# Package 'miscFuncs'

March 22, 2024

**Maintainer** Benjamin M. Taylor <benjamin.taylor.software@gmail.com>

**License** GPL-3

**Title** Miscellaneous Useful Functions Including LaTeX Tables, Kalman Filtering, QQplots with Simulation-Based Confidence Intervals and Development Tools

**Type** Package

**LazyLoad** yes

**Author** Benjamin M. Taylor

**Description** Implementing various things including functions for LaTeX tables, the Kalman filter, QQ-plots with simulation-based confidence intervals, web scraping, development tools, relative risk and odds rati, GARCH(1,1) Forecasting.

**Version** 1.5-8

**Date** 2024-03-21

**Depends** roxygen2, mvtnorm, extraDistr

**Imports** stats

**Suggests** bayesGARCH

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-03-22 15:30:03 UTC

## R topics documented:

---

`.onAttach` *.onAttach function*

---

## Description

A function to print a welcome message on loading package

## Usage

```
.onAttach(libname, pkgname)
```

### Arguments

| | |
|---|---|
| libname | libname argument |
| pkgname | pkgname argument |

### Value

...

---

## bin *bin function*

---

### Description

A function to convert decimal to binary

### Usage

```
bin(n)
```

### Arguments

| | |
|---|---|
| n | a non-negative integer |

### Value

the binary representation stored in a vector.

---

## colour_legend *colour_legend function*

---

### Description

A function to

### Usage

```
colour_legend(palette, suffix = "", dir = ".")
```

### Arguments

| | |
|---|---|
| palette | X |
| suffix | X |
| dir | X |

### Value

...

`cor_taylor` *cor_taylor function*

## Description

A function to compute Taylor's correlation coefficient ;-)

## Usage

```
cor_taylor(X)
```

## Arguments

| | |
|---|---|
| `X` | a numeric matrix with number of rows bigger than the number of columns |

## Value

Taylor's correlation coefficient, a number between 0 and 1 expressing the amount of dependence between multiple variables.

`cospulse` *cospulse function*

## Description

A function to

## Usage

```
cospulse(x, tau = pi)
```

## Arguments

| | |
|---|---|
| `x` | X |
| `tau` | pulse duration |

## Value

...

## cosrsaw *cosrsaw function*

### Description

A function to

### Usage

```
cosrsaw(x)
```

### Arguments

| | |
|---|---|
| x | X |

### Value

...

## cossaw *cossaw function*

### Description

A function to

### Usage

```
cossaw(x)
```

### Arguments

| | |
|---|---|
| x | X |

### Value

...

## costri *costri function*

### Description

A function to

### Usage

```
costri(x)
```

### Arguments

| | |
|---|---|
| x | X |

### Value

...

## daynames *daynames function*

### Description

A function to

### Usage

```
daynames()
```

### Value

...

EKFadvance *EKFadvance function*

## Description

A function to perform one iteration of ther EKF. Currently UNDER DEVELOPMENT.

## Usage

```
EKFadvance(
  obs,
  oldmean,
  oldvar,
  phi,
  phi.arglist,
  psi,
  psi.arglist,
  W,
  V,
  loglik = FALSE,
  na.rm = FALSE
)
```

## Arguments

| | |
|---|---|
| obs | observations |
| oldmean | old mean |
| oldvar | old variance |
| phi | Function computing a Taylor Series approximation of the system equation. Can include higher (ie 2nd order and above) terms. |
| phi.arglist | arguments for function phi |
| psi | Function computing a Taylor Series approximation of the observation equation. Can include higher (ie 2nd order and above) terms. |
| psi.arglist | arguments for function psi |
| W | system noise matrix |
| V | observation noise matrix |
| loglik | whether or not to compute the pseudo-likelihood |
| na.rm | logical, whether or not to handle NAs. Defult is FALSE. Set to TRUE if there are any missing values in the observed data. |

## Value

list containing the new mean and variance, and if specified, the likelihood

## fcastGARCH *fcastGARCH function*

### Description

A function to forecast forwards using MCMC samples from the bayesGARCH function from the bayesGARCH package.

### Usage

```
fcastGARCH(y, parmat, l)
```

### Arguments

| | |
|---|---|
| y | vector of log-returns used in fitting the model via bayesGARCH |
| parmat | a matrix of MCMC samples from the bayesGARCH function e.g. "out$chain1" where "out" is the output of the fitted model and "chain1" is the desired chain |
| l | number of lags to forecast forward |

### Details

Suggest thinning MCMC samples to get, say 1000, posterior samples (this can be done post-hoc)

See also the function lr2fact for converting log-returns to a factor. Apply this to the output of fcastGARCH in order to undertake forecasting on the scale of the original series (i.e. not the log returns). Quantiles may be computed across the MCMC iterations and then all one needs to do is to multiply the result by the last observed value in the original series (again, not the log returns)

### Value

forcast log returns and also forecast y

## generic *generic function*

### Description

A function to generate roxygen templates for generic funtions and associated methods.

### Usage

```
generic(gen, methods = NULL, sp = 3, oname = "obj")
```

### Arguments

| | |
|---|---|
| gen | character string giving the name of an S3 generic. |
| methods | character vector: a list of methods for which to provide templates |
| sp | the amont of space to put in between functions |
| oname | name of the generic object |

### Value

roxygen text printed to the console.

---

## genharmonic *genharmonic function*

---

### Description

A function to create harmonic terms ready for a harmonic regression model to be fitted.

### Usage

```
genharmonic(
  df,
  tname,
  base,
  num,
  sinfun = sin,
  cosfun = cos,
  sname = "s",
  cname = "c",
  power = FALSE
)
```

### Arguments

| | |
|---|---|
| df | a data frame |
| tname | a character string, the name of the time variable. Note this variable will be converted using the function as.numeric |
| base | the period of the first harmonic e.g. for harmonics at the sub-weekly level, one might set base=7 if time is measured in days |
| num | the number of harmonic terms to return |
| sinfun | function to compute sin-like components in model. Default is sin, but alternatives include sintri, or any other periodic function defined on [0,2pi] |
| cosfun | function to compute sin-like components in model. Default is cos, but alternatives include costri, or any other periodic function defined on [0,2pi] offset to sinfun by pi/2 |

| | |
|---|---|
| sname | the prefix of the sin terms, default 's' returns variables 's1', 's2', 's3' etc. |
| cname | the prefix of the cos terms, default 's' returns variables 's1', 's2', 's3' etc. |
| power | logical, if FALSE (the default) it will return the standard Fourier series with sub-harmonics at 1, 1/2, 1/3, 1/4 of the base periodicicy. If TRUE, a power series will be used instead, with harmonics 1, 1/2, 1/4, 1/8 etc. of the base frequency. |

## Value

a data frame with the time variable in numeric form and the harmonic components

---

`genIntegratedharmonic` *genIntegratedharmonic function*

---

## Description

A function to generate basis vectors for integrated Fourier series.

## Usage

```
genIntegratedharmonic(
  df,
  t1name,
  t2name,
  base,
  num,
  sname = "bcoef",
  cname = "acoef",
  power = FALSE
)
```

## Arguments

| | |
|---|---|
| df | a data frame containing a numeric time variable of interest |
| t1name | a character string, the name of the variable in df containing the start time of the intervals |
| t2name | a character string, the name of the variable in df containing the end time of the intervals |
| base | the fundamental period of the signal, e.g. if it repeats over 24 hours and time is measured in hours, then put 'base = 24'; if the period is 24 hours but time is measured in days, then use 'base = 1/7' |
| num | number of sin and cosine terms to compute |
| sname | character string, name for cosine terms in Fourier series (not integrated) |
| cname | character string, name for sine terms in Fourier series (not integrated) |
| power | legacy functionality, not used here |

## Details

If the non-integrated Fourier series is:

f(t) = sum_k a_k sin(2 pi k t / P) + b_k cos(2 pi k t / P)

then
int_t1^t2 f(s) ds = sum_k a_k (base/(2 pi k))*(cos(2 pi k t1 / P) - cos(2 pi k t2 / P)) +
b_k (base/(2 pi k))*(sin(2 pi k t2 / P)-sin(2 pi k t1 / P))
where P is the funcamental period, or 'base', as referred to in the function arguments

## Value

a data frame containing the start and end time vectors, together with the sin and cosine terms

---

`getstrbetween` *getstrbetween function*

---

## Description

A function used in web scraping. Used to simplify the searching of HTML strings for information.

## Usage

```
getstrbetween(linedata, start, startmark, endmark, include = FALSE)
```

## Arguments

| | |
|---|---|
| `linedata` | a string |
| `start` | integer, where to start looking in linedata |
| `startmark` | character string. a pattern identifying the start mark |
| `endmark` | character string. a pattern identifying the end mark |
| `include` | include the start and end marks? |

## Value

the first string after start and between the start and end marks

## getwikicoords *getwikicoords function*

### Description

A function to return the lat/lon coordinates of towns in the UK from Wikipedia. Does not always work. Sometimes the county has to be specified too.

### Usage

```
getwikicoords(place, county = NULL, rmslash = TRUE)
```

### Arguments

| | |
|---|---|
| place | character, ther name of the town |
| county | character, the county it is in |
| rmslash | remove slash from place name. Not normally used. |

### Value

The lat/lon coordinates from Wikipedia

## hCreate *hCreate function*

### Description

A function used in the forecasting of GARCH(1,1) models

### Usage

```
hCreate(pars, y, T = length(y))
```

### Arguments

| | |
|---|---|
| pars | parameters for the GARCH model, these would come from an MCMC run |
| y | vector of log returns |
| T | this is the length of y; allow this to be pre-computed |

### Value

vector of h's

KFadvance *KFadvance function*

## Description

A function to compute one step of the Kalman filter. Embed in a loop to run the filter on a set of data.

## Usage

```
KFadvance(
  obs,
  oldmean,
  oldvar,
  A,
  B,
  C,
  D,
  E,
  F,
  W,
  V,
  marglik = FALSE,
  log = TRUE,
  na.rm = FALSE
)
```

## Arguments

| | |
|---|---|
| obs | Y(t) |
| oldmean | mu(t-1) |
| oldvar | Sigma(t-1) |
| A | matrix A |
| B | column vector B |
| C | matrix C |
| D | matrix D |
| E | column vector E |
| F | matrix F |
| W | state noise covariance |
| V | observation noise covariance |
| marglik | logical, whether to return the marginal likelihood contribution from this observation |
| log | whether or not to return the log of the likelihood contribution. |
| na.rm | na.rm logical, whether or not to handle NAs. Defult is FALSE. Set to TRUE if there are any missing values in the observed data. |

## Details

The model is: (note that Y and theta are COLUMN VECTORS)

theta(t) = A*theta(t-1) + B + C*W (state equation)

Y(t) = D*theta(t) + E + F*V (observation equation)

W and V are the covariance matrices of the state and observation noise. Prior is normal,

N(mu(t-1),Sigma(t-1))

Result is the posterior, N(mu(t),Sigma(t)), together with the likelihood contribution Prob(Y(t)|Y(t-1))

## Value

list containing the new mean and variance, and if specified, the likelihood

---

`KFadvanceAR2` *KFadvanceAR2 function*

---

## Description

A function to compute one step of the Kalman filter with second order AR state evolution. Embed in a loop to run the filter on a set of data.

## Usage

```
KFadvanceAR2(
  obs,
  oldmean,
  oldermean,
  oldvar,
  oldervar,
  A,
  A1,
  B,
  C,
  D,
  E,
  F,
  W,
  V,
  marglik = FALSE,
  log = TRUE,
  na.rm = FALSE
)
```

## Arguments

| | |
|---|---|
| obs | Y(t) |
| oldmean | mu(t-1) |
| oldermean | mu(t-2) |
| oldvar | Sigma(t-1) |
| oldervar | Sigma(t-2) |
| A | A matrix A |
| A1 | A matrix A1 |
| B | column vector B |
| C | matrix C |
| D | matrix D |
| E | column vector E |
| F | matrix F |
| W | state noise covariance |
| V | observation noise covariance |
| marglik | logical, whether to return the marginal likelihood contribution from this observation |
| log | whether or not to return the log of the likelihood contribution. |
| na.rm | na.rm logical, whether or not to handle NAs. Defult is FALSE. Set to TRUE if there are any missing values in the observed data. |

## Details

The model is: (note that Y and theta are COLUMN VECTORS)

theta(t) = A*theta(t-1) + A1*theta(t-2) + B + C*W (state equation)

Y(t) = D*theta(t) + E + F*V (observation equation)

W and V are the covariance matrices of the state and observation noise. Priors are normal,

N(mu(t-1),Sigma(t-1)) and N(mu(t-2),Sigma(t-2))

Result is the posterior, N(mu(t),Sigma(t)), together with the likelihood contribution Prob(Y(t)|Y(t-1))

## Value

list containing the new mean and variance, and if specified, the likelihood

## KFtemplates *KFtemplates function*

### Description

A function to print KFfit and KFparest templates to the console. See vignette("miscFuncs") for more information

### Usage

```
KFtemplates()
```

### Value

Tust prints to the console. This can be copied and pasted into a text editor for further manipulation.

## latexformat *latexformat function*

### Description

A function to format text or numeric variables using scientific notation for LaTeX documents.

### Usage

```
latexformat(x, digits = 3, scientific = -3, ...)
```

### Arguments

| | |
|---|---|
| x | a numeric, or character |
| digits | see ?format |
| scientific | see ?format |
| ... | other arguments to pass to the function format |

### Value

...

`latextable` *latextable function*

## Description

A very useful function to create a LaTeX table from a matrix. Rounds numeric entries and also replaces small numbers with standard index form equivalents.

## Usage

```
latextable(
  x,
  digits = 3,
  scientific = -3,
  colnames = NULL,
  rownames = NULL,
  caption = NULL,
  narep = " ",
  laststr = "",
  intable = TRUE,
  manualalign = NULL,
  file = "",
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | a matrix, or object that can be coerced to a matrix. x can include mixed character and numeric entries. |
| `digits` | see help file for format |
| `scientific` | see help file for format |
| `colnames` | optional column names set to NULL (default) to automatically use column names of x. NOTE! if rownames is not NULL present, colnames must include an entry for the rownames i.e. it should be a vector of length the number of columns of x plus 1. |
| `rownames` | optional row names set to NULL (default) to automatically use row names of x |
| `caption` | optional caption, not normally used |
| `narep` | string giving replacement for NA entries in the matrix |
| `laststr` | string to write at end, eg note the double backslash!! |
| `intable` | output in a table environment? |
| `manualalign` | manual align string e.g. 'ccc' or 'llccc' |
| `file` | connection to write to, default is " which writes to the console; see ?write for further details |
| `...` | additional arguments passed to format |

## Details

To get a backslash to appear, use a double backslash

Just copy and paste the results into your LaTeX document.

## Value

prints the LaTeX table to screen, so it can be copied into reports

## Examples

```
latextable(as.data.frame(matrix(1:4,2,2)))
```

---

`lr2fact` *lr2fact function*

---

## Description

Apply this to the output of fcastGARCH in order to undertake forecasting on the scale of the original series (i.e. not the log returns). Quantiles may be computed across the MCMC iterations and then all one needs to do is to multiply the result by the last observed value in the original series (again, not the log returns)

## Usage

```
lr2fact(mod)
```

## Arguments

| | |
|---|---|
| `mod` | the output of fcastGARCH |

## Value

the multiplicative factors.

## method *method function*

### Description

A function to generate a roxygen template for a method of a generic S3 function. Normally, this would be called from the function generic, see ?generic

### Usage

```
method(meth, gen, oname = "obj")
```

### Arguments

| | |
|---|---|
| meth | character, the name of the method |
| gen | character the associated generic method |
| oname | name of object |

### Value

a roxygen template for the method.

## monthnames *monthnames function*

### Description

A function to

### Usage

```
monthnames()
```

### Value

...

## print22 *print22 function*

### Description

A function to print details of the 2 by 2 table for use with the function twotwoinfo.

### Usage

```
print22()
```

### Value

prints the names of the arguments of twotwofunction info to screen in their correct place in the 2 by 2 table

### See Also

twotwoinfo

## qqci *qqci function*

### Description

A function to compare quantiles of a given vector against quantiles of a specified distribution. The function outputs simulation-based confidence intervals too. The option of zero-ing the plot (rather than visualising a diagonal line (which can be difficult to interpret) and also standardising (so that varying uncertainty around each quantile appears equal to the eye) are also given.

### Usage

```
qqci(
  x,
  rfun = NULL,
  y = NULL,
  ns = 100,
  zero = FALSE,
  standardise = FALSE,
  qts = c(0.025, 0.975),
  llwd = 2,
  lcol = "red",
  xlab = "Theoretical",
  ylab = "Sample",
  alpha = 0.02,
  cicol = "black",
```

```
  cilwd = 1,
  ...
)
```

## Arguments

| | |
|---|---|
| x | a vector of values to compare |
| rfun | a function accepting a single argument to generate samples from the comparison distribution, the default is rnorm |
| y | an optional vector of samples to compare the quantiles against. In the case this is non-null, the function rfun will be automatically chosen as bootstrapping y with replacement and sample zise the same as the length of x. You must specify exactly one of rfun or y. |
| ns | the number of simulations to generate: the more simulations, the more accurate the confidence bands. Default is 100 |
| zero | logical, whether to zero the plot across the x-axis. Default is FALSE |
| standardise | logical, whether to standardise so that the variance around each quantile is made constant (this can help in situations where the confidence bands appear very tight in places) |
| qts | vector of probabilities giving which sample-based empirical quantiles to add to the plot. Default is c(0.025,0.975) |
| llwd | positive numeric, the width of line to plot, default is 2 |
| lcol | colour of line to plot, default is red |
| xlab | character, the label for the x-axis |
| ylab | character, the label for the y-axis |
| alpha | controls transparency of samples (coloured blue) |
| cicol | colour of confidence band lines, default is black |
| cilwd | width of confidence band lines, default is 1 |
| ... | additional arguments to pass to matplot |

## Value

Produces a QQ-plot with simulation-based confidence bands

## Examples

```
qqci(rnorm(1000))
qqci(rnorm(1000),zero=TRUE)
qqci(rnorm(1000),zero=TRUE,standardise=TRUE)
```

## roxbc *roxbc function*

### Description

A function to build and check packages where documentation has been compiled with roxygen. Probably only works in Linux.

### Usage

```
roxbc(name, checkflags = "--as-cran")
```

### Arguments

| | |
|---|---|
| name | package name |
| checkflags | string giving optional check flags to R CMD check, default is –as-cran |

### Value

builds and checks the package

## roxbuild *roxbuild function*

### Description

A function to build packages where documentation has been compiled with roxygen. Probably only works in Linux.

### Usage

```
roxbuild(name)
```

### Arguments

| | |
|---|---|
| name | package name |

### Value

builds and checks the package

`roxtext` *roxtext function*

## Description

A function to generate roxygen documentation templates for functions for example,

## Usage

```
roxtext(fname)
```

## Arguments

| | |
|---|---|
| `fname` | the name of a function as a character string or as a direct reference to the function |

## Details

would generate a template for this function. Note that functions with default arguments that include quotes will throw up an error at the moment, just delete these bits from the string, and if shold work.

## Value

minimal roxygen template

`sinpulse` *sinpulse function*

## Description

A function to

## Usage

```
sinpulse(x, tau = pi)
```

## Arguments

| | |
|---|---|
| `x` | X |
| `tau` | pulse duration |

## Value

...

## sinrsaw *sinrsaw function*

### Description

A function to

### Usage

```
sinrsaw(x)
```

### Arguments

| | |
|---|---|
| x | X |

### Value

...

## sinsaw *sinsaw function*

### Description

A function to

### Usage

```
sinsaw(x)
```

### Arguments

| | |
|---|---|
| x | X |

### Value

...

## sintri *sintri function*

### Description

A function to

### Usage

```
sintri(x)
```

### Arguments

| | |
|---|---|
| x | X |

### Value

...

## timeop *timeop function*

### Description

A function to time an operation in R

### Usage

```
timeop(expr)
```

### Arguments

| | |
|---|---|
| expr | an expression to evaluate |

### Value

The time it took to evaluate the expression in seconds

`twotwoinfo` *twotwoinfo function*

## Description

A function to compute and diplay information about 2 by 2 tables for copying into LaTeX documents. Computes odds ratios and relative risks together with confidence intervals for 2 by 2 table and prints to screen in LaTeX format. The funciton will try to fill in any missing values from the 2 by 2 table. Type print22() at the console to see what each argument refers to.

## Usage

```
twotwoinfo(
  e1 = NA,
  u1 = NA,
  o1t = NA,
  e2 = NA,
  u2 = NA,
  o2t = NA,
  et = NA,
  ut = NA,
  T = NA,
  lev = 0.95,
  LaTeX = TRUE,
  digits = 3,
  scientific = -3,
  ...
)
```

## Arguments

| | |
|---|---|
| `e1` | type print22() at the console |
| `u1` | type print22() at the console |
| `o1t` | type print22() at the console |
| `e2` | type print22() at the console |
| `u2` | type print22() at the console |
| `o2t` | type print22() at the console |
| `et` | type print22() at the console |
| `ut` | type print22() at the console |
| `T` | type print22() at the console |
| `lev` | significance level for confidence intervals. Default is 0.95 |
| `LaTeX` | whether to print the 2 by 2 information as LaTeX text to the screen, including the table, odds ratio, relative risk and confidence intervals |
| `digits` | see ?format |

| | |
|---|---|
| `scientific` | see ?format |
| `...` | other arguments passed to function format |

## Value

Computes odds ratios and relative risks together with confidence intervals for 2 by 2 table and prints to screen in LaTeX format.

## See Also

print22

---

# vdc *vdc function*

---

## Description

A function to generate a Van der Corput sequence of numbers.

## Usage

```
vdc(n)
```

## Arguments

| | |
|---|---|
| `n` | the length of the sequence |

## Value

Van der Corput sequence of length n

---

# yhIterate *yhIterate function*

---

## Description

A function to perform forecasting of the series, used by fcastGARCH

## Usage

```
yhIterate(i, current, pars, eps, omega)
```

## Arguments

| | |
|---|---|
| i | the index of the forward lags |
| current | current matrix of (y,h) |
| pars | parameters for the GARCH model, these would come from an MCMC run |
| eps | matrix of Gaussian noise, dimension equal to number of MCMC iterations by the number of forecast lags |
| omega | matrix of Inverse Gamma noise, dimension equal to number of MCMC iterations by the number of forecast lags |

## Value

two column matrix containing forecast y (1st column) and updated h (2nd column)

# Index