Package 'eeptools'

October 21, 2018

Type Package

Title Convenience Functions for Education Data

Version 1.2.1

Description Collection of convenience functions to make working with administrative records easier and more consistent. Includes functions to clean strings, and identify cut points. Also includes three example data sets of administrative education records for learning how to process records with errors.

License GPL-3

Depends R (>= 2.15.1), ggplot2

Imports arm, data.table, vcd, maptools

Suggests testthat, stringr, knitr, rmarkdown, MASS

LazyData true

VignetteBuilder knitr

RoxygenNote 6.0.1

URL https://github.com/jknowles/eeptools

BugReports https://github.com/jknowles/eeptools/issues

NeedsCompilation no

Author Jason P. Becker [ctb],
Jared E. Knowles [aut, cre]

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

Repository CRAN

Date/Publication 2018-10-21 16:20:02 UTC

R topics documented:

`age_calc` *Function to calculate age from date of birth.*

Description

his 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)
```

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	character, which units of age should be calculated? allowed values are days, months, and years
precise	logical indicating whether or not to calculate with leap year and leap second precision

Value

A numeric vector of ages the same length as the dob vector

Author(s)

Jason P. Becker

Source

This function was developed in part from this response on the R-Help mailing list.

See Also

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

Examples

```
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
```

---

autoplot.lm *A function to replicate the basic plot function for linear models in ggplot2*

Description

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

Usage

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

Arguments

object	a linear model object from lm
which	which of the tests do we want to display output from
mfrow	Describes the layout of the resulting function in the plot frames
...	additional parameters to pass through

Value

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

References

Modified from: http://librestats.com/2012/06/11/autoplot-graphical-methods-with-ggplot2/

See Also

plot.lm which this function mimics

Examples

```
# Univariate
a <- runif(1000)
b <- 7 * a + rnorm(1)
mymod <- lm(b~a)
autoplot(mymod)
# Multivariate
data(mpg)
mymod <- lm(cty~displ + 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(fn, keepPDF = TRUE, keepRnw = TRUE, keepRproj = TRUE)
```

Arguments

fn	a filename for your .Rnw file
keepPDF	Logical. Should function save PDF files with filename fn. Default is TRUE.
keepRnw	Logical. Should function save Rnw files with filename fn. Default is TRUE.
keepRproj	Logical. Should function save .Rproj files with filename fn. Default is TRUE.

Value

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

---

crosstabplot *Draw a visual crosstab (mosaic plot) with shading for correlations and labels in each cell.*

Description

Improves labeling of mosaic plots over mosaic from the vcd package

Usage

```
crosstabplot(data, rowvar, colvar, varnames, title = NULL, subtitle = NULL,
  label = FALSE, shade = TRUE, ...)
```

Arguments

data	a data object, matrix or dataframe, that contains the categorical variables to compose the crosstab
rowvar	a character value for the column in data that will be displayed on the rows of the crosstab
colvar	a character value for the column in data that will be displayed in columns of the crosstab
varnames	a character vector of length two with the labels for rowvar and colvar respectively
title	a character vector of length one that contains the main title for the plot
subtitle	a character vector of length one that contains the subtitle displayed beneath the plot
label	logical, if TRUE cells will be labeled, else they will not
shade	logical, if TRUE cells will be shaded with Pearson residuals
...	additional arguments to crosstabs e.g. digits

Value

A mosaic plot

Source

http://www.rexdouglass.com/blog:3

See Also

mosaic which this function wraps crosstabs which does the data manipulation for the crosstab

Examples

```
df <- data.frame(cbind(x=seq(1,3,by=1), y=sample(LETTERS[6:8],60,replace=TRUE)),
fac=sample(LETTERS[1:4], 60, replace=TRUE))
varnames<-c('Quality','Grade')
myCT <- crosstabs(df, rowvar = "x",colvar = "fac", varnames = varnames, digits =2)
crosstabplot(df, rowvar = "x",colvar = "fac", varnames = varnames,
title = 'My Plot', subtitle = 'Foo', label = FALSE, shade = TRUE, digits = 3)
```

---

crosstabs *Build a list of crosstabulations from a dataset*

Description

Build a list of crosstabulations from a dataset

Usage

```
crosstabs(data, rowvar, colvar, varnames, digits = 2)
```

Arguments

data	a data object, matrix or dataframe, that contains the categorical variables to compose the crosstab
rowvar	a character value for the column in data that will be displayed on the rows of the crosstab
colvar	a character value for the column in data that will be displayed in columns of the crosstab
varnames	a character vector of length two with the labels for rowvar and colvar respectively
digits	an integer for how much to round the proportion calculations by, default is 2

Value

a list with crosstab calculations

Examples

```
df<-data.frame(cbind(x=seq(1,3,by=1), y=sample(LETTERS[6:8],60,replace=TRUE)),
fac=sample(LETTERS[1:4], 60, replace=TRUE))
varnames<-c('Quality','Grade')
myCT <- crosstabs(df, rowvar = "x",colvar = "fac", varnames = varnames, digits =2)
```

cutoff *A function to calculate thresholds of cumulative sums 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, na.rm = TRUE)
```

Arguments

x	a numeric vector, missing values are allowed
cutoff	a user defined numeric value to stop the cutoff specified as a proportion 0 to 1
na.rm	logical, should missing values be excluded?

Details

Calculates the distance through a numeric vector before a certain proportion of the sum 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

Examples

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

decomma *Remove commas from numeric fields and return them as numerics*

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.

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

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

Author(s)

Jared E. Knowles

See Also

factor, levels to understand the R implementation of factors.

Examples

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

eeptools *Evaluation of educational policy tools*

Description

Make common tasks for educational evaluation easier to do!

Details

Package:	eeptools
Type:	Package
Version:	1.2.0
Date:	2018-06-01
License:	GPL-3

his 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 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

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)

makenum(gender)
gender <- factor(gender)
defac(gender)
```

`gelmansim` *Generate prediction intervals for model functions*

Description

Generate prediction intervals from R models following Gelman and Hill

Usage

```
gelmansim(mod, newdata, n.sims, 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
n.sims	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

Modified from Gelman and Hill 2006. Data Analysis Using Regression and Multilevel/Hierarchical Models. Cambridge University Press.

Examples

```
 #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, n.sims=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, n.sims=200, na.omit=TRUE)

## End(Not run)
```

---

ggmapmerge *A depcrecated method for fortifying SpatialPolygonsDataFrames for plotting*

Description

Convenience function for fortifying SpatialPolygonsDataFrames for ggplot2 plotting.

Usage

```
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

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

## End(Not run)
```

isid *A function to check if a set of variables form a unique ID in a dataframe.*

Description

When passed a set of variable names and a dataframe, this function returns a check TRUE/FALSE whether or not the variables together uniquely identify a row in the dataframe.

Usage

```
isid(data, vars, verbose = FALSE)
```

Arguments

data	A dataframe.
vars	A character vector specifying the column names in the dataframe to check as unique.
verbose	A logical, default FALSE. If TRUE, isid will tell you how many rows you need and how many your variables uniquely identify

Value

TRUE or FALSE. TRUE indicates the variables uniquely identify the rows. FALSE indicates they do not.

Author(s)

Jared E. Knowles

Examples

```
data(stuatt)
isid(stuatt, vars = c("sid"))
isid(stuatt, vars = c("sid", "school_year"))
isid(stuatt, vars = c("sid", "school_year"), verbose = TRUE)
```

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
time	The variable representing time periods
periods	A scalar for the number of periods to be lagged in the data. Can be negative to indicate leading variable.
values	The names of the variables to be lagged

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)
```

`leading_zero` *Function to add leading zeroes to maintain fixed width.*

Description

This function ensures that fixed width data is the right length by padding zeroes to the front of values. This is a common problem with fixed width data after importing into R as non-character type.

Usage

```
leading_zero(x, digits = 2)
```

Arguments

`x`	a vector of numeric data that should be fixed width but is missing leading zeroes.
`digits`	an integer representing the desired width of `x`

Details

If `x` contains negative values then the width specified by digits will include one space taken up for the negative sign. The function does not trim values that are longer than digits, so the vector produced will not have a uniform width if `nchar(x) > d`

Value

A character vector of length `digits`

Author(s)

Jason P. Becker

Jared E. Knowles

Examples

```
a <- seq(1,10)
a <- leading_zero(a, digits = 3)
a
```

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

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

```
makenum(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 numeric

Note

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

Examples

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

mapmerge — *A depcrecated method for converting polygons to dataframes Combine 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 a dataframe
xid	Name of ID variable in the SpatialPolygonsDataFrame
yid	Name of ID variable in the 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)
```

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

```
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

A vector with the maximum value or with an NA of the proper type

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),na.rm=TRUE)
max_mis(c(NA,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

```
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_t95` 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

```
data(midsch)
head(midsch)
```

`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

```
moves_calc(df, enrollby, exitby, gap = 14, sid = "sid", schid = "schid",
  enroll_date = "enroll_date", exit_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.
exitby	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, of days, 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.

Details

enrollby and exitby are specified automatically if not defined. They are assigned to the default dates of -09-15 and -06-01 of the min and max year respectively.

Value

a data.frame

Author(s)

Jason P. Becker

Examples

```
## 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='%Y-%m-%d'),
                    exit_date = as.Date(c('2004-08-26',
                                        '2005-04-10',
                                        '2005-06-15',
                                        '2004-11-02',
                                        '2005-01-10',
                                        '2005-03-01',
                                        '2005-06-15',
                                        '2005-05-30',
                                        NA,
                                        '2005-06-15'),
                                      format='%Y-%m-%d'))
moves <- moves_calc(df)
moves
moves <- moves_calc(df, enrollby='2004-10-15', gap=22)
moves
moves <- moves_calc(df, enrollby='2004-10-15', exitby='2005-05-29')
moves

## End(Not run)
```

`nth_max` *Find the nth maximum value*

Description

Find the nth maximum value

Usage

```
nth_max(x, n = 1)
```

Arguments

`x`	a vector of numeric values
`n`	which max to return

Value

the value of the nth most maximum value in a vector

Note

If `n` is smaller/larger than `0`/`length(unique(x))` the error 'index outside bounds' is thrown.

Examples

```
x <- c(1:20, 20:1)
nth_max(x, n = 1) #20
nth_max(x, n = 2) #19
```

`profpoly` *Creates a proficiency polygon in ggplot2 for showing assessment categories*

Description

Creates a proficiency polygon in ggplot2 for showing assessment categories

Usage

```
profpoly(data)
```

Arguments

`data`	a data.frame produced by `profpoly.data`

Value

a ggplot2 object that can be printed or saved

See Also

`geom_polygon` which this function wraps

Examples

```
grades<-c(3,4,5,6,7,8)
g <- length(grades)
LOSS <- rep(200, g)
HOSS <- rep(650, g)
basic <- c(320,350,370,390,420,440)
minimal <- basic-30
prof <- c(380,410,430,450,480,500)
adv <- c(480,510,530,550,580,600)
z <- profpoly.data(grades, LOSS, minimal, basic, proficient = prof,
                   advanced = adv, HOSS)
profpoly(z)
```

profpoly.data *Creates a data frame suitable for building custom polygon layers in ggplot2 objects*

Description

Creates a data frame suitable for building custom polygon layers in ggplot2 objects

Usage

```
profpoly.data(grades, LOSS, minimal, basic, proficient, advanced, HOSS)
```

Arguments

grades	a vector of tested grades in sequential order
LOSS	is a vector of the lowest obtainable scale score on an assessment by grade
minimal	is a vector of the floor of the minimal assessment category by grade
basic	is a vector of the floor of the basic assessment category by grade
proficient	is a vector of the floor of the proficient assessment category by grade
advanced	is a vector of the floor of the advanced assessment category by grade
HOSS	is a vector of the highest obtainable scale score by grade

Value

a dataframe for adding a polygon to layers in other ggplot2 plots

See Also

geom_polygon which this function assists

Examples

```
grades<-c(3,4,5,6,7,8)
g<-length(grades)
LOSS<-rep(200,6)
HOSS<-rep(650,6)
basic<-c(320,350,370,390,420,440)
minimal<-basic-30
prof<-c(380,410,430,450,480,500)
adv<-c(480,510,530,550,580,600)

z<-profpoly.data(grades,LOSS,minimal,basic,
               proficient = prof,advanced = adv, HOSS)
z
```

`remove_char` *A 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

```
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 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

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

`retained_calc` *Function to calculate whether a student has repeated a grade.*

Description

This function calculates whether or not a student has repeated a grade. It returns a `data.frame` with the student ID and a character vector with `Y` representing they repeated the grade and `N` that they had not.

Usage

```
retained_calc(df, sid = "sid", grade = "grade", grade_val = 9)
```

Arguments

`df`	a data.frame containing minimally a student identifier and their grade.
`sid`	a character that indicates the name of the student id attribute in `df`. The default value is `sid`.
`grade`	a character that indicates the name of the student grade attribute in `df`. The default value is `grade`.
`grade_val`	a numeric vector that contains the value of the grade that is being checked for retention. The default value is `9`.

Value

a data.frame

Author(s)

Jason P. Becker

Examples

```
x <- data.frame(sid = c(101, 101, 102, 103, 103, 103, 104),
                grade = c(9, 10, 9, 9, 9, 10, 10))
retained_calc(x)
```

`statamode` *A function to mimic the mode function in Stata.*

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

```
statamode(x, method = c("last", "stata", "sample"))
```

Arguments

`x`	a vector, missing values are allowed
`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 original x vector. 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

```
a <- c(month.name, month.name)
statamode(a, method="stata") # returns "." to show no unique mode; useful for ddply
statamode(a ,method="sample") # randomly pick one
a <- c(LETTERS, "A" , "A")
statamode(a)
```

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

```
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`

`hs_diploma_date` a factor with levels `12/2/2008` `12/21/2008` `4/14/2008` `4/18/2008` ...

Details

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

Source

Available from the Strategic Data Project online at http://sdp.cepr.harvard.edu/toolkit-effective-data-use

References

Visit the Strategic Data Project online at: http://sdp.cepr.harvard.edu/

Examples

```
data(stuatt)
head(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

```
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

`proflvl` 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)
head(stulevel)
```

---

theme_dpi *a deprecated ggplot2 theme developed for PDF and PNG for use at the Wisconsin Department of Public Instruction*

Description

This is a custom ggplot2 theme developed for the Wisconsin Department of Public Instruction. This function is now deprecated.

Usage

```
theme_dpi(base_size = 16, base_family = "")
```

Arguments

`base_size`	numeric, specify the font size as a numeric value, default is 16
`base_family`	character, 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

Source

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

See Also

his uses `unit` from the grid package extensively. See also `theme_bw` from the ggplot2 package.

`theme_dpi_map` *a deprecated ggplot2 theme developed for PDF or SVG maps*

Description

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

Usage

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

Arguments

`base_size`	numeric, specify the font size, default is 14
`base_family`	character, 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

Source

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

See Also

his uses unit from the grid package extensively. See also theme_bw from the ggplot2 package.

`theme_dpi_map2` *an alternate deprecated ggplot2 theme developed for PDF or SVG maps*

Description

This is a deprecated 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`	numeric, specify the font size, default is 14
`base_family`	character, 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

Source

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

See Also

his uses unit from the grid package extensively. See also theme_bw from the ggplot2 package.

`theme_dpi_mapPNG` *an deprecated ggplot2 theme developed for PNG or JPG maps*

Description

This is a deprecated 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`	numeric, specify the font size, default is 18
`base_family`	character, 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

Source

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

See Also

his uses `unit` from the grid package extensively. See also `theme_bw` from the ggplot2 package.

thresh *A 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, na.rm = TRUE)
```

Arguments

x	a numeric vector, missing values are allowed
cutoff	numeric, the number of elements to look at
na.rm	logical, should missing values be excluded?

Details

Calculates the proportion of a numeric vector reached after sorting the vector in ascending order and stopping at the specified count

Value

A numeric proportion

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
```

Index