

# Package ‘aakmisc’

August 17, 2021

## R topics documented:

db . . . . .	1
lazyload . . . . .	4
matrix2latex . . . . .	5
numbers2words . . . . .	5
plotMatrix . . . . .	6
random . . . . .	7
scinot . . . . .	9
trnc . . . . .	10
tunnel . . . . .	10
<b>Index</b>	<b>12</b>

---

db	<i>Interface with databases</i>
----	---------------------------------

---

## Description

Interface with project databases.

## Usage

```
writtenDBTable(  
  name,  
  value,  
  overwrite = FALSE,  
  append = FALSE,  
  row.names = FALSE,  
  host = getOption("aakmisc.dbhost", "localhost"),  
  dbname = getOption("aakmisc.dbname", NULL),  
  port = getOption("aakmisc.port", 5432),  
  user = getOption("aakmisc.user", NULL),  
  ...  
)
```

```
)

getQuery(
    statement,
    host = getOption("aakmisc.dbhost", "localhost"),
    dbname = getOption("aakmisc.dbname", NULL),
    port = getOption("aakmisc.port", 5432),
    user = getOption("aakmisc.user", NULL),
    ...
)

getMLEs(
    host = getOption("aakmisc.dbhost", "localhost"),
    dbname = getOption("aakmisc.dbname", NULL),
    port = getOption("aakmisc.port", 5432),
    user = getOption("aakmisc.user", NULL),
    ...
)

recMLEs(
    mle,
    host = getOption("aakmisc.dbhost", "localhost"),
    dbname = getOption("aakmisc.dbname", NULL),
    port = getOption("aakmisc.port", 5432),
    user = getOption("aakmisc.user", NULL),
    ...
)

recScript(
    files,
    host = getOption("aakmisc.dbhost", "localhost"),
    dbname = getOption("aakmisc.dbname", NULL),
    port = getOption("aakmisc.port", 5432),
    user = getOption("aakmisc.user", NULL),
    ...
)

dropScript(
    script,
    host = getOption("aakmisc.dbhost", "localhost"),
    dbname = getOption("aakmisc.dbname", NULL),
    port = getOption("aakmisc.port", 5432),
    user = getOption("aakmisc.user", NULL),
    ...
)

listScripts(
    host = getOption("aakmisc.dbhost", "localhost"),
```

```

    dbname = getOption("aakmisc.dbname", NULL),
    port = getOption("aakmisc.port", 5432),
    user = getOption("aakmisc.user", NULL),
    ...
)

catScript(
  script,
  file = "",
  host = getOption("aakmisc.dbhost", "localhost"),
  dbname = getOption("aakmisc.dbname", NULL),
  port = getOption("aakmisc.port", 5432),
  user = getOption("aakmisc.user", NULL),
  ...
)

```

### Arguments

name, value	Name and contents of table to create.
overwrite, append, row.names	See <a href="#">dbWriteTable</a> .
host	Hostname on which to connect to the PostgreSQL server.
dbname	Name of PostgreSQL database.
port	Port on which to connect to PostgreSQL database. If NULL, a random port number will be used.
user	Username to use in connecting to PostgreSQL database. If NULL, Sys.getenv("USER") will be used.
...	Additional arguments will be passed to <a href="#">dbConnect</a> .
statement	SQL statement passed to <a href="#">dbGetQuery</a> .
mle	A data-frame of MLEs to be recorded.
files	Files containing R scripts to be recorded.
script	Name of script.
file	File to which the script will be written. See <a href="#">cat</a> .

### Author(s)

Aaron A. King

### Examples

```

## Not run:
startTunnel()
listScripts()
stopTunnel()

## End(Not run)

```

---

 lazyload

*Functions for lazy-loading knitr caches.*


---

## Description

These functions are helpful for loading cached chunks into an interactive session.

## Usage

```
lazyload_cache_dir(
  path = "./cache",
  envir = parent.frame(),
  ask = FALSE,
  verbose = getOption("verbose", FALSE),
  full.names = TRUE,
  ...
)

lazyload_cache_labels(
  labels,
  path = "./cache/",
  envir = parent.frame(),
  verbose = getOption("verbose", FALSE),
  filter,
  full.names = TRUE,
  ...
)
```

## Arguments

path	the path to the cache directory
envir	the environment to load the objects into
ask	if TRUE, interactively ask whether to load each database discovered in path
verbose	if TRUE, display the names of chunk labels being loaded
full.names	use the full name, i.e., include the path, for the chunk label? This argument is passed to <a href="#">list.files</a> .
...	additional arguments passed to <a href="#">list.files</a>
labels	character vector; chunk labels to load
filter	optional function; passed to <a href="#">lazyLoad</a> . When called on a character vector of object names, this function should return a logical vector: objects for which this is TRUE will be loaded.

## Details

Use `lazyload_cache_dir` to load a whole directory of cached objects.

Use `lazyload_cache_labels` to load and explicit set of cached chunks.

**Value**

Both functions return NULL, invisibly.

**Author(s)**

Peter DeWitt (<https://github.com/dewittpe>).

---

matrix2latex	<i>matrix2latex</i>
--------------	---------------------

---

**Description**

Format a matrix for latex.

**Usage**

```
matrix2latex(x, type = "pmatrix")
```

**Arguments**

x	matrix
type	latex matrix environment

---

numbers2words	<i>Convert integers to English words.</i>
---------------	---

---

**Description**

numbers2words spells out integers in English. The code is lifted from Andy Teucher [https://github.com/ateucher/useful\\_code/blob/master/R/numbers2words.r](https://github.com/ateucher/useful_code/blob/master/R/numbers2words.r), who in turn stole it from John Fox. It has been improved somewhat by AAK

**Usage**

```
numbers2words(x)
```

**Arguments**

x	integer to format.
---	--------------------

**Examples**

```
numbers2words(49968883)
numbers2words(c(85999000, 54, 540, 5400, 54000, 540000))
numbers2words(1e13+3)
```

---

plotMatrix

*A scatterplot matrix with densities on the diagonal.*


---

## Description

A special scatterplot matrix.

## Usage

```
plotMatrix(data, ...)

## S3 method for class 'list'
plotMatrix(
  data,
  marg.exp = 0.02,
  labels = names(data),
  alpha = 1,
  pch = 16,
  size = unit(2, "mm"),
  ...
)

## S3 method for class 'data.frame'
plotMatrix(
  data,
  marg.exp = 0.02,
  labels = names(data),
  alpha = 1,
  pch = 16,
  size = unit(2, "mm"),
  ...
)

## S3 method for class 'aakplot'
print(x, newpage = is.null(vp), vp = NULL, ...)
```

## Arguments

data	Data to plot.
...	optional arguments, passed to <a href="#">hist</a> .
marg.exp	Fraction by which to expand the plot at the margins.
labels	Names of variables plotted.
alpha, pch, size	Refer to the plotted points in the scatterplots.
x	plotMatrix object to display.

**newpage**            logical; if TRUE, `grid.newpage()` will be called before the graphics are drawn.

**vp**                viewport to use. See [viewport](#).

### Author(s)

Aaron A. King <kingaa at umich dot edu>

### Examples

```
## Not run:
x <- data.frame(a=rexp(n=1000,rate=1/3),b=rnorm(1000))
mutate(x,c=a+b^2,d=a-b^3) -> x

print(plotMatrix(x,alpha=0.2))

g <- plotMatrix(
  x[-2],
  labels=c(
    expression(alpha),
    expression(beta),
    expression(phi)
  ),
  alpha=0.3
)
print(g)

print(plotMatrix(as.list(x),alpha=0.2,breaks='scott'))

## End(Not run)
```

### Description

Functions for generating and working with truly random seeds.

### Usage

```
random.org(n = 10, rnd = "new")

urandom(n = 10)

rngControl(expr, seed = NULL)

rngSeeds(n, seed = NULL)
```

**Arguments**

n	Number of integers required.
rnd	random.org parameter
expr	Expression to be evaluated with RNG control.
seed	RNG seed.

**Details**

random.org gets seeds from [random.org](http://www.random.org).

urandom gets seeds locally from ‘/dev/urandom’ on \*nix systems.

rngControl is a function to control RNG for the evaluation of an expression.

rngSeeds generates RNG seeds using [runif](#). It is included for situations when neither [random.org](#) nor [urandom](#) is available.

**Value**

integers suitable for use as RNG seeds

**Author(s)**

Aaron A. King

**References**

<http://www.random.org>

**Examples**

```
## Not run:
random.org(n=5)
seed <- urandom(n=1)
seeds <- rngSeeds(5, seed=seed)
set.seed(seed)
runif(5)
rngControl(runif(5), seed=seed[1])
rngControl(runif(5), seed=seed[1])
runif(5)
set.seed(seed)
runif(5)
runif(5)

## End(Not run)
```



---

`scinot`*Scientific notation.*

---

## Description

Format using scientific notation.

## Usage

```
scinot(  
  x,  
  digits = 2,  
  format = c("expression", "latex", "math"),  
  simplify = FALSE  
)
```

## Arguments

<code>x</code>	number(s) to format.
<code>digits</code>	number of significant digits in mantissa.
<code>format</code>	format specification. <code>format="expression"</code> results in an R expression. <code>format="latex"</code> results in a latex expression. <code>format="math"</code> is like "latex" but wraps the text in "\$".
<code>simplify</code>	logical. If <code>simplify=TRUE</code> , then $1 \times 10^n$ is simplified to $10^n$ .

## Author(s)

Aaron A. King

## See Also

[scientific](#)

## Examples

```
x <- c(0.0309595, 8577676.441, 10000)  
scinot(x[2], 4)  
scinot(x[1], 2, "latex")  
sapply(x, scinot, digits=3, format='math')  
scinot(x, digits=0, simplify=FALSE)  
scinot(x, digits=0, simplify=TRUE)
```

---

trnc	<i>Truncation of plots.</i>
------	-----------------------------

---

**Description**

Truncate to the specified window.

**Usage**

```
trnc(x, range = c(0, 1), only.finite = TRUE)
```

**Arguments**

- x                    Numeric vector of values to manipulate.
- range                Numeric vector of length two giving desired output range.
- only.finite        if TRUE (the default), will only modify finite values.

**Details**

trnc is a function for truncating data to a specified window. It is suitable for use in `scale_{x,y}_{continuous,discrete}`, for example.

**Author(s)**

Aaron A. King <kingaa at umich dot edu>

**See Also**

[censor](#)

**Examples**

```
trnc(c(-1,0.5,1,2,NA))
```

---

tunnel	<i>ssh tunneling</i>
--------	----------------------

---

**Description**

Setting up ssh tunnels for database access.

**Usage**

```
startTunnel(  
    port = NULL,  
    remotehost = getOption("aakmisc.remotehost", NULL),  
    user = getOption("aakmisc.user", NULL),  
    sleep = 5  
)  
  
stopTunnel(..., pid = getOption("aakmisc.tunnelpid", NULL))
```

**Arguments**

port	Port on which to connect to PostgreSQL database. If NULL, a random port number will be used.
remotehost	Hostname of PostgreSQL server. An ssh tunnel to this host will be created.
user	Username to use in connecting to PostgreSQL database. If NULL, Sys.getenv("USER") will be used.
sleep	Time in seconds to sleep after initiating the ssh tunnel.
...	Additional arguments will be passed to <a href="#">dbConnect</a> .
pid	ID of ssh tunnel process. Set automatically by startTunnel.

# Index

- \* **internal**
  - tunnel, [10](#)
- \* **programming**
  - lazyload, [4](#)
- cat, [3](#)
- catScript (db), [1](#)
- censor, [10](#)
- cleanup (tunnel), [10](#)
- db, [1](#)
- dbConnect, [3](#), [11](#)
- dbGetQuery, [3](#)
- dbWriteTable, [3](#)
- dropScript (db), [1](#)
- getMLEs (db), [1](#)
- getQuery (db), [1](#)
- hist, [6](#)
- lazyLoad, [4](#)
- lazyload, [4](#)
- lazyload\_cache\_dir (lazyload), [4](#)
- lazyload\_cache\_labels (lazyload), [4](#)
- list.files, [4](#)
- listScripts (db), [1](#)
- matrix2latex, [5](#)
- numbers2words, [5](#)
- plotMatrix, [6](#)
- print.aakplot (plotMatrix), [6](#)
- random, [7](#)
- random.org, [8](#)
- recMLEs (db), [1](#)
- recScript (db), [1](#)
- rngControl (random), [7](#)
- rngSeeds (random), [7](#)
- runif, [8](#)
- scientific, [9](#)
- scinot, [9](#)
- startTunnel (tunnel), [10](#)
- stopTunnel (tunnel), [10](#)
- trnc, [10](#)
- tunnel, [10](#)
- urandom, [8](#)
- urandom (random), [7](#)
- viewport, [7](#)
- writeDBTable (db), [1](#)