Package 'utilitR'

December 17, 2017

Type Package

Title Lots of useful functions

Version 0.1.0

Author JuG

Maintainer JuG <julien.godet@unistra.fr>

Description Useful functions

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

R topics documented:

cleanUpAttempt *Attempt to clean-up messy vectors*

Description

An attempt to clean-up messy vectors

Usage

```
cleanUpAttempt(messy, cluster = FALSE, ngroup = NULL)
```

Arguments

messy	a factor vector

Details

use stringdist package. examples from https://cran.r-project.org/web/packages/rrefine/vignettes/rrefine-vignette.html

Value

void or cleaned-up factor

Author(s)

JuG

Examples

```
x <- c("Y", "Y", "Yes", "N", "No","No","No","No","No","Nope","Yes","Yes","Yes")
cleanUpAttempt(x)
cleanUpAttempt(x,ngroup =  2)
cleanUpAttempt(messy = x,ngroup =  2, cluster = TRUE)


# install.packages("devtools")
devtools::install_github("vpnagraj/rrefine")
library(rrefine)
cleanUpAttempt(messy = lateformeeting$what.day.whas.it,ngroup=5)
cleaned <-cleanUpAttempt(messy = lateformeeting$what.day.whas.it,ngroup = 5,cluster=TRUE)
table(cleaned)

cleanUpAttempt(messy = lateformeeting$was.i.on.time.for.work,ngroup=2)

raw <- c("persistante modérée à sévère", "Persistante modérée a sévère","légère", "persistante modérée à sévè
cleanUpAttempt(messy = raw,ngroup=4)
#' cleanUpAttempt(messy = raw)
```

closeRefine *Close connection to openRefine API*

Description

Get openRefine cleaned data.frame and close

Usage

```
closeRefine(dataFrameOutput = TRUE, kill = FALSE)
```

Arguments

dataFrameOutput	if TRUE, return a data.frame (a tibble otherwise)
kill	if TRUE delete openRefine project

Details

...

Value

data.frame or tibble

Author(s)

JuG

Examples

```
cleanedData <- closeRefine(dataFrameOutput = TRUE)
```

factOrNot *Define class of data.frame variables*

Description

Merge several data.frames

Usage

```
factOrNot(datafr, arg = "factor", ind)
```

Arguments

datafr	data.frame
arg	class to be evaluated c(factor","Date","integer","numeric")
ind	if TRUE return index, else return names

Value

list or names of columns

Author(s)

JuG

`HDIofICDF` *Compute Highest Density Interval limits*

Description

Compute Highest Density interval limits.

Usage

```
HDIofICDF(ICDFname, credMass = 0.95, tol = 1e-08, ...)
```

Arguments

`ICDFname`	R's name for the inverse cumulative density function of the distribution
`credMass`	the desired mass of the HDI region.
`tol`	tolerance passed to R's optimize function

Details

Code from J.K.Kruschke - Doing bayesian data analysis. Adapted and corrected from Greg Snow's TeachingDemos package.

Value

HDI

Author(s)

JuG

Examples

```
#ICDFname must be explicitly named
HDIofICDF( qbeta , shape1 = 30 , shape2 = 12 )

#does not work
HDIofICDF( qbeta , 30 , 12 )
```

lmB *Fitting Linear Models using bayesian inference*

Description

Fitting Linear Models using bayesian inference

Usage

```
lmB(formula, data, graphOutput = TRUE, nIter = 10000, thin = 1)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which blm is called.
graphOutput	regression parameters graphical output (MCMC Trace and posterior density)
nIter	number of iterations
thin	thinning interval for monitors

Details

Models for lm are specified symbolically. A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response.

Value

regression parameters

Author(s)

JuG

Examples

```
data(mtcars)
summary(lm(mpg~ cyl + vs+gear+carb,data=mtcars))
lmB(mpg~ cyl + vs+gear+carb,data=mtcars,nIter=50000)
lmB(mpg~ .,data=mtcars,nIter=50000,graphOutput=FALSE)
```

`lmBselect` *Variable selection and fitting Linear Models using bayesian inference.*

Description

Variable selection and fitting Linear Models using bayesian inference.

Usage

```
lmBselect(formula, data, graphOutput = TRUE, nIter = 10000, thin = 1,
  effect = "random")
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which blm is called.
`graphOutput`	regression parameters graphical output (MCMC Trace and posterior density)
`nIter`	number of iterations
`thin`	thinning interval for monitors
`effect`	"fixed", "random" or "randomPrior" effect for variable selection

Details

Models for lm are specified symbolically. A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. The effect "randomPrior" adds a beta prior for the model inclusion probability. This induces a distribution for the number of included variables which has longer tails than the binomial distribution, allowing the model to learn about the degree of sparsity.

Value

regression parameters

Author(s)

JuG

References

1. O'Hara, R. and Sillanpaa, M. (2009) A review of Bayesian variable selection methods: what, how and which. Bayesian Analysis, 4(1):85-118.
2. Kuo, L. and Mallick, B. (1998) Variable selection for regression models. Sankhya B, 60(1):65-81.

Examples

```
data(mtcars)
summary(lm(mpg~ cyl + vs+gear+carb,data=mtcars))
lmB(mpg~ cyl + vs+gear+carb,data=mtcars,nIter=50000)
lmB(mpg~ .,data=mtcars,nIter=50000,graphOutput=FALSE)
mod <- lm(mpg~ .,data=mtcars)
stepAIC(mod,direction = "both")
```

---

merge.all *Merge several data.frames*

Description

Merge several data.frames

Usage

```
## S3 method for class 'all'
merge(by, ...)
```

Arguments

by	variable to merge on

Value

merged data.frame

Author(s)

JuG

Examples

```
a <- data.frame("USUBJID" = 1:10, val1 = rnorm(10,1,.2))
b <- data.frame("USUBJID" = 1:10, val2 = LETTERS[1:10])
c <- data.frame("USUBJID" = 1:10, val3 = gl(n = 5,k = 2,length = 10))
merged<-merge.all(by="USUBJID",a,b,c)
```

---

na.count *Count missing data*

Description

Count missing data

Usage

```
na.count(data)
```

Arguments

data	a vector (numeric, factor or ordered), a matrix, a data.frame or a list

Details

key function `nafun <- function(x){sum(is.na(x))}`

Value

vector of missing value per variable

Author(s)

JuG

Examples

```
#with data.frame
data3 <- data.frame(a = c(1,2,3), b= c("e",NA,3), c = 1:3)
na.count(data3)
#with list
data4 <- list(a = c(1,2,3), b= c("e",NA))
na.count(data4)
```

---

newFun *Print a skeleton for a new function in the console*

Description

Print a skeletton for a new function in the console

Usage

```
newFun(name, createFile = FALSE, path = NULL)
```

Arguments

name	A name for the function

Details

>Sys.setenv(TZ="Europe/Berlin") >Sys.setlocale("LC_TIME", "C"); to get UK date format

Value

void

Author(s)

JuG

Examples

```
newFun("newFunction")
newFun("newFunction",createFile = TRUE)
```

`nlsB` *Determine the nonlinear (weighted) MCMC estimates of the parameters of a nonlinear model.*

Description

Determine the nonlinear (weighted) MCMC estimates of the parameters of a nonlinear model.

Usage

```
nlsB(formula, data, graphOutput = TRUE, nIter = 10000, thin = 1)
```

Arguments

`formula`	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
`data`	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which blm is called.
`graphOutput`	regression parameters graphical output (MCMC Trace and posterior density)
`nIter`	number of iterations
`thin`	thinning interval for monitors

Details

Models for lm are specified symbolically. A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response.

Value

non-linear regression parameters

Author(s)

JuG

Examples

```
data(mtcars)
summary(lm(mpg~ cyl + vs+gear+carb,data=mtcars))
lmB(mpg~ cyl + vs+gear+carb,data=mtcars,nIter=50000)
lmB(mpg~ .,data=mtcars,nIter=50000,graphOutput=FALSE)
```

openRefine *Connect to openRefine API*

Description

Connect to openRefine API to import, export or delete a project in OpenRefine directly from an R script

Usage

```
openRefine(data)
```

Arguments

data	a data.frame

Details

Connect to openRefine API https://cran.r-project.org/web/packages/rrefine/vignettes/rrefine-vignette.html. openRefine must be installed on the local machine http://openrefine.org/download.html.

Value

void

Author(s)

JuG

Examples

```
devtools::install_github("vpnagraj/rrefine")
library(rrefine)
lateformeeting[1:10,]
openRefine(lateformeeting)
```

piEst *Estimate population proportion and interval limits*

Description

Estimate population proportion and interval limits from observede counts of succes and trials.

Usage

```
piEst(y = 12, n = 20, beta.priors = c(0.5, 0.5), credMass = 0.95,
  showPlot = FALSE, ...)
```

Arguments

y	counts of successes
n	counts of trials
beta.priors	vector of 2 non-negative parameters of the Beta prior distribution (shapes - by default Jeffrey's c(0.5,0.5))
credMass	mass of the HDI region
showPlot	if TRUE, show a graphical representation of π distribution

Value

Proportion estimates with HDI, CI (quantile c(.025,0.975)), and binomial CI (from binom.test)

Author(s)

JuG

Examples

```
piEst(y = 3, n = 12, showPlot = TRUE)
```

---

plotCorMat *Plot correlation matrix*

Description

Plot correlation matrix

Usage

```
plotCorMat(cormat, cexlab = 12, angle.label = 45)
```

Arguments

cormat	correlation matrix
cexlab	label size
angle.label	cex.label size

Value

plot

Author(s)

JuG

Examples

```
#correlation matrix - simulated (or alternative fact)
corMat <- matrix(runif(400,-1,1),ncol=20)
diag(corMat)<-1
colnames(corMat) <- rownames(corMat) <- paste("Vble",1:20,sep='')
#plot CorMat
plotCorMat(corMat)

# Correlation matrix from mtcars
corMat2 <-cor(mtcars)
plotCorMat(corMat2)
```

`relWeights` *Functions for predictor selection*

Description

Predictor selection.The total amount of variance accounted for by the model is divided amoong the predicator variables.

Usage

```
relWeights(fit, ...)
```

Arguments

`fit`	a models lm object.

Value

graphic for rredictor selection

Author(s)

JuG

Examples

```
form = as.formula("Fertility ~ .")
allsubreg(formula = form, data = swiss, nbest=6)

if(!require('MASS')){install.packages('MASS')}
library('MASS')
fit1 <- lm(Fertility ~ ., data = swiss)
relWeights(fit1, col="lightgrey",las=1)
```

stripPlot *Plot a customized stipchart*

Description

Plot a customized stipchart

Usage

```
stripPlot(data, formula, col = NULL, addBoxplot = FALSE, jitter = 0.3,
  grid = FALSE, xlab = NULL, ylab = NULL, xlim = NULL, ylim = NULL,
  ...)
```

Arguments

formula	a formula, such as y ~ grp, where y is a numeric vector of data values to be split into groups according to the grouping variable grp (usually a factor)
col	if col is non-null it is assumed to contain colors to be used
addBoxplot	if TRUE, a bxplot (package beeswarm) is drawn for each factor modalities
grid	if TRUE, draw a grid
datafr	data.frame

Value

customized stripchart graphics

Author(s)

JuG

Examples

```
dtf <- data.frame(val1 = rnorm(1000,1,.2), fact2 = LETTERS[gl(n = 5,k = 200,length = 1000)])
if(!require(colorRamps)){install.packages('colorRamps')}
require(colorRamps)
stripPlot(data = dtf, val1 ~ fact2, addBoxplot = TRUE,col=matlab.like(12))

if(!require(RColorBrewer)){install.packages('RColorBrewer')}
library("RColorBrewer")
stripPlot(data = dtf, val1 ~ fact2,col=brewer.pal(n = 5, name = "Dark2"), jitter = .2,grid = T)
```

subReg *Functions for model selection*

Description

Model selection by exhaustive search. All subset regression are evaluated to help choosing the best minimalist model. R in action p 211.

Usage

```
subReg(formula, data, nbest)
```

Arguments

formula	The models fit by, e.g., the lm and glm functions specified in a compact symbolic form.
data	data.frame
nbest	number of subsets of each size to record

Value

graphic for best model selection based on Adjusted R-squre

Author(s)

JuG

Examples

```
form = as.formula("Fertility ~ .")

if(!require('MASS')){install.packages('MASS')}
library('MASS')
fit1 <- lm(form, data = swiss)
stepAIC(fit1,direction = "both")

subReg(form, data=swiss, nbest = 3)

relweights(fit1, col="lightgrey",las=1)
```

summaryOR *Odds ratio summary statistics #' @author JuG*

Description

Fitting Linear Models using bayesian inference

Usage

```
summaryOR(glm.fit, lateX = TRUE)
```

Arguments

glm.fit	an object of class "glm.fit" calculated with a "binomial "link function.
lateX	if TRUE, return lateX xtable of the result table

Details

Provide Odds ratio (exponential of the glm.fit coefficients) and their confidence intervals (given by confint). Stars are "***" if p-value < 0.001, "**" if 0.001 < p-value <= 0.01, "*" if 0.01 < p-value <= 0.05, "." if 0.05 < p-value <=0.1

Examples

```
y <- rbinom(n = 50, size = 1, prob = .3)
x1 <- rnorm(50, 5,1)
x2 <- runif(50, 0,5) * (1 + y)
x3 <- rnorm(50, 12,5)
 Data <- data.frame(y,x1,x2,x3)

 logist <- glm(y~., data=Data, family="binomial")
 summary(logist)
 summaryOR(logist,lateX=FALSE)
```

tonum *Convert ill-encoded factor to numeric*

Description

Convert ill-encoded factor to numeric

Usage

```
tonum(data, comma2period = TRUE)
```

Arguments

data	a vector (character, factor, oredered) or a data.frame
comma2period	change "," to "."

Details

Apply `as.numeric(as.character())`

Value

numeric() vector or matrixaze

Author(s)

JuG

Examples

```
devtools::install_github("vpnagraj/rrefine")
library(rrefine)
lateformeeting$sleephours
tonum(lateformeeting$sleephours)
```

---

unaccent *Transform accented into unaccented characters*

Description

Transform accented into unaccented characters

Usage

```
unaccent(text)
```

Arguments

`text`	an in put that can be coerced as character

Details

from http://data.hypotheses.org/564

Value

text

Author(s)

JuG

Examples

```
unaccent(c("é","à","è","ù","ç",'ü',"a,"))
unaccent("À Süßen (Ba.-Wü.), j'ai mangé des Curry Würstchen")
```

Index