

# Package ‘basr’

June 22, 2018

**Version** 0.12

**Date** 2018-06-22

**Title** Basic, but hopefully useful, functions

**Description** This package provides a bunch of basic functions for a variety of usage.

**Author** Mathieu Basille, contributions from Samuel Brown, Marc in the box, Clement Calenge, Michael Hallquist, Jean Lobry, Emiel van Loon, Kevin Wright

**Maintainer** Mathieu Basille <basille@ufl.edu>

**Suggests** devtools

**License** GPL (>= 3)

**URL** <http://ase-research.org/basille/basr>

**RoxygenNote** 6.0.1

## R topics documented:

|                        |    |
|------------------------|----|
| basr . . . . .         | 2  |
| capwords . . . . .     | 2  |
| colNA . . . . .        | 3  |
| confint . . . . .      | 4  |
| cv . . . . .           | 5  |
| dynamitePlot . . . . . | 6  |
| extrange . . . . .     | 8  |
| getcolors . . . . .    | 8  |
| manual . . . . .       | 9  |
| memUse . . . . .       | 10 |
| mv . . . . .           | 11 |
| ncut . . . . .         | 12 |
| nsubset . . . . .      | 13 |
| q . . . . .            | 14 |
| reclass . . . . .      | 14 |
| save.image . . . . .   | 15 |

|                         |           |
|-------------------------|-----------|
| savepkglist . . . . .   | 16        |
| se . . . . .            | 17        |
| summaryIC . . . . .     | 18        |
| table . . . . .         | 19        |
| togray . . . . .        | 19        |
| writeFunction . . . . . | 20        |
| <b>Index</b>            | <b>22</b> |

---

|      |                          |
|------|--------------------------|
| basr | <i>Utility functions</i> |
|------|--------------------------|

---

**Description**

basr package

**Details**

This package provides a bunch of basic, but hopefully useful, functions for a variety of usage. For a list of documented functions, use `library(help = "basr")`

**Author(s)**

Mathieu Basille <basille@ufl.edu>, contributions from Samuel Brown, Marc in the box, Jean Lobry, Kevin Wright

---

|          |                     |
|----------|---------------------|
| capwords | <i>Capitalizing</i> |
|----------|---------------------|

---

**Description**

Capitalizing - every first letter of a word is changed to upper case.

**Usage**

`capwords(s, strict = FALSE)`

**Arguments**

- |        |  |
|--------|--|
| s      | A character vector, or an object that can be coerced to character by <code>as.character</code> . |
| strict | Logical: other letters than the first are converted to lower case                                |

**Value**

A character vector of the same length and with the same attributes as `x` (after possible coercion).

**Author(s)**

From the help page of [chartr](#)

**Examples**

```
capwords(c("using AIC for model selection"))
## -> [1] "Using AIC For Model Selection"
capwords(c("using AIC", "for MODEL selection"), strict = TRUE)
## -> [1] "Using Aic" "For Model Selection"
##           ^^^      ^^^^^
##           'bad'    'good'
```

---

colNA

*Number of NAs by column*

---

**Description**

Returns the number of NAs for each column of a data frame.

**Usage**

```
colNA(x)
```

**Arguments**

x                      A data frame.

**Value**

A numeric vector

**Author(s)**

Mathieu Basille <basille@ufl.edu>

**Examples**

```
set.seed(123)
(df <- data.frame(matrix(sample(c(NA, 1:10), 100, TRUE), ncol = 4)))
colNA(df)
```

---

confint

*Confidence Intervals for Model Parameters*


---

## Description

Modified version of the [confint](#) function, which displays the coefficients in addition to the CIs, and allows for more control on display parameters. A plot argument and function allow to graph the coefficients and their CIs.

## Usage

```
confint(object, parm, level = 0.95, order = FALSE, groups, plot = FALSE,
...)
```

```
## S3 method for class 'confint'
plot(x, mar = c(5, 7, 3, 1) + 0.1, col = NULL,
     main = attr(x, "model"), pch = 19, add.signif = FALSE, ...)
```

## Arguments

|            |  |
|------------|--|
| order      | Logical. If TRUE, the results are ordered by descending order on the coefficient value.  |
| groups     | A factor in the sense that <code>as.factor(f)</code> defines the groups, or a list of such factors in which case their interaction is used for the groups. See <a href="#">split</a> . |
| plot       | Whether to plot the results.   |
| ...        | Further arguments passed to points.  |
| x          | A data.frame of class confint.   |
| mar        | The number of lines of margin, can be useful if the coefficient names do not fit in the left margin. See <a href="#">par</a> for more details.   |
| col        | The color of each coefficient + CI; gray by default. If "groups", the color of each (sorted) group; use a hcl palette by default.  |
| main       | The title of the plot.   |
| pch        | The symbol to be used for the points. See <a href="#">par</a> for more details.  |
| add.signif | Logical, whether to add an asterik for variables with CIs non-overlapping with zero.   |
| ...        | Further arguments passed to points.  |

## Value

A data frame providing the CI and coefficients.

## Author(s)

Mathieu Basille <basille@ufl.edu>

**See Also**

[confint](#) for more details on other parameters.

**Examples**

```
## Example of linear model
fit <- lm(100/mpg ~ disp + hp + wt + am, data = mtcars)
## Standard 'confint' function
stats::confint(fit)
## Same results with modified function
confint(fit)
## Argument 'level'
stats::confint(fit, level = .9)
confint(fit, level = .9)
## Argument 'order'
confint(fit, order = TRUE)
## Argument 'groups'
confint(fit, groups = c(3, 1, 1, 1, 2))
## Argument 'level', "'order' and 'groups' simultaneously
confint(fit, level = .9, order = TRUE, groups = c(3, 1, 1, 1, 2))
## Argument 'parm'
stats::confint(fit, "am")
confint(fit, "am")
##'
## Plot of the results
plot(confint(fit, order = TRUE, groups = c(3, 1, 1, 1, 2)))
confint(fit, order = TRUE, groups = c(3, 1, 1, 1, 2), plot = TRUE)
confint(fit, order = TRUE, groups = c(3, 1, 1, 1, 2), plot = TRUE,
  col = c("blue", "red", "green"), pch = 18, cex = 2)
confint(fit, order = TRUE, groups = c(3, 1, 1, 1, 2), level = 0.9,
  plot = TRUE, add.signif = TRUE)
```

---

cv

*Coefficient of variation*


---

**Description**

This function computes the coefficient of variation (i.e.  $sd / mean$ ) of the values in `x`. If `ci` is `TRUE` then confidence intervals are also computed.

**Usage**

```
cv(x, na.rm = FALSE, ci = FALSE, conf.level = 0.95,
  method = c("mckaymod", "mckay", "naive"))
```

**Arguments**

|                         |   |
|-------------------------|---|
| <code>x</code>          | A numeric vector  |
| <code>na.rm</code>      | Logical. Should missing values be removed?  |
| <code>ci</code>         | Logical. Should confidence intervals be computed?   |
| <code>conf.level</code> | Confidence level of the interval.   |
| <code>method</code>     | The method to compute the confidence interval. Either the naive (naive), the McKay (mckay) or the modified McKay (mckaymod, default) approximation. |

**Value**

If `ci`, returns a list with the coefficient of variation. in the first element and the confidence interval in the second.

**Original URL**

<http://tolstoy.newcastle.edu.au/R/e2/help/07/06/19043.html>

**Author(s)**

From Kevin Wright, modified by Mathieu Basille <basille@ufl.edu>

**References**

Vangel, M. G. (1996) Confidence intervals for a normal coefficient of variation. The American Statistician, 50: 21-26

**Examples**

```
xx <- 1:10
cv(xx)
sd(xx)/mean(xx)
cv(xx, ci = TRUE)
```

---

dynamitePlot

*Dynamite Plots*


---

**Description**

Creates dynamite plots.

**Usage**

```
dynamitePlot(height, error, names.arg = NULL, significance = NA,
  ylim = c(0, maxLim), sym = FALSE, head = 0.7, lwd = par("lwd"),
  cex.sig = 1.2, ...)
```

**Arguments**

|              |   |
|--------------|---|
| height       | A vector of values describing the heights of the rectangular bars which make up the plot.   |
| error        | A vector of values indicating the length of error bars.   |
| names.arg    | A vector of names to be plotted below each bar or group of bars. If this argument is omitted, then the names are taken from the names attribute of height.      |
| significance | A character vector giving the group significance for each value.  |
| ylim         | Limits for the y axis. By default, ylim uses <code>c(0, maxLim)</code> , where <code>maxLim</code> is the maximum height + error multiplied by a factor of 1.1. |
| sym          | Logical. Whether to draw lower error bars.  |
| head         | A numeric, which gives the approximate width of the head, relative to the bar width.  |
| lwd          | The line width of the error bars, a <code>_positive_</code> number, defaulting to <code>par("lwd")</code> (usually 1).  |
| cex.sig      | The magnification to be used for significance groups relative to the current setting of <code>cex</code> (which defaults to 1).                                 |
| ...          | Arguments to be passed to <code>barplot</code> .  |

**Original URL**

<http://the-praise-of-insects.blogspot.ca/2012/04/dynamite-plots-in-r.html>

**Note**

Ben Bolker wrote an extensive discussion of the advantages and disadvantages of dynamite plots here: <http://emdbolker.wikidot.com/blog:dynamite>

**Author(s)**

Samuel Brown, modified by Mathieu Basille <basille@ufl.edu>

**Examples**

```
values <- c(1, 2, 5, 4)
errors <- c(0.25, 0.5, 0.33, 0.12)
names <- paste("Trial", 1:4)
sig <- c("a", "a", "b", "b")
dynamitePlot(values, errors)
par(mar = c(3, 5, 1, 1) + .1)
dynamitePlot(values, errors, names.arg = names, significance = sig,
  ylab = "Values", sym = TRUE, cex.lab = 1.5, cex.axis = 1.2,
  cex.names = 1.2, cex.sig = 1.5, space = c(0, 0.2, 0.8, 0.2),
  lwd = 2, head = 0, col = c(grey(0.5), "white"), border = c(NA,
    "black"))
```

extrange

*Extended range*

---

**Description**

Returns the range extended by a given proportion.

**Usage**

```
extrange(x, percent = 0.1, na.rm = FALSE)
```

**Arguments**

|         |  |
|---------|--|
| x       | A numeric vector.                              |
| percent | The proportion to be added to the range.       |
| na.rm   | Logical, indicating if NA's should be omitted. |

**Details**

If the regular range returns a single value, the proportion is computed on this value itself (and not on the range).

**Author(s)**

Mathieu Basille <basille@ufl.edu>

**Examples**

```
extrange(0:10)
extrange(0:10, percent = .5)
extrange(-10:10)
extrange(rep(10, 3))
```

---

getcolors*Choosing colors visually*

---

**Description**

Allows for the selection of n colors by using a simplified color swatch.

**Usage**

```
getcolors(n)
```

**Arguments**

|   |                                |
|---|--------------------------------|
| n | The number of colors to choose |
|---|--------------------------------|



## Details

`getcolors` allows selection with a mouse using the `locator` function. Following selection, a second plot opens showing how these colors look next to each other and on a background gradient of black to white. The function uses an RGB color model: Red increases on the y-axis, Green increases on the x-axis, and Blue is a repeated sequence of levels across the x-axis.

## Value

A character vector with elements of 7 or 9 characters, `"#"` followed by the red, blue, green and optionally alpha values in hexadecimal (after rescaling to 0 ... 255). The optional alpha values range from 0 (fully transparent) to 255 (opaque).

## Original URL

<http://menugget.blogspot.com/2013/01/choosing-colors-visually-with-getcolors.html>

## Author(s)

Marc in the box

## Examples

```
## Not run:
set.seed(1)
n <- 100
x <- seq(n)
y1 <- cumsum(rnorm(n))
y2 <- cumsum(rnorm(n))
y3 <- cumsum(rnorm(n))
y4 <- cumsum(rnorm(n))
ylim <- range(c(y1, y2, y3, y4))

cols <- getcolors(4)

plot(x, y1, ylim = ylim, t = "l", col = cols[1], lwd = 3, ylab = "")
lines(x, y2, col = cols[2], lwd = 3)
lines(x, y3, col = cols[3], lwd = 3)
lines(x, y4, col = cols[4], lwd = 3)
legend("topleft", legend = paste("y", 1:4, sep = ""), col = cols,
      lwd = 3)
## End(Not run)
```

## Description

Generate package reference manual. This function requires the `devtools` package.

**Usage**

```
manual(pkg = ".", path = NULL, preview = TRUE, overwrite = FALSE)
```

**Arguments**

|           |   |
|-----------|---|
| pkg       | package description, can be path or package name. See <a href="#">as.package</a> for more information |
| path      | path in which to produce package. If NULL, defaults to the root directory of the package.             |
| preview   | preview generated PDF file  |
| overwrite | overwrite output file if it exists  |

**Author(s)**

Mathieu Basille <basille@ufl.edu>

---

memUse

*Memory usage of saved objects*


---

**Description**

Show memory usage of saved objects.

**Usage**

```
memUse(pos = 1, pattern)

## S3 method for class 'memUse'
print(x, sort = c("size", "alphabetical"),
      decreasing = ifelse(sort == "size", TRUE, FALSE), n = 10, bytes = FALSE,
      ...)
```

**Arguments**

|            |   |
|------------|---|
| pos        | Which environment to use to list the saved objects (as a position in the search list). See <a href="#">ls</a> . |
| pattern    | An optional regular expression. See <a href="#">ls</a> .  |
| x          | An object of class memUse.  |
| sort       | Whether to sort the object list by size or alphabetical order.  |
| decreasing | Logical. Should the sort order be increasing or decreasing?   |
| n          | A single integer, giving the number of objects to display.  |
| bytes      | Logical. Whether to display the raw size in bytes.  |

**Value**

A data frame of classe memUse providing object names, class and memory usage; or 0 if no object is present in the specified environment.

**Original URL**

<http://stackoverflow.com/a/9839949>. See also the complete Stack Overflow thread: <https://stackoverflow.com/questions/1358003/tricks-to-manage-the-available-memory-in-an-r-session>

**Author(s)**

Michael Hallquist, modified by Mathieu Basille <basille@ufl.edu>

**Examples**

```
## Create some data
rand <- rnorm(100)
data(iris)
foo <- function(x) print(x)
##'
## Check memory usage (and raw numbers in bytes)
memUse()
memUse()$bytes
```

---

mv

*Rename an R object.*

---

**Description**

Rename an R object.

**Usage**

```
mv(from, to)
```

**Arguments**

|      |  |
|------|--|
| from | The name of an R object, with or without quotes. |
| to   | The new name, with or without quotes.            |

**Author(s)**

Jean Lobry

**Examples**

```
bla <- 2
ls()
mv(bla, bli)
bli
ls()
```

---

ncut

---

*Cut into classes of equal size*


---

**Description**

Cut a numeric vector into k classes of (roughly) equal size.

**Usage**

```
ncut(x, k = 10, labels = FALSE)
```

**Arguments**

|        |  |
|--------|--|
| x      | numeric vector to cut.   |
| k      | the number of classes.   |
| labels | Labels for the levels of the resulting category. By default (FALSE), labels are constructed using simple integer codes; if NULL, labels are build using "(a,b]" interval notation. Alternatively, a custom vector of length k can be used. |

**Value**

A vector of k classes.

**Author(s)**

Mathieu Basille <basille@ufl.edu> and Emiel van Loon

**Examples**

```
bla <- rnorm(100)
summary(bla)
bli <- ncut(bla)
table(bli)
head(ncut(bla, labels = NULL))
head(ncut(bla, labels = LETTERS[1:10]))
```

---

|         |  |
|---------|--|
| nsubset | <i>Subset according to frequency of a column</i> |
|---------|--|

---

## Description

Subset an object based on the frequency of a column (factor or not).

## Usage

```
nsubset(x, col, n, sign = c(">=", "<=", ">", "<", "=="), select,
        drop = FALSE)
```

## Arguments

|        |   |
|--------|---|
| x      | object to be subsetted (e.g. data.frame, matrix, Spatial*DataFrame, etc.).  |
| col    | name of the column which stores the grouping factor (without quotes).   |
| n      | the reference number for the frequency of the grouping factor (must be a numeric of length 1).                      |
| sign   | any comparison sign (>= by default, <=, >, <, or ==).   |
| select | expression, indicating columns to select (either name(s) without quote or numeric indicating the column number(s)). |
| drop   | passed on to '[' indexing operator.   |

## Value

A subset of the object (with the same class).

## Author(s)

Mathieu Basille <basille@ufl.edu>

## Examples

```
set.seed(1)
bla <- data.frame(value = rnorm(100), group = sample(letters[1:4],
  size = 100, replace = TRUE, prob = (1:4) * 10))
table(bla$group)
bli <- nsubset(bla, group, 25)
table(bli$group)
blo <- nsubset(bla, group, 25, sign = ">", select = 1, drop = TRUE)
head(blo)
length(blo)
```

---

|   |                               |
|---|-------------------------------|
| q | <i>Terminate an R Session</i> |
|---|-------------------------------|

---

### Description

A modified version of [quit](#) or its alias [q](#). See [quit](#) for the function details.

### Usage

```
q(save = "default", status = 0, runLast = TRUE)
```

```
quit(save = "default", status = 0, runLast = TRUE)
```

### Details

If `save = "yes"`, the list of attached packages is automatically saved in a file `.Rpackages`. See [savepkglist](#) for more details.

### Author(s)

R Core Team, modified by Mathieu Basille <basille@ufl.edu>

---

|         |   |
|---------|---|
| reclass | <i>Reclassify the values of a vector.</i> |
|---------|---|

---

### Description

Reclassify given values of a vector by new values. Note that all values need not to be documented, only the ones that need to be modified.

### Usage

```
reclass(x, from, to = NULL, factor = FALSE, ...)
```

### Arguments

|        |   |
|--------|---|
| x      | A character or numeric vector.  |
| from   | A vector describing the values to change from, or a matrix of reclassification with two columns (from, to). |
| to     | A vector describing the values to change to, or nothing if from is a matrix.                                |
| factor | Logical, whether to return a factor (default is FALSE).   |
| ...    | Additional arguments passed to factor.  |

**Value**

A vector with the same length as x.

**Author(s)**

Mathieu Basille <basille@ufl.edu>

**Examples**

```
(bla <- rep(1:5, 3))
reclass(bla, c(3, 4), c(7, 3))
reclass(bla, c(3, 4), c("a", "b"))

## Conversion as a factor
reclass(bla, c(3, 4), c("a", "b"), factor = TRUE)
(bli <- rep(letters[1:5], 3))
reclass(bli, c("b", "d"), c(1, 2))

## With a matrix of reclassification
(mat <- matrix(c("b", "d", 1, 2), ncol = 2))
reclass(bli, mat)

## Fast computation time on large vectors
blu <- rpois(1e6, 10)
system.time(reclass(blu, c(3, 4), c(7, 3)))
```

---

 save.image

*Save the current workspace*


---

**Description**

A modified version of [save.image](#) that allows to save the commands history and the list of attached packages. See [save.image](#) for the function details.

**Usage**

```
save.image(file = ".RData", version = NULL, ascii = FALSE,
  compress = !ascii, safe = TRUE, hist = TRUE, h.file = ".Rhistory",
  pkglist = TRUE, p.file = ".Rpackages")
```

**Arguments**

|         |   |
|---------|---|
| hist    | Logical. Whether to save or not the commands history.   |
| h.file  | The name of the file in which to save the history, or from which to load it. The path is relative to the current working directory.                   |
| pkglist | Logical. Whether to save or not the list of attached packages (default is TRUE).  |
| p.file  | The name of the file in which to save the list of attached packages, or from which to load it. The path is relative to the current working directory. |

**Author(s)**

R Core Team, modified by Mathieu Basille <basille@ufl.edu>

**See Also**

[savehistory](#) to save the commands history, and [savepkglist](#) to save the list of attached packages.

---

savepkglist

*Load or save the list of attached packages*

---

**Description**

Display, save or load the list of attached packages.

**Usage**

```
savepkglist(file = ".Rpackages")
```

```
attpkglist()
```

```
loadpkglist(file = ".Rpackages")
```

```
.loadpkglist()
```

**Arguments**

|      |   |
|------|---|
| file | The name of the file in which to save the list of attached packages, or from which to load it. The path is relative to the current working directory. |
|------|---|

**Details**

attpkglist simply lists all attached packages (i.e. not base packages).

savepkglist saves the list of all attached packages in a file, with one package per line.

loadpkglist loads a list of packages from a file. The file should contain one package name per line, without quotes, and no empty line. If the packages are not installed, the function sends a warning.

.loadpkglist automatically loads the .Rpackages file at startup (see the Note below).

**Note**

To automatically load a .Rpackages list at startup, add this in your .Rprofile:

```
### Load packages at the start of R if the package list exists  
basr:::loadpkglist()
```

Essentially, the function appends the list of packages at the end of the defaultPackages option (see [options](#) for this option; see also [Startup](#) for more details about the initialization at start of an R session).



**Author(s)**

Mathieu Basille <basille@ufl.edu>

**Examples**

```
## Not run: savepkglist(file = "list.Rpackages")
## Not run: attpkglist()
## Not run: loadpkglist()
```

---

| se | <i>Standard errors</i> |
|----|------------------------|
|----|------------------------|

---

**Description**

This function computes the standard error (i.e.  $\text{sd} / \sqrt{n}$ ) of the values in `x`. If `na.rm` is `TRUE` then missing values are removed before computation proceeds.

**Usage**

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

**Arguments**

|                    |   |
|--------------------|---|
| <code>x</code>     | A numeric vector or an R object which is coercible to one by <code>as.vector</code> . |
| <code>na.rm</code> | Logical. Should missing values be removed?  |

**Original URL**

<http://cran.r-project.org/doc/manuals/R-intro.html>

**Author(s)**

From the Writing R Extensions manual, modified by Mathieu Basille <basille@ufl.edu>

**See Also**

[var](#) and [sd](#) for the variance and standard deviation.

**Examples**

```
bla <- rnorm(1000, sd = 100)
sd(bla)
sqrt(var(bla)/length(bla))
se(bla)

is.na(bla) <- 200:300
sd(bla, na.rm = TRUE)
se(bla, na.rm = TRUE)
```

summaryIC

*Summary of AIC/BIC***Description**

Summarizes IC differences and weights to identify plausible models (models with highest empirical support).

**Usage**

```
summaryIC(x, delta = 2)
```

**Arguments**

|       |   |
|-------|---|
| x     | A data frame with one row per model, and one column giving an information-theoretic index (AIC or BIC), such as returned by <a href="#">AIC</a> or <a href="#">BIC</a> with several models. |
| delta | The difference threshold to identify models with similar support.   |

**Value**

The input data frame with additional columns `delta` giving IC differences with the best model, best identifying the best model(s) (\* for the absolute best model, + for models within the threshold), and `omega` giving AIC weights.

**Author(s)**

Mathieu Basille <basille@ufl.edu>

**Examples**

```
## Prepare two models:
lm1 <- lm(Fertility ~ . , data = swiss)
lm2 <- update(lm1, . ~ . -Examination)

## Check AIC:
AIC(lm1, lm2)

## Summary of AIC and BIC:
summaryIC(AIC(lm1, lm2))
summaryIC(BIC(lm1, lm2))
```

---

|       |  |
|-------|--|
| table | <i>Modified table function to handle NAs</i> |
|-------|--|

---

**Description**

A slight modification of the [table](#) function, to include NA values in the table by default. See [table](#) for details of the function.

**Usage**

```
table(..., exclude = if (useNA == "no") c(NA, NaN), useNA = c("ifany", "no",  
  "always"), dnn = list.names(...), deparse.level = 1)
```

**Arguments**

useNA                      Whether to include NA values in the table. Default is now ifany.

**Author(s)**

R Core Team, modified by Mathieu Basille <basille@ufl.edu>

**Examples**

```
d <- factor(rep(c("A", "B", "C"), 10), levels = c("A", "B", "C",  
  "D", "E"))  
is.na(d) <- 3:4  
d  
table(d)
```

---

|        |   |
|--------|---|
| togray | <i>Convert continuous variable to grey levels</i> |
|--------|---|

---

**Description**

Convert a continuous variable to the corresponding levels of grey.

**Usage**

```
togray(x, min = 0.1, max = 0.9, alpha = NULL, inverse = FALSE,  
  sqrt = FALSE)  
  
togrey(x, min = 0.1, max = 0.9, alpha = NULL, inverse = FALSE,  
  sqrt = FALSE)
```

**Arguments**

|         |  |
|---------|--|
| x       | A numeric vector.  |
| min     | The minimum grey level.  |
| max     | The maximum grey level.  |
| alpha   | The opacity.   |
| inverse | Logical. By default, bigger is darker. If <code>inverse = TRUE</code> , bigger is lighter. |
| sqrt    | Logical. Applies a square root transformation to get more progressive grey levels.         |

**Value**

A vector of colors of the same length as x.

**Author(s)**

From Clement Calenge, modified by Mathieu Basille <basille@ufl.edu>

**Examples**

```
bla <- runif(10000)
plot(bla, col = togray(bla, 0, 1), pch = 20)
plot(bla, col = togray(bla, 0, 1, sqrt = TRUE), pch = 20)
plot(bla, col = togray(bla, 0, 1, alpha = 0.5), pch = 20)
```

---

writeFunction

*Function output*

---

**Description**

Prints a function to a file.

**Usage**

```
writeFunction(fun, file = NULL)
```

**Arguments**

|      |   |
|------|---|
| fun  | A function.   |
| file | A character string naming a file. By default, write the function in <fun>.R in the working directory. |

**Author(s)**

Mathieu Basille <basille@ufl.edu>

**Examples**

```
f1 <- function(x) {  
  ## Comment  
  print(x)  
}  
writeFunction(f1)  
rm(f1)  
source("f1.R")  
file.remove("f1.R")  
f1(3)
```

# Index

`.loadpkglist (savepkglist)`, 16

AIC, 18

`as.package`, 10

`attpkglist (savepkglist)`, 16

`basr`, 2

`basr-package (basr)`, 2

BIC, 18

`capwords`, 2

`chartr`, 3

`colNA`, 3

`confint`, 4, 4, 5

`cv`, 5

`dynamitePlot`, 6

`exrange`, 8

`getcolors`, 8

`loadpkglist (savepkglist)`, 16

`ls`, 10

manual, 9

`memUse`, 10

`mv`, 11

`ncut`, 12

`nsubset`, 13

options, 16

`par`, 4

`plot.confint (confint)`, 4

`print.memUse (memUse)`, 10

`q`, 14, 14

`quit`, 14

`quit (q)`, 14

`reclass`, 14

`save.image`, 15, 15

`savehistory`, 16

`savepkglist`, 14, 16, 16

`sd`, 17

`se`, 17

`split`, 4

Startup, 16

`summaryIC`, 18

`table`, 19, 19

`togray`, 19

`togrey (togray)`, 19

`var`, 17

`writeFunction`, 20