

Package ‘kyotil’

July 2, 2014

LazyLoad yes

LazyData yes

Version 2014.4-1

Date

Title Utility Functions by Youyi & Krisz

Author Youyi Fong <yfong@fhcrc.org>, Krisztian Sebestyen <ksebestyen@gmail.com>

Maintainer Youyi Fong <yfong@fhcrc.org>

Depends R (>= 3.0.0)

Imports

Suggests RUnit, knitr, lme4, nlme, xtable, MASS

VignetteBuilder knitr

Description A miscellaneous set of functions for printing, plotting, kernels, etc. Additional contributors are acknowledged on individual function help pages.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-04-01 11:52:47

R topics documented:

base.functions	2
getK	4
kyotil	5
math.functions	6
matrix.array.functions	7
matrix2	9

plotting	10
print.functions	14
random.functions	15
regression.model.functions	16
stat.functions	19
string.functions	20
testing.functions	21

Index	23
--------------	-----------

base.functions	<i>Some Base Functions</i>
----------------	----------------------------

Description

binary returns binary representation of an integer. binary2 returns binary representatin of an integer with leading 0, the length of string is n. mysystem can call any exe file that is in the PATH f2c convert temperature from f to c/

Usage

binary(i)

binary2(i, n)

f2c(f)

ftoi(f)

keepWarnings(expr)

meanmed(x, na.rm = FALSE)

methods4(classes, super = FALSE, ANY = FALSE)

myaggregate(x, by, FUN, new.col.name = "aggregate.value", ...)

myreshapewide(formula, dat, idvar = NULL)

mysapply(X, FUN, ..., simplify = TRUE, USE.NAMES = TRUE, ret.mat = TRUE)

myscale(x)

mysystem(cmd, ...)

mytapply(X, INDEX, FUN = NULL, ..., simplify = TRUE)

```
read.csv(file, header = TRUE, ...)  
read.tsv(file, header = TRUE, sep = "\t", ...)  
table.prop(x, y, digit = 1, style = 2, whole.table.add.to.1=FALSE)  
unix()
```

Arguments

i
n
f
formula a formula object.
expr
x
na.rm
classes
super
ANY
by
whole.table.add.to.1
 Boolean
new.col.name
...
dat
idvar
X
simplify
USE.NAMES
ret.mat
cmd
INDEX
file
header
sep
y
digit
style
FUN

Examples

```
binary(5) ### 101
binary2(5, 4)
```

getK

getK

Description

getK calculates the kernel matrix between X and itself and returns a n by n matrix. Alternatively, it calculates the kernel matrix between X and X2 and returns a n by n2 matrix.

Usage

```
getK (X, kernel, para=NULL, X2=NULL, C = NULL)
```

Arguments

X	covariate matrix with dimension n by d. Note this is not the paired difference of covariate matrix.
kernel	string specifying type of kernel: polynomial or $p(1 + \langle x, y \rangle)^{\text{para}}$, rbf or $r \exp(-\text{para} * \ x - y\ ^2)$, linear or $l \langle x, y \rangle$, ibs or $i 0.5 * \text{mean}(2.0 - x - y)$ or $\text{sum}(w * (2.0 - x - y)) / \text{sum}(w)$, with $x[i], y[i]$ in 0,1,2 and weights 'w' given in 'para'. hamming or h for $\text{sum}(x == y)$ with $x[i], y[i]$ binary, no default.
para	parameter of the kernel function. for ibs or hamming, para can be a vector of weights.
X2	optional second covariate matrix with dimension n2 by d
C	logical. If TRUE, kernels are computed by custom routines in C, which may be more memory efficient, and faster too for ibs and hamming kernels.

Details

IBS stands for 'Identical By State'. If 'x', 'y' are in in 0,1,2 then

$\text{IBS}(x, y) = 0$ if $|x - y| = 2$, 1 if $|x - y| = 1$, 2 if $|x - y| = 0$, or $\text{IBS}(x, y) = 2.0 - |x - y|$.

$K(u, v) = \text{sum}(\text{IBS}(u[i], v[i])) / 2K$ where $K = \text{length}(u)$.

The 'hamming' kernel is the equivalent of the 'ibs' kernel for binary data. Note that 'hamming' kernel is based on hamming similarity(!), not on dissimilarity distance.

Within in the code, C is default to TRUE for ibs and hamming kernels and FALSE otherwise.

Value

A kernel matrix.

Author(s)

Youyi Fong <youyifong@gmail.com>
Krisztian Sebestyen <>
Shuxin Yin <>

Examples

```
X = cbind(x1=rnorm(n=5), x2=rnorm(n=5))
dim(X)
X2 = cbind(x1=rnorm(n=3), x2=rnorm(n=3))
dim(X2)

K = getK(X,"linear")
dim(K)

K = getK(X,"linear",X2=X2)
dim(K)
K1 = getK(X2,"1",X2=X)
dim(K1)
all(K==t(K1))

# RBF kernel
K = getK(X,"rbf",para=1,X2=X2)
K1 = getK(X2,"r",para=1,X2=X)
all(K==t(K1))

# IBS kernel for ternary data
X <- as.matrix(expand.grid(0:2,0:2))
K = getK(X, kernel = 'ibs')

# add weight
w = runif(ncol(X))
K = getK(X, kernel = 'ibs', para = w)

# IBS kernel for binary data via option 'h' for 'hamming similarity measure'
X <- as.matrix(expand.grid(0:1,0:1))
K=getK(X, kernel = 'h')
```

Description

Utility functions by Youyi Fong and Krisz Sebestyen, and some functions copied from other packages for convenience (acknowledged on their manual pages).

Most useful functions: mypostscript/mypdf, mytex,

See the Index link below for a list of available functions.

`math.functions`*Math Functions*

Description

H calculates entropy.

Usage

```
as.binary(n, base = 2, r = FALSE)
```

```
binom.coef(n, m)
```

```
expit(x)
```

```
logDiffExp(logx1, logx2)
```

```
logit(x)
```

```
logMeanExp(logx, B = NULL)
```

```
logSumExp(logx)
```

```
logSumExpFor2(logx, logy)
```

```
permn(x, fun = NULL, ...)
```

```
Stirling2(n, m)
```

Arguments

n

base

r

m

x

logx1

logx2
logx
B
logy
fun
...

Examples

```
H(rep(1/5,5))  
H(rep(3,5))
```

matrix.array.functions

Matrix and Array Functions

Description

concatList returns a string that concatenates the elements of the input list or array

Usage

```
AR1(p, w)  
  
concatList(lis, sep = "")  
  
EXCH(p, rho)  
  
fill.jagged.array(a)  
  
getMidPoints(x)  
  
getUpperRight(matri, func = NULL)  
  
last(x, n = 1, ...)  
  
mix(a, b)  
  
## S3 method for class 'data.frame'  
rep(x, times = 1, ...)  
  
## S3 method for class 'matrix'  
rep(x, times = 1, each = 1, by.row = TRUE, ...)  
  
## S3 method for class 'matrix.block'
```

```
rep(x, times = 2, ...)  
shift.left(x, k = 1)  
shift.right(x, k = 1)  
thin.rows(dat, thin.factor = 10)  
ThinRows(dat, thin.factor = 10)  
tr(m)
```

Arguments

p
w
lis list or array
sep
rho
a
x
matri
func
n
...
b
times
each
by.row
k
dat
thin.factor
m

Examples

```
concatList(1:3, "_")
```


Description

DXD computes $D \% \% X \% \% D$, where D is a diagonal matrix. symprod computes $S \% \% X$ for symmetric S . txSy computes $t(x) \% \% S \% \% y$ for symmetric S .

Usage

```
DXD(d1, X, d2)

symprod(S, X)

txSy(x, S, y)

.as.double(x, stripAttributes = FALSE)
```

Arguments

d1	a diagonal matrix or an array
d2	a diagonal matrix or an array
x	array
y	array
S	symmetric matrix
X	matix
stripAttributes	boolean

Details

.as.double does not copying whereas as.double(x) for older versions of R when using .C(DUP = FALSE) make duplicate copy of x. In addition, even if x is a 'double', since x has attributes (dim(x)) as.double(x) duplicates

The functions do not check whether S is symmetric. If it is not symmetric, then the result will be wrong. DXD offers a big gain, while symprod and txSy gains are more incremental.

Author(s)

Krisztian Sebestyen

Examples

```
d1=1:3
d2=4:6
X=matrix(1:9,3,3)
all(DXD(d1, X, d2) == diag(d1) %** X %** diag(d2))
```

```
S=matrix(c(1,2,3,2,4,5,3,5,8),3,3)
X=matrix(1:9,3,3)
all( symprod(S, X) == S %** X )
```

```
x=1:3
y=4:6
S=matrix(c(1,2,3,2,4,5,3,5,8),3,3)
txSy(x, S, y) == drop(t(x)%**S%**y)
```

plotting

Plotting Functions

Description

mypostscript and mypdf sets the width and height based on mfrow input.

Usage

```
abline.pt.slope(pt1, slope, ...)
```

```
abline.pts(pt1, pt2 = NULL)
```

```
butterfly.plot(dat, dat2 = NULL, add = FALSE, xaxislabels = rep("", 4), x.ori = 0,
  xlab = "", ylab = "", cex.axis = 1, ...)
```

```
empty.plot()
```

```
getMfrow(len)
```

```
myforestplot(dat, xlim = NULL, xlab = "", main = "", col.1 = "red", col.2 = "blue",
  plot.labels=TRUE,order=FALSE,decreasing=FALSE)
```

```
my.interaction.plot(dat, x.ori = 0, xaxislabels = rep("", 2), cex.axis = 1, add = FALSE,
  xlab = "", ylab = "", pcol = NULL, lcol = NULL, ...)
```

```
myboxplot(object, ...)  
  
## S3 method for class 'formula'  
myboxplot(formula, data, cex = 0.5, ylab = "", xlab = "", main = "",  
           box = TRUE, at = NULL, pch = 1, col = 1, test = c("t", "w", "f", "k"),  
           reshape.formula=NULL, ...)  
  
## S3 method for class 'list'  
myboxplot(object, cex = 0.5, ylab = "", xlab = "", main = "", box = TRUE,  
           at = NULL, pch = 1, col = 1, test = c("t", "w"), ...)  
  
mylegend(legend, x, lty = NULL, bty = "n", ...)  
  
mymatplot(x, y, make.legend = TRUE, legend = NULL, legend.x = 9, lty = 1:5, pch = NULL,  
           col = 1:6, legend.title = NULL, legend.cex = 1, xlab = NULL, ylab = "",  
           draw.x.axis = TRUE, bg = NA, legend.inset=0, lwd=1, at=NULL, ...)  
  
mypairs(dat, method = "pearson")  
  
mypdf(...)  
  
mypostscript(file = "temp", mfrow = c(1, 1), mfcol = NULL, width = NULL, height = NULL,  
             ext = "eps", oma = NULL, mar = NULL, main.outer = FALSE, save2file = TRUE, ...)  
  
panel.cor(x, y, digits = 2, prefix = "", cex.cor, method = "pearson", ...)  
  
panel.hist(x, ...)  
  
panel.nothing(x, ...)  
  
corplot(object, ...)  
  
## Default S3 method:  
corplot(object, y, ...)  
  
## S3 method for class 'formula'  
corplot(formula, data, main = "", method = c("pearson", "spearman"),  
         col = 1, cex = 0.5, add.diagonal.line = TRUE, add.lm.fit=FALSE,add=FALSE,log="",  
         same.xyylim=FALSE, xlim=NULL,ylim=NULL,...)
```

Arguments

```
pt1  
slope  
...  
pt2  
dat
```

lwd line width.
legend.inset legend inset
dat2
add
log
add.lm.fit
reshape.formula
 a formula object.
xaxislabels
x.ori
xlab
ylab
cex.axis
len
same.xyylim Boolean. Whether xlim and ylim should be the same
xlim
ylim
main
col.1
col.2
pcol
lcol
object
formula
data
cex
box
at
pch
col
test
legend
x
lty
bty
make.legend
legend.x
legend.title

```
legend.cex
draw.x.axis
bg
method
file
mfrow
mfcol
width
height
ext
oma
mar
main.outer
save2file
y
digits
prefix
cex.cor
plot.labels    Boolean
order          Boolean
decreasing     Boolean
add.diagonal.line
```

Details

myboxplot shows data points along with boxes. The data points are jittered and the pattern of jittering is made reproducible in repeated calls.

myforestplot is modified from code from Allan deCamp/SCHARP. dat should have three columns. first column should be point estimate, second and third lci and uci, fourth p value. col.1 is the color used for CIs that do not include null, col.2 is used for CIs that do include null. If order is TRUE, the rows are ordered by the first column of dat. decreasing can be used to change the behavior of order.

Examples

```
mypdf(mfrow=c(1,2))
  plot(1:10)
  plot(1:10)
dev.off()
```

print.functions	<i>Print Functions</i>
-----------------	------------------------

Description

roundup prints a specified number of digits after decimal point even if 0s are needed at the end.
format.int prints a specified number of digits before decimal point even if 0s are needed at the beginning.

Usage

```
## S3 method for class 'int'  
format(x, digits, fill = "0", ...)  
  
make.latex.coef.table(models, model.names = NULL, row.major = FALSE, round.digits = NULL)  
  
mytex(dat = NULL, file.name = "temp", digits = NULL, display = NULL, align = "r",  
      append = FALSE, preamble = "", include.rownames = TRUE, include.dup.rownames = FALSE,  
      floating = FALSE, lines = TRUE, ...)  
  
mytex.begin(file.name, preamble = "")  
  
mytex.end(file.name)  
  
mywrite(x, ...)  
  
mywrite.csv(x, file = "tmp", row.names = FALSE, digits = NULL, ...)  
  
roundup(value, digits)
```

Arguments

value
digits
fill
models
model.names
row.major
round.digits
dat
file.name
display
align

```
append
preamble
include.rownames

include.dup.rownames

floating
lines
...
x
file
row.names
```

Examples

```
roundup (3.1, 2) # 3.10
format.int(3, 2) # 03
```

random.functions	<i>Random Functions</i>
------------------	-------------------------

Description

rbern generates Bernoulli random variables.

Usage

```
dbern(x, prob, log = FALSE)
dcorbern(x, p, a, log = FALSE)
dmixnorm(x, mix.p, sd1, sd2, log = FALSE)
dnorm.norm.gamma(x, p, same.distr = FALSE, log = FALSE)
rbern(n, prob, generalized = FALSE)
rejective.sampling(N, n, pik)
rnorm.cor(n, mu, sd, alpha)
rnorm.norm.gamma(n, mu.0, lambda, alpha, beta)
```

Arguments

x
prob
log
p
a
mix.p
sd1
sd2
same.distr
n
generalized
N
pik
mu
sd
alpha
mu.0
lambda
beta

Examples

```
set.seed(1)
rbern(n=10, p=1/2)
rbern(n=2, p=c(.999, .001))
```

regression.model.functions

Regression Model Functions

Description

getFormattedSummary prints a table of regression coefficient estimates and standard errors.

Usage

```
getFormattedSummary(fits, type = 1, est.digits = 2, se.digits = 2, random = FALSE)

getVarComponent(object, ...)

getFixedEf(object, ...)

## S3 method for class 'coxph'
getFixedEf(object, ...)

## S3 method for class 'gam'
getFixedEf(object, ...)

## S3 method for class 'gee'
getFixedEf(object, ...)

## S3 method for class 'geese'
getFixedEf(object, ...)

## S3 method for class 'glm'
getFixedEf(object, exp = FALSE, ...)

## S3 method for class 'inla'
getFixedEf(object, ...)

## S3 method for class 'lm'
getFixedEf(object, ...)

## S3 method for class 'lme'
getFixedEf(object, ...)

## S3 method for class 'logistf'
getFixedEf(object, exp = FALSE, ...)

## S3 method for class 'matrix'
getFixedEf(object, ...)

## S3 method for class 'MIresult'
getFixedEf(object, ...)

## S3 method for class 'hyperpar.inla'
getVarComponent(object, transformation = NULL, ...)

## S3 method for class 'matrix'
getVarComponent(object, ...)

interaction.table(fit, v1, v2, v1.type = "continuous", v2.type = "continuous",
```

```

    logistic.regression = TRUE)

## S3 method for class 'geese'
coef(object, ...)

## S3 method for class 'geese'
predict(object, x, ...)

## S3 method for class 'geese'
residuals(object, y, x,...)

## S3 method for class 'geese'
vcov(object, ...)

## S3 method for class 'logistf'
vcov(object, ...)

```

Arguments

```

...
object
fit
exp
fits
type
est.digits
se.digits
random
transformation
v1
v2
v1.type
v2.type
logistic.regression

x
y

```

Details

getFormattedSummary: from a list of fits, say lmer, inla fits, return formatted summary controlled by "type". For a matrix, return Monte Carlo variance random=TRUE returns variance components type=1: est type=2: est (se) type=3: est (2.5 percent, 97.5 percent) type=4: est se

getFixedEf returns a matrix, first column coef, second column se,

getFixedEf.matrix used to get mean and sd from a jags or winbugs sample, getVarComponent.matrix and getFixedEf.matrix do the same thing. Each column of samples is a variable

Examples

```
## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
## Page 9: Plant Weight Data.
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
glm.D9 <- glm(weight ~ group)
getFormattedSummary (list(lm.D9, glm.D9))
```

stat.functions

Stat Functions

Description

H calculates entropy.

Usage

```
H(p, logbase = c("e", "2"))
mutual.info(two.way.table, logbase = c("e", "2"))
cor.mixed(x, ...)
cor.mixed.default(x, na.fun, method = c("pearson", "spearman"))
cor.mixed.vector(x, y, na.fun, method = c("pearson", "spearman"))
cor.mixed.formula(formula, data, na.fun, method = c("pearson", "spearman"))
info.cor(two.way.table)
yule.y(two.by.two.matrix)
kappa.cor(two.by.two.matrix, weight = c(1, 1), maximum = FALSE)
l.measure(two.by.two.matrix)
```

Arguments

p either a count vector or a probability vector, but can not be a vector of membership indicator

logbase
two.way.table
x
...
na.fun
method
y
formula
data
two.by.two.matrix

weight
maximum

Examples

```
H(rep(1/5,5))  
H(rep(3,5))
```

string.functions *String Functions*

Description

`%+%` concatenates its arguments and returns a string.

Usage

```
a %+% b  
  
contain(s1, s2)  
  
endsWith(s1, s2)  
  
escapeUnderline(name)  
  
fileStem(file.name)  
  
firstIndex(s1, s2)  
  
getExt(file.name)  
  
getFileStem(file.name)
```

```
getStem(file.name)

lastIndex(s1, s2)

myprint(object, ...)

## Default S3 method:
myprint(..., newline = TRUE, digits = 3)

remove.prefix(s, sep = "_")

startsWith(s1, s2)
```

Arguments

```
a
b
s1
s2
name
file.name
object
...
newline
digits
s
sep
```

Examples

```
x=1
x %+% "b" %+% "c"
```

testing.functions

Testing Functions

Description

Testing functions.

Usage

```
hosmerlem(y, yhat, g = 10)

quick.t.test(x, y, var.equal = FALSE)

## S3 method for class 'test'
sign(x)

tukey.mtest(mu, ms, n)

vector.t.test(mean.x, mean.y, var.x, var.y, n)
```

Arguments

```
y
yhat
g
x
var.equal
mu
ms
n
mean.x
mean.y
var.x
var.y
```

Examples

```
sign.test(runif(10))
```

Index

.as.double (matrix2), 9
%+(string.functions), 20

abline.pt.slope (plotting), 10
abline.pts (plotting), 10
AR1 (matrix.array.functions), 7
array.functions
 (matrix.array.functions), 7
as.binary (math.functions), 6

base.functions, 2
binary (base.functions), 2
binary2 (base.functions), 2
binom.coef (math.functions), 6
butterfly.plot (plotting), 10

coef.geese
 (regression.model.functions),
 16
concatList (matrix.array.functions), 7
contain (string.functions), 20
cor.mixed (stat.functions), 19
coplot (plotting), 10

dbern (random.functions), 15
dcorbern (random.functions), 15
dmixnorm (random.functions), 15
dnorm.norm.gamma (random.functions), 15
DXD (matrix2), 9

empty.plot (plotting), 10
endsWith (string.functions), 20
escapeUnderline (string.functions), 20
EXCH (matrix.array.functions), 7
expit (math.functions), 6

f2c (base.functions), 2
fileStem (string.functions), 20
fill.jagged.array
 (matrix.array.functions), 7
firstIndex (string.functions), 20

format.int (print.functions), 14
ftoi (base.functions), 2

getExt (string.functions), 20
getFileStem (string.functions), 20
getFixedEf
 (regression.model.functions),
 16
getFixedEf2
 (regression.model.functions),
 16
getFormattedSummary
 (regression.model.functions),
 16
getK, 4
getMfrow (plotting), 10
getMidPoints (matrix.array.functions), 7
getStem (string.functions), 20
getUpperRight (matrix.array.functions),
 7
getVarComponent
 (regression.model.functions),
 16

H (stat.functions), 19
hosmerlem (testing.functions), 21

info.cor (stat.functions), 19
interaction.table
 (regression.model.functions),
 16

kappa.cor (stat.functions), 19
keepWarnings (base.functions), 2
kyotil, 5

l.measure (stat.functions), 19
last (matrix.array.functions), 7
lastIndex (string.functions), 20
logDiffExp (math.functions), 6
logit (math.functions), 6

- logMeanExp (math.functions), 6
- logSumExp (math.functions), 6
- logSumExpFor2 (math.functions), 6
- make.latex.coef.table
 - (print.functions), 14
- math.functions, 6
- matrix.array.functions, 7
- matrix.functions
 - (matrix.array.functions), 7
- matrix2, 9
- meanmed (base.functions), 2
- methods4 (base.functions), 2
- mix (matrix.array.functions), 7
- mutual.info (stat.functions), 19
- my.interaction.plot (plotting), 10
- myaggregate (base.functions), 2
- myboxplot (plotting), 10
- myforestplot (plotting), 10
- mylegend (plotting), 10
- mymatplot (plotting), 10
- mypairs (plotting), 10
- mypdf (plotting), 10
- mypostscript (plotting), 10
- myprint (string.functions), 20
- myreshapewide (base.functions), 2
- mysapply (base.functions), 2
- myscale (base.functions), 2
- mysystem (base.functions), 2
- mytapply (base.functions), 2
- mytex (print.functions), 14
- mywrite (print.functions), 14
- panel.cor (plotting), 10
- panel.hist (plotting), 10
- panel.nothing (plotting), 10
- permn (math.functions), 6
- plotting, 10
- predict.geese
 - (regression.model.functions), 16
- print.functions, 14
- quick.t.test (testing.functions), 21
- random.functions, 15
- rbern (random.functions), 15
- read.csv (base.functions), 2
- read.tsv (base.functions), 2
- regression.model.functions, 16
- rejective.sampling (random.functions), 15
- remove.prefix (string.functions), 20
- rep.data.frame
 - (matrix.array.functions), 7
- rep.matrix (matrix.array.functions), 7
- residuals.geese
 - (regression.model.functions), 16
- rnorm.cor (random.functions), 15
- rnorm.norm.gamma (random.functions), 15
- roundup (print.functions), 14
- shift.left (matrix.array.functions), 7
- shift.right (matrix.array.functions), 7
- sign.test (testing.functions), 21
- startsWith (string.functions), 20
- stat.functions, 19
- Stirling2 (math.functions), 6
- string.functions, 20
- symprod (matrix2), 9
- table.prop (base.functions), 2
- testing.functions, 21
- thin.rows (matrix.array.functions), 7
- ThinRows (matrix.array.functions), 7
- tr (matrix.array.functions), 7
- tukey.mtest (testing.functions), 21
- txSy (matrix2), 9
- unix (base.functions), 2
- vcov.geese
 - (regression.model.functions), 16
- vcov.logistf
 - (regression.model.functions), 16
- vector.t.test (testing.functions), 21
- yule.y (stat.functions), 19