Package ‘eeptools’

February 19, 2015

Type Package
Title Convenience functions for education data
Version 0.3.1
Date 2013-12-10
Description Collection of convenience functions to make working with administrative records easier and more consistent. Includes functions to clean strings, identify cutpoints, and quickly combine shapefiles and dataframes for plotting. Includes four alternative themes for ggplot2 as well as a wrapper for exporting graphics for inclusion in MS Office products. Also includes three example datasets of administrative education records for learning how to process records with errors.
License GPL-3
Depends R (>= 2.15.1), ggplot2, MASS
Imports stringr, maptools, grid, arm, data.table, memisc
Suggests reshape, sp, plyr, mapproj, testthat
Author Jason P. Becker [ctb], Jared E. Knowles [aut, cre]
Maintainer Jared E. Knowles <jknowles@gmail.com>
NeedsCompilation no
Repository CRAN
Date/Publication 2014-02-23 21:07:50

R topics documented:

eeptools-package ........................................................................... 2
Evalution of educational policy tools

Description
Make common tasks for educational evaluation easier to do!

Details

Package: eeptools
Type: Package
Version: 0.1
Date: 2012-07-21
License: GPL-3

This package has a number of useful shortcuts for common tasks. It includes some themes for ggplot2 plots, processing arbitrary text files of data, calculating student characteristics, and finding
age_calc

thresholds within vectors. Future development work will include methods for tuning and evaluating early warning system models.

Note

This package is still in beta and function names may change in the next release.

Author(s)

Jared E Knowles
Jared E. Knowles <jknowles@gmail.com>

References

This package is developed to support the work at state and local education agencies for evaluating education policies.

Examples

gender<-c("M","M","M","F","F","F")
statamode(gender)
statamode(gender[1:5])

missing_data<-c(NA,NA,NA)
max_mis(missing_data)

destring(gender)
gender<-factor(gender)
defac(gender)

age_calc

Function to calculate age from date of birth.

Description

This function calculates age in days, months, or years from a date of birth to another arbitrary date. This returns a numeric vector in the specified units.

Usage

age_calc(dob, enddate = Sys.Date(), units = "months", precise = TRUE)
autoplot.lm

Arguments

- **dob**: a vector of class Date representing the date of birth/start date
- **enddate**: a vector of class Date representing the when the observation’s age is of interest, defaults to current date.
- **units**: a string that indicates units of age
- **precise**: logical indicating whether or not to calculate with leap year and leap second precision

Value

- a vector of class numeric

Author(s)

Jason P. Becker

Source

This function was developed in part from this response on the R-Help mailing list. https://stat.ethz.ch/pipermail/r-help/2009-November/217649.html

See Also

See also **difftime** which this function uses and mimics some functionality but at higher unit levels.

Examples

```r
a <- as.Date(seq(as.POSIXct('1987-05-29 018:07:00'), len=26, by="21 day"))
b <- as.Date(seq(as.POSIXct('2002-05-29 018:07:00'), len=26, by="21 day"))

age <- age_calc(a, units='years')
age
age <- age_calc(a, units='months')
age
age <- age_calc(a, as.Date('2005-09-01'))
age
```

Description

This uses ggplot2 to replicate the plot functionality for lm in ggplot2 and allow themes.
autoplot.lm

Usage

```r
## S3 method for class 'lm'
autoplot(object, which = c(1:6), mfrow = c(3, 2), ...)
```

Arguments

- **object**: a linear model object
- **which**: which of the tests do we want to display output from
- **mfrow**: Describes the layout of the resulting function in the plot frames
- **...**: functions to pass through

Value

A ggplot2 object that mimics the functionality of a plot of linear model.

Author(s)

Jared E. Knowles

References


See Also

- `plot.lm` which this function mimics

Examples

```r
# Univariate
a <- runif(1000)
b <- 7 * a + rnorm(1)
mymod <- lm(b ~ a)
autoplot(mymod)

# Multivariate

data(mpg)
mymod <- lm(cty ~ disp + cyl + drv, data = mpg)
autoplot(mymod)
```
cleanTex  

*Remove Unwanted LaTeX files after building document*

**Description**

Convenience function for cleaning up your directory after running pdflatex

**Usage**

`cleanTex(mydoc, keepPDF = NULL, keepRnw = NULL, keepRproj = NULL)`

**Arguments**

- **mydoc** a filename for your .Rnw file
- **keepPDF** Logical. Should function preserve PDF files with filename `mydoc`. Default is TRUE.
- **keepRnw** Logical. Should function save Rnw files with filename `mydoc`. Default is TRUE
- **keepRproj** Logical. Should function save .Rproj files with filename `mydoc`. Default is TRUE

**Value**

Nothing. All files except the .tex, .pdf and .Rnw are removed from your directory.

---

cutoff  

*function to calculate thresholds in a vector*

**Description**

This function tells us how far we have to go before reaching a cutoff in a variable by sorting the vector, then finding how far to go. Note that the cutoff is expressed in percentage terms (Fixed cumulative sum)

**Usage**

`cutoff(x, cutoff)`

**Arguments**

- **x** a numeric vector, missing values are allowed
- **cutoff** cutoff a user defined numeric value to stop the cutoff specified as a proportion 0 to 1
Details
Calculates the distance through a numeric vector before a certain proportion is reached by sorting
the vector and calculating the cumulative proportion of each element.

Value
An integer for the minimum number of elements necessary to reach cutoff.

Author(s)
Jared E. Knowles

See Also
cumsum which this function uses

Examples

```r
# for vector
a <- rnorm(100, mean = 6, sd = 1)
cutoff(a, .7) # return minimum number of elements to account 70 percent of total

## The function is currently defined as
function (x, cutoff)
{
  x <- x[order(-x)]
xb <- cumsum(x)
xc <- xb/sum(x, na.rm = T)
  length(xc[xc < cutoff])
}
```

decomma

<table>
<thead>
<tr>
<th>decomma</th>
<th>Remove commas from numeric fields and return them as numerics</th>
</tr>
</thead>
</table>

Description
A shortcut function to strip commas out of numeric fields imported from other software and convert
them into numeric vectors that can be operated on. This assumes decimal point as opposed to
decimal comma notation.

Usage
decomma(x)

Arguments

| x | A character vector containing numbers with commas that should be coerced into being numeric. |
defac

Details
This function assumes decimal point notation for numbers. For more information, see http://en.wikipedia.org/wiki/Decimal_mark#Countries_using_Arabic_numerals_with_decimal_point.

Value
A numeric vector.

Author(s)
Jared E. Knowles

Examples
input <- c("10,243", "11,212", "7,011", "5443", "500")
output <- decomma(input)
is.numeric(output)

defac
convert a factor to a character string safely

Description
This is a shortcut function to convert a factor to a character variable without having to type as.character().

Usage
defac(x)

Arguments
x
a factor to be turned into a character

Value
a character string of the vector input

Author(s)
Jared E. Knowles

See Also
factor, levels to understand the R implementation of factors.


**destring**

*a function to convert numeric factors into numeric class objects*

---

**Examples**

```r
a <- as.factor(LETTERS)
summary(a)
b <- defac(a)
class(b)
```

---

**Description**

This function allows you to convert directly from a numeric factor to the numeric class in R and strip away the underlying level index of a factor. This makes it safer to convert from factors to numeric characters directly without accidentally misassigning numbers.

**Usage**

```r
destring(x)
```

**Arguments**

- `x` a factor with numeric levels

**Details**

This function should only be used on factors where all levels are valid numbers that can be coerced into a numeric class.

**Value**

a vector of class numeric

**warning**

This will force all levels to be converted to characters and then to numeric objects. Leading zeroes will be stripped off and commas will cause errors.

**Author(s)**

Jared E. Knowles

**See Also**

- `character`
dropbox_source

Examples

```r
a <- ordered(c(1, 3, '09', 7, 5))
b <- destring(a)
class(b)
b
a
```

Description

Allows the user to source R scripts from a public Dropbox folder using the URL provided by the Dropbox shared URL tool. Parses the URL and then sources the script in one easy line.

Usage

```r
dropbox_source(myurl)
```

Arguments

- **myurl**  
  A character string for the full URL of the Dropbox share

Details

Must be the full URL including https://

Value

None. The script is sourced.

Note

This function works with Dropbox as of the release date. If functionality breaks, please e-mail the package maintainer to push an update.

Author(s)

Jared Knowles

See Also

source
**gemansim**

**Examples**

```r
## Not run:
dropbox_source("https://dropbox.com/public/somekey/cooldata.csv")
## End(Not run)
```

---

**gelmansim**  
*Generate prediction intervals for model functions*

**Description**

Generate prediction intervals from R models following Gelman and Hill

**Usage**

```r
gelmansim(mod, newdata, nsims, na.omit = TRUE)
```

**Arguments**

- **mod**  
  Name of a model object such as `lm`, `glm`, or `merMod`
- **newdata**  
  Sets of new data to generate predictions for
- **nsims**  
  Number of simulations per case
- **na.omit**  
  Logical indicating whether to remove NAs from `newdata`

**Details**

Currently gelmansim does not work for `lm` objects because of the way `sim` in the `arm` package handles variable names for these objects. It is recommended users use `glm` in these cases.

**Value**

A dataframe with `newdata` and prediction intervals

**References**

Examples

```r
# Examples of "sim"
set.seed(1)
J <- 15
n <- J*(J+1)/2
group <- rep(1:J, 1:J)
mu.a <- 5
sigma.a <- 2
a <- rnorm(J, mu.a, sigma.a)
b <- -3
x <- rnorm(n, 2, 1)
sigma.y <- 6
y <- rnorm(n, a[group] + b*x, sigma.y)
u <- runif(J, 0, 3)
y123.dat <- cbind(y, x, group)
# Linear regression
x1 <- y123.dat[,2]
y1 <- y123.dat[,1]
M1 <- glm(y1 ~ x1)
cases <- data.frame(x1 = seq(-2, 2, by=0.1))
sim.results <- gelmansim(M1, newdata=cases, nsims=200, na.omit=TRUE)
## Not run:

dat <- as.data.frame(y123.dat)
M2 <- glm(y1 ~ x1 + group, data=dat)
cases <- expand.grid(x1 = seq(-2, 2, by=0.1),
                     group=seq(1, 14, by=2))
sim.results <- gelmansim(M2, newdata=cases, nsims=200, na.omit=TRUE)

## End(Not run)
```

---

**ggmapmerge**

Fortify a SpatialPolygonsDataFrame

**Description**

Convenience function for fortifying SpatialPolygonsDataFrames for ggplot2 plotting.

**Usage**

```r
ggmapmerge(mapobj, xid)
```

**Arguments**

- `mapobj` Name of an S4 SpatialPolygonsDataFrame
- `xid` Name of ID variable in the SpatialPolygonsDataFrame
Details

This function requires maptools to be loaded and gpclibPermit to be TRUE. This is because it depends on the fortify method in ggplot2.

Value

An S3 dataframe suitable for using in a gggplot2 map

Examples

```r
## Not run:
xx <- maptools::readShapePoly(system.file("shapes/sids.shp", package="maptools")[[1]], IDvar="FIPSNO")
plotobj<-ggmapmerge(xx,"FIPS")
## End(Not run)
```

---

**lag_data**

Create a lag

Description

Lag variables by an arbitrary number of periods even if the data is grouped

Usage

`lag_data(df, group, time, periods, values)`

Arguments

- `df` A dataframe with groups, time periods, and a variable to be lagged
- `group` The grouping factor in the dataframe
- `values` The names of the variables to be lagged
- `time` The variable representing time periods
- `periods` A scalar for the number of periods to be lagged in the data

Value

A dataframe with a newly created variable lagged
Examples

test_data <- expand.grid(id = sample(letters, 10),
    time = 1:10)
test_data$value1 <- rnorm(100)
test_data$value2 <- runif(100)
test_data$value3 <- rpois(100, 4)
group <- "id"
time <- "time"
values <- c("value1", "value2")
vars <- c(group, time, values)
periods <- 2
newdat <- lag_data(test_data, group="id", time="time",
    values=c("value1", "value2"), periods=3)

mapmergeCombine an S4 polygon object with a dataframe

Description

Convenience function for merging dataframes and S4 spatial polygon objects.

Usage

mapmerge(mapobj, data, xid, yid)

Arguments

mapobj Name of an S4 SpatialPolygonsDataFrame
data Name of an S3 dataframe
xid Name of ID variable in the SpatialPolygonsDataFrame
yid Name of ID variable in the S3 dataframe

Value

A SpatialPolygonsDataFrame with new variables attached from supplied dataframe

Examples

## Not run:
xx <- maptools::readShapePoly(system.file("shapes/sids.shp", package="maptools")[[1]], IDvar="FIPSNO")
yy<-as(xx,"data.frame")
yy$newvar<-sample(letters,nrow(yy),replace=TRUE)
yy<-subset(yy,select=c("FIPS","newvar"))
newpoly<-mapmerge(xx,yy,xid="FIPS",yid="FIPS")

## End(Not run)
**mapmerge2**

*Combine an S4 polygon object with a dataframe*

**Description**

Convenience function for merging dataframes and S4 spatial polygon objects.

**Usage**

```r
mapmerge2(mapobj, data, xid, yid)
```

**Arguments**

- `mapobj` Name of an S4 SpatialPolygonsDataFrame
- `data` Name of an S3 dataframe
- `xid` Name of ID variable in the SpatialPolygonsDataFrame
- `yid` Name of ID variable in the S3 dataframe

**Value**

A SpatialPolygonsDataFrame with new variables attached from supplied dataframe

**Examples**

```r
## Not run:
xx <- maptools::readShapePoly(system.file("shapes/sids.shp", package="maptools")[[1]], IDvar="FIPSNO")
yy <- as(xx, "data.frame")
yy$newvar <- sample(letters, nrow(yy), replace=TRUE)
newpoly <- mapmerge(xx, yy, xid="FIPS", yid="FIPS")
## End(Not run)
```

---

**max_mis**

*A function to safely take the maximum of a vector that could include only NAs.*

**Description**

When computing the maximum on arbitrary subsets of data, some of which may only have missing values, it may be necessary to take the maximum of a vector of NAs. This replaces the behavior that returns Inf or -Inf and replaces it with simply returning an NA.

**Usage**

```r
max_mis(x)
```
Arguments

x A vector of data that a maximum can be taken of.

Details

This function only returns valid results for vectors with a mix of NA and numeric values.

Value

If any valid numeric values are in the vector it returns the maximum. If all NAs, it returns NA.

Author(s)

Jared E. Knowles

See Also

See also max which this function wraps.

Examples

max(c(7, NA, 3, 2, 0), na.rm=TRUE)
max_mis(c(7, NA, 3, 2, 0))
max(c(NA, NA, NA), na.rm=TRUE)
max_mis(c(NA, NA, NA))

midsch

A dataframe of aggregate test scores for schools in a Midwest state.

Description

This data comes from publicly available aggregated test scores of a large midwestern state. Each row represents scores for school A in grade X and then scores in school A and grade X+1. Additionally, some regression diagnostics and results from a predictive model of test scores in grade X+1 are included.

Usage

data(midsch)
Format

A data frame with 19985 observations on the following 16 variables.

- **district_id**: a numeric vector
- **school_id**: a numeric vector
- **subject**: a factor with levels `math`, `read` representing the subject of the test scores in the row
- **grade**: a numeric vector
- **n1**: a numeric vector for the count of students in the school and grade in t
- **ss1**: a numeric vector for the scale score in t
- **n2**: a numeric vector for the count of students in the school and grade in t+1
- **ss2**: a numeric vector for the mean scale score in t+1
- **predicted**: a numeric vector of the predicted ss2 for this observation
- **residuals**: a numeric vector of residuals from the predicted ss2
- **resid_z**: a numeric vector of standardized residuals
- **resid_t**: a numeric vector of studentized residuals
- **cooks**: a numeric vector of cooks D for the residuals
- **test_year**: a numeric vector representing the year the test was taken
- **tprob**: a numeric vector representing the probability of a residual appearing
- **flagged_t9U**: a numeric vector

Details

These data were fit with a statistical model by a large newspaper to investigate unusual gains in test scores. Fifty separate models were fit representing all unique combinations of grade, year, and subject.

Examples

```r
data(midsch)
```

```r
moves_calc
```

Function to calculate the number of times a student has changed schools.

Description

This function calculates the number of times a student has changed schools, including accounting for gaps in enrollment data. It returns a `data.table` with the student ID and the number of student moves.

Usage

```r
moves_calc(df, enrollby, exitby, gap, sid, schid, enroll_date, exit_date)
```
Arguments

df  a data.frame containing minimally a student identifier, school identifier, enrollment date, and exit date.
enrollby  a date that determines the earliest a student can enroll for the first time without being credited with having moved at least once.
extby  a date that determines the latest a student can exit for the final time without being credited with having moved at least once.
gap  a number that represents the largest gap between an exit date and the next enrollment date that can occur without indicating the student moved to a third school not contained within the data set. The default value is 14.
sid  a character that indicates the name of the student id attribute in df. The default value is sid.
schid  a character that indicates the name of the school id attribute in df. The default value is schid.
enroll_date  a character that indicates the name of the enrollment date attribute in df. The default value is enroll_date.
exit_date  a character that indicates the name of the student id attribute in df. The default value is exit_date.

Value

a data.table

Author(s)

Jason P. Becker

Examples

```r
# Not run:
df <- data.frame(sid = c(rep(1,3), rep(2,4), 3, rep(4,2)),
schid = c(1, 2, 2, 2, 3, 1, 1, 1, 3, 1),
enroll_date = as.Date(c('2004-08-26',
  '2004-10-01',
  '2005-05-01',
  '2004-09-01',
  '2004-11-03',
  '2005-01-11',
  '2005-04-02',
  '2004-09-26',
  '2004-09-01',
  '2005-02-02'),
  format=''
exit_date = as.Date(c('2004-08-26',
  '2005-04-10',
  '2005-06-15',
  '2004-11-02',
  '2005-01-10',
```
\textit{plotForWord} \hspace{1cm} \textit{Save a graphic as a Windows MetaFile}

\textbf{Description}

Convenience function for producing Windows MetaFile graphics for Microsoft documents

\textbf{Usage}

\texttt{plotForWord(x, fn)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \hspace{1cm} A stored plot object to be printed
  \item \texttt{fn} \hspace{1cm} A string for the filename without any file extension
\end{itemize}

\textbf{Details}

Uses parameters designed for import into Word and PowerPoint documents. These are a width of 12, height of 9, pointsize of 16, and using Windows serif fonts.

\textbf{Value}

Nothing. Saves a .wmf with filename \texttt{fn} in the working directory.

\textbf{Note}

Works with \texttt{ggplot} objects in the \texttt{ggplot2} package. Saves plots to the working directory.
Examples

```r
## Not run:
data(cars)
library(ggplot2) # works with ggplot objects
myplot <- qplot(speed~dist, data=cars)
plotForWord(myplot, "speedanddistance")

## End(Not run)
```

### remove_char

Function to replace an arbitrary character like a "*" in redacted data with an NA in R

#### Description

Redacted education data files often have a "*" character. When importing into R this is a problem, which this function solves in a simple step by replacing "*" with NA, and then converting the vector to numeric.

#### Usage

```r
remove_char(x, char)
```

#### Arguments

- **x**: a vector of data that should be numeric but contains characters indicating redaction forcing R to read it as character
- **char**: The character or character string that should be removed from the vector.

#### Value

Returns a vector of the same length as the input vector that is numeric with NAs in place of the character.

#### Note

Future versions could be modified to accommodate other indicators of redacted data.

#### Author(s)

Jared E. Knowles

#### Examples

```r
a <- c(1, 5, 3, 6, "*", 2, 5, "*", "*")
b <- remove_char(a, "*")
as.numeric(b)
```
Description

This function mimics the functionality of the mode function in Stata. It does this by calculating the modal category of a vector and replacing tied categories with a "." to represent a single mode does not exist.

Usage

\[
\text{statamode}(x, \text{method})
\]

Arguments

- \textit{x}: a numeric vector, missing values are allowed
- \textit{method}: a character vector of length 1 specifying the way to break ties in cases where more than one mode exists; either "stata", "sample", or "last". "stata" provides a "." if more than one mode exists. "sample" randomly samples from among the tied values for a single mode. "last" takes the final modal category appearing in the data.

Details

Specifying method="stata" will result in ties for the mode being replaced with a "." character. Specifying "sample" will result in the function randomly sampling among the tied values and picking a single value. Finally, specifying "last" will result in the function picking the value that appears last in the dataset. The default behavior is stata.

Value

The modal value of a vector if a unique mode exists, else output determined by method

Author(s)

Jared E. Knowles

See Also

table which this function uses

Examples

\[
\begin{align*}
\text{# for vectors} \\
\text{a<-c(month.name,month.name)} \\
\text{statamode(a,method="stata") # returns "." to show no unique mode; useful for dply} \\
\text{statamode(a,method="sample") # randomly pick one} \\
\text{a<-c(LETTERS,"A","A")} \\
\text{statamode(a)}
\end{align*}
\]
stuatt

**Student Attributes from the Strategic Data Project Toolkit**

**Description**

A synthetic dataset of student attributes from the Strategic Data Project which includes records with errors to practice data cleaning and implementing business rules for consistency in data.

**Usage**

`data(stuatt)`

**Format**

A data frame with 87534 observations on the following 9 variables.

- `sid`: a numeric vector of the unique student ID
- `school_year`: a numeric vector of the school year
- `male`: a numeric vector indicating 1 = male
- `race_ethnicity`: a factor with levels A B H M/O W
- `birth_date`: a numeric vector of the student birthdate
- `first_9th_school_year_reported`: a numeric vector of the first year a student is reported in 9th grade
- `hs_diploma`: a numeric vector
- `hs_diploma_type`: a factor with levels Alternative Diploma College Prep Diploma Standard Diploma

**Details**

This is the non-clean version of the data to allow for implementing business rules to clean data.

**Source**


**References**

Visit the Strategic Data Project online at: [http://www.gse.harvard.edu/sdp](http://www.gse.harvard.edu/sdp)
**Examples**

```r
data(stuatt)
## maybe str(stuatt); plot(stuatt) ...
```

---

**stulevel**  
* A synthetic data set of K-12 student attributes.

---

**Description**

A small dataset of synthetic data on K-12 students with 2700 observations. 1200 individual students are represented, nested within 4 districts and 2 schools.

**Usage**

```r
data(stulevel)
```

**Format**

A data frame with 2700 observations on the following 32 variables.

- `X` a numeric vector
- `school` a numeric vector
- `stuid` a numeric vector
- `grade` a numeric vector
- `schid` a numeric vector
- `dist` a numeric vector
- `white` a numeric vector
- `black` a numeric vector
- `hisp` a numeric vector
- `indian` a numeric vector
- `asian` a numeric vector
- `econ` a numeric vector
- `female` a numeric vector
- `ell` a numeric vector
- `disab` a numeric vector
- `sch_fay` a numeric vector
- `dist_fay` a numeric vector
- `luck` a numeric vector
- `ability` a numeric vector
- `measerr` a numeric vector
- `teachq` a numeric vector
year a numeric vector
attday a numeric vector
schoolscore a numeric vector
district a numeric vector
schoolhigh a numeric vector
schoolavg a numeric vector
schoollow a numeric vector
readSS a numeric vector
mathSS a numeric vector
prof1vl a factor with levels advanced basic below basic proficient
race a factor with levels A B H I W

Details

This data is synthetically generated to reflect student test scores and demographic attributes.

Source

The script to generate this synthetic dataset can be found and modified at https://github.com/jknowles/r_tutorial_ed

Examples

data(stulevel)

theme_dpi

Description

This is a custom ggplot2 theme developed for the Wisconsin Department of Public Instruction

Usage

theme_dpi(base_size = 16, base_family = "")

Arguments

base_size specify the font size as a numeric value, this value is optional
base_family specify the font family, this value is optional

Details

All values are optional
theme_dpi_map

Value

A theme object which is a list of attributes applied to a ggplot2 object.

Author(s)

Jared E. Knowles

References

For more information see https://github.com/hadley/ggplot2/wiki/Themes

See Also

This uses unit from the gridExtra package extensively. See also theme_bw from the ggplot2 package.

Examples

```r
qplot(mpg, wt, data=mtcars) # standard
qplot(mpg, wt, data=mtcars) + theme_dpi()
```

theme_dpi_map

a ggplot2 theme developed for PDF or SVG maps

Description

This is a custom ggplot2 theme developed for the Wisconsin Department of Public Instruction for making PDF maps

Usage

```r
theme_dpi_map(base_size = 14, base_family = "")
```

Arguments

- `base_size`: specify the font size as a numeric value, this value is optional
- `base_family`: specify the font family, this value is optional

Details

All values are optional

Value

A theme object which is a list of attributes applied to a ggplot2 object.
Author(s)
Jared E. Knowles

References
For more information see https://github.com/hadley/ggplot2/wiki/Themes

See Also
This uses unit from the gridExtra package extensively. See also theme_bw from the ggplot2 package.

Examples

# Data
crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)
require(reshape) # for melt
crimesm <- melt(crimes, id = 1)

# No DPI theme
states_map <- map_data("state")
ggplot(crimes, aes(map_id = state)) + geom_map(aes(fill = Murder), map = states_map) +
expand_limits(x = states_map$long, y = states_map$lat) + labs(title="USA Crime")

# Draw map
last_plot() + coord_map()

# DPI theme
ggplot(crimesm, aes(map_id = state)) + geom_map(aes(fill = value), map = states_map) +
expand_limits(x = states_map$long, y = states_map$lat) + facet_wrap(~ variable) + theme_dpi_map()

theme_dpi_map2

Description
This is a custom ggplot2 theme developed for the Wisconsin Department of Public Instruction for making PDF maps

Usage
theme_dpi_map2(base_size = 14, base_family = "")

Arguments
base_size specify the font size as a numeric value, this value is optional
base_family specify the font family, this value is optional
theme_dpi_mapPNG

Details

All values are optional

Value

A theme object which is a list of attributes applied to a ggplot2 object.

Author(s)

Jared E. Knowles

References

For more information see https://github.com/hadley/ggplot2/wiki/Themes

See Also

This uses unit from the gridExtra package extensively. See also theme_bw from the ggplot2 package.

Examples

```r
# Data
crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)
require(reshape)  # for melt
crimesm <- melt(crimes, id = 1)

# No DPI theme
states_map <- map_data("state")
ggplot(crimes, aes(map_id = state)) + geom_map(aes(fill = Murder), map = states_map) +
expand_limits(x = states_map$long, y = states_map$lat) + labs(title="USA Crime")

# Draw map
last_plot() + coord_map()+theme_dpi_map2()

# DPI theme
ggplot(crimesm, aes(map_id = state)) + geom_map(aes(fill = value), map = states_map) +
expand_limits(x = states_map$long, y = states_map$lat) + facet_wrap(~ variable)+theme_dpi_map2()
```

---

theme_dpi_mapPNG  
an alternate ggplot2 theme developed for PNG or JPG maps

Description

This is a custom ggplot2 theme developed for the Wisconsin Department of Public Instruction for making PNG or JPG maps
Usage

`theme_dpi_mapPNG(base_size = 18, base_family = "")`

Arguments

- `base_size`: specify the font size as a numeric value, this value is optional
- `base_family`: specify the font family, this value is optional

Details

All values are optional

Value

A theme object which is a list of attributes applied to a ggplot2 object.

Author(s)

Jared E. Knowles

References

For more information see https://github.com/hadley/ggplot2/wiki/Themes

See Also

This uses `unit` from the gridExtra package extensively. See also `theme_bw` from the ggplot2 package.

Examples

```r
# Data
crimes <- data.frame(state = tolower(rownames(USArests)), USArests)
require(reshape) # for melt
crimesm <- melt(crimes, id = 1)

# No DPI theme
states_map <- map_data("state")
ggplot(crimes, aes(map_id = state)) + geom_map(aes(fill = Murder), map = states_map) + expand_limits(x = states_map$long, y = states_map$lat) + labs(title="USA Crime")

# Draw map
last_plot() + coord_map()+theme_dpi_mapPNG()

# DPI theme
ggplot(crimesm, aes(map_id = state)) + geom_map(aes(fill = value), map = states_map) + expand_limits(x = states_map$long, y = states_map$lat) + facet_wrap(~ variable)+
theme_dpi_mapPNG()
```
thresh  

function to return the maximum percentage of the cumulative sum represented by a subset of the vector

Description

Returns the proportion of the cumulative sum represented by the number of elements in the vector a user specifies. This allows the user to identify the maximum proportion of the total that only X number of elements may represent in the vector.

Usage

thresh(x, cutoff)

Arguments

- x a numeric vector, missing values are allowed
- cutoff the number of elements to look at

Value

An integer for the proportion of the vector’s sum reached at cutoff

Author(s)

Jared E. Knowles

See Also

cutoff which this function is related to

Examples

# for vector
a <- rnorm(100, mean = 6, sd = 1)
thresh(a, 8) # return minimum number of elements to account 70 percent of total
# for df

## The function is currently defined as
function (x, cutoff)
{
  x <- x[order(-x)]
  xb <- cumsum(x)
  xc <- xb/sum(x, na.rm = T)
  xc[cutoff]
}
Index

*Topic \textasciitildeaplot
  theme_dpi, 24
  theme_dpi_map, 25
  theme_dpi_map2, 26
  theme_dpi_mapPNG, 27

*Topic \textasciitildekwd1
  max_mis, 15

*Topic \textasciitildekwd2
  max_mis, 15
  thresh, 29

*Topic \textasciitildemanip
  cutoff, 6
  remove_char, 20
  statamode, 21
  thresh, 29

*Topic \textasciitildemisc
  theme_dpi, 24
  theme_dpi_map, 25
  theme_dpi_map2, 26
  theme_dpi_mapPNG, 27

*Topic \textasciitildutilities
  dropbox_source, 10

*Topic \textasciitildeaplot
  theme_dpi, 24
  theme_dpi_map, 25
  theme_dpi_map2, 26
  theme_dpi_mapPNG, 27

*Topic \textasciitildeutilities
  dropbox_source, 10

*Topic \textasciitildeaplot
  theme_dpi, 24
  theme_dpi_map, 25
  theme_dpi_map2, 26
  theme_dpi_mapPNG, 27

*Topic \textasciitildemanip
  cutoff, 6
  remove_char, 20
  statamode, 21
  thresh, 29

*Topic \textasciitildemisc
  theme_dpi, 24
  theme_dpi_map, 25
  theme_dpi_map2, 26
  theme_dpi_mapPNG, 27

*Topic \textasciitildutilities
  dropbox_source, 10

*Topic datasets
  midsch, 16
  stuatt, 22
  stulevel, 23

*Topic package
  eeptools-package, 2

  age_calc, 3
  autoplot, 4
  character, 9
  cleanTex, 6
  cumsum, 7
  cutoff, 6, 29
  data.table, 17
  decomma, 7
  defac, 8
  destring, 9
  difftime, 4
  dropbox_source, 10
  eeptools (eeptools-package), 2
  eeptools-package, 2
  factor, 8
  fortify, 13
  gelmansim, 11
  ggmapmerge, 12
  ggpplot, 19
  ggpplot2, 13, 19
  glm, 11
  gpc/libPermit, 13
  lag_data, 13
  levels, 8
  lm, 11
  mapmerge, 14
  mapmerge2, 15
  max, 16
  max_mis, 15
  midsch, 16
  moves_calc, 17
  plot.lm, 5
  plotForWord, 19
  remove_char, 20
  sim, 11
  source, 10
  statamode, 21
  stuatt, 22
  stulevel, 23
  table, 21
  theme_bw, 25–28
INDEX

theme_dpi, 24
theme_dpi_map, 25
theme_dpi_map2, 26
theme_dpi_mapPNG, 27
thresh, 29

unit, 25–28