.75,
  h2w,
  sideways = FALSE,
  res = 300,
  ...
)

bmp_fit(
  file,
  pt = 10,
  width = 1,
  height,
  w2h = 1.75,
  h2w,
  sideways = FALSE,
  res = 300,
  ...
)

```

Arguments

<code>file</code>	The name of the file to which export the figure.
<code>pt</code>	The size of the text, in pt, once the figure is inserted in your final document. The default is 10. This means that all text appearing in the plot with <code>cex = 1</code> will appear with 10pt-sized fonts in your document.
<code>width</code>	The width of the graph, expressed in percentage of the width of the body-text of the document in which it will be inserted. Default is 1, which means that the graph will take 100% of the text width. It can also be equal to a character of the type "100%" or "80%". Alternatively, the following units are valid. Relative sizes: "pw" (page width), "tw" (text width), "ph" (page height), "th" (text height). Absolute sizes: "in", "cm", and "px".
<code>height</code>	Numeric between 0 and 1 or character scalar. The height of the graph, expressed in percentage of the height of the body-text of the document in which it will be inserted. Default is missing, and the height is determined by the other argument <code>w2h</code> . This argument should range between 0 and 1. It can also be equal to a character of the type "100%" or "80%". Alternatively, the following units are valid. Relative sizes: "pw" (page width), "tw" (text width), "ph" (page height), "th" (text height). Absolute sizes: "in", "cm", and "px".
<code>w2h</code>	Numeric scalar. Used to determine the height of the figure based on the width. By default it is equal to 1.75 which means that the graph will be 1.75 larger than tall. Note that when argument <code>sideways = TRUE</code> , the default for the height becomes 90%.

h2w	Numeric scalar, default is missing. Used to determine the aspect ratio of the figure.
sideways	Logical, defaults to FALSE. If the figure will be placed in landscape in the final document, then sideways should be equal to TRUE. If TRUE, then the argument width now refers to the height of the text, and the argument height to its width.
res	Numeric, the resolution in ppi. Default is 300.
...	Other arguments to be passed to bmp , png , jpeg , or tiff . For example: antialias, bg, etc.

Value

This function does not return anything. It connects the output of the R graphics engine to a file.

Setting the page size

You can set the page size with the function [setFplot_page](#), which defines the size of the page and its margins to deduce the size of the body of the text in which the figures will be inserted. By default the page is considered to be US-letter with *normal* margins (not too big nor thin).

It is important to set the page size appropriately to have a final plotting-text size guaranteed once the figure is inserted in the document.

Examples

```
# This function creates figures made to be inserted
# in a Latex document (US-letter with "normal" margins)
# By default, the figures should take 100% of the
# text width. If so, the size of the text in the figures
# will be exact.

tmpFile = file.path(tempdir(), "png_examples.png")

png_fit(tmpFile, pt = 8)
plot(1, 1, type = "n", ann = FALSE)
text(1, 1, "This text will be displayed in 8pt.")
fit.off()

png_fit(tmpFile, pt = 12)
plot(1, 1, type = "n", ann = FALSE)
text(1, 1, "This text will be displayed in 12pt.")
fit.off()

png_fit(tmpFile, pt = 12, sideways = TRUE)
plot(1, 1, type = "n", ann = FALSE)
text(1, 1, "This text will be displayed in 12pt if in sideways.")
fit.off()

# If we reduce the end plot width but keep font size constant
# this will lead to a very big font as compared to the plot
png_fit(tmpFile, pt = 8, width = "50%")
```

```
plot(1, 1, type = "n", ann = FALSE)
text(1, 1, "This text will be displayed in 8pt\nif the graph is 50% of the text width.")
fit.off()
```

setFplot_dict

Sets/gets the dictionary used in fplot

Description

Sets/gets the default dictionary used to rename the axes/moderator variables in the functions of the package fplot. The dictionaries are used to relabel variables (usually towards a fancier, more explicit formatting) that can be useful not to explicitly use the arguments xlab/ylab when exporting graphs. By setting the dictionary with setFplot_dict, you can avoid providing the argument dict in fplot functions.

Usage

```
setFplot_dict(dict)
```

```
getFplot_dict
```

Arguments

dict	A named character vector. E.g. to change my variable named "us_md" and "state" to (resp.) "\$ million" and "U.S. state", then use dict = c(us_md="\$ million", state = "U.S. state").
------	---

Format

An object of class function of length 1.

Details

This function stores a named vector in the option "fplot_dict". The dictionary is automatically accessed by all fplot functions.

Value

The function setFplot_dict() does not return anything, it only sets an option after checking the format of the arguments.

The function getFplot_dict() returns a named vector representing the dictionary set in setFplot_dict().

Author(s)

Laurent Berge

Examples

```
data(airquality)
setFplot_dict(c(Ozone = "Ozone (ppb)"))
plot_distr(Ozone ~ Month, airquality, weight.fun = mean)
```

setFplot_distr	<i>Sets the defaults of plot_distr</i>
----------------	--

Description

The default values of most arguments of [plot_distr](#) can be set with `setFplot_distr`.

Usage

```
setFplot_distr(
  sorted,
  log,
  top,
  yaxis.num,
  col,
  border = "black",
  mod.method,
  within,
  total,
  at_5,
  labels.tilted,
  other,
  cumul = FALSE,
  centered = TRUE,
  weight.fun,
  int.categorical,
  dict = NULL,
  mod.title = TRUE,
  labels.angle,
  cex.axis,
  trunc = 20,
  trunc.method = "auto",
  reset = FALSE
)

getFplot_distr()
```

Arguments

sorted	Logical: should the first elements displayed be the most frequent? By default this is the case except for numeric values put to log or to integers.
--------	---

log	Logical, only used when the data is numeric. If TRUE, then the data is put to logarithm beforehand. By default numeric values are put to log if the log variation exceeds 3.
top	What to display on the top of the bars. Can be equal to "frac" (for shares), "nb" or "none". The default depends on the type of the plot. To disable it you can also set it to FALSE or the empty string.
yaxis.num	Whether the y-axis should display regular numbers instead of frequencies in percentage points. By default it shows numbers only when the data is weighted with a different function than the sum. For conditionnal distributions, a numeric y-axis can be displayed only when <code>mod.method = "sideTotal"</code> , <code>mod.method = "splitTotal"</code> or <code>mod.method = "stack"</code> , since for the within distributions it does not make sense (because the data is rescaled for each moderator).
col	A vector of colors, default is close to paired. You can also use "set1" or "paired".
border	Outer color of the bars. Defaults is "black". Use NA to remove the borders.
mod.method	A character scalar: either i) "split", the default for categorical data, ii) "side", the default for data in logarithmic form or numeric data, or iii) "stack". This is only used when there is more than one moderator. If "split": there is one separate histogram for each moderator case. If "side": moderators are represented side by side for each value of the variable. If "stack": the bars of the moderators are stacked onto each other, the bar heights representing the distribution in the total population. You can use the other arguments <code>within</code> and <code>total</code> to say whether the distributions should be within each moderator or over the total distribution.
within	Logical, default is missing. Whether the distributions should be scaled to reflect the distribution within each moderator value. By default it is TRUE if <code>mod.method</code> is different from "stack".
total	Logical, default is missing. Whether the distributions should be scaled to reflect the total distribution (and not the distribution within each moderator value). By default it is TRUE only if <code>mod.method="stack"</code> .
at_5	Equal to FALSE, "roman" or "line". When plotting categorical variables, adds a small Roman number under every 5 bars (<code>at_5 = "roman"</code>), or draws a thick axis line every 5 bars (<code>at_5 = "line"</code>). Helps to get the rank of the bars. The default depends on the type of data – Not implemented when there is a moderator.
labels.tilted	Whether there should be tilted labels. Default is FALSE except when the data is split by moderators (see <code>mod.method</code>).
other	Logical. Should there be a last column counting for the observations not displayed? Default is TRUE except when the data is split.
cumul	Logical, default is FALSE. If TRUE, then the cumulative distribution is plotted.
centered	Logical, default is TRUE. For numeric data only and when <code>sorted=FALSE</code> , whether the histogram should be centered on the mode.
weight.fun	A function, by default it is <code>sum</code> . Aggregate function to be applied to the weight with respect to variable and the moderator. See examples.
int.categorical	Logical. Whether integers should be treated as categorical variables. By default they are treated as categorical only when their range is small (i.e. smaller than 1000).

dict	A dictionary to rename the variables names in the axes and legend. Should be a named vector. By default it s the value of getFplot_dict(), which you can set with the function setFplot_dict .
mod.title	Character scalar. The title of the legend in case there is a moderator. You can set it to TRUE (the default) to display the moderator name. To display no title, set it to NULL or FALSE.
labels.angle	Only if the labels of the x-axis are tilted. The angle of the tilt.
cex.axis	Cex value to be passed to biased labels. By defaults, it finds automatically the right value.
trunc	If the main variable is a character, its values are truncated to trunc characters. Default is 20. You can set the truncation method with the argument trunc.method.
trunc.method	If the elements of the x-axis need to be truncated, this is the truncation method. It can be "auto", "right" or "mid".
reset	Logical scalar, default is FALSE. Whether the defaults should be reset.

Value

The function `setFplot_distr()` does not return anything, it only sets the default parameters for the function [plot_distr\(\)](#).

The function `getFplot_distr()` returns a named list containing the arguments that have been set with the function `setFplot_distr()`.

See Also

[plot_distr](#), [pdf_fit](#), [fit.off](#).

Examples

```
# Changing the default color set for plot_distr only
my_col = c("#36688D", "#F3CD05", "#F49F05", "#F18904", "#BDA589")

setFplot_distr(col = my_col, mod.method = "split", border = NA)

plot_distr(~ Petal.Length | Species, iris)

# Back to normal
setFplot_distr(reset = TRUE)

plot_distr(~ Petal.Length | Species, iris)
```

setFplot_page	<i>Sets the target page size for figure exporting</i>
---------------	---

Description

The package `fplot` offers some functions (e.g. `pdf_fit` or `png_fit`) to export figures, with a guarantee to obtain the desired point size for the plotting text. The function `setFplot_page` sets the target page size (once and for all). This is important for the accuracy of the export, although the default values should be working well most of the time.

Usage

```
setFplot_page(
  page = "us",
  margins = "normal",
  units = "tw",
  pt = 10,
  w2h = 1.75,
  reset = FALSE
)
```

```
getFplot_page()
```

Arguments

page	What is the page size of the document? Can be equal to "us" (for US letter, the default) or "a4". Can also be a numeric vector of length 2 giving the width and the height of the page in inches . Or can be a character string of the type: "8.5in,11in" where the width and height are separated with a comma, note that only centimeters (cm), inches (in) and pixels (px) are accepted as units—further: you can use the unit only once.
margins	The bottom/left/top/right margins of the page. This is used to obtain the dimension of the body of the text. Can be equal to "normal" (default, which corresponds to 2cm/2.5cm/2cm/2.5cm), or to "thin" (1.5/1/1/1cm). Can be a numeric vector of length 1: then all margins are the same given size in inches . Can also be a numeric vector of length 2 or 4: 2 means first bottom/top margins, then left/right margins; 4 is bottom/left/top/right margins, in inches. Last, it can be a character vector of the type "2,2.5,2,2.5cm" with the margins separated by a comma or a slash, and at least one unit appearing: either cm, in or px.
units	The default units when using the functions <code>pdf_fit</code> , <code>png_fit</code> , etc. Defaults to "tw" (text width) which is a fraction of the size of the text. Alternatives can be "pw" (page width), and "in", "cm", "px".
pt	The size of the text, in pt, once the figure is inserted in your final document. The default is 10. This means that all text appearing in the plot with <code>cex = 1</code> will appear with 10pt-sized fonts in your document.

w2h	Numeric scalar. Used to determine the height of the figure based on the width. By default it is equal to 1.75 which means that the graph will be 1.75 larger than tall. Note that when argument <code>sideways = TRUE</code> , the default for the height becomes 90%.
reset	Logical, default is FALSE. Whether arguments should be reset to default before applying modifications.

Details

This function sets the option "fplot_export_opts" after parsing the arguments. This option is then automatically accessed by the functions used to export graphs [export_graph_start\(\)](#).

Value

The function `setFplot_page()` does not return anything. It sets an R option containing the page parameters.

The function `getFplot_page()` returns the named list of page parameters which has been set in `setFplot_page()`.

See Also

Exporting functions: [pdf_fit](#), [png_fit](#). The function closing the connection and showing the obtained graph in the viewer: [fit.off](#).

Examples

```
#
# How to set the page size
#

# All examples below provide the same page size
setFplot_page(page = "us")
setFplot_page(page = "8.5in, 11in")
setFplot_page(page = "8.5/11in")
setFplot_page(page = c(8.5, 11))

# All examples below provide the same margins
setFplot_page(margins = "normal")
setFplot_page(margins = "2cm, 2.5cm, 2cm, 2.5cm")
setFplot_page(margins = "2/2.5/2/2.5cm")
setFplot_page(margins = c(2, 2.5) / 2.54) # cm to in
setFplot_page(margins = c(2, 2.5, 2, 2.5) / 2.54)
```

`us_pub_econ`*Publication data sample*

Description

This data reports the publications of U.S. institutions in the field of economics between 1985 and 1990.

Usage

```
data(us_pub_econ)
```

Format

`us_pub_econ` is a data table with 30,756 observations and 6 variables.

- `paper_id`: Numeric identifier of the publication.
- `year`: Year of publication.
- `institution`: Institution of the authors of the publication.
- `journal`: Journal/conference name.
- `jnl_top_25p`: 0/1 variable of whether the journal belongs to the top 25% in terms of average cites.
- `jnl_top_5p`: 0/1 variable of whether the journal belongs to the top 5% in terms of average cites.

Source

The source is Microsoft Academic Graph (see reference).

References

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June (Paul) Hsu, and Kuansan Wang. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion). ACM, New York, NY, USA, 243-246.

Index

- * **datasets**
 - setFplot_dict, [22](#)
 - us_pub_econ, [28](#)
- bmp, [4](#), [21](#)
- bmp_fit(png_fit), [19](#)
- export_graph_end(export_graph_start), [2](#)
- export_graph_end(), [5](#), [6](#), [19](#)
- export_graph_start, [2](#)
- export_graph_start(), [5](#), [6](#), [19](#), [27](#)
- fit.off, [2](#), [5](#), [7](#), [8](#), [15](#), [25](#), [27](#)
- fplot(fplot-package), [2](#)
- fplot-package, [2](#)
- getFplot_dict(setFplot_dict), [22](#)
- getFplot_distr(setFplot_distr), [23](#)
- getFplot_page(setFplot_page), [26](#)
- jpeg, [4](#), [21](#)
- jpeg_fit, [5](#), [6](#)
- jpeg_fit(png_fit), [19](#)
- legend, [18](#)
- pdf, [6](#), [7](#)
- pdf_fit, [2](#), [5](#), [6](#), [6](#), [15](#), [25–27](#)
- plot_box, [2](#), [9](#), [15](#)
- plot_distr, [2](#), [12](#), [23](#), [25](#)
- plot_distr(), [25](#)
- plot_lines, [2](#), [15](#), [17](#)
- png, [4](#), [19](#), [21](#)
- png_fit, [2](#), [5](#), [6](#), [15](#), [19](#), [26](#), [27](#)
- regex, [11](#)
- setFplot_dict, [11](#), [15](#), [18](#), [22](#), [25](#)
- setFplot_distr, [15](#), [23](#)
- setFplot_page, [2](#), [4–6](#), [8](#), [21](#), [26](#)
- tiff, [4](#), [21](#)
- tiff_fit(png_fit), [19](#)
- us_pub_econ, [28](#)