# Package 'export'

May 8, 2026

**Type** Package

**Title** Streamlined Export of Graphs and Data Tables

**Version** 0.3.2

**Maintainer** Tom Wenseleers <tom.wenseleers@kuleuven.be>

**Depends** R (>= 3.0)

**Imports** stats, utils, datasets, grDevices, officer (>= 0.2.2), rvg (>= 0.1.8), xtable (>= 1.8-2), flextable (>= 0.4.3), xml2 (>= 1.2.0), stargazer (>= 5.2.1), openxlsx (>= 4.0.17), broom (>= 0.4.4), devEMF (>= 3.8)

**Suggests** rgl (>= 0.99.16), ggplot2 (>= 1.0)

**Description** Easily export 'R' graphs and statistical output to 'Microsoft Office' / 'LibreOffice', 'Latex' and 'HTML' Documents, using sensible defaults that result in publication-quality output with simple, straightforward commands. Output to 'Microsoft Office' is in editable 'DrawingML' vector format for graphs, and can use corporate template documents for styling. This enables the production of standardized reports and also allows for manual tidy-up of the layout of 'R' graphs in 'Powerpoint' before final publication. Export of graphs is flexible, and functions enable the currently showing R graph or the currently showing 'R' stats object to be exported, but also allow the graphical or tabular output to be passed as objects. The package relies on package 'officer' for export to 'Office' documents,and output files are also fully compatible with 'LibreOffice'. Base 'R', 'ggplot2' and 'lattice' plots are supported, as well as a wide variety of 'R' stats objects, via wrappers to xtable(), broom::tidy() and stargazer(), including aov(), lm(), glm(), lme(), glmnet() and coxph() as well as matrices and data frames and many more...

**License** GPL-2

**BugReports** https://github.com/tomwenseleers/export/issues

**RoxygenNote** 7.2.2

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Tom Wenseleers [aut, cre], Christophe Vanderaa [aut]

**Repository** CRAN
**Date/Publication** 2025-08-18 10:00:11 UTC

# Contents



---

`graph2bitmap` *Save currently active R graph to bitmap format*

---

## Description

Save the currently active R graph or a graph passed as an object or function to bitmap format with sensible defaults

## Usage

```
graph2bitmap(
  x = NULL,
  file = "Rplot",
  fun = NULL,
  type = c("PNG", "JPG", "TIF"),
  aspectr = NULL,
  width = NULL,
  height = NULL,
  dpi = 300,
  scaling = 100,
  font = ifelse(Sys.info()["sysname"] == "Windows", "Arial", "Helvetica")[[1]],
  bg = "white",
  cairo = TRUE,
  tiffcompression = c("lzw", "rle", "jpeg", "zip", "lzw+p", "zip+p"),
  jpegquality = 99,
  ...
)

graph2png(...)

graph2tif(...)

graph2jpg(...)
```

## Arguments

| | |
|---|---|
| x | given ggplot2 plot or lattice plot object to export; if set to NULL the currently active R graph will be exported; not supported for base R plots. |
| file | name of output file. Any extension is ignored and added according to the requested output type. If file already exists it is overwritten. |
| fun | plot passed on as a function used to create it; useful especially for base R plots. |
| type | desired output type - PNG, TIF or JPG are currently supported. PNG is the preferred format, as it is a lossless format, and compresses better than TIF. |
| aspectr | desired width to height aspect ratio. If set to NULL, the aspect ratio of the graphics device is used. Can also be combined with one value for either the desired width or height of the graph. |
| width | desired width in inches; can be combined with a desired aspect ratio aspectr. |
| height | desired height in inches; can be combined with a desired aspect ratio aspectr. |
| dpi | desired output in dpi; defaults to 600 dpi. |
| scaling | scale width & height by a certain percentage. |
| font | desired font to use for labels in PNG and TIFF output; defaults to "Arial" on Windows systems and to "Helvetica" on other systems. |
| bg | desired background colour, e.g. "white" or "transparent". |
| cairo | logical, specifying whether or not to use Cairographics for export. |
| tiffcompression | compression to use for TIF files. |
| jpegquality | quality of JPEG compression. |
| ... | any other options are passed on to grDevices' png, tiff, or jpeg function (according to the supplied type). |

## Value

No return value

## Functions

- graph2png(): Save currently active R graph to png file
- graph2tif(): Save currently active R graph to TIF file
- graph2jpg(): Save currently active R graph to JPEG file

## Author(s)

Tom Wenseleers

## See Also

graph2office, graph2vector, graph2svg, graph2pdf, graph2eps

## Examples

```
# Create a file name
filen <- tempfile(pattern = "ggplot")
# or
# filen <- paste("YOUR_DIR/ggplot")

# Generate graphical output
library(ggplot2)
library(datasets)
x <- qplot(Sepal.Length, Petal.Length, data = iris,
           color = Species, size = Petal.Width, alpha = I(0.7))
plot.fun <- function() {
  print(qplot(Sepal.Length, Petal.Length, data = iris,
              color = Species, size = Petal.Width, alpha = 0.7))
}

# There are 3 ways to use graph2bitmap():
### 1. Pass the plot as an object
graph2png(x = x, file = filen, dpi = 400, height = 5, aspectr = 4)
graph2tif(x = x, file = filen, dpi = 400, height = 5, aspectr = 4)
graph2jpg(x = x, file = filen, dpi = 400, height = 5, aspectr = 4)
### 2. Get the plot from current screen device
x
graph2png(file = filen, dpi = 400, height = 5, aspectr = 4)
graph2tif(file = filen, dpi = 400, height = 5, aspectr = 4)
graph2jpg(file = filen, dpi = 400, height = 5, aspectr = 4)
### 3. Pass the plot as a function
graph2png(file = filen, fun = plot.fun, dpi = 400, height = 5, aspectr = 4)
graph2tif(file = filen, fun = plot.fun, dpi = 400, height = 5, aspectr = 4)
graph2jpg(file = filen, fun = plot.fun, dpi = 400, height = 5, aspectr = 4)
```

---

`graph2office` *Save currently active R graph to Microsoft Office / LibreOffice format*

---

## Description

Save the currently active R graph or a graph passed as an object or function to Microsoft Office / LibreOffice format with sensible defaults

## Usage

```
graph2office(
  x = NULL,
  file = "Rplot",
  fun = NULL,
  type = c("PPT", "DOC"),
  append = FALSE,
  aspectr = NULL,
  width = NULL,
```

```
  height = NULL,
  scaling = 100,
  paper = "auto",
  orient = ifelse(type[1] == "PPT", "landscape", "auto"),
  margins = c(top = 0.5, right = 0.5, bottom = 0.5, left = 0.5),
  center = TRUE,
  offx = 1,
  offy = 1,
  upscale = FALSE,
  vector.graphic = TRUE,
  ...
)

graph2ppt(...)

graph2doc(...)
```

## Arguments

| | |
|---|---|
| x | given ggplot2 plot or lattice plot object to export; if set to NULL the currently active R graph will be exported; not supported for base R plots. |
| file | name of output file. Any extension is ignored and added according to the requested output type. |
| fun | plot passed on as a function used to create it; useful especially for base R plots. |
| type | desired output type - DOC for Word document, PPT for Powerpoint. |
| append | logical value - if TRUE and type=PPT it will append the graph to the given file, where file can also be a given corporate template. If append=FALSE any existing file will be overwritten. Currently ignored in Word export. |
| aspectr | desired width to height aspect ratio. If set to NULL, the aspect ratio of the active graphics device is used. |
| width | desired width in inches; can be combined with a desired aspect ratio aspectr. |
| height | desired height in inches; can be combined with a desired aspect ratio aspectr. |
| scaling | scale width & height by a certain percentage. |
| paper | desired paper size to use - "A5" to "A1" for Powerpoint export, or "A5" to "A3" for Word output; default "auto" automatically selects the paper size that fits your graph. Graphs that are too large to fit on a given paper size are scaled down. |
| orient | desired paper orientation - "auto", "portrait" or "landscape"; default to "auto" for Word output and to "landscape" for Powerpoint. |
| margins | vector with the desired margins that should be left blank in |
| center | logical specifying whether or not to center the graph in the exported Powerpoint. |
| offx | if center is set to FALSE, the desired x offset at which to place one's graph in Powerpoint output. |
| offy | if center is set to FALSE, the desired y offset at which to place one's graph in Powerpoint output. |

| | |
|---|---|
| upscale | logical specifying whether or not to upscale one's graph to make it page-filling (excluding the margins). Note that scaling may result in a different look of one's graph relative to how it looks on the screen due to the change in size. |
| vector.graphic | logical specifying whether or not to output in vectorized format. This avoids pixelated images in the document. Note that for PowerPoint, the image can be edited after first ungrouping the plot elements. If set to FALSE, the plot is rasterized to PNG bitmap format at a resolution of 300 dpi. |
| ... | any other options are passed on to rvg's dml_pptx function if type == "DOC" or to devEMF's emf function if type == "PPT" (only when vector.graphics == TRUE). |

## Value

No return value

## Functions

- graph2ppt(): Save currently active R graph to a Microsoft Office PowerPoint/LibreOffice Impress presentation
- graph2doc(): Save currently active R graph to a Microsoft Office Word/LibreOffice Writer document

## Author(s)

Tom Wenseleers, Christophe Vanderaa

## See Also

graph2vector, graph2svg, graph2pdf, graph2eps, graph2bitmap, graph2png, graph2tif, graph2jpg

## Examples

```
# Create a file name
filen <- tempfile(pattern = "ggplot") # or
# filen <- paste("YOUR_DIR/ggplot")

# Generate graphical output
library(ggplot2)
library(datasets)
x=qplot(Sepal.Length, Petal.Length, data = iris,
        color = Species, size = Petal.Width, alpha = I(0.7))
plot.fun = function(){
  print(qplot(Sepal.Length, Petal.Length, data = iris,
              color = Species, size = Petal.Width, alpha = I(0.7)))
}

# There are 3 ways to use graph2office():
### 1. Pass the plot as an object
graph2ppt(x=x, file=filen)
graph2doc(x=x, file=filen, aspectr=0.5)
```

```
### 2. Get the plot from current screen device
if (interactive()) {
  x
  graph2ppt(file=filen, width=9, aspectr=2, append = TRUE)
  graph2doc(file=filen, aspectr=1.7, append =TRUE)
  # Note this requires a graphical device
}
### 3. Pass the plot as a function
if (interactive()) {
  graph2ppt(fun=plot.fun, file=filen, aspectr=0.5, append = TRUE)
  graph2doc(fun=plot.fun, file=filen, aspectr=0.5, append = TRUE)
  # Note this requires a graphical device
}

### Formatting options:
# Disable vectorized image export (export as a bitmap)
graph2ppt(x=x, file=filen, vector.graphic=FALSE, width=9,
          aspectr=sqrt(2), append = TRUE)
# Fill the slide with graph
graph2ppt(x=x, file=filen, margins=0, upscale=TRUE, append=TRUE)
# etc...
```

---

graph2vector *Save currently active R graph to vector format*

---

## Description

Save the currently active R graph or a graph passed as an object or function to vector format with sensible defaults

## Usage

```
graph2vector(
  x = NULL,
  file = "Rplot",
  fun = NULL,
  type = "SVG",
  aspectr = NULL,
  width = NULL,
  height = NULL,
  scaling = 100,
  font = ifelse(Sys.info()["sysname"] == "Windows", "Arial", "Helvetica")[[1]],
  bg = "white",
  colormodel = "rgb",
  cairo = TRUE,
  fallback_resolution = 600,
  ...
)
```

```
graph2svg(...)

graph2pdf(...)

graph2eps(...)
```

## Arguments

| | |
|---|---|
| x | given ggplot2 plot or lattice plot object to export; if set to NULL the currently active R graph will be exported; not supported for base R plots. |
| file | name of output file. Any extension is ignored and added according to the requested output type. If file already exists it is overwritten. |
| fun | plot passed on as a function used to create it; useful especially for base R plots. |
| type | desired output type - SVG, PDF or EPS are currently supported. SVG is the preferred format, and good for editing in Inkscape; PDF is good for printing; EPS is sometimes requested by journals, though lower quality, especially when semi-transparency is used, as this is rasterized to bitmap. graph2office is recommended for vector output to Microsoft Office. |
| aspectr | desired width to height aspect ratio. If set to NULL, the aspect ratio of the graphics device is used. Can also be combined with one value for either the desired width or height of the graph. |
| width | desired width in inches; can be combined with a desired aspect ratio aspectr. |
| height | desired height in inches; can be combined with a desired aspect ratio aspectr. |
| scaling | scale width & height by a certain percentage. |
| font | desired font to use for labels; defaults to "Arial" on Windows systems and to "Helvetica" on other systems. Fonts are embedded by default in EPS output. |
| bg | desired background colour, e.g. "white" or "transparent". |
| colormodel | desired colormodel in pdf or eps output when cairo=FALSE; currently allowed values are "rgb" (default), "cmyk", "srgb", "srgb+gray", "rgb-nogray", and "gray" (or "grey"). |
| cairo | logical indicating whether or not to use the cairo graphics device for output to PDF or EPS, defaults to TRUE, thereby allowing for simulated semi-transparency in EPS output, by rasterizing semi-transparent sections, and automated font embedding. |
| fallback_resolution | resolution in dpi to use to rasterize non-supported vector graphics (e.g. semi-transparent vector elements in EPS) output). |
| ... | any other options are passed on to svg, cairo_pdf, cairo_ps, pdf or postscript. |

## Value

No return value

## Functions

- `graph2svg()`: Save currently active R graph to SVG format
- `graph2pdf()`: Save currently active R graph to PDF format
- `graph2eps()`: Save currently active R graph to EPS format

## Author(s)

Tom Wenseleers

## See Also

graph2office, graph2bitmap, graph2png, graph2tif, graph2jpg

## Examples

```
# Create a file name
filen <- tempfile(pattern = "ggplot") # or
# filen <- paste("YOUR_DIR/ggplot")

# Generate graphical output
library(ggplot2)
library(datasets)
x=qplot(Sepal.Length, Petal.Length, data = iris,
        color = Species, size = Petal.Width, alpha = I(0.7))
plot.fun <- function(){
  print(qplot(Sepal.Length, Petal.Length, data = iris,
              color = Species, size = Petal.Width, alpha = 0.7))
}

# There are 3 ways to use graph2vector():
### 1. Pass the plot as an object
graph2svg(x=x, file=filen, aspectr=2, font = "Times New Roman",
          height = 5, bg = "white")
graph2pdf(x=x, file=filen, aspectr=2, font = "Arial",
          height = 5, bg = "transparent")
graph2eps(x=x, file=filen, aspectr=2, font = "Arial",
          height = 5, bg = "transparent")
### 2. Get the plot from current screen device
if (interactive()) { # Because the example uses screen devices
  x
  graph2svg(file=filen, aspectr=2, font = "Arial",
            height = 5, bg = "transparent")
  graph2pdf(file=filen, aspectr=2, font = "Times New Roman",
            height = 5, bg = "white")
  graph2eps(file=filen, aspectr=2, font = "Times New Roman",
            height = 5, bg = "white")
}
### 3. Pass the plot as a function
if (interactive()) { # Because the example uses screen devices
  graph2svg(file=filen, fun = plot.fun, aspectr=2, font = "Arial",
            height = 5, bg = "transparent")
```

```
  graph2pdf(file=filen, fun=plot.fun, aspectr=2, font = "Arial",
            height = 5, bg = "transparent")
  graph2eps(file=filen, fun=plot.fun, aspectr=2, font = "Arial",
            height = 5, bg = "transparent")
}
```

---

`rgl2bitmap` *Save currently active rgl 3D graph to bitmap format*

---

## Description

Save currently active rgl 3D graph to bitmap format in current orientation

## Usage

```
rgl2bitmap(file = "Rplot", type = c("PNG"))

rgl2png(...)
```

## Arguments

| | |
|---|---|
| `file` | name of output file. Any extension is ignored and added according to the requested output type. If file already exists it is overwritten. |
| `type` | desired output type - currently only `PNG` is supported. |
| `...` | passing the `rgl2png` arguments to `rgl2bitmap` |

## Value

No return value

## Functions

- `rgl2png()`: Save currently active rgl 3D graph to PNG format

## Author(s)

Tom Wenseleers

## Examples

```
# Create a file name
filen <- tempfile(pattern = "rgl") # or
# filen <- paste("YOUR_DIR/rgl")

# Generate a 3D plot using 'rgl'
x = y = seq(-10, 10, length = 20)
z = outer(x, y, function(x, y) x^2 + y^2)
```

```
rgl::persp3d(x, y, z, col = 'lightblue')

# Save the plot as a png
rgl2png(file = filen)
# Note that omitting 'file' will save in current directory
```

---

`table2office` *Export statistical output to a table in Microsoft Office / LibreOffice format*

---

## Description

Export currently showing R stats object or stats object obj to a Microsoft Office / LibreOffice table

## Usage

```
table2office(
  x = NULL,
  file = "Rtable",
  type = c("PPT", "DOC"),
  append = FALSE,
  digits = 2,
  digitspvals = NULL,
  trim.pval = 1e-16,
  width = NULL,
  height = NULL,
  offx = 1,
  offy = 1,
  font = ifelse(Sys.info()["sysname"] == "Windows", "Arial", "Helvetica")[[1]],
  pointsize = 12,
  add.rownames = FALSE
)

table2ppt(...)

table2doc(...)
```

## Arguments

| | |
|---|---|
| `x` | given R stats object to export; if set to `NULL` the output of the previous R command will be exported. |
| `file` | name of output file. The .pptx or .docx extension is added automatically. |
| `type` | desired output type - `"PPT"` for PowerPoint and `"DOC"` for Word. |
| `append` | logical value - if `TRUE` and `type="PPT"` or `"DOC"` it will append the table to the given file, where file can also be a given corporate. If `append=FALSE` any existing file will be overwritten. |

| | |
|---|---|
| digits | number of digits after the comma (for all numeric columns except p-values or degrees of freedom) |
| digitspvals | number of digits after the comma (for p-values only). The default is equal to digits. |
| trim.pval | a threshold below which the p-values are trimmed as "< trim.pval". |
| width | desired width of table in inches. If the given width exceeds the page or slide width, the table width becomes the page/slide width. |
| height | desired height of table in inches. If the given height exceeds the page or slide height, the table height becomes the page/slide height. |
| offx | x offset in inches to specify horizontal location of table (only for type=="PPT"). |
| offy | y offset in inches to specify vertical location of table (only for type=="PPT"). |
| font | desired font to use for output table; defaults to "Arial" on Windows systems and to "Helvetica" on other systems. |
| pointsize | desired font point size. |
| add.rownames | logical specifying whether or not to add row names. |
| ... | Further arguments to be passed to table2office. |

## Details

Columns corresponding to degrees of freedom (with header "Df" or "df") are always given as integers. Objects that can be exported with table2office are all those supported by xtable and tidy. The function will first use xtable to format the data. If the data class is not supported by xtable the function will then use tidy. The data classes suported by xtable are:

- anova
- aov
- aovlist
- data.frame
- glm
- gmsar
- lagImpact
- lm
- matrix
- prcomp
- sarlm
- sarlm.pred
- spautolm
- sphet
- splm
- stsls
- summary.aov

- summary.aovlist
- summary.glm
- summary.gmsar
- summary.lm
- summary.prcomp
- summary.sarlm
- summary.spautolm
- summary.sphet
- summary.splm
- summary.stsls
- table
- ts
- zoo

The data classes suported by tidy are:

- aareg
- acf
- Arima
- betareg
- biglm
- binDesign
- binWidth
- brmsfit
- btergm
- cch
- character
- cld
- coeftest
- confint.glht
- cv.glmnet
- default
- density
- dgCMatrix
- dgTMatrix
- dist
- emmGrid
- ergm
- felm

- fitdistr
- ftable
- gam
- Gam
- gamlss
- geeglm
- glht
- glmnet
- glmRob
- gmm
- htest
- ivreg
- kappa
- kde
- kmeans
- Line
- Lines
- list
- lme
- lmodel2
- lmRob
- logical
- lsmobj
- manova
- map
- Mclust
- merMod
- mle2
- muhaz
- multinom
- nlrq
- nls
- NULL
- numeric
- orcutt
- pairwise.htest
- plm

- poLCA
- Polygon
- Polygons
- power.htest
- pyears
- rcorr
- ref.grid
- ridgelm
- rjags
- roc
- rowwise_df
- rq
- rqs
- sparseMatrix
- SpatialLinesDataFrame
- SpatialPolygons
- SpatialPolygonsDataFrame
- spec
- speedlm
- stanfit
- stanreg
- summary.glht
- summaryDefault
- survdiff
- survexp
- survfit
- survreg
- tbl_df
- TukeyHSD

## Value

flextable object

## Functions

- table2ppt(): Export statistical output to a table in a Microsoft Office PowerPoint/ LibreOffice Impress presentation
- table2doc(): Export statistical output to a table in a Microsoft Office Word/ LibreOffice Writer document

## Author(s)

Tom Wenseleers, Christophe Vanderaa

## See Also

table2tex, table2html, table2spreadsheet

## Examples

```
# Create a file name
filen <- tempfile(pattern = "table_aov") # or
# filen <- paste("YOUR_DIR/table_aov")

# Generate ANOVA output
fit=aov(yield ~ block + N * P + K, data = npk) # 'npk' dataset from base 'datasets'

# Save ANOVA table as a PPT
### Option 1: pass output as object
x=summary(fit)
if (interactive())
    table2ppt(x=x,file=filen, digits = 1, digitspvals = 3)
### Option 2: get output from console
summary(fit)
if (interactive())
    table2ppt(x=x,file=filen, width=5, font="Times New Roman", pointsize=14,
          digits=4, digitspvals=1, append=TRUE) # append table to previous slide

# Save ANOVA table as a DOC file
if (interactive())
    table2doc(x=x,file=filen, digits = 1, digitspvals = 3)
summary(fit)
if (interactive())
    table2doc(file=filen, width=3.5, font="Times New Roman", pointsize=14,
          digits=4, digitspvals=1, append=TRUE) # append table at end of document
```

---

# table2spreadsheet *Export statistical output to a table in spreadsheet compatible format (.xlsx or .csv)*

---

## Description

Export currently showing R stats object or stats object obj to a Microsoft Excel / LibreOffice Calc or comma-separated value file

## Usage

```
table2spreadsheet(
  x = NULL,
  file = "Rtable",
```

```
  type = c("XLS", "CSV", "CSV2"),
  append = FALSE,
  sheetName = "new sheet",
  digits = 2,
  digitspvals = 2,
  trim.pval = 1e-16,
  add.rownames = FALSE,
  ...
)

table2excel(...)

table2csv(...)

table2csv2(...)
```

## Arguments

| | |
|---|---|
| x | given R stats object to export; if set to NULL the output of the previous R command will be exported. |
| file | name of output file. The .xlsx or .csv extension is added automatically. |
| type | desired output type - "XLS" for Excel and "CSV"/"CSV2" for CSV file. Note that type="CSV2" will generate a CSV file where the value separator is a semi-colon (";") and the decimal separator is a comma (",") |
| append | logical value - if TRUE and type="XLS" it will add a new woorksheet to the given file, where file can also be a given corporate. append=FALSE any existing file will be overwritten. |
| sheetName | a string giving the name of the new sheet that is created (only for type=="XLS"). It must be unique (case insensitive) from any existing sheet name in the file. |
| digits | number of significant digits to show for all columns except for the column with p values. |
| digitspvals | number of significant digits to show for columns with p values. |
| trim.pval | a threshold below which the p-values are trimmed as "< trim.pval". |
| add.rownames | logical specifying whether or not to add row names. |
| ... | extra options are passed on to createStyle for the formatting of the woorksheet. This is only applicable for type=="XLS". |

## Details

Columns corresponding to degrees of freedom (with header "Df" or "df") are always given as integers. Objects that can be exported with table2office are all those supported by xtable and tidy. The function will first use xtable to format the data. If the data class is not supported by xtable the function will then use tidy. The data classes suported by xtable are:

- anova
- aov

- aovlist
- data.frame
- glm
- gmsar
- lagImpact
- lm
- matrix
- prcomp
- sarlm
- sarlm.pred
- spautolm
- sphet
- splm
- stsls
- summary.aov
- summary.aovlist
- summary.glm
- summary.gmsar
- summary.lm
- summary.prcomp
- summary.sarlm
- summary.spautolm
- summary.sphet
- summary.splm
- summary.stsls
- table
- ts
- zoo

The data classes suported by tidy are:

- aareg
- acf
- Arima
- betareg
- biglm
- binDesign
- binWidth
- brmsfit

- btergm
- cch
- character
- cld
- coeftest
- confint.glht
- cv.glmnet
- default
- density
- dgCMatrix
- dgTMatrix
- dist
- emmGrid
- ergm
- felm
- fitdistr
- ftable
- gam
- Gam
- gamlss
- geeglm
- glht
- glmnet
- glmRob
- gmm
- htest
- ivreg
- kappa
- kde
- kmeans
- Line
- Lines
- list
- lme
- lmodel2
- lmRob
- logical

- lsmobj
- manova
- map
- Mclust
- merMod
- mle2
- muhaz
- multinom
- nlrq
- nls
- NULL
- numeric
- orcutt
- pairwise.htest
- plm
- poLCA
- Polygon
- Polygons
- power.htest
- pyears
- rcorr
- ref.grid
- ridgelm
- rjags
- roc
- rowwise_df
- rq
- rqs
- sparseMatrix
- SpatialLinesDataFrame
- SpatialPolygons
- SpatialPolygonsDataFrame
- spec
- speedlm
- stanfit
- stanreg
- summary.glht

- summaryDefault
- survdiff
- survexp
- survfit
- survreg
- tbl_df
- TukeyHSD

## Value

A data frame

## Functions

- table2excel(): Export statistical output to a table in a Microsoft Office Excel/ LibreOffice Calc spreadsheet
- table2csv(): Export statistical output to a table in a CSV format ("," for value separation and "." for decimal)
- table2csv2(): Export statistical output to a table in a CSV format (";" for value separation and "," for decimal)

## Author(s)

Tom Wenseleers, Christophe Vanderaa

## See Also

table2tex, table2html, table2office

## Examples

```
# Create a file name
filen <- tempfile(pattern = "table_aov") # or
# filen <- paste("YOUR_DIR/table_aov")

# Generate ANOVA output
fit=aov(yield ~ block + N * P + K, data = npk) # 'npk' dataset from base 'datasets'
x=summary(fit)

# Save ANOVA table as a CSV
### Option 1: pass output as object
table2csv(x=x,file=filen, digits = 1, digitspvals = 3)
### Option 2: get output from console
summary(fit)
table2csv(file=filen, digits = 2, digitspvals = 4)

# Save ANOVA table as an Excel
# Without formatting of the worksheet
x
```

```
table2excel(file=filen, sheetName="aov_noformatting",
            digits = 1, digitspvals = 3)
# With formatting of the worksheet
table2excel(x=x,file=filen, sheetName="aov_formated",
            append = TRUE, add.rownames=TRUE, fontName="Arial",
            fontSize = 14, fontColour = rgb(0.15,0.3,0.75),
            border=c("top", "bottom"), fgFill = rgb(0.9,0.9,0.9),
            halign = "center", valign = "center", textDecoration="italic")
```

---

`table2tex` *Export statistical output to HTML or Latex table*

---

## Description

Export currently showing R stats object or stats object obj to a HTML or Latex table

## Usage

```
table2tex(
  x = NULL,
  file = "Rtable",
  type = "TEX",
  digits = 2,
  digitspvals = 2,
  trim.pval = 1e-16,
  summary = FALSE,
  standAlone = TRUE,
  add.rownames = FALSE,
  ...
)

table2html(...)
```

## Arguments

| | |
|---|---|
| `x` | given R stats object or list of stats objects to export; if set to `NULL` the output of the previous R command will be exported. |
| `file` | name of output file. The appropriate extension is added automatically. |
| `type` | desired output type - `"TEX"` for Latex and `"HTML"` for HTML. |
| `digits` | number of significant digits to show for all columns except for the column with p values. |
| `digitspvals` | number of significant digits to show for columns with p values. |
| `trim.pval` | a threshold below which the p-values are trimmed as "< `trim.pval`". |
| `summary` | logical indicating whether or not to summarize data files. |

| | |
|---|---|
| standAlone | logical indicating whether exported Latex code should be standalone compilable, or whether it will be pasted into another document. |
| add.rownames | logical indicating whether the names of the rows should be added to the table (inserting a column before first column). |
| ... | extra options are passed on to stargazer. |

## Details

Objects that can be exported are all those supported by xtable, tidy (see table2office for an extensive list of supported methods), or stargazer. The models supported by stargazer are:

- aftreg (eha)
- anova (stats)
- aov (stats)
- aovlist (stats)
- arima (stats)
- betareg (betareg)
- binaryChoice (sampleSelection)
- bj (rms)
- brglm (brglm)
- censReg (censReg)
- coeftest (lmtest)
- coxph (survival)
- coxreg (eha)
- clm (ordinal)
- clogit (survival)
- cph (rms)
- dynlm (dynlm)
- ergm (ergm)
- errorsarlm (spdev)
- felm (lfe)
- gam (mgcv)
- garchFit (fGarch)
- gee (gee)
- glm (stats)
- Glm (rms)
- glmer (lme4)
- glmrob (robustbase)
- gls (nlme)
- Gls (rms)

- gmm (gmm)
- heckit (sampleSelection)
- hetglm (glmx)
- hurdle (pscl)
- ivreg (AER)
- lagarlm (spdep)
- lm (stats)
- lme (nlme)
- lmer (lme4)
- lmrob (robustbase)
- lrm (rms)
- maBina (erer)
- mclogit (mclogit)
- mlogit (mlogit)
- mnlogit (mnlogit)
- mlreg (eha)
- multinom (nnet)
- nlme (nlme)
- nlmer (lme4)
- ols (rms)
- pgmm (plm)
- phreg (eha)
- plm (plm)
- pmg (plm)
- polr (MASS)
- psm (rms)
- rem.dyad (relevent)
- rlm (MASS)
- rq (quantreg)
- Rq (rms)
- selection (sampleSelection)
- svyglm (survey)
- survreg (survival)
- tobit (AER)
- weibreg (eha)
- zeroin (pscl)
- relogit (zelig)
- cloglog.net (zelig)
- gamma.net (zelig)
- probit.net (zelig)
- logit.net (zelig)

## Value

No return value

## Functions

- `table2html()`: Export statistical output to HTML table

## Author(s)

Tom Wenseleers, Christophe Vanderaa

## See Also

table2office ,table2ppt, table2doc, stargazer

## Examples

```
# Create a file name
filen <- tempfile(pattern = "table_aov") # or
# filen <- paste("YOUR_DIR/table_aov")

# Generate ANOVA output
fit=aov(yield ~ block + N * P + K, data = npk) # 'npk' dataset from base 'datasets'
x=summary(fit)

# Export to Latex in standAlone format
if (interactive()) table2tex(x=x,file=filen)
# Export to Latex to paste in tex document
summary(fit) # get output from the console
if (interactive()) table2tex(file=filen, standAlone = FALSE)

# Export to HTML
if (interactive()) table2html(x=x,file=filen) # or
summary(fit) # get output from the console
if (interactive()) table2html(file=filen)
```

# Index