

Package ‘xkcd’

July 2, 2014

Type Package

Title Plotting ggplot2 graphics in a XKCD style

Version 0.0.3

Date 2014-03-05

Author Emilio Torres Manzanera

Maintainer Emilio Torres Manzanera <torres@uniovi.es>

Description This package allows the representation of ggplot2 graphs using the XKCD style.

License GPL-3

Depends Hmisc, ggplot2

Suggests methods, sysfonts, showtext

URL <http://xkcd.r-forge.r-project.org/>

Repository CRAN

Repository/R-Forge/Project xkcd

Repository/R-Forge/Revision 35

Repository/R-Forge/DateTimeStamp 2014-03-05 06:08:30

Date/Publication 2014-03-05 16:46:13

NeedsCompilation no

R topics documented:

xkcd-package	2
createdefaultmappinganddata	3
doforeachrow	4
mappingjoin	5
pointscircumference	6
pointssegment	7
theme_xkcd	8
xkcdaxis	9
xkcdline	10
xkcdman	11
xkcdrect	13
Index	15

xkcd-package	<i>Plot ggplot graphics in a XKCD style</i>
--------------	---

Description

XKCD is a webcomic of romance, sarcasm, math, and language created by Randall Munroe. This package tries to give a satisfactory answer to the question How can we make xkcd style graphs in R?. The package provides a set of ggplot2 functions for plotting data in a XKCD style.

Details

Package: xkcd
 Type: Package
 Version: 0.0.1
 Date: 2013-04-27
 License: GPL3

This package introduces three main functions: xkcdman, xkcdline and xkcdrect. The xkcdman function creates a XKCD man and the other one plot pseudo handwritten objects. These functions are derived from the geom_path of the package ggplot2. All the functionalities of the ggplot system can be used. See the vignettes for detailed examples.

Author(s)

Emilio Torres Manzanera
 Maintainer: Emilio Torres Manzanera <torres@uniovi.es>

References

Hadley Wickham 2012. ggplot2 <http://ggplot2.org/>

Randall Munroe. A webcomic of romance, sarcasm, math, and language <http://xkcd.com/>
Various Authors 2012. How can we make xkcd style graphs in R? <http://stackoverflow.com/questions/12675147/how-can-we-make-xkcd-style-graphs-in-r>

See Also

[geom_path](#)

Examples

```
p <- ggplot() + geom_point(aes(mpg, wt), data=mtcars) + theme_xkcd()
```

createdefaultmappinganddata

Creates a canonical mapping and its associated data base

Description

This function transforms the mapping and the data base to get a canonical mapping where the aesthetic variables are equal to the names of the data base.

Usage

```
createdefaultmappinganddata(mapping, data, mandatoryarguments=c("x", "y"))
```

Arguments

mapping	Set of aesthetics.
data	Default data frame. It is expected that all layers are included in the data set.
mandatoryarguments	Arguments that must appear in the mapping

Details

It creates an aesthetic layer with the default names. If it is necessary, it changes the names of the data base.

Value

A list with a mapping of aesthetics and a data frame.

Note

Internal use.

Author(s)

Emilio Torres Manzanera

See Also[aes_string](#)**Examples**

```

data <- data.frame(x1=1,y1=2)
mapping <- aes(x = x1 + y1, y = y1) # default names: x and y
mapping
## List of 2
## $ x: language x1 + y1
## $ y: symbol y1

createdefaultmappinganddata(mapping,data)
## $mapping
## List of 2
## $ x: symbol x
## $ y: symbol y

## $data
##  x y x1 y1
## 1 3 2  1  2

```

`doforeachrow`*Applies a function to each row of a data frame*

Description

This function applies a function to each row of a data frame. All variables of the data frame and the optional arguments are passed as arguments to the function.

Usage

```
doforeachrow(data, fun, doitalsoforoptargs, ...)
```

Arguments

<code>data</code>	A data frame.
<code>fun</code>	A non-empty character string naming the function to be called.
<code>doitalsoforoptargs</code>	Logical. See Details.
<code>...</code>	Other arguments.

Details

It applies a function to each row of the data frame. The variables of the data frame and the optional arguments are passed as arguments. If there are common names among the data frame and the optional arguments, those values of the data frame are used.

If `doitalsoforoptargs` is `TRUE`, then try to use the row of the optional arguments, if it is possible (only when the length of the optional argument is equal to the number of rows of the data base). Otherwise, (`doitalsoforoptargs = FALSE` or the lengths are different), the whole optional arguments are passed to the function.

Value

A list for each row with the values returned by the function.

Note

Generic function.

Examples

```
fun <- "seq"
data <- data.frame(from=c(1,10), to=c(2,11))
doforeachrow(data, fun, TRUE)
## [[1]]
## [1] 1 2

## [[2]]
## [1] 10 11

doforeachrow(data, fun, TRUE, length.out=c(4))
## [[1]]
## [1] 1.000000 1.333333 1.666667 2.000000

## [[2]]
## [1] 10.00000 10.33333 10.66667 11.00000

doforeachrow(data, fun, TRUE, length.out=c(3,5))
## [[1]]
## [1] 1.0 1.5 2.0

## [[2]]
## [1] 10.00 10.25 10.50 10.75 11.00
```

mappingjoin

Joins two sets of aesthetics

Description

This function joins two sets of aesthetics.

Arguments

x	A numeric value. Coordinate x of the center of the circumference.
y	A numeric value. Coordinate y of the center of the circumference.
diameter	A numeric value. Diameter fo the circumference.
ratioxy	A numeric value. Ratio x to y.
npoints	Number of points (including the limits).
alpha	Numeric. Angle to begin to calculate the circumferece.

Details

It calculates a circumference in a XKCD style.

By default, the data are smoothed using a Bezier curve. It gives 60 points.

Value

A data frame with the points of the circumference.

Examples

```
plot(pointscircunference(x=0, y=0), ylim=c(-1,1), xlim=c(-1,1))
plot(pointscircunference(x=0, y=0), ylim=c(-1,1), xlim=c(-2,2))
plot(pointscircunference(x=0, y=0, ratioxy=2), ylim=c(-1,1), xlim=c(-2,2))
```

pointssegment	<i>Interpolates between two points</i>
---------------	--

Description

This function calculates intermediate points between two points. The intermidate points can be jittered and smoothed with a Bezier curve.

Usage

```
pointssegment(xbegin, ybegin, xend, yend,
              npoints = 10, xjitteramount= 0, yjitteramount=0, bezier = TRUE)
```

Arguments

xbegin	A numeric value. Coordinate x of the point from which to interpolate.
ybegin	A numeric value. Coordinate y of the point from which to interpolate.
xend	A numeric value. Coordinate x of the point to which to interpolate.
yend	A numeric value. Coordinate y of the point to which to interpolate.
npoints	Number of points (including the limits).
xjitteramount, yjitteramount	Numeric. Amount of jitter.
bezier	Logical. Use or not the Bezier curves to smooth the jittered data.

Details

It calculates the intermediate points between two points. If there are no jitter amounts, then there are no interpolations.

By default, the data are smoothed using a Bezier curve. It gives 30 points.

Value

A data frame with the interpolated values.

See Also

[jitter](#)

Examples

```
plot(pointssegment(xbegin=0, ybegin=0, xend=10, yend=10))
plot(pointssegment(xbegin=0, ybegin=0, xend=10, yend=10,
  xjitteramount=2,yjitteramount=2))
plot(pointssegment(xbegin=0, ybegin=0, xend=10, yend=10,
  xjitteramount=2,yjitteramount=2,bezier=FALSE))
```

theme_xkcd

Creates a XKCD theme

Description

This function creates a XKCD theme for ggplot graphics.

Usage

```
theme_xkcd()
```

Arguments

None

Details

It uses the XKCD fonts. You must install this type of fonts. See Note.

Value

A theme with white background and fonts of the XKCD family.

Note

To install the fonts (See the vignette):

```
download.file("http://simonsoftware.se/other/xkcd.ttf", dest="xkcd.ttf") system("mkdir ~/.fonts") sys-
tem("cp xkcd.ttf -t ~/.fonts") ##library(sysfonts) font.add("xkcd", regular = "xkcd.ttf")
```


References

fibosworld 2013. Change fonts in ggplot2, and create xkcd style graphs <http://fibosworld.wordpress.com/2013/02/17/change-fonts-in-ggplot2-and-create-xkcd-style-graphs/>

Various Authors 2012. How can we make xkcd style graphs in R? <http://stackoverflow.com/questions/12675147/how-can-we-make-xkcd-style-graphs-in-r>

<http://simonsoftware.se/other/xkcd.ttf>

Examples

```
## Not run:  
p <- ggplot() + geom_point(aes(mpg, wt), data=mtcars) + theme_xkcd()  
p  
  
## End(Not run)
```

xkcdaxis

Plots the axis

Description

This function plots the axis.

Usage

```
xkcdaxis(xrange, yrange, ...)
```

Arguments

xrange	The range of X axe.
yrange	The range of Y axe.
...	Other arguments.

Details

It plots the axis of the graph.

Value

A layer with the axis.

Examples

```
## Not run:
xrange <- range(mtcars$mpg)
yrange <- range(mtcars$wt)
p <- ggplot() + geom_point(aes(mpg, wt), data=mtcars) +
  xkcdaxis(xrange,yrange)
p

## End(Not run)
```

xkcdline

Draws handwritten lines

Description

This function draws ggplot2 segments or circles in a handwritten style.

Usage

```
xkcdline(mapping, data, typexkcdline = "segment", mask = TRUE, ...)
```

Arguments

mapping	Mapping between variables and aesthetics generated by aes. See Details.
data	Dataset used in this layer.
typexkcdline	A string value. If it is segment, it draws a segment. If it is circumference, it plots a circumference.
mask	Logical. If it is true, it erases the pictures that are under the line.
...	Optional arguments.

Details

It draws a segment or a circumference in a XKCD style. If it is a segment, the following aesthetics are required:

1. xbegin: x position of the point from.
2. ybegin: y position of the point from.
3. xend: x position of the point to.
4. yend: y position of the point to.

If it is a circumference, the following aesthetics are required:

1. x: x position of the center.
2. y: y position of the center.
3. diameter: diameter of the circumference.

Additionally, you can use the aesthetics of geom_path, pointssegment and pointscircumference.

Value

A ggplot layer.

See Also

[aes](#), [geom_path](#), [pointssegment](#), [pointscircumference](#)

Examples

```
data <- data.frame(x1=c(1,2), y1=c(10,20), xend=c(2.5,0.5),
  yend=c(20,10), model=c("low","high"))
```

```
ggplot() + xkcdline(mapping=aes(xbegin=x1 +y1, ybegin=y1, xend =xend, yend= yend,
  color = model), data=data)
```

```
ggplot() + xkcdline(mapping=aes(xbegin=x1 +y1, ybegin=y1, xend =xend, yend= yend,
  color = model), data=data) + facet_grid(. ~ model)
```

```
ggplot() + xkcdline(mapping=aes(x=x1 +y1, y=y1, diameter =xend), data=data, type="circumference")
```

xkcdman

Draws a handwritten man

Description

This function draws ggplot2 man in a handwritten style.

Usage

```
xkcdman(mapping, data, ...)
```

Arguments

mapping	Mapping between variables and aesthetics generated by aes. See Details.
data	Dataset used in this layer.
...	Optional arguments.

Details

It draws a man in a XKCD style. The following aesthetics are required:

1. x: x position of the center of the head.
2. y: y position of the center of the head.
3. scale: scale of the man. It is the size of the man (in units of the Y axis).
4. ratioxy: Ratio x to y of the graph (Use ratioxy <- diff(xrange) / diff(yrange))

5. `angleofspine`: angle between the spine and a horizontal line that passes by the center of the head.
6. `anglerighthumerus`, `anglelefthumerus`: angle between the right/left humerus and a horizontal line that passes by the top of the spine.
7. `anglerightradius`, `angleleftradius`: angle between the right/left radius and a horizontal line that passes by the end of the right/left humerus.
8. `anglerightleg`, `angleleftleg`: angle between the right/left leg and a horizontal line that passes by the end of the spine.
9. `angleofneck`: angle between the begin of spine and a horizontal line that passes by the center of the head.

Angles are in radians.

Additionally, you can use the aesthetics of `geom_path`, and `xkcdline`.

Value

A `ggplot` layer.

Note

See `vignette("xkcd-intro")` for a graphical representation of the angles used in this function.

See Also

[aes](#), [geom_path](#), [xkcdline](#)

Examples

```

datascaled <- data.frame(x=c(-3,3),y=c(-30,30))
p <- ggplot(data=datascaled, aes(x=x,y=y)) + geom_point()
xrange <- range(datascaled$x)
yrange <- range(datascaled$y)
ratioxy <- diff(xrange) / diff(yrange)

mapping <- aes(x=x,
              y=y,
              scale=scale,
              ratioxy=ratioxy,
              angleofspine = angleofspine,
              anglerighthumerus = anglerighthumerus,
              anglelefthumerus = anglelefthumerus,
              anglerightradius = anglerightradius,
              angleleftradius = angleleftradius,
              anglerightleg = anglerightleg,
              angleleftleg = angleleftleg,
              angleofneck = angleofneck,
              color = color )

dataman <- data.frame( x= c(-1,0,1), y=c(-10,0,10),

```

```

scale = c(10,7,5),
ratioxy = ratioxy,
angleofspine = seq(- pi / 2, -pi/2 + pi/8, l=3) ,
anglerighthumerus = -pi/6,
anglelefthumerus = pi + pi/6,
anglerightradius = 0,
angleleftradius = runif(3,- pi/4, pi/4),
angleleftleg = 3*pi/2 + pi / 12 ,
anglerightleg = 3*pi/2 - pi / 12,
angleofneck = runif(3, min = 3 * pi / 2 - pi/10 , max = 3 * pi / 2 + pi/10),
color=c("A","B","C"))

p + xkcdman(mapping,dataman)

```

xkcdrect

Plots rectangles

Description

This function plots rectangles.

Usage

```
xkcdrect(mapping, data,...)
```

Arguments

mapping	Mapping between variables and aesthetics generated by aes. See Details.
data	Dataset used in this layer.
...	Optional arguments.

Details

It plots rectangles. The following aesthetics are required:

1. xmin
2. ymin
3. xmax
4. ymax

Additionally, you can use the aesthetics of `geom_rect`, `geom_path`, and `pointssegment`.

Value

A layer with the rectangles.

See Also

[geom_rect](#)

Examples

```
volunteers <- data.frame(year=c(2007:2011),
                          number=c(56470, 56998,59686, 61783, 64251))
p <- ggplot() + xkcdrect(aes(xmin = year,
                             xmax= year +0.3,
                             ymin=number,
                             ymax = number + 3600),
                        volunteers,
                        fill="red", colour="black")
p
```

Index

*Topic **manip**

- createdefaultmappinganddata, 3
- doforeachrow, 4
- mappingjoin, 5
- theme_xkcd, 8
- xkcdaxis, 9

*Topic **package**

- xkcd-package, 2

aes, 11, 12

aes_string, 4, 6

createdefaultmappinganddata, 3

doforeachrow, 4

geom_path, 3, 11, 12

geom_rect, 13

jitter, 8

mappingjoin, 5

pointscircunference, 6

pointscircunference, 11

pointscircunference

(pointscircunference), 6

pointssegment, 7, 11

theme_xkcd, 8

xkcd (xkcd-package), 2

xkcd-package, 2

xkcdaxis, 9

xkcdline, 10, 12

xkcdman, 11

xkcdrect, 13