

# Package ‘ggiraphExtra’

July 22, 2025

**Type** Package

**Title** Make Interactive 'ggplot2'. Extension to 'ggplot2' and 'ggiraph'

**Version** 0.3.0

**Maintainer** Keon-Woong Moon <cardiomoon@gmail.com>

**URL** <https://github.com/cardiomoon/ggiraphExtra>

**Depends** R (>= 2.10)

**Imports** ggplot2 (>= 2.2.0), ggiraph (>= 0.3.2), scales, reshape2,  
plyr, mycor, ppcor, grid, mgcv, sjlabelled, sjmisc, stringr,  
tidyr, purrr, dplyr, magrittr, RColorBrewer

**Suggests** TH.data, ggforce, webshot, ztable, moonBook, maps, gcookbook,  
knitr, rmarkdown, testthat

## Description

Collection of functions to enhance 'ggplot2' and 'ggiraph'. Provides functions for exploratory plots. All plot can be a 'static' plot or an 'interactive' plot using 'ggiraph'.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Keon-Woong Moon [aut, cre]

**Repository** CRAN

**Date/Publication** 2020-10-06 07:00:02 UTC

## Contents

addLabelDf . . . . .	3
browsers . . . . .	3
coord_radar . . . . .	4
getMapping . . . . .	4

ggAncova . . . . .	5
ggArea . . . . .	6
ggBar . . . . .	7
ggBoxplot . . . . .	8
ggCatepillar . . . . .	9
ggChoropleth . . . . .	10
ggCLE . . . . .	11
ggCor . . . . .	12
ggDensity . . . . .	14
ggDonut . . . . .	15
ggDot . . . . .	16
ggEffect . . . . .	18
ggErrorBar . . . . .	19
ggHeatmap . . . . .	20
ggHSD . . . . .	22
ggPair . . . . .	23
ggPie . . . . .	24
ggPieDonut . . . . .	25
ggPoints . . . . .	26
ggPredict . . . . .	28
ggRadar . . . . .	29
ggRose . . . . .	30
ggSpine . . . . .	31
ggViolin . . . . .	33
makeEq . . . . .	34
model2df . . . . .	35
myscale . . . . .	35
myscale2 . . . . .	36
newColName . . . . .	36
num2cut . . . . .	37
num2factorDf . . . . .	37
p2chr . . . . .	38
palette2colors . . . . .	38
pastecolon . . . . .	38
pastecomma . . . . .	39
rescale_df . . . . .	39
rose . . . . .	40
subcolors . . . . .	40
summarySE . . . . .	41
taco . . . . .	41
theme_clean . . . . .	42
theme_clean2 . . . . .	42
unselectNumeric . . . . .	43
<b>Index</b>	<b>44</b>

---

addLabelDf	<i>Add value labels to the data.frame</i>
------------	---

---

**Description**

Add value labels to the data.frame

**Usage**

```
addLabelDf(data, mapping = NULL)
```

**Arguments**

data	A data.frame
mapping	Set of aesthetic mappings created by aes or aes_.

---

browsers	<i>Browser market share 2011</i>
----------	----------------------------------

---

**Description**

A phony dataset measuring browser market share

**Usage**

```
browsers
```

**Format**

A data.frame with 12 rows and 3 columns

**browser** browser

**version** browser version

**share** market share, in percentage

---

coord_radar	<i>The radar coordinate system is a modification of polar coordinate system, commonly used for radar chart</i>
-------------	--

---

**Description**

The radar coordinate system is a modification of polar coordinate system, commonly used for radar chart

**Usage**

```
coord_radar(theta = "x", start = 0, direction = 1)
```

**Arguments**

theta	variable to map angle to (x or y)
start	offset of starting point from 12 o'clock in radians
direction	1, clockwise; -1, counterclockwise

---

getMapping	<i>extract variable name from mapping, aes</i>
------------	--

---

**Description**

extract variable name from mapping, aes

**Usage**

```
getMapping(mapping, varname)
```

**Arguments**

mapping	aesthetic mapping
varname	variable name to extract

**Value**

variable name in character

**Examples**

```
require(ggplot2)
mapping=aes(colour=sex)
mapping=aes(x=c(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width))
getMapping(mapping, "colour")
getMapping(mapping, "x")
```

---

`ggAncova`*Make an interactive plot for an ANCOVA model*

---

**Description**

Make an interactive plot for an ANCOVA model

**Usage**

```
ggAncova(x, ...)  
  
## Default S3 method:  
ggAncova(x, mapping, use.label = TRUE, use.labels = TRUE, ...)  
  
## S3 method for class 'formula'  
ggAncova(x, data, ...)  
  
## S3 method for class 'lm'  
ggAncova(x, label = NULL, digits = 1, interactive = FALSE, ...)
```

**Arguments**

<code>x</code>	an object
<code>...</code>	additional arguments passed to the generic function
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data
<code>data</code>	a <code>data.frame</code>
<code>label</code>	A character string of column name be assigned to the label
<code>digits</code>	An integer indicating the number of decimal places
<code>interactive</code>	A logical value. If <code>TRUE</code> , an interactive plot will be returned

**Methods (by class)**

- `default`: Make an interactive plot for an ANCOVA model
- `formula`: Make an interactive plot for an ANCOVA model
- `lm`: Make an interactive plot for an ANCOVA model

**Examples**

```
require(moonBook)  
require(ggplot2)  
require(ggiraph)  
ggAncova(radial, aes(age, NTAV, color=sex), interactive=TRUE)  
fit=lm(NTAV~age+HBP, data=radial)
```

```
ggAncova(fit, interactive=TRUE)
ggAncova(NTAV~age+DM, data=radial)
```

---

ggArea

*Draw an interactive area plot*

---

## Description

Draw an interactive area plot

## Usage

```
ggArea(
  data,
  mapping,
  position = "stack",
  palette = "Blues",
  reverse = TRUE,
  alpha = 0.4,
  size = 0.3,
  use.label = TRUE,
  use.labels = TRUE
)
```

## Arguments

data	A data.frame
mapping	Set of aesthetic mappings created by aes or aes_.
position	Either "stack" or "fill"
palette	A character string indicating the color palette
reverse	If true, reverse palette colors
alpha	Transparency
size	Line size
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data

## Value

An area plot

## Examples

```
require(gcookbook)
require(ggplot2)
ggArea(uspopcode, aes(x=Year, y=Thousands, fill=AgeGroup))
ggArea(uspopcode, aes(x=Year, y=Thousands, fill=AgeGroup), position="fill")
```

---

`ggBar`*Draw an interactive barplot*

---

## Description

Draw an interactive barplot

## Usage

```
ggBar(  
  data,  
  mapping,  
  stat = "count",  
  position = "stack",  
  palette = NULL,  
  horizontal = FALSE,  
  yangle = 0,  
  xangle = 0,  
  maxylev = 6,  
  addlabel = FALSE,  
  labelsize = 5,  
  polar = FALSE,  
  reverse = FALSE,  
  use.label = TRUE,  
  use.labels = TRUE,  
  interactive = FALSE,  
  ...  
)
```

## Arguments

<code>data</code>	A <code>data.frame</code>
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>stat</code>	The statistical transformation to use on the data for this layer, as a string <code>c("count", "identity")</code>
<code>position</code>	Position adjustment. One of the <code>c("fill", "stack", "dodge")</code>
<code>palette</code>	A character string indicating the color palette
<code>horizontal</code>	A logical value. If <code>TRUE</code> , a horizontal bar plot will be returned
<code>yangle</code>	An integer. The value will be used adjust the angle of <code>axis.text.y</code>
<code>xangle</code>	An integer. The value will be used adjust the angle of <code>axis.text.x</code>
<code>maxylev</code>	integer indicating threshold of unique value to be treated as a categorical variable
<code>addlabel</code>	A logical value. If <code>TRUE</code> , label will be added to the plot
<code>labelsize</code>	label size
<code>polar</code>	A logical value. If <code>TRUE</code> , <code>coord_polar()</code> function will be added

<code>reverse</code>	If true, reverse palette colors
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data
<code>interactive</code>	A logical value. If TRUE, an interactive plot will be returned
<code>...</code>	other arguments passed on to <code>geom_bar_interactive</code> .

**Value**

An interactive barplot

**Examples**

```
require(moonBook)
require(ggplot2)
require(ggiraph)
require(plyr)
ggBar(acs, aes(x=Dx, fill=smoking), interactive=TRUE, width=1, colour="white", size=0.2, polar=TRUE)
ggBar(acs, aes(x=Dx, fill=smoking), position="fill", addlabel=TRUE, horizontal=TRUE, width=0.5)
ggBar(acs, aes(x=Dx, fill=smoking), position="dodge", interactive=TRUE, addlabel=TRUE)
ggBar(acs, aes(x=Dx, fill=smoking), position="fill", addlabel=TRUE)
ggBar(rose, aes(x=Month, fill=group, y=value), stat="identity", polar=TRUE, palette="Reds", width=1,
      color="black", size=0.1, reverse=TRUE, interactive=TRUE)
```

---

`ggBoxplot`

*Draw boxplots of a data.frame*

---

**Description**

Draw boxplots of a data.frame

**Usage**

```
ggBoxplot(
  data,
  mapping = NULL,
  rescale = FALSE,
  horizontal = FALSE,
  interactive = FALSE,
  addMean = TRUE,
  position = 0.9,
  use.label = TRUE,
  use.labels = TRUE,
  ...
)
```



**Arguments**

data	a data.frame
mapping	Set of aesthetic mappings created by aes or aes_.
rescale	if true, rescale the data.frame
horizontal	if true, horizontal boxplots will be made
interactive	A logical value. If TRUE, an interactive plot will be returned
addMean	Whether add mean point on the plot
position	An integer. Uses as argument of position_dodge()
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data
...	other arguments passed on to geom_boxplot_interactive

**Examples**

```
require(ggplot2)
require(ggiraph)
require(reshape2)
ggBoxplot(mtcars, rescale=TRUE)
ggBoxplot(mtcars, aes(x=c(mpg, cyl, disp, hp, drat), color=am), rescale=TRUE)
ggBoxplot(mtcars, aes(x=c(mpg, cyl, disp, hp, drat))), rescale=TRUE)
ggBoxplot(mtcars, rescale=TRUE, interactive=TRUE)
ggBoxplot(mtcars, horizontal=TRUE, interactive=TRUE)
```

---

ggCatepillar

---

*Make an interactive catepillar plot*


---

**Description**

Make an interactive catepillar plot

**Usage**

```
ggCatepillar(
  data,
  mapping,
  errorbar = "se",
  interactive = FALSE,
  digits = 1,
  flip = FALSE,
  use.label = TRUE,
  use.labels = TRUE
)
```

**Arguments**

data	a data.frame
mapping	Set of aesthetic mappings created by aes or aes_.
errorbar	which value is displayed with errorbar : "se" or "sd"
interactive	A logical value. If TRUE, an interactive plot will be returned
digits	An integer indicating the number of decimal places
flip	Logical. If TRUE, coord_flip() function is used to make a horizontal plot
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data

**Value**

An interactive catepillar plot

**Examples**

```
require(moonBook)
require(ggiraph)
require(ggplot2)
ggCatepillar(acs, aes(Dx, age, color=HBP))
ggCatepillar(acs, aes(c(Dx, sex), age, color=HBP), interactive=TRUE, flip=TRUE, use.labels=FALSE)
ggCatepillar(acs, aes(age, height, color=sex), errorbar=FALSE, interactive=TRUE)
```

---

ggChoropleth

*Draw an interactive choropleth map*

---

**Description**

Draw an interactive choropleth map

**Usage**

```
ggChoropleth(
  data,
  mapping,
  map,
  palette = "OrRd",
  reverse = FALSE,
  color = "grey50",
  title = "",
  digits = 1,
  interactive = FALSE,
  ...
)
```

**Arguments**

data	a data.frame
mapping	Set of aesthetic mappings created by aes or aes_. Passed on geom_map_interactive. Required mappings are map_id and fill. Possible mapping is facet.
map	a map maybe a result of map_data()
palette	A palette name used for discrete fill var, Default value is "OrRd"
reverse	If true, reverse palette colors
color	A name of color of polygon, Default value is "grey50"
title	A title
digits	An integer indicating the number of decimal places
interactive	Logical. If positive an interactive map will be made
...	other arguments passed on to geom_map_interactive

**Examples**

```
#crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)
#require(ggplot2)
#require(ggiraph)
#require(maps)
#require(mapproj)
#require(reshape2)
#require(RColorBrewer)
#states_map <- map_data("state")
#ggChoropleth(crimes,aes(fill=Murder,map_id=state),map=states_map,interactive=TRUE)
#ggChoropleth(crimes,aes(fill=c(Murder,Rape),map_id=state),map=states_map,interactive=TRUE)
#ggChoropleth(crimes,aes(map_id=state),map=states_map,palette="OrRd",interactive=TRUE)
```

---

ggCLE

*Draw a cleveland dot plot*


---

**Description**

Draw a cleveland dot plot

**Usage**

```
ggCLE(
  data,
  mapping,
  reorderByX = TRUE,
  no = NULL,
  start = 0.99,
  interactive = FALSE,
  decreasing = TRUE,
  use.label = TRUE,
```

```

    use.labels = TRUE,
    ...
  )

```

### Arguments

<code>data</code>	a data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>reorderByX</code>	If true, the data is reordered by x variable
<code>no</code>	Number of data be drawn in plot
<code>start</code>	start point of x axis as ratio to minimum x variable
<code>interactive</code>	A logical value. If TRUE, an interactive plot will be returned
<code>decreasing</code>	Should the sort order be increasing or decreasing?
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data
<code>...</code>	other arguments passed on to <code>geom_point_interactive</code>

### Examples

```

require(ggplot2)
require(ggiraph)
ggCLE(data=mtcars, aes(x=mpg), decreasing=FALSE, interactive=TRUE)
ggCLE(data=mtcars, aes(x=mpg, color=am, facet=am), interactive=TRUE)
if(requireNamespace("gcookbook", quietly=TRUE)){
  require(gcookbook)
  ggCLE(data=tophitters2001, aes(x=avg, y=name, color=lg, facet=lg), no=30, interactive=TRUE)
}

```

---

ggCor

*Draw a heatmap of correlation test*

---

### Description

Draw a heatmap of correlation test

### Usage

```

ggCor(
  data,
  what = 1,
  label = 0,
  colors = NULL,
  title = TRUE,
  mode = 2,
  digits = 2,

```

```

    interactive = FALSE,
    yreverse = TRUE,
    xangle = 45,
    yangle = 0,
    use.label = FALSE,
    ...
  )

```

## Arguments

data	A data.frame
what	if 1, correlation, if 2, partial correlation, if 3, semi-partial correlation
label	if 0, no label(default), if 1, use r value as label, if 2, use r value with significant mark as label
colors	colors for low, mid and high correlation values
title	if true, add title to the heatmap
mode	1 or 2
digits	The number of decimal place
interactive	A logical value. If TRUE, an interactive plot will be returned
yreverse	If true, reverse y axis
xangle	x-axis text angle
yangle	y-axis text angle
use.label	Logical whether or not use label in case of labelled data
...	further arguments to be passed to cor.test

## Examples

```

require(mycor)
require(ggplot2)
require(ggiraph)
require(ppcor)
ggCor(iris,digits=3,label=3)
ggCor(iris,what=3,digits=3,label=3)
ggCor(iris,label=3,interactive=TRUE)
ggCor(mtcars,interactive=TRUE)
ggCor(mtcars,mode=2,interactive=TRUE)
ggCor(iris,method="pearson",interactive=TRUE)

```

---

`ggDensity`*Make a density plot with histogram*

---

## Description

Make a density plot with histogram

## Usage

```
ggDensity(  
  data,  
  mapping,  
  linecolor = "red",  
  addhist = TRUE,  
  use.label = TRUE,  
  use.labels = TRUE  
)
```

## Arguments

<code>data</code>	a data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>linecolor</code>	Color of density curve
<code>addhist</code>	Whether add histogram or not
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data

## Examples

```
require(ggplot2)  
require(moonBook)  
ggDensity(acs, aes(x=age))  
ggDensity(acs, aes(x=age, color=sex, fill=sex), addhist=FALSE)  
ggDensity(acs, aes(x=age, color=sex, fill=sex))  
ggDensity(acs, aes(x=age, fill=sex), addhist=FALSE)  
ggDensity(acs, aes(x=age, color=sex))
```

---

`ggDonut`*Draw a Donut plot*

---

**Description**

Draw a Donut plot

**Usage**

```
ggDonut(  
  data,  
  mapping,  
  addDonutLabel = TRUE,  
  showRatio = TRUE,  
  polar = TRUE,  
  labelposition = 1,  
  labelsize = 3,  
  title = "",  
  use.label = TRUE,  
  use.labels = TRUE,  
  alpha = 0.7,  
  interactive = FALSE,  
  palette = NULL,  
  reverse = FALSE,  
  xmin = 3,  
  xmax = 4,  
  start = 3 * pi/2,  
  direction = 1,  
  colour = "white",  
  explode = NULL,  
  explodePos = 0.5,  
  ...  
)
```

**Arguments**

<code>data</code>	A data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>addDonutLabel</code>	A logical value. If TRUE, labels are added to the Donuts
<code>showRatio</code>	A logical value. If TRUE, Ratios are added to the DonutLabels
<code>polar</code>	A logical value. If TRUE, <code>coord_polar()</code> function will be added
<code>labelposition</code>	A number indicating the label position
<code>labelsize</code>	label size. default value is 3
<code>title</code>	Plot title
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data

<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data
<code>alpha</code>	transparency of <code>geom_rect</code>
<code>interactive</code>	A logical value. If TRUE, an interactive plot will be returned
<code>palette</code>	A character string indicating the color palette
<code>reverse</code>	If true, reverse palette colors
<code>xmin</code>	minimum x position
<code>xmax</code>	maximum x position
<code>start</code>	offset of starting point from 12 o'clock in radians
<code>direction</code>	1, clockwise; -1, counterclockwise
<code>colour</code>	colour of <code>geom_rect</code>
<code>explode</code>	number of donuts to explode
<code>explodePos</code>	explode position
<code>...</code>	further arguments to be passed to <code>geom_rect_interactive</code>

**Value**

An interactive Pie and Donut plot

**Examples**

```
require(ggplot2)
require(ggiraph)
require(ply)
ggDonut(browsers, aes(donuts=version, count=share))
ggDonut(browsers, aes(donuts=version, count=share), palette="Reds", explode=c(2,4,6), labelposition=0)
```

---

ggDot

*Draw a Wilkinson dot plot*

---

**Description**

Draw a Wilkinson dot plot

**Usage**

```
ggDot(
  data,
  mapping,
  stackdir = "center",
  binaxis = "y",
  binwidth = 0.5,
  method = "dotdensity",
  position = 0.2,
  boxwidth = 0.25,
```



```

    boxfill = NULL,
    use.label = TRUE,
    use.labels = TRUE,
    ...
  )

```

## Arguments

data	a data.frame
mapping	Set of aesthetic mappings created by aes or aes_.
stackdir	which direction to stack the dots. "up" (default), "down", "center", "center-hole" (centered, but with dots aligned)
binaxis	The axis to bin along, "x" (default) or "y"
binwidth	When method is "dotdensity", this specifies maximum bin width. When method is "histodot", this specifies bin width. Defaults to 1/30 of the range of the data
method	"dotdensity" (default) for dot-density binning, or "histodot" for fixed bin widths (like stat_bin)
position	Position adjustment. If 0, no adjustment.
boxwidth	The width of boxplot
boxfill	Fill color of boxplot
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data
...	other arguments passed on to geom_dotplot

## Examples

```

require(ggplot2)
if(requireNamespace("gcookbook",quietly=TRUE)){ # for data heightweight
  require(gcookbook)
  ggDot(heightweight,aes(sex,heightIn,fill=sex),boxfill="white",binwidth=0.4)
  ggDot(heightweight,aes(heightIn))
  ggDot(heightweight,aes(x=heightIn,fill=sex))
}
require(moonBook) #for use data radial
ggDot(radial,aes(x=sex,y=height,fill=sex),boxfill="white",position=0,binwidth=1,boxwidth=1)
ggDot(radial,aes(x=height,fill=sex),binwidth=1)
ggDot(acs,aes(x=sex,y=age,color=sex))
ggDot(acs,aes(x=Dx,y=age,color=Dx))

```

---

ggEffect	<i>Visualize the effect of interaction between two continuous independent variables on a response variable</i>
----------	--

---

## Description

Visualize the effect of interaction between two continuous independent variables on a response variable

## Usage

```
ggEffect(x, ...)

## Default S3 method:
ggEffect(x, mapping, use.label = TRUE, use.labels = TRUE, ...)

## S3 method for class 'formula'
ggEffect(x, data, ...)

## S3 method for class 'lm'
ggEffect(
  x,
  no = 1,
  probs = c(0.1, 0.5, 0.9),
  point = TRUE,
  xvalue = NULL,
  digits = 2,
  use.rownames = FALSE,
  interactive = FALSE,
  ...
)
```

## Arguments

x	Object to ggEffect
...	additional arguments passed to the generic function
mapping	Set of aesthetic mappings created by aes or aes_.
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data
data	A data.frame
no	an integer
probs	A vector of probability weights for obtaining the elements of the vector being sampled. Default value is c(0.10,0.5,0.90)
point	A logical value. If TRUE, draw points

xvalue	A numeric vector
digits	An integer indicating the number of decimal places
use.rownames	If TRUE, use rownames in label
interactive	A logical value. If TRUE, an interactive plot will be returned

**Value**

An interactive plot showing interaction

**Methods (by class)**

- `default`: Visualize the effect of interaction between two continuous independent variables on a response variable
- `formula`: Visualize the effect of interaction between two continuous independent variables on a response variable
- `lm`: Visualize the effect of interaction between two continuous independent variables on a response variable

**Examples**

```
require(ggplot2)
require(ggiraph)
ggEffect(mtcars, aes(x=wt, y=mpg, color=hp))
ggEffect(mtcars, aes(x=wt, y=mpg, color=hp), interactive=TRUE)
require(moonBook)
ggEffect(acs, aes(x=height, y=weight, color=smoking))
ggEffect(acs, aes(x=height, y=weight, color=smoking), interactive=TRUE)
require(ggplot2)
require(ggiraph)
require(moonBook)
ggEffect(NTAV~age*smoking, data=radial)
require(moonBook)
require(ggplot2)
require(ggiraph)
fit=lm(age~sex*smoking, data=acs)
ggEffect(fit, interactive=TRUE)
ggEffect(radial, aes(x=age, y=NTAV, color=smoking))
ggEffect(radial, aes(x=age, y=NTAV, color=smoking), interactive=TRUE)
```

---

ggErrorBar

*Make an interactive bar plot with error bar*

---

**Description**

Make an interactive bar plot with error bar

**Usage**

```
ggErrorBar(
  data,
  mapping,
  interactive = FALSE,
  digits = 1,
  mode = 2,
  errorbar = "se",
  use.label = TRUE,
  use.labels = TRUE
)
```

**Arguments**

<code>data</code>	A <code>data.frame</code>
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>interactive</code>	A logical value. If <code>TRUE</code> , an interactive plot will be returned
<code>digits</code>	An integer indicating the number of decimal places
<code>mode</code>	if 2, two-sided error bar will be displayed, if 1 one-sided errorbar will be displayed
<code>errorbar</code>	which value is displayed with errorbar : "se" or "sd"
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data

**Value**

An interactive caterpillar plot

**Examples**

```
require(ggplot2)
require(ggiraph)
ggErrorBar(mpg, aes(x=drv, y=cty))
ggErrorBar(mpg, aes(x=drv, y=hwy, color=cyl), mode=1, interactive=TRUE, errorbar="sd")
```

---

ggHeatmap

*Make an interactive Heatmap*

---

**Description**

Make an interactive Heatmap

**Usage**

```
ggHeatmap(
  data,
  mapping,
  stat = "count",
  palette = "Blues",
  reverse = FALSE,
  addlabel = FALSE,
  polar = FALSE,
  interactive = FALSE,
  yangle = 0,
  color = "grey50",
  size = 0.1,
  use.label = TRUE,
  use.labels = TRUE,
  ...
)
```

**Arguments**

<code>data</code>	A data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>stat</code>	The statistical transformation to use on the data for this layer, as a string <code>c("count","identity")</code>
<code>palette</code>	A palette name used for discrete fill var, Default value is "Blues"
<code>reverse</code>	If true, reverse palette colors
<code>addlabel</code>	A logical value. If TRUE, label will be added to the plot
<code>polar</code>	A logical value. If TRUE, <code>coord_polar()</code> function will be added
<code>interactive</code>	A logical value. If TRUE, an interactive plot will be returned
<code>yangle</code>	A integer. The value will be used adjust the angle of <code>axis.text.y</code>
<code>color</code>	Color argument passed on to <code>geom_rect_interactive</code> .
<code>size</code>	Size argument passed on to <code>geom_rect_interactive</code> .
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data
<code>...</code>	other arguments passed on to <code>geom_rect_interactive</code> .

**Value**

An interactive barplot

**Examples**

```
require(moonBook)
require(ggplot2)
require(ggiraph)
require(sjmisc)
```

```

ggHeatmap(acs, aes(x=Dx, y=smoking), addlabel=TRUE, interactive=TRUE)
ggHeatmap(acs, aes(x=sex, y=Dx, fill=age), addlabel=TRUE, interactive=TRUE)
ggHeatmap(rose, aes(x=Month, y=group, fill=value), stat="identity", addlabel=TRUE)
ggHeatmap(rose, aes(x=Month, y=group, fill=value), addlabel=TRUE)
ggHeatmap(taco, aes(x=AgeGroup, y=Filling, fill=Rating, facet=ShellType), color="grey50", stat="identity")

```

---

ggHSD

*Draw Tukey Honest Significant Differences plot*


---

### Description

Draw Tukey Honest Significant Differences plot

### Usage

```
ggHSD(tukey, no = 1, digits = 2, interactive = FALSE)
```

### Arguments

tukey	A object of class "TukeyHSD", the result of TukeyHSD()
no	An integer specify the order of list
digits	integer indicating the number of decimal places
interactive	A logical value. If TRUE, an interactive plot will be returned

### Value

A (interactive) ggplot

### Examples

```

require(ggplot2)
fm1 <- aov(breaks ~ wool + tension, data = warpbreaks)
result=TukeyHSD(fm1, "tension", ordered = TRUE)
str(result)
ggHSD(result)
ggHSD(result, interactive=TRUE)

```

---

`ggPair`*Make an interactive scatter and line plot*

---

## Description

Make an interactive scatter and line plot

## Usage

```
ggPair(  
  data,  
  mapping = NULL,  
  rescale = FALSE,  
  idcolor = TRUE,  
  horizontal = FALSE,  
  use.label = FALSE,  
  use.labels = TRUE,  
  includeFactor = TRUE,  
  includeAll = FALSE,  
  interactive = FALSE  
)
```

## Arguments

<code>data</code>	a data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>rescale</code>	if true, rescale the data.frame
<code>idcolor</code>	Logical. If TRUE, row numbers uses as a color variable
<code>horizontal</code>	Logical. If TRUE, <code>coord_flip()</code> function is used to make a horizontal plot
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data
<code>includeFactor</code>	Logical. Whether or not include factor variables
<code>includeAll</code>	Logical. Whether or not include all variables
<code>interactive</code>	Logical. If TRUE, an interactive plot using <code>girafe()</code> function will be returned

## Examples

```
require(ggplot2)  
require(ggiraph)  
require(sjmisc)  
require(moonBook)  
ggPair(iris,rescale=TRUE,horizontal=TRUE)  
ggPair(acs,aes(colour=smoking),horizontal=TRUE,rescale=TRUE)  
ggPair(radial,aes(color=male),horizontal=TRUE,rescale=TRUE)  
ggPair(mtcars,horizontal=TRUE,rescale=TRUE)
```

```
ggPair(iris,rescale=TRUE,horizontal=TRUE,interactive=TRUE)
ggPair(iris,aes(color=Species),rescale=TRUE,interactive=TRUE)
ggPair(iris,aes(x=c(Sepal.Length,Sepal.Width),color=Species),horizontal=TRUE,interactive=TRUE)
```

ggPie

*Draw a pie plot***Description**

Draw a pie plot

**Usage**

```
ggPie(
  data,
  mapping,
  addPieLabel = TRUE,
  showRatioPie = TRUE,
  showRatioPieAbove10 = TRUE,
  title = "",
  labelposition = 1,
  polar = TRUE,
  use.label = TRUE,
  use.labels = TRUE,
  interactive = FALSE
)
```

**Arguments**

<code>data</code>	A data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>addPieLabel</code>	A logical value. If TRUE, labels are added to the Pies
<code>showRatioPie</code>	A logical value. If TRUE, Ratios are added to the PieLabels
<code>showRatioPieAbove10</code>	A logical value. If TRUE, labels are added to the Pies with ratio above 10.
<code>title</code>	Plot title
<code>labelposition</code>	A number indicating the label position
<code>polar</code>	A logical value. If TRUE, <code>coord_polar()</code> function will be added
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data
<code>interactive</code>	A logical value. If TRUE, an interactive plot will be returned

**Value**

An interactive pie plot



## Examples

```
require(ggplot2)
require(ggiraph)
require(plyr)
require(moonBook)
ggPie(data=browsers,aes(pies=browser ,count=share))
ggPie(data=acs,aes(pies=Dx))
```

---

ggPieDonut

*Draw a Pie and Donut plot*

---

## Description

Draw a Pie and Donut plot

## Usage

```
ggPieDonut(
  data,
  mapping,
  addPieLabel = TRUE,
  addDonutLabel = TRUE,
  showRatioDonut = TRUE,
  showRatioPie = TRUE,
  showRatioPieAbove10 = TRUE,
  title = "",
  labelposition = 1,
  polar = TRUE,
  use.label = TRUE,
  use.labels = TRUE,
  interactive = FALSE
)
```

## Arguments

data	A data.frame
mapping	Set of aesthetic mappings created by aes or aes_.
addPieLabel	A logical value. If TRUE, labels are added to the Pies
addDonutLabel	A logical value. If TRUE, labels are added to the Donuts
showRatioDonut	A logical value. If TRUE, Ratios are added to the DonutLabels
showRatioPie	A logical value. If TRUE, Ratios are added to the PieLabels
showRatioPieAbove10	A logical value. If TRUE, labels are added to the Pies with ratio above 10.
title	Plot title
labelposition	A number indicating the label position

polar	A logical value. If TRUE, coord_polar() function will be added
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data
interactive	A logical value. If TRUE, an interactive plot will be returned

**Value**

An interactive Pie and Donut plot

**Examples**

```
require(ggplot2)
require(ggiraph)
require(ply)
require(moonBook)
ggPieDonut(acs, aes(pies=Dx, donuts=smoking))
ggPieDonut(acs, aes(pies=smoking))
ggPieDonut(browsers, aes(pies=browser, donuts=version, count=share))
ggPieDonut(browsers, aes(x=c(browser, version), y=share), interactive=TRUE)
```

---

ggPoints

---

*Make an interactive scatterplot with regression line(s)*


---

**Description**

Make an interactive scatterplot with regression line(s)

**Usage**

```
ggPoints(
  data,
  mapping,
  smooth = TRUE,
  se = TRUE,
  method = "auto",
  formula = y ~ x,
  fullrange = FALSE,
  level = 0.95,
  use.count = FALSE,
  maxfactorno = 6,
  digits = 2,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  use.label = TRUE,
  use.labels = TRUE,
  tooltip = NULL,
```

```

    interactive = FALSE,
    ...
  )

```

### Arguments

data	a data.frame
mapping	Set of aesthetic mappings created by aes or aes_.
smooth	Logical. Add regression lines to the scatter plot
se	Logical. display confidence interval around linear regression? (TRUE by default)
method	smoothing method (function) to use, eg. "lm", "glm", "gam", "loess", "rlm"
formula	formula to use in smoothing function, eg. $y \sim x$ , $y \sim \text{poly}(x, 2)$ , $y \sim \log(x)$
fullrange	should the fit span the full range of the plot, or just the data
level	level of confidence interval to use (0.95 by default)
use.count	Logical. If true use geom_count instead of geom_point_interactive
maxfactorno	An integer. Maximum unique number of a numeric vector treated as a factor
digits	integer indicating the number of decimal places
title	The text for plot title
subtitle	The text for plot subtitle
caption	The text for plot caption
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data
tooltip	A character string of column name be included in tooltip. Default value is NULL
interactive	A logical value. If TRUE, an interactive plot will be returned
...	other arguments passed on to geom_point

### Examples

```

require(ggplot2)
require(ggiraph)
require(plyr)
ggPoints(aes(x=wt,y=mpg,fill=am),data=mtcars)
ggPoints(aes(x=wt,y=mpg),data=mtcars)
ggPoints(aes(x=wt,y=mpg,fill=am),data=mtcars,method="lm",interactive=TRUE)
ggPoints(aes(x=wt,y=mpg,color=am),data=mtcars,interactive=TRUE)

```

---

`ggPredict`*Visualize predictions from the multiple regression models.*

---

## Description

Visualize predictions from the multiple regression models.

## Usage

```
ggPredict(  
  fit,  
  colorn = 4,  
  point = NULL,  
  jitter = NULL,  
  se = FALSE,  
  show.summary = FALSE,  
  colorAsFactor = FALSE,  
  digits = 2,  
  interactive = FALSE,  
  ...  
)
```

## Arguments

<code>fit</code>	a model object for which prediction is desired.
<code>colorn</code>	Integer. Number of subgroups of color variables.
<code>point</code>	Logical. Whether or not draw each point
<code>jitter</code>	Logical. Whether or not jitter points
<code>se</code>	Logical. Whether or not draw se
<code>show.summary</code>	Logical. Whether or not show summary
<code>colorAsFactor</code>	Logical. Whether or not treat color variable as categorical variable
<code>digits</code>	An integer indicating the number of decimal places
<code>interactive</code>	A logical value. If TRUE, an interactive plot will be returned
<code>...</code>	additional arguments affecting the predictions produced.

## Examples

```
require(moonBook)  
require(ggplot2)  
require(ggiraph)  
require(plyr)  
fit=lm(NTAV~age*weight,data=radial)  
fit=lm(NTAV~age*weight*DM,data=radial)  
fit=lm(NTAV~age+DM,data=radial)  
ggPredict(fit,interactive=TRUE)
```

```

require(TH.data)
fit=glm(cens~pnodes*horTh,data=GBSG2,family=binomial)
ggPredict(fit,se=TRUE)
fit1=glm(cens~pnodes*age,data=GBSG2,family=binomial)
ggPredict(fit1)
ggPredict(fit1,colorn=100,jitter=FALSE,interactive=TRUE)
fit2=glm(cens~pnodes*age*horTh,data=GBSG2,family=binomial)
ggPredict(fit2,colorn=100,jitter=FALSE,interactive=TRUE)

```

ggRadar

*Draw a radar chart***Description**

Draw a radar chart

**Usage**

```

ggRadar(
  data,
  mapping = NULL,
  rescale = TRUE,
  legend.position = "top",
  colour = "red",
  alpha = 0.3,
  size = 3,
  ylim = NULL,
  scales = "fixed",
  use.label = FALSE,
  interactive = FALSE,
  ...
)

```

**Arguments**

data	A data.frame
mapping	Set of aesthetic mappings created by aes or aes_.
rescale	A logical value. If TRUE, all continuous variables in the data.frame are rescaled.
legend.position	Legend position. One of c("top","bottom","left","right","none")
colour	A name of color to be assigned as a color variable
alpha	Any numbers from 0 (transparent) to 1 (opaque)
size	Point size
ylim	A numeric vector of length 2, giving the y coordinates ranges.
scales	should Scales be fixed ("fixed", the default), free ("free"), or free in one dimension ("free_x", "free_y")

<code>use.label</code>	Logical. Whether or not use column label
<code>interactive</code>	A logical value. If TRUE, an interactive plot will be returned
<code>...</code>	other arguments passed on to <code>geom_point</code>

**Value**

An interactive radar plot

**Examples**

```
require(ggplot2)
require(ggiraph)
require(ply)
require(reshape2)
require(moonBook)
require(sjmisc)
ggRadar(data=iris, aes(group=Species))
ggRadar(data=mtcars, interactive=TRUE)
ggRadar(data=mtcars, aes(colour=am, facet=cyl), interactive=TRUE)
ggRadar(data=acs, aes(colour=Dx, facet=Dx))
ggRadar(iris, aes(x=c(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width)))
```

---

ggRose

*Draw an interactive Rose plot*

---

**Description**

Draw an interactive Rose plot

**Usage**

```
ggRose(data, mapping, palette = "Reds", color = "black", size = 0.1, ...)
```

**Arguments**

<code>data</code>	A data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>palette</code>	A character string indicating the color palette
<code>color</code>	Bar colour
<code>size</code>	Bar size
<code>...</code>	other arguments passed on to <code>geom_bar_interactive</code> .

**Value**

An interactive Rose plot

**Examples**

```
require(moonBook)
require(ggplot2)
require(ggiraph)
require(plyr)
ggRose(rose,aes(x=Month,fill=group,y=value),stat="identity",interactive=TRUE)
ggRose(acs,aes(x=Dx,fill=smoking),interactive=TRUE)
```

---

ggSpine

*Draw an interactive spinogram*

---

**Description**

Draw an interactive spinogram

**Usage**

```
ggSpine(  
  data,  
  mapping,  
  stat = "count",  
  position = "fill",  
  palette = "Blues",  
  interactive = FALSE,  
  polar = FALSE,  
  reverse = FALSE,  
  width = NULL,  
  maxylev = 6,  
  digits = 1,  
  colour = "black",  
  size = 0.2,  
  addlabel = TRUE,  
  labelsize = 5,  
  minlabelgroup = 0.04,  
  minlabel = 2,  
  hide.legend = TRUE,  
  ylabelMean = FALSE,  
  sec.y.axis = FALSE,  
  use.label = TRUE,  
  use.labels = TRUE,  
  labeller = NULL,  
  facetbycol = TRUE,  
  xangle = NULL,  
  yangle = NULL,  
  xreverse = FALSE,  
  yreverse = FALSE,  
  xlab = NULL,
```

```

    filllab = NULL,
    family = NULL,
    ...
)

```

### Arguments

<code>data</code>	A data.frame
<code>mapping</code>	Set of aesthetic mappings created by <code>aes</code> or <code>aes_</code> .
<code>stat</code>	The statistical transformation to use on the data for this layer, as a string <code>c("count", "identity")</code>
<code>position</code>	Position adjustment. One of the <code>c("fill", "stack", "dodge")</code>
<code>palette</code>	A character string indicating the color palette
<code>interactive</code>	A logical value. If TRUE, an interactive plot will be returned
<code>polar</code>	A logical value. If TRUE, <code>coord_polar()</code> function will be added
<code>reverse</code>	If true, reverse palette colors
<code>width</code>	Bar width
<code>maxylev</code>	integer indicating threshold of unique value to be treated as a categorical variable
<code>digits</code>	integer indicating the number of decimal places
<code>colour</code>	Bar colour
<code>size</code>	Bar size
<code>addlabel</code>	A logical value. If TRUE, label will be added to the plot
<code>labelsize</code>	label size
<code>minlabelgroup</code>	minimal threshold of label group. Default is 0.04
<code>minlabel</code>	minimal threshold of label. Default is 2
<code>hide.legend</code>	A logical value. If TRUE, the legend is removed and y labels are recreated
<code>ylabelMean</code>	Logical. If TRUE, y axis labels are positioned at mean value.
<code>sec.y.axis</code>	Logical. If TRUE, secondary y axis is shown at the right side.
<code>use.label</code>	Logical. Whether or not use column label in case of labelled data
<code>use.labels</code>	Logical. Whether or not use value labels in case of labelled data
<code>labeller</code>	A function that takes one data frame of labels and returns a list or data frame of character vectors.
<code>facetbycol</code>	Logical. If TRUE, facet by column.
<code>xangle</code>	angle of axis label
<code>yangle</code>	angle of axis label
<code>xreverse</code>	Logical. Whether or not reverse x-axis
<code>yreverse</code>	Logical. Whether or not reverse y-axis
<code>xlab</code>	Label for x-axis
<code>filllab</code>	Label for fill aes
<code>family</code>	font family
<code>...</code>	other arguments passed on to <code>geom_rect_interactive</code> .



**Value**

An interactive spinogram

**Examples**

```
require(moonBook)
require(ggplot2)
acs$Dx=factor(acs$Dx,levels=c("Unstable Angina","NSTEMI","STEMI"))
ggSpine(data=acs,aes(x=age,fill=Dx,facet=sex),palette="Reds")
ggSpine(data=acs,aes(x=age,fill=Dx,facet=sex),facetbycol=FALSE,minlabelgroup=0.02)
ggSpine(data=acs,aes(x=age,fill=Dx),palette="Reds")
ggSpine(data=acs,aes(x=smoking,fill=Dx),palette="Reds")
ggSpine(data=acs,aes(x=DM,fill=Dx,facet=sex),palette="Reds")
ggSpine(data=acs,aes(x=Dx,fill=smoking,facet=sex),palette="Reds")
ggSpine(data=acs,aes(x=DM,facet=smoking,fill=Dx),sec.y.axis=TRUE)
ggSpine(data=acs,aes(x=DM,facet=smoking,fill=Dx),facetbycol=FALSE)
ggSpine(mtcars,aes(x=gear,fill=carb),interactive=TRUE)
ggSpine(mtcars,aes(x=gear,fill=carb,facet=am))
ggSpine(data=acs,aes(x=Dx,fill=smoking),position="dodge")
ggSpine(data=acs,aes(x=Dx,fill=smoking),position="stack")
```

---

ggViolin

*Draw violin plots of a data.frame*


---

**Description**

Draw violin plots of a data.frame

**Usage**

```
ggViolin(
  data,
  mapping = NULL,
  rescale = FALSE,
  horizontal = FALSE,
  alpha = 0.1,
  addBoxplot = TRUE,
  addMean = TRUE,
  use.label = TRUE,
  use.labels = TRUE,
  ...
)
```

**Arguments**

data	a data.frame
mapping	Set of aesthetic mappings created by aes or aes_.
rescale	if true, rescale the data.frame

horizontal	if true, horizontal boxplots will be made
alpha	An integer. Default value is 0.1.
addBoxplot	Whether add boxplots on the plot
addMean	Whether add mean point on the plot
use.label	Logical. Whether or not use column label in case of labelled data
use.labels	Logical. Whether or not use value labels in case of labelled data
...	other arguments passed on to geom_boxplot_interactive

### Examples

```
require(ggplot2)
require(ggiraph)
require(reshape2)
ggViolin(iris)
ggViolin(iris,aes(fill=Species),rescale=TRUE)
ggViolin(mtcars,aes(x=c(mpg,cyl,disp,hp,drat),color=am),rescale=TRUE)
ggViolin(mtcars,aes(x=c(mpg,cyl,disp,hp,drat)),rescale=TRUE)
```

---

makeEq	<i>Make a regression equation of a model</i>
--------	--

---

### Description

Make a regression equation of a model

### Usage

```
makeEq(model, digits = 2)
```

### Arguments

model	A model of class "lm" or "glm" or "loess"
digits	integer indicating the number of decimal places

---

model2df	<i>Make a data.frame of yhat with a model</i>
----------	---

---

**Description**

Make a data.frame of yhat with a model

**Usage**

```
model2df(model, x = NULL, n = 100)
```

**Arguments**

model	A model of class "lm" or "glm" or "loess"
x	A optional vector of explanatory variable
n	number of observations.

---

myscale	<i>Rescale a vector with which minimum value 0 and maximum value 1</i>
---------	--

---

**Description**

Rescale a vector with which minimum value 0 and maximum value 1

**Usage**

```
myscale(x)
```

**Arguments**

x	A numeric vector
---	------------------

---

myscale2	<i>Rescale a vector with which minimum value 0 and maximum value 1</i>
----------	--

---

**Description**

Rescale a vector with which minimum value 0 and maximum value 1

**Usage**

```
myscale2(x, minx = 0, maxx = 1)
```

**Arguments**

x	A numeric vector
minx	The minimum value
maxx	The maximum value

---

newColName	<i>find new column name</i>
------------	-----------------------------

---

**Description**

find new column name

**Usage**

```
newColName(df)
```

**Arguments**

df	a data.frame
----	--------------

---

num2cut	<i>Computing breaks for make a histogram of a continuous variable</i>
---------	---

---

**Description**

Computing breaks for make a histogram of a continuous variable

**Usage**

```
num2cut(x)
```

**Arguments**

x                    A continuous variables

**Value**

A list contains a factor and a numeric vector

---

num2factorDf	<i>Make numeric column of a data.frame to factor</i>
--------------	--

---

**Description**

Make numeric column of a data.frame to factor

**Usage**

```
num2factorDf(data, colnames, maxfactorno = 6)
```

**Arguments**

data                a data.frame  
colnames            Column names to be converted  
maxfactorno        maximum unique value of column

p2chr *Convert p values to character*

---

**Description**

Convert p values to character

**Usage**

```
p2chr(x)
```

**Arguments**

x                    A vector

---

palette2colors *Extract colors from a palette*

---

**Description**

Extract colors from a palette

**Usage**

```
palette2colors(name, reverse = FALSE)
```

**Arguments**

name                A palette name from the RColorBrewer package  
reverse             if true, reverse colors

---

pastecolon *Paste character vectors separated by colon*

---

**Description**

Paste character vectors separated by colon

**Usage**

```
pastecolon(...)
```

**Arguments**

...                 Arguments passed on to paste()

---

pastecomma	<i>Add comma to vectors</i>
------------	-----------------------------

---

**Description**

Add comma to vectors

**Usage**

```
pastecomma(...)
```

**Arguments**

...           Argument passed to paste0

---

rescale_df	<i>Rescale all numeric variables of a data.frame except grouping variable</i>
------------	---

---

**Description**

Rescale all numeric variables of a data.frame except grouping variable

**Usage**

```
rescale_df(data, groupvar = NULL)
```

**Arguments**

data           A data.frame  
groupvar       A column name used as a grouping variable

**Value**

A rescaled data.frame

---

rose	<i>Rose sales among 7 groups in a year</i>
------	--

---

**Description**

A phony dataset representing rose sales

**Usage**

```
rose
```

**Format**

An object of class `data.frame` with 84 rows and 3 columns.

**Details**

@format A `data.frame` with 84 rows and 3 columns

**group** group A to G

**Month** Month 1 to 12

**value** Rose sales amount

---

subcolors	<i>Make a subcolors according to the mainCol</i>
-----------	--

---

**Description**

Make a subcolors according to the mainCol

**Usage**

```
subcolors(.dta, main, mainCol)
```

**Arguments**

.dta	A <code>data.frame</code>
main	A character string of column name used as a main variable
mainCol	A main color



---

summarySE	<i>Summarize a continuous variable by groups with mean, sd and SE</i>
-----------	---

---

**Description**

Summarize a continuous variable by groups with mean, sd and SE

**Usage**

```
summarySE(
  data = NULL,
  measurevar,
  groupvars = NULL,
  conf.interval = 0.95,
  na.rm = TRUE,
  .drop = TRUE
)
```

**Arguments**

data	A data.frame
measurevar	A name of variable to measure a mean and sd
groupvars	Name(s) of variable used as a grouping variables
conf.interval	confidence interval
na.rm	A logical value indicating whether or not remove NA values
.drop	should combinations of variables that do not appear in the input data be preserved (FALSE) or dropped (TRUE, default)

**Value**

A data.frame summarized a continuous variable by groups with mean, sd and SE

---

taco	<i>Taco ratings by age group</i>
------	----------------------------------

---

**Description**

Taco ratings by ShellType, AgeGroup and Filling source: [Communicating experiment results with R](#)

**Usage**

```
taco
```

**Format**

An object of class `data.frame` with 136 rows and 4 columns.

**Details**

@format A `data.frame` with 136 rows and 4 columns

**ShellType** Hard or Soft

**Fillings** Fillings of taco

**AgeGroup** AgeGroup One of the `c("<13","13-20","21-39","40+",)`

**Rating** A numeric. Rating of taco

---

theme_clean	<i>Clean theme for PieDonut plot</i>
-------------	--------------------------------------

---

**Description**

Clean theme for PieDonut plot

**Usage**

```
theme_clean(base_size = 12)
```

**Arguments**

`base_size` An integer, default 12.

---

theme_clean2	<i>Clean theme for ggCor</i>
--------------	------------------------------

---

**Description**

Clean theme for ggCor

**Usage**

```
theme_clean2(base_size = 12, xangle = 45, yangle = 0)
```

**Arguments**

<code>base_size</code>	base font size
<code>xangle</code>	x-axis text angle
<code>yangle</code>	y-axis text angle

---

unselectNumeric	<i>Unselect numeric column of a data.frame</i>
-----------------	--

---

**Description**

Unselect numeric column of a data.frame

**Usage**

```
unselectNumeric(data, colnames, maxfactorno = 6)
```

**Arguments**

data	a data.frame
colnames	Column names to be converted
maxfactorno	maximum unique value of column

# Index

- \* **datasets**
  - browsers, [3](#)
  - rose, [40](#)
  - taco, [41](#)
- addLabelDf, [3](#)
- browsers, [3](#)
- coord\_radar, [4](#)
- getMapping, [4](#)
- ggAncova, [5](#)
- ggArea, [6](#)
- ggBar, [7](#)
- ggBoxplot, [8](#)
- ggCatepillar, [9](#)
- ggChoropleth, [10](#)
- ggCLE, [11](#)
- ggCor, [12](#)
- ggDensity, [14](#)
- ggDonut, [15](#)
- ggDot, [16](#)
- ggEffect, [18](#)
- ggErrorBar, [19](#)
- ggHeatmap, [20](#)
- ggHSD, [22](#)
- ggPair, [23](#)
- ggPie, [24](#)
- ggPieDonut, [25](#)
- ggPoints, [26](#)
- ggPredict, [28](#)
- ggRadar, [29](#)
- ggRose, [30](#)
- ggSpine, [31](#)
- ggViolin, [33](#)
- makeEq, [34](#)
- model2df, [35](#)
- myscale, [35](#)
- myscale2, [36](#)
- newColName, [36](#)
- num2cut, [37](#)
- num2factorDf, [37](#)
- p2chr, [38](#)
- palette2colors, [38](#)
- pastecolon, [38](#)
- pastecomma, [39](#)
- rescale\_df, [39](#)
- rose, [40](#)
- subcolors, [40](#)
- summarySE, [41](#)
- taco, [41](#)
- theme\_clean, [42](#)
- theme\_clean2, [42](#)
- unselectNumeric, [43](#)