# Package 'fplot'

November 21, 2019

**Type** Package

**Title** Automatic Distribution Graphs Using Formulas

**Version** 0.2.0

**Imports** stats, graphics, utils, Formula, Rcpp

**Suggests** knitr, rmarkdown, fixest

**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 boxplots.

**License** GPL-3

**SystemRequirements** C++11

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** yes

**Author** Laurent Berge [aut, cre]

**Maintainer** Laurent Berge <laurent.berge@uni.lu>

**Repository** CRAN

**Date/Publication** 2019-11-21 16:10:02 UTC

## R topics documented:

---

`fplot-package` *Aggregate/conditional graphs and automatic layout using formulas*

---

## 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 are: `plot_distr` to draw distributions, `plot_lines` to represent the (usually temporal) evolution of variables, `plot_box` for conditional boxplots, and `plot_bar` for aggregate bar plots.

## Author(s)

**Maintainer**: Laurent Berge <laurent.berge@uni.lu>

---

`plot_bar` *Barplot with aggregate and moderator possibilities*

---

## Description

This functions draws a barplot in which the variables can have been aggregated beforehand.

## Usage

```
plot_bar(fml, data, agg, fun = mean, dict = getFplot_dict(),
  order = FALSE, maxBins = 50, show0 = TRUE, cex.text = 0.7,
  isDistribution = FALSE, yaxis.show = TRUE, labels.tilted,
  trunc = 20, trunc.method = "auto", max_line, hgrid = TRUE,
  onTop = "nb", showOther = TRUE, inCol = "#386CB0",
  outCol = "white", xlab, ylab, ...)
```

## Arguments

| | |
|---|---|
| `fml` | A formula of the form: `var ~ agg` with `var` the variable, `agg` the variable over which to aggregate that will appear in the x-axis. |
| `data` | A data.frame containing all the variables in the formula argument. |
| `agg` | In the case `fml` is a vector, `agg` can be a vector of values over which to aggregate the main variable. |
| `fun` | A function for the aggregation. Default is `mean`. |

| | |
|---|---|
| dict | A dictionnary 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`. |
| order | Defaults to `FALSE`. Should the data be ordered w.r.t. frequency? |
| maxBins | Defaults to 50. All other information that does not fit is put into the bin "other". |
| show0 | Default to `FALSE`. Should the 0 be kept? By default, all the 0s are dropped. |
| cex.text | The size of the text appearing on the top of the bins. Defaults to 0.7. |
| isDistribution | Defaults to `FALSE`. It impacts the y-axis display. If it's a distribution, then percentages are shown on the y-axis. |
| yaxis.show | Defaults to `FALSE`. Should the y-axis labels be displayed? |
| labels.tilted | Whether there should be tilted labels. Default is `FALSE` except when the data is split by moderators (see `mod.method`). |
| trunc | If the main variable is a character, its values are truncaded 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", "trimRight" or "trimRight". |
| max_line | Integer, default is 1. By defaults the labels of the x-axis can be displayed on several lines (from -1 to 1). This arguments says how far can the algorithm for the placement of the labels go downwards in the x-axis region. |
| hgrid | Default `TRUE`. Should the horizontal grid be displayed? |
| onTop | 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. |
| showOther | Default is `TRUE`. In the case the number of bins is lower than the number of cases, should the remaining obervations be displayed by a bar? |
| inCol | Color of the interior of the bars. Default to blue. |
| outCol | Color of the border of the bars. Defatult is black. |
| xlab | The x-axis labels. By default it is the name of the aggregating variable. |
| ylab | The y-axis labels. By default it is the name of the x variable combined with the function applied to it. |
| ... | Other arguments to be passed to `barplot`. |

## Value

Invisibly returns the aggregated data.

## Author(s)

Laurent Berge

## Examples

```
plot_bar(Sepal.Length~Species, iris)

plot_bar(Sepal.Length~Species, iris, fun = var, labels.tilted = TRUE)
```

---

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_case, dict_moderator,
  order_case, order_moderator, addMean, mean.col = "darkred",
  mean.pch = 18, mean.cex = 2, mod.title, labels.tilted, trunc = 20,
  trunc.method = "auto", max_line, dict = getFplot_dict(), ...)
```

## Arguments

| | |
|---|---|
| fml | A formula of the type: var ~ case \| moderator. Note that if a formula is provided then the argument 'data' must be provided. |
| 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 is 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_case | A named character vector. If provided, it changes the values of the variable 'case' to the ones contained in the vector dict_case. Example: I want to change my variable named "a" to "Australia" and "b" to "Brazil", then I used dict=c(a="Australia",b="Brazil"). |
| dict_moderator | A named character vector. If provided, it changes the values of the variable 'moderator' to the ones contained in the vector dict_moderator. Example: I want to change my variable named "a" to "Australia" and "b" to "Brazil", then I used dict=c(a="Australia",b="Brazil"). |
| 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. By default it is equal to the moderator name (possibly modified by the argument dict) if the moderator is numeric, and empty if the moderator is *not* numeric. You can set it to TRUE 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 mod.method). |
| trunc | If the main variable is a character, its values are truncaded 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", "trimRight" or "trimRight". |
| max_line | Option for the x-axis, how far should the labels go. Default is 1 for normal labels, 2 for tilted labels. |
| dict | A dictionnary 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. |
| ... | Other parameters to be passed to plot. |

## Value

Invisibly returns the coordinates of the x-axis.

### Author(s)

Laurent Berge

### Examples

```
m = iris
m$period = sample(1:4, 150, TRUE)

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

---

`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, maxFirst, toLog, maxBins,
  bin.size, legend_options = list(), onTop, yaxis.show = TRUE,
  yaxis.num, col, outCol = "black", mod.method, mod.select, tick_5,
  labels.tilted, addOther, cumul = FALSE, plot = TRUE, sep,
  centered = TRUE, weight.fun, int.categorical, dict = getFplot_dict(),
  mod.title, 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: `var ~ moderator \| weight`. If there are no moderator nor weights, you can use directly a vector, or use `fml = var ~ 1`. To use weights and no moderator, use `fml = var ~ 1 \| weight`. 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. |
| `maxFirst` | 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. |
| `toLog` | 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. |

| | |
|---|---|
| maxBins | 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. |
| onTop | 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 mod.method = "sideTotal", mod.method = "splitTotal" or mod.method = "stack", 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". |
| outCol | Outer color of the bars. Defaults is "black". |
| mod.method | A character scalar: either "splitWithin", the default for categorical data, "splitTotal", "sideWithin", the default for data in logarithmic form or numeric data, "sideTotal" or "stack". This is only used when there is more than one moderator. If within: the bars represent the distribution within each moderator class; if total, the heights of the bar represent the share in the total distribution. 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 (in this case the within distribution does not make sense). |
| 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. |
| tick_5 | Logical. When plotting categorical variables, adds small discrete ticks every 5 bars, and bigger ticks every 10 bars. Helps to get the rank of the bars. By default it is equal to TRUE when strictly more than 10 bars have to be plotted. |
| labels.tilted | Whether there should be tilted labels. Default is FALSE except when the data is split by moderators (see mod.method). |
| addOther | 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 maxFirst=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 dictionnary 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. By default it is equal to the moderator name (possibly modified by the argument dict) if the moderator is numeric, and empty if the moderator is *not* numeric. You can set it to TRUE 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 truncaded 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", "trimRight" or "trimRight". |
| ... | Other elements to be passed to plot. |

## Author(s)

Laurent Berge

## Examples

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

# 1) Let's plot the distribution of publications by institutions:
plot_distr(institution~1, 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(institution ~ 1 | jnl_top_5p, 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(institution ~ 1 | jnl_top_5p, 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, addOther = 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 in the same axis:
plot_distr(Sepal.Length ~ Species, iris, mod.method = "sideWithin")
```

---

`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, 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 = getFplot_dict(), mod.title, ...)
```

## 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'. |

| | |
|---|---|
| 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. |
| 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 \8codec(19, 17, 15, 8, 5, 4, 3, 1). |
| legend_options | A list containing additional parameters for the function legend – only concerns the moderator. Not 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 dictionnary 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. By default it is equal to the moderator name (possibly modified by the argument dict) if the moderator is numeric, and empty if the moderator is *not* numeric. You can set it to TRUE to display the moderator name. To display no title, set it to NULL or FALSE. |
| ... | Other arguments to be passed to the function plot. |

## Author(s)

Laurent Berge

## Examples

```
df = iris
df$period = sample(1:4, 150, TRUE)

plot_lines(Petal.Length ~ period|Species, df)

plot_lines(Petal.Length ~ Species, df)
```

---

`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.) "$ miilion" and "U.S. state", then use `dict = c(us_md="$ million",state = "U.S. state")`. |

## Author(s)

Laurent Berge

## Examples

```
library(fixest)
data(trade)
setFplot_dict(c(Origin = "Country of Origin", Euros = "Exportations"))
plot_distr(Origin~1|Euros, trade)
setFplot_dict()
```

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