# Package 'Tmisc'

October 12, 2022

**Title** Turner Miscellaneous

**Version** 1.0.0

**Maintainer** Stephen Turner <vustephen@gmail.com>

**Description** Miscellaneous utility functions for data manipulation, data tidying, and working with gene expression data.

**URL** https://github.com/stephenturner/Tmisc, https://stephenturner.github.io/Tmisc/

**Depends** R (>= 3.1.2)

**Imports** dplyr, tibble, utils, rstudioapi, methods, magrittr, stats

**License** GPL-3

**LazyData** true

**RoxygenNote** 7.1.0

**Suggests** ggplot2, reshape2

**NeedsCompilation** no

**Author** Stephen Turner [aut, cre] (<https://orcid.org/0000-0001-9140-9028>)

**Repository** CRAN

**Date/Publication** 2020-09-16 13:50:04 UTC

## R topics documented:

---

`addins` *Insert text at current position.*

---

## Description

Call these function as an addin to insert desired text at the cursor position. After installing Tmisc, hit the Addins menu, and optionally add a keyboard shortcut, e.g., Command+Shift+I, Alt+-, etc.

---

`are_all_equal` *Are all equal?*

---

## Description

Are all the elements of a numeric vector (approximately) equal?

## Usage

```
are_all_equal(x, na.rm = FALSE)
```

### Arguments

| | |
|---|---|
| x | A numeric vector. |
| na.rm | Remove missing values (FALSE by default; NAs in x will return NA). |

### Value

Logical, whether all elements of a numeric vector are equal.

### Examples

```
are_all_equal(c(5,5,5))
are_all_equal(c(5,5,5,6))
are_all_equal(c(5,5,5,NA,6))
are_all_equal(c(5,5,5,NA,6), na.rm=TRUE)
5==5.000000001
identical(5, 5.000000001)
are_all_equal(c(5L, 5, 5.000000001))
```

---

## corner *Print the top left corner of a data frame*

---

### Description

Prints the first n rows and columns of a data frame or matrix.

### Usage

```
corner(x, n = 5)
```

### Arguments

| | |
|---|---|
| x | A data.frame. |
| n | The number of rows/columns to print. |

### Value

The corner of the data frame

### Examples

```
corner(mtcars)
corner(iris, n=4)
```

## counts2fpkm *Fragments per kilobase per million*

### Description

Takes a count matrix and a vector of gene lengths and returns an optionally log2-transformed FPKM matrix. Modified from edgeR.

### Usage

```
counts2fpkm(x, length, log = FALSE, prior.count = 0.25)
```

### Arguments

| | |
|---|---|
| x | a matrix of counts |
| length | a vector of length nrow(x) giving length in bases |
| log | logical, if TRUE, then log2 values are returned. |
| prior.count | average count to be added to each observation to avoid taking log of zero. Used only if log=TRUE. |

### Value

A matrix of FPKM values.

### Examples

```
## Not run:
library(readr)
library(dplyr)
countdata <- read_csv("http://files.figshare.com/2439061/GSE37704_featurecounts.csv")
counts <- countdata %>% select(countdata, starts_with("SRR")) %>% as.matrix
counts2fpkm(counts, countdata$length)

## End(Not run)
```

## deseqresult2tbl *Tidy DESeq2 result*

### Description

Returns a tidy version of a DESeq2 results table.

### Usage

```
deseqresult2tbl(deseqresult, colname = "ensgene")
```

### Arguments

| | |
|---|---|
| deseqresult | Results from running results(dds) on a DESeqDataSet object. |
| colname | The name of the column you want to use for what DESeq puts in the row name. |

### Value

a tidy version of the DESeq2 results.

### Examples

```
## Not run:
res <- results(dds)
res <- deseqresult2tbl

## End(Not run)
```

---

## dokuwiki *Create tables in dokuwiki format*

---

### Description

Prints the supplied data frame or matrix using Dokuwiki's table syntax, optionally copying the data to the clipboard (Mac OS X only).

### Usage

```
dokuwiki(x, headersep = "^", sep = "|", clip = TRUE, ...)
```

### Arguments

| | |
|---|---|
| x | A data.frame. |
| headersep | The separator used between entries in the header row. |
| sep | The separator used between entries in all other rows. |
| clip | Whether or not to write the returned table to the clipboard (currently only supported on Mac OS X). |
| ... | Further arguments passed to write.table. |

### Examples

```
dokuwiki(head(iris), clip=FALSE)
dokuwiki(head(mtcars), clip=FALSE, row.names=TRUE)
```

## ellipses *Truncate a data frame with ellipses.*

### Description

Prints the specified number of rows of a data frame, followed by a row of ellipses. Useful for piping to `knitr::kable()` for printing a truncated table in a markdown document.

### Usage

```
ellipses(df, n = 5L)
```

### Arguments

| | |
|---|---|
| `df` | A data.frame. |
| `n` | The number of rows to show before an ellipses row. |

### Value

A data frame truncated by a row of ellipses.

### Examples

```
## Not run:
ellipses(mtcars, 5)

## End(Not run)
```

## fisherp *Fisher's method to combine p-values.*

### Description

Uses Fisher's method to combine p-values from different tests.

### Usage

```
fisherp(x)
```

### Arguments

| | |
|---|---|
| `x` | A vector of p-values between 0 and 1. |

### Value

A combined p-value.

### Examples

```
fisherp(c(.042, .02, .001, 0.01, .89))
```

---

## gghues *Emulate ggplot2 default hues*

---

### Description

This will emulate ggplot2's hues, which are equally spaced hues around the color wheel, starting from 15.

### Usage

```
gghues(n, start = 15)
```

### Arguments

| | |
|---|---|
| n | The Numeric; number of hues to generate. |
| start | Numeric; the place on the color wheel to start. ggplot2 default is 15. |

### Value

A vector of hues

### Examples

```
n <- 10
gghues(3)
barplot(rep(1,n), col=gghues(n), names=gghues(n))
barplot(rep(1,n), col=gghues(n, start=15+180), names=gghues(n, start=15+180))
```

---

## gg_na *Plot missing data*

---

### Description

Plots missing data as holes on a black canvas.

### Usage

```
gg_na(df)
```

### Arguments

| | |
|---|---|
| df | A data.frame. |

## Examples

```
# What a mess.
# Feature 10 is missing a lot. Observations 25 and 35 are completely missing.
# Most of features 40-45 are missing, except for the first few observations.
set.seed(2016-07-12)
x <- matrix(1, nrow=50, ncol=50)
x[sample(prod(dim(x)), 100)] <- NA
x <- data.frame(x)
x$X10[sample(length(x$X10), 25)] <- NA
x[c(25, 35), ] <- NA
x[1:40, 40:45] <- NA
gg_na(x)
```

---

`gt2refalt` *Two-letter genotype from VCF GT*

---

## Description

Get a two-letter genotype from a VCF GT field. Current implementation is quick and dirty, and only accepts 0/0, 0/1, or 1/1. Any other input to gt will return a missing value.

## Usage

```
gt2refalt(gt, ref, alt)
```

## Arguments

| | |
|---|---|
| `gt` | The genotype field (must be 0/0, 0/1, or 1/1). |
| `ref` | The reference allele. |
| `alt` | The alternate allele. |

## Value

Returnvalue

## Examples

```
gt2refalt(gt="0/0", ref="R", alt="A")
gt2refalt(gt="0/1", ref="R", alt="A")
gt2refalt(gt="1/1", ref="R", alt="A")
gt2refalt(gt="0/2", ref="R", alt="A")
gt2refalt(gt="./.", ref="R", alt="A")
```

## jsd *Jensen-Shannon divergence*

### Description

Calculates a distance matrix from a matrix of probability distributions using Jensen-Shannon divergence. Adapted from https://enterotype.embl.de/enterotypes.html#dm.

### Usage

```
jsd(M, pseudocount = 1e-06, normalizeCounts = FALSE)
```

### Arguments

| | |
|---|---|
| M | a probability distribution matrix, e.g., normalized transcript compatibility counts. |
| pseudocount | a small number to avoid division by zero errors. |
| normalizeCounts | logical, whether to attempt to normalize by dividing by the column sums. Set to TRUE if this is, e.g., a count matrix. |

### Value

A Jensen-Shannon divergence-based distance matrix.

### Examples

```
set.seed(42)
M <- matrix(rpois(100, lambda=100), ncol=5)
colnames(M) <- paste0("sample", 1:5)
rownames(M) <- paste0("gene", 1:20)
Mnorm <- apply(M, 2, function(x) x/sum(x))
Mjsd <- jsd(Mnorm)
# equivalently
Mjsd <- jsd(M, normalizeCounts=TRUE)
Mjsd
plot(hclust(Mjsd))
```

## lmp *Linear model p-value*

### Description

Extract F-test p-value from a linear model object. Can also use `broom::glance(fit)`. Originally described at https://web.archive.org/web/20200829213926/https://gettinggeneticsdone.blogspot.com/2011/01/rstats-function-for-extracting-f-test-p.html.

### Usage

```
lmp(modelobject)
```

### Arguments

| | |
|---|---|
| modelobject | A model object of class lm. |

### Value

The p-value on the f-test of a linear model object testing the null hypothesis that R^2==0.

### Examples

```
# simulate some (e.g. SNP genotype) data
set.seed(42)
n=20
d=data.frame(x1=rbinom(n,2,.5), x2=rbinom(n,2,.5))
d=transform(d, y=x1+x2+rnorm(n))
#fit the linear model
fit=lm(y ~ x1 + x2, data=d)
#shows that the F-test is 0.006641
summary(fit)
#can't access that p-value using this
names(summary(fit))
# this doesn't work either
names(fit)
lmp(fit)
```

---

## lowestnonzero *Lowest nonzero values*

---

### Description

Sometimes want to plot p-values (e.g., volcano plot or MA-plot), but if a statistical test returns a zero p-value, this causes problems with visualization on the log scale. This function returns a vector where the zero values are equal to the smallest nonzero value in the vector.

### Usage

```
lowestnonzero(x)
```

### Arguments

| | |
|---|---|
| x | A vector of p-values between 0 and 1. |

### Value

A vector of p-values where zero values are exchanged for the lowest non-zero p-value in the original vector.

## Examples

```
lowestnonzero(c(.042, .02, 0, .001, 0, .89))
```

---

`lsa` *Improved list of objects*

---

## Description

Improved list of objects. Sorts by size by default. Adapted from https://stackoverflow.com/q/1358003/654296.

## Usage

```
lsa(
  pos = 1,
  pattern,
  order.by = "Size",
  decreasing = TRUE,
  head = TRUE,
  n = 10
)
```

## Arguments

| | |
|---|---|
| `pos` | numeric. Position in the stack. |
| `pattern` | Regex to filter the objects by. |
| `order.by` | character. Either 'Type', 'Size', 'PrettySize', 'Rows', or 'Columns'. This will dictate how the output is ordered. |
| `decreasing` | logical. Should the output be displayed in decreasing order? |
| `head` | logical. Use head on the output? |
| `n` | numeric. Number of objects to display is head is TRUE. |

## Value

A data.frame with type, size in bytes, human-readable size, rows, and columns of every object in the environment.

## Author(s)

Dirk Eddelbuettel, Tony Breyal

## Examples

```
## Not run:
a <- rnorm(100000)
b <- matrix(1, 1000, 100)
lsa()

## End(Not run)
```

---

`mat2df` *Matrix to pairwise data frame*

---

## Description

Turns a distance matrix into a data frame of pairwise distances.

## Usage

```
mat2df(M)
```

## Arguments

| | |
|---|---|
| `M` | a square pairwise matrix (e.g., of distances). |

## Value

Data frame with pairwise distances.

## Examples

```
set.seed(42)
M <- matrix(rnorm(25), nrow=5)
M
mat2df(M)
M <- matrix(rnorm(25), nrow=5, dimnames=list(letters[1:5], letters[1:5]))
M
mat2df(M)
```

Mode *Mode.*

## Description

Returns the mode of a vector. First in a tie wins (see examples).

## Usage

```
Mode(x, na.rm = FALSE)
```

## Arguments

| | |
|---|---|
| x | A vector. |
| na.rm | Remove missing values before calculating the mode (FALSE by default). NAs are counted just like any other element. That is, an NA in the vector won't necessarily result in a return NA. See the first example. |

## Value

A combined p-value.

## Examples

```
Mode(c(1,2,2,3,3,3, NA))
Mode(c(1,2,2,3,3,3, NA), na.rm=TRUE)
Mode(c(1,2,2,3,3,3, NA, NA, NA, NA))
Mode(c(1,2,2,3,3,3, NA, NA, NA, NA), na.rm=TRUE)
Mode(c("A", "Z", "Z", "B", "B"))
```

nn *Get names and class of all columns in a data frame*

## Description

Get names and class of all columns in a data frame in a friendly format.

## Usage

```
nn(df)
```

## Arguments

| | |
|---|---|
| df | A data.frame. |

### Value

A data.frame with index and class.

### Author(s)

Stephen Turner

### Examples

```
nn(iris)
```

---

## o *Open the current working directory on mac*

---

### Description

Opens the current working directory on mac.

### Usage

```
o()
```

### Examples

```
## Not run:
o()

## End(Not run)
```

---

## peek *Peek at the top of a text file*

---

### Description

This returns a character vector which shows the top n lines of a file.

### Usage

```
peek(x, n = 5)
```

### Arguments

| | |
|---|---|
| x | a filename |
| n | the number of lines to return |

## Examples

```
## Not run:
filename <- tempfile()
x<-matrix(round(rnorm(10^4),2),1000,10)
colnames(x)=letters[1:10]
write.csv(x,file=filename,row.names=FALSE)
peek(filename)

## End(Not run)
```

---

`propmiss` *Missing stats*

---

## Description

Returns the number of missing values, total length, and proportion missing values for each variable in a data.frame

## Usage

```
propmiss(df)
```

## Arguments

| | |
|---|---|
| `df` | A data.frame. |

## Value

A data.frame with missingness stats.

## Examples

```
## Not run:
propmiss(data.frame(a=1:5, b=c(6,NA,NA,9,10)))

## End(Not run)
```

## quartet *Anscombe's Quartet data (tidy)*

### Description

Tidy version of built-in Anscombe's Quartet data. Four datasets that have nearly identical linear regression properties, yet appear very different when graphed.

### Usage

```
quartet
```

### Format

Data frame with columns.

## read.cb *Read from the clipboard*

### Description

Read tabular data from the clipboard.

### Usage

```
read.cb(header = TRUE, ...)
```

### Arguments

| | |
|---|---|
| header | A logical value indicating whether the file contains the names of the variables as its first line. Overrides the default header=FALSE option in read.table(). |
| ... | Further arguments to be passed to read.table |

### Value

A data.frame

### Examples

```
## Not run:
# To read CSV data with a header from the clipboard:
read.cb(header=TRUE, sep=',')

## End(Not run)
```

## rownames_to_symprobe *Rownames to symbol-probeID*

### Description

This function takes an `exprs(eset)` matrix where the rownames are probeset IDs and takes an annotated topTable output where you have an ID and Symbol column and outputs a character vector with symbol_probeid for each probeid in rownames(exprs(eset)). You can use this such that the output on a heatmap contains the gene names concatenated to the probe ID in case you have multiple symbols with the same probeID.

### Usage

```
rownames_to_symprobe(exprset, tt)
```

### Arguments

| | |
|---|---|
| `exprset` | The output of `exprs(eset)`. |
| `tt` | A topTable object. |

### Value

Character vector of the gene symbol with the probe ID.

### Examples

```
## Not run:
rownames_to_symprobe(esprs(eset), topTable(fit, number=nrow(fit)))

## End(Not run)
```

## saveit *Rename objects while saving.*

### Description

Allows you to rename objects as you save them. See https://stackoverflow.com/a/21248218/654296.

Allows you to rename objects as you save them. See https://stackoverflow.com/a/21248218/654296.

### Usage

```
saveit(..., file = stop("'file' must be specified"))

saveit(..., file = stop("'file' must be specified"))
```

### Arguments

| | |
|---|---|
| ... | Objects to save. |
| file | Filename/path where data will be saved. |

### Examples

```
## Not run:
foo <- 1
saveit(bar=foo, file="foobar.Rdata")

## End(Not run)

## Not run:
foo <- 1
saveit(bar=foo, file="foobar.Rdata")

## End(Not run)
```

---

## sicb *Write sessionInfo to the clipboard*

---

### Description

Writes output of `sessionInfo()` to the clipboard. Only works on Mac.

### Usage

```
sicb()
```

### Examples

```
## Not run:
# Write sessionInfo() to the clipboard on mac.
sicb()

## End(Not run)
```

## strSort *Sort characters in a string*

### Description

Alphabetically sorts characters in a string. Vectorized over x.

### Usage

```
strSort(x)
```

### Arguments

| | |
|---|---|
| x | A string to sort. |

### Value

A sorted string.

### Examples

```
strSort("cba")
strSort("zyxcCbB105.a")
strSort(c("cba", "zyx"))
strSort(c("cba", NA))
```

## Tcols *A palette of 17 diverging colors*

### Description

17 diverging colors created by combining the Set1 and Dark2 palettes from RColorBrewer.

### Usage

```
Tcols
```

### Format

Vector of 17 diverging colors.

### Source

R Color brewer: `c(brewer.pal(9, "Set1"), brewer.pal(8, "Dark2"))`.

## Examples

```
## Not run:
 barplot(rep(1, 17), col=Tcols, axes=F, names=c(rep("Set1", 9), rep("Dark2", 8)), horiz=TRUE, las=2)

## End(Not run)
```

---

`Thist` *Histograms with overlays*

---

## Description

Plot a histogram with either a normal distribution or density curve overlay.

## Usage

```
Thist(x, overlay = "normal", col = "gray80", ...)
```

## Arguments

| | |
|---|---|
| `x` | A numeric vector. |
| `overlay` | Either "normal" (default) or "density" indicating whether a normal distribution or density curve should be plotted on top of the histogram. |
| `col` | Color of the histogram bars. |
| `...` | Other arguments to be passed to `hist()`. |

## Examples

```
set.seed(42)
x <- rnorm(1000, mean=5, sd=2)
Thist(x)
Thist(x, overlay="density")
Thist(x^2)
Thist(x^2, overlay="density", breaks=50, col="lightblue2")
```

---

`Tmisc` *Tmisc*

---

## Description

Stephen Turner's miscellaneous functions

## Author(s)

Stephen Turner

## Tpairs *Better scatterplot matrices.*

### Description

A matrix of scatter plots with rugged histograms, correlations, and significance stars. Much of the functionality borrowed from `PerformanceAnalytics::chart.Correlation()`.

### Usage

```
Tpairs(x, histogram = TRUE, gap = 0, ...)
```

### Arguments

| | |
|---|---|
| x | A numeric matrix or data.frame. |
| histogram | Overlay a histogram on the diagonals? |
| gap | distance between subplots, in margin lines. |
| ... | arguments to be passed to or from other methods. |

### Examples

```
Tpairs(iris[-5])
Tpairs(iris[-5], pch=21, bg=Tcols[factor(iris$Species)])
Tpairs(iris[-5], pch=21, bg=gghues(3)[factor(iris$Species)], gap=1)
```

## %like% *x like y*

### Description

Returns a logical vector of elements of x matching the regex y.

### Usage

```
x %like% pattern
```

### Arguments

| | |
|---|---|
| x | a vector (numeric, character, factor) |
| pattern | a vector (numeric, character, factor), matching the mode of x |

### Value

A logical vector with length equal to x of things in x that are like y.

## See Also

%like%, %nlike%, %nin%,

## Examples

```
(Name <- c("Mary","George","Martha"))
Name %in% c("Mary")
Name %like% "^Mar"
Name %nin% c("George")
Name %nlike% "^Mar"
```

---

# %nin% *x not in y*

---

## Description

Returns a logical vector of elements of x that are not in y.

## Usage

```
x %nin% table
```

## Arguments

| | |
|---|---|
| x | a vector (numeric, character, factor) |
| table | a vector (numeric, character, factor), matching the mode of x |

## Value

A logical vector with length equal to x of things in x that aren't in y.

## See Also

%like%, %nlike%, %nin%,

## Examples

```
1:10 %nin% seq(from=2, to=10, by=2)
c("a", "b", "c") %nin% c("a", "b")
letters[letters %nin% unlist(strsplit("pack my box with five dozen liquor jugs", ""))]
```

`%nlike%` *x not like y*

## Description

Returns a logical vector of elements of x not matching the regex y.

## Usage

```
x %nlike% pattern
```

## Arguments

| | |
|---|---|
| x | a vector (numeric, character, factor) |
| pattern | a vector (numeric, character, factor), matching the mode of x |

## Value

A logical vector with length equal to x of things in x that aren't like y.

## See Also

%like%, %nlike%, %nin%,

## Examples

```
(Name <- c("Mary","George","Martha"))
Name %in% c("Mary")
Name %like% "^Mar"
Name %nin% c("George")
Name %nlike% "^Mar"
```

# Index