Package ‘pathdiagram’

October 2, 2019

Type Package
Title Basic Functions for Drawing Path Diagrams
Version 0.1.9.1
Date 2013-07-28
Author Gaston Sanchez
Maintainer Gaston Sanchez <gaston.stat@gmail.com>
Description Implementation of simple functions to draw basic path diagrams just for visualization purposes.

URL http://www.gastonsanchez.com
Depends R (>= 3.0), shape
Suggests knitr
VignetteBuilder knitr
License GPL-3
Collate 'arrow.R' 'draw.R' 'latent.R' 'manifest.R' 'wall.R'
NeedsCompilation no
Repository CRAN
Date/Publication 2019-10-02 15:27:59 UTC

R topics documented:

  arrow ................................................................. 2
  draw ................................................................. 3
  latent ............................................................... 4
  manifest ........................................................... 6
  wall ................................................................. 7

Index 9
**Description**

Use this function to draw connecting arrows between manifest and latent variables.

**Usage**

```r
arrow(from, to, start = "east", end = "west",
      length = 0.1, angle = 10, code = 2, col = "#d2def1",
      lwd = 3, ...)```

**Arguments**

- `from` An object of either class "manifest" or "latent". This object is the origin of the arrow.
- `to` An object of either class "manifest" or "latent". This object is the destination of the arrow.
- `start` Character string to specify the starting direction of the arrow. Options are "north", "south", "east", "west".
- `end` Character string to specify the ending direction of the arrow. Options are "north", "south", "east", "west".
- `length` length of the edges of the arrow head (in inches).
- `angle` angle from the shaft of the arrow to the edge of the arrow head.
- `code` integer code, determining kind of arrows to be drawn.
- `col` color of the arrow.
- `lwd` width of the arrow.
- `...` other arguments passed on to `arrows`.

**Author(s)**

Gaston Sanchez

**See Also**

`manifest`, `latent`, `draw`

**Examples**

```r
## Not run:
# latent variables
attack = latent("ATTACK", x=0.35, y=0.7, rx=0.08, ry=0.06)
defense = latent("DEFENSE", x=0.35, y=0.3, rx=0.08, ry=0.06)
success = latent("SUCCESS", x=0.65, y=0.5, rx=0.08, ry=0.06)```
# open wall
wall()

# draw latent variables
draw(attack)
draw(defense)
draw(success)

# add arrows
arrow(from=attack, to=succes, start="east", end="west")
arrow(from=defense, to=succes, start="east", end="west")

## End(Not run)

draw

Draw manifest and latent variables

Description

Use this function to draw either manifest or latent variables on a plot.

Usage

draw(obj)

Arguments

obj An object of either class "manifest" or "latent"

Author(s)

Gaston Sanchez

See Also

manifest, latent

Examples

## Not run:
# manifest variables
ingredients = list(
  eggs = manifest("eggs", x=0.3, y=0.7, width=0.10, height=0.08),
  milk = manifest("milk", x=0.3, y=0.6, width=0.10, height=0.08),
  flour = manifest("flour", x=0.3, y=0.5, width=0.10, height=0.08),
  sugar = manifest("sugar", x=0.3, y=0.4, width=0.10, height=0.08),
  butter = manifest("butter", x=0.3, y=0.3, width=0.10, height=0.08))

# latent variables
pancakes = latent("PANCAKES", x=0.6, y=0.6, rx=0.09, ry=0.07)
waffles = latent("WAFFLES", x=0.6, y=0.4, rx=0.09, ry=0.07)

# open wall
wall()

title("Toy Path Diagram", col.main="gray20")
# draw manifest variables
for (i in 1:length(ingredients)) {
  draw(ingredients[[i]])
}

# draw latent variables
draw(pancakes)
draw(waffles)
# draw arrows
for (i in 1:length(ingredients)) {
  arrow(from=ingredients[[i]], to=pancakes, start="east", end="west")
  arrow(from=ingredients[[i]], to=waffles, start="east", end="west")
}

## End(Not run)

---

**latent**

*Set specifications of a latent variable*

**Description**

Use this function to specify the graphic characteristics of a latent variable. The specifications will be used by the function `draw` to plot latent variables (in a path diagram).

**Usage**

```r
latent(label = "latent", x = 0.5, y = 0.5, rx = 0.05,
ry = 0.05, border = "white", lwd = 2, fill = "#5f8bd7",
col = "white", cex = 1, vfont = NULL, font = 2,
family = "sans")
```

**Arguments**

- `label` A character string with the label to be displayed.
- `x` x-axis coordinate for center of ellipse.
- `y` y-axis coordinate for center of ellipse.
- `rx` long radius of ellipse.
- `ry` short radius of ellipse.
- `border` color of the border.
- `lwd` width of border line.
- `fill` color to fill the ellipse.
**latent**

- `col` color of the label.
- `cex` numeric character expansion of the label.
- `vfont` font family of the label.
- `font` An integer specifying which font to use for the label. See `par` family
- `family` The name of a font family for drawing text. Standard values are "serif", "sans" and "mono".

**Details**

Latent variables are drawn as ellipses using the function `plotellipse`.

**Value**

An object of class "latent", which is a list with the specified parameters to draw latent variables.

**Author(s)**

Gaston Sanchez

**See Also**

`manifest`, `draw`

**Examples**

```r
## Not run:
# latent variables
attack = latent("ATTACK", x=0.35, y=0.7, rx=0.08, ry=0.06)
defense = latent("DEFENSE", x=0.35, y=0.3, rx=0.08, ry=0.06)
success = latent("SUCCESS", x=0.65, y=0.5, rx=0.08, ry=0.06)

# own wall
wall()
title("Drawing three latent variables", col.main="gray20")

draw(attack)
draw(defense)
draw(success)

## End(Not run)
```
manifest  

Set specifications of a manifest variable

Description

Use this function to specify the graphic characteristics of a manifest variable. The specifications will be used by the function draw to plot manifest variables (in a path diagram).

Usage

manifest(label = "manifest", x = 0.5, y = 0.5,  
width = NULL, height = 0.1, border = "white",  
fill = "#9dbafa", lwd = 1, col = "gray20", cex = 1,  
vfont = NULL, font = 1, family = "sans")

Arguments

label  A character string with the label to be displayed.
x  x-axis coordinate for center of rectangle.
y  y-axis coordinate for center of rectangle.
width  width of the rectangle.
height  height of the rectangle.
border  color of the border.
fill  color to fill the rectangle.
lwd  width of the border.
col  color of the label.
cex  numeric character expansion of the label.
vfont  font family of the label.
font  An integer specifying which font to use for the label. See par
family  The name of a font family for drawing text. Standard values are "serif", "sans" and "mono".

Details

Manifest variables are drawn as rectangles.

Value

An object of class "manifest", which is a list with the specified parameters to draw manifest variables.

Author(s)

Gaston Sanchez
## Not run:

```r
# manifest variables
eggs = manifest("eggs", x=0.3, y=0.7, width=NULL, height=0.08)
milk = manifest("milk", x=0.4, y=0.6, width=NULL, height=0.08)
flour = manifest("flour", x=0.5, y=0.5, width=NULL, height=0.08)
sugar = manifest("sugar", x=0.6, y=0.4, width=NULL, height=0.08)
butter = manifest("butter", x=0.7, y=0.3, width=NULL, height=0.08)

# open wall
wall()
wall()
title("Five manifest variables", col.main="gray20")

# draw manifest variables
draw(eggs)
draw(milk)
draw(flour)
draw(sugar)
draw(butter)
```

## End(Not run)

---

### wall

**Open a new frame for a path diagram**

#### Description

Use this function to open a white canvas to start drawing a path diagram. By default, wall opens a new plot window from 0 to 1 in both axes.

#### Usage

```r
wall(xlim = c(0, 1), ylim = c(0, 1), xpd = TRUE, ...)
```

#### Arguments

- **xlim**: Numeric vector of length 2 giving the x coordinate range. Default `xlim = c(0,1)`.
- **ylim**: Numeric vector of length 2 giving the y coordinate range. Default `ylim = c(0,1)`.
- **xpd**: Logical value to indicate if all plotting is clipped to the figure region. The default is `c(1,1,1,1)`.
- **...**: other graphical arguments passed on to `plot.window`.

#### Details

wall calls `plot.new()` and `plot.window()` to open a new plot frame.
Author(s)

Gaston Sanchez

See Also

manifest, latent, draw

Examples

```r
## Not run:
# latent variables
attack = latent("ATTACK", x=0.35, y=0.7, rx=0.08, ry=0.06)
defense = latent("DEFENSE", x=0.35, y=0.3, rx=0.08, ry=0.06)
success = latent("SUCCESS", x=0.65, y=0.5, rx=0.08, ry=0.06)

# open diagram
wall()

# draw latent variables
draw(attack)
draw(defense)
draw(success)

# add arrows
arrow(from=attack, to=success, start="east", end="west")
arrow(from=defense, to=success, start="east", end="west")

## End(Not run)
```
Index

arrow, 2

draw, 2, 3, 5, 7, 8

latent, 2, 3, 4, 7, 8

manifest, 2, 3, 5, 6, 8

par, 5, 6

plot.window, 7

plotellipse, 5

wall, 7