

# Package: rplotengine (via r-universe)

September 8, 2024

**Type** Package

**Version** 1.0-9

**Date** 2023-03-13

**Title** R as a Plotting Engine

**Depends** R (>= 2.6.2), xtable

**Description** Generate basic charts either by custom applications, or from a small script launched from the system console, or within the R console. Two ASCII text files are necessary: (1) The graph parameters file, which name is passed to the function 'rplotengine()'. The user can specify the titles, choose the type of the graph, graph output formats (e.g. png, eps), proportion of the X-axis and Y-axis, position of the legend, whether to show or not a grid at the background, etc. (2) The data to be plotted, which name is specified as a parameter ('data\_filename') in the previous file. This data file has a tabulated format, with a single character (e.g. tab) between each column. Optionally, the file could include data columns for showing confidence intervals.

**License** GPL (>= 2)

**URL** <http://www.umh.es>

**NeedsCompilation** no

**Author** Pedro-Pablo Garrido Abenza [aut, cre]

**Maintainer** Pedro-Pablo Garrido Abenza <pgarrido@umh.es>

**Repository** CRAN

**Date/Publication** 2023-03-13 10:40:02 UTC

## Contents

rplotengine .....	2
<b>Index</b>	<b>7</b>

---

 rplotengine

*R as a plotting engine*


---

### Description

Generate a set of graphs as defined in the file specified.

### Usage

```
rplotengine (args_file = "mygraph.arg")
```

### Arguments

**args\_file** A character string giving the name of the text file containing the parameters for the graph to be plotted. The format of such input file is described in detail below.

### Details

The **input file** name passed to **rplotengine** is a plain text file with a list of pairs parameter=value, one-per-line. The meaning of each parameter is explained below:

**title** Main title that can be shown in the graph. It is required to specify show\_titles=1; otherwise the value will be ignored.

**subtitle** Subtitle that can be shown in the graph. It is required to specify show\_titles=1; otherwise the value will be ignored.

**x\_axis\_title** Title for the X-axis.

**y\_axis\_title** Title for the Y-axis.

**col\_x\_values** Column number of the values for the X-axis within the input data file (1-based). DEPRECATED: since 1.0-6 the first column (1) is always the values for the X-axis (this argument will be ignored).

**col\_y\_values** List with the columns number within the input data file corresponding to the series to be plotted. In case of using confidence intervals, each column within the input data file should be followed by another column containing its corresponding confidence interval values, but that column is not specified here, and this is why a typical value for this argument is something like: 2,4,6,8 (being 3,5,7,9 the corresponding columns for the confidence intervals). DEPRECATED: since 1.0-6 all columns in the data file will be used (this argument will be ignored).

**series\_names** Name for the series to be plotted as should be shown in the legend. The number of names specified should match with the number of columns specified in the argument col\_y\_values. In case of showing an overall serie (total\_serie greater than 0), an additional name for it should be specified too.

**x\_factor** Value which the X-axis values will be multiplied by (usually 1).

**y\_factor** Value which the Y-axis values will be multiplied by (usually 1).

- total\_serie** Value ranged from 0 to 4, to indicate if an additional summary serie should be shown. The values mean: 0-none; 1-sum of all the series; 2-average of all the series; 3-constant value (specified by `total_value`); 4-proportional to the x value (proportion specified by `total_value`). In case of a total serie be shown (argument `total_serie` greater than 0), an additional name need to be specified in the list of names (argumento `series_names`).
- total\_value** Value used for the argument `total_serie` in certain cases (3 and 4); otherwise, it is ignored.
- x\_min** Minimum value for the X-axis (usually 0). Specify 'automatic' (without the quotes) in order to consider the minimum value from the data.
- x\_max** Maximum value for the X-axis. It is recommended to adjust automatically the graph depending on the data by assigning the value 'automatic' (without the quotes).
- y\_min** Minimum value for the Y-axis (usually 0). Specify 'automatic' (without the quotes) in order to consider the minimum value from the data. When using logarithmic scale in the y-axis (`y_log=1`), this value cannot be 0.0; the script try to fix this automatically.
- y\_max** Maximum value for the Y-axis (usually -1). It is recommended to adjust automatically the graph depending on the data by assigning the value 'automatic' (without the quotes).
- y\_log** Value 0/1 to indicate if the Y-axis should be shown in logarithm scale (usually 0).
- smooth\_data** Smooth data; 0.0 (data as is) to 1.0 (maximum).
- show\_titles** Value 0/1 to indicate if title and subtitle are shown.
- show\_grid** Value 0/1 to indicate if a grid are shown in the background.
- show\_confint** Value 0/1 to indicate if confidence interval are shown.
- confint\_as\_percentage** Value 0/1 to indicate if the values for the confidence interval within the input data file are expressed as absolute values (0) or as a % value (1, the margin need to be calculated).
- text\_size\_title** Font size for the title. The value specified will be a proportion to the normal text (1.0); for a greater font size specify values > 1.0; for smaller font size specify values < 1.0.
- text\_size\_subtitle** Font size for the subtitle. The value specified will be a proportion to the normal text (1.0), as explained before.
- text\_size\_axis\_ticks** Font size for the X-axis and Y-axis numbers. The value specified will be a proportion to the normal text (1.0), as explained before.
- text\_size\_axis\_titles** Font size for the X-axis and Y-axis titles. The value specified will be a proportion to the normal text (1.0), as explained before.
- text\_size\_legend** Font size for the series names in the legend. The value specified will be a proportion to the normal text (1.0), as explained before.
- pos\_legend** Place for the legend in the graph, within a 3x3 grid with cells numbered from 1 to 9: 1-top left corner, 2-top middle; 3-top right corner; ... ; 9-bottom right corner. Specify a 0 value if a legend is not required.
- graph\_type** Type of graph to generate (0:lines; 1:stacked bars).
- lines\_width** Width of lines for series (default: 1.0); total series will be shown x3.
- hotspots\_size** Size of the hotspots (default: 1.0).
- width\_factor** Graph width. The value specified will be a proportion to the normal size (1.0); for a greater font size specify values > 1.0; for smaller font size specify values < 1.0.

**height\_factor** Graph height. The value specified will be a proportion to the normal size (1.0): for a greater font size specify values > 1.0; for smaller font size specify values < 1.0.

**dpi** Dots per inch; by default, dpi=72. This parameter affects to the width and height of the chart.

**latex\_digits** Decimal digits for tables exported to latex (requires the 'xtable' R package).

**verbose** Value 0/1 to indicate if debug messages are printed on the screen.

**data\_filename** Path and name for the input data file. Path may be absolute or relative to the current directory.

**graph\_filename** Path and name for the graphs to be generated (without suffix). Path may be absolute or relative to the current directory.

**graph\_fileext\_seq** List of suffixes for each output format to generate. For example: png,eps,pdf. The latter suffix 'pdf' will generate a PDF file with embedded fonts; the other formats not embed the fonts used. The final graph file names generated will be the specified with graph\_filename joined to each of these suffixes.

The following is an **example of input file** called 'mygraph.arg'.

```
# -----
# File: mygraph.arg
# -----
title=QoE - PSNR (AI mode; all tiles; varying BGT)
subtitle=Varying Background Traffic
x_axis_title=Background packets per second
y_axis_title=PSNR (dB)
col_x_values=1
col_y_values=2,4,6,8
series_names=Serie 1, Serie 2, Serie 3, Serie 4, Maximum, Acceptable
x_factor=1.0
y_factor=1.0
total_serie=3, 3
total_value=36.14, 27.0
x_min=automatic
x_max=automatic
y_min=14.5
y_max=36.5
y_log=0
smooth_data=0
show_titles=0
show_grid=1
show_confint=0
confint_as_percentage=0
text_size_title=1.0
text_size_subtitle=1.2
text_size_axis_ticks=1.6
text_size_axis_titles=1.8
text_size_legend=1.6
pos_legend=7
graph_type=2
```

```

lines_width=2
hotspots_size=3
width_factor=1.0
height_factor=1.0
dpi=72
latex_digits=3
verbose=1
data_filename=mydata.txt
graph_filename=mygraph
graph_fileext_seq=png,eps,pdf

```

In the previous file, a data file called 'mydata.txt' is specified with the `data_filename` parameter. The following is an **example of input data file** (slightly modified for clarity), which includes columns for the 95% confidence intervals for each statistic:

```

0 36.141 36.142 36.139 36.139
25 29.271 30.083 31.751 31.839
50 22.076 23.907 26.977 27.684
75 15.383 19.44 22.346 23.899

```

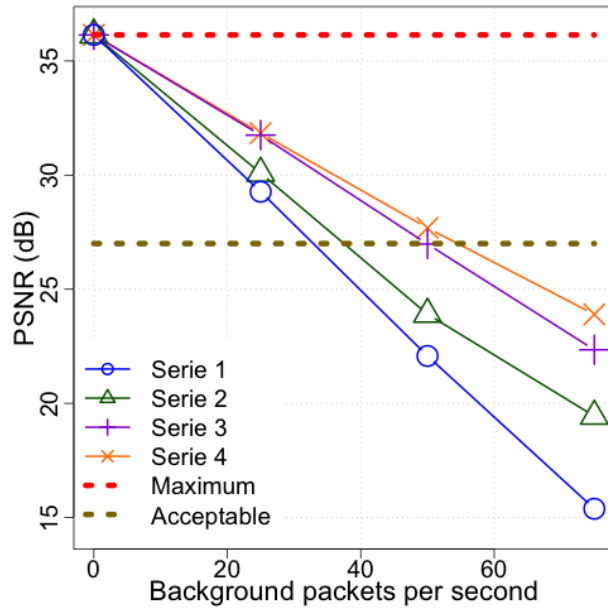
As we can see, column #1 is the X-axis values, as set with the `col_x_values` parameter. On the other hand, four series are going to be shown, corresponding to the columns specified by `col_y_values` parameter: 2,4,6,8. The rest of columns (3,5,7,9) are the confidence intervals computed for each serie, respectively.

As a result of executing the command `rplotengine("mygraph.arg")` within the R console, the following graphs files will be generated: 'mygraph.png', 'mygraph.eps' and 'mygraph.pdf'. In addition, a LaTeX file 'mygraph.tex' will be written with the same values as in the input data file but formatted as a table ready for be imported from a LaTeX document. **NOTE:** the package 'xtable' should be installed and loaded before: `install.packages('xtable');library('xtable')`.

The same can be achieved from the system console with the command:

```
Rscript -e "library('xtable');source('rplotengine.R');rplotengine('mygraph.arg')".
```

The following graph 'mygraph.png' is the result of the previous command with the parameter file 'mygraph.arg' and data file 'mydata.txt' shown:



### Value

This function return a logical value indicating if the operation ended successfully (TRUE) or not (FALSE).

### Author(s)

Pedro-Pablo Garrido Abenza (<pgarrido@umh.es>  
 Miguel Hernandez University (Elche, Spain)

### Examples

```
## Generate the graphs as defined in the 'mygraph.arg' file.
rplotengine ()
## Generate the graphs as defined in the 'mygraph.arg' file.
rplotengine ("mygraph.arg")
```

# Index

rplotengine, [2](#)