Package: multcompView (via r-universe)

September 1, 2024

Type Package

Title Visualizations of Paired Comparisons

Version 0.1-10

Date 2024-04-04

Author Spencer Graves, Hans-Peter Piepho and Luciano Selzer with help
from Sundar Dorai-Raj

Maintainer Luciano Selzer <luciano.selzer@gmail.com>

Description Convert a logical vector or a vector of p-values or a
correlation, difference, or distance matrix into a display
identifying the pairs for which the differences were not
significantly different. Designed for use in conjunction with
the output of functions like TukeyHSD, dist{stats}, simint,
simtest, csimint, csimtest{multcomp}, friedmanmc,
kruskalmc{pgirmess}.

License GPL

Imports grid

Suggests multcomp, pgirmess, MASS

RoxygenNote 7.2.3

Repository https://lselzer.r-universe.dev

RemoteUrl https://github.com/lselzer/multcompview

RemoteRef HEAD

RemoteSha db4cdc06f9fec9e7a39ab0ab119cdfef4d95711e

Contents

---

multcompView-package   *Summarize multiple paired comparisons*

Description

Convert a logical vector or a vector of p-values or a difference or distance matrix into a display
identifying the pairs for which the differences were not significantly different or for which the
difference exceeded a threshold.

Details

Package:	multcompView
Type:	Package
Version:	0.1-1
Date:	2006-08-06
License:	GPL

Convert a logical vector or a vector of p-values or a difference or distance matrix into either a
letter-based display using "multcompLetters" or a graphic roughly like a "T" using "multcompTs"
to identify factor levels or similar groupings that are or are not significantly different. Designed for
use in conjunction with the output of functions like TukeyHSD, diststats, simint, simtest, csimint,
csimtestmultcomp, friedmanmc, kruskalmcpgirmess.

Author(s)

Spencer Graves and Hans-Peter Piepho with help from Sundar Dorai-Raj

Maintainer: Spencer Graves <spencer.graves@prodsyse.com>

References

Piepho, Hans-Peter (2004) "An Algorithm for a Letter-Based Representation of All-Pairwise Com-
parisons", Journal of Computational and Graphical Statistics, 13(2)456-466.

John R. Donaghue (2004) "Implementing Shaffer's multiple comparison procedure for a large num-
ber of groups", pp. 1-23 in Benjamini, Bretz and Sarkar (eds) Recent Developments in Multiple
Comparison Procedures (Institute of Mathematical Statistics Lecture Notes-Monograph Series vol.
47)

Examples

```
dif3 <- c(FALSE, FALSE, TRUE)
names(dif3) <- c("a-b", "a-c", "b-c")
multcompTs(dif3)
multcompLetters(dif3)

library(MASS)
multcompBoxplot(Postwt~Treat, data=anorexia)
```

---

extract_p   *Extracts p-values*

Description

For a given object it will look for the column named "p adj" or "difference" and extract its value keeping its names

Usage

```
extract_p(x)

## Default S3 method:
extract_p(x)

## S3 method for class 'TukeyHSD'
extract_p(x)

## S3 method for class 'mc'
extract_p(x)
```

Arguments

x   A object that has p-values or logical values.

Value

A named vector with p-values or logical values.

Methods (by class)

- `extract_p(default)`:
- `extract_p(TukeyHSD)`: extract p values from a TukeyHSD object
- `extract_p(mc)`:

Author(s)

Luciano Selzer

See Also

multcompLetters multcompTs

Examples

```
experiment <- data.frame(treatments = gl(11, 20, labels = c("dtl", "ctrl", "treat1",
               "treat2", "treatA2", "treatB", "treatB2",
               "treatC", "treatD", "treatA1", "treatX")),
            y = c(rnorm(20, 10, 5), rnorm(20, 20, 5), rnorm(20, 22, 5), rnorm(20, 24, 5),
               rnorm(20, 35, 5), rnorm(20, 37, 5), rnorm(20, 40, 5), rnorm(20, 43, 5),
               rnorm(20, 45, 5), rnorm(20, 60, 5), rnorm(20, 60, 5)))
exp_tukey <- TukeyHSD(exp_aov <- aov(y  ~ treatments, data = experiment))

extract_p(exp_tukey)

if(require(pgirmess)){
extract_p(kruskalmc(y ~ treatments, data = experiment))
}
```

---

multcompBoxplot	*boxplot with multcomp graphics*

Description

Create boxplots with multcompTs and / or multcompLetters

Usage

```
multcompBoxplot(
  formula,
  data,
  horizontal = TRUE,
  compFn = "TukeyHSD",
  sortFn = "mean",
  decreasing = TRUE,
 plotList = list(boxplot = list(fig = c(0, 0.75, 0, 1)), multcompTs = list(fig = c(0.7,
   0.85, 0, 1)), multcompLetters = list(fig = c(0.87, 0.97, 0.03, 0.98), fontsize = 20,
     fontface = "bold"))
)
```

Arguments

formula   a two sided formula like "y~z", where both "y" and "z" are columns of the
                 data.frame "data", "y" is numeric, and "z" is a factor. This will be passed as
                 the first argument for both 'boxplot' and 'compFn', and so must work in both
                 contexts. NOTE: Any more complicated formula may produce errors or unan-
                 ticipated results.

data	A data.frame for evaluating 'formula'.
horizontal	TRUE for horizontal boxplots and vertical multcompTs and / or multcompLetters; FALSE for the opposite.
compFn	a function whose output will serve as the the only non-default input to either 'multcompTs' or 'multcompLetters'. The default "TukeyHSD" actually translates to 'TukeyHSD(aov(formula, data))[[1]][, "p adj"]'.
sortFn	If sortFn is a function or a character string naming a function, it is used to summarize the subset of y corresponding to each level of z into a single number, which will then be used to sort the levels of z according to the argument 'decreasing'. This step is skipped if sortFn is NULL or NA or if it is neither a function nor a character string that might name a function. If sortFn is a character string but a function by that name is not found in the search path, multcompBoxplot stops with, 'Error in do.call(sortFn, list(x = x)) : could not find function ...'.
decreasing	If the levels of z are to be sorted using the output of 'sortFn', this is uses as the 'decreasing' in 'order' to sort the levels of z for plotting.
plotList	A list with names in c("boxplot", "multcompTs", "multcompLetters"). Replicates are allowed. If present, they produce, e.g., multiple "multcompTs" side by side. This can be used to compare the visual effects of different arguments to "plot.multcompTs". Each component of 'plotList' is itself a list of arguments to pass to either "boxplot", "plot.multcompTs" or "plot.multcompLetters". Placement can be controlled via 'fig' arguments passed (indirectly) of the form 'c(x1, x2, y1, y2)'. If(horizontal==TRUE), fig gives the coordinates of the figure region in the display region of the plot device, as described on the [par](#) help page; if(horizontal==FALSE), fig[c(3,4,1,2)] gives c(x1, x2, y1, y2) for placement of that portion of the plot.

Details

For formula = y~z, if 'sortFn' is a function or the name of a function, 'multcompBoxplot' starts by applying sortFn to the subsets of y corresponding to each level of z, and then sorting those summaries in increasing or decreasing order, per 'decreasing'. If 'sortFn' is NULL or NA, this sort step is skipped.

'multcompBoxplot' then creates 'boxplot' as specified in 'plotList'. Next, 'compFn' is called to generate comparisons to feed to the functions ([multcompTs](#) and / or [multcompLetters](#), whose output is then passed to ([plot.multcomp](#)) for plotting. Components of the relevant sublists of 'plotList' are made available to [par](#) or (for [plot.multcompLetters](#)) to [gpar](#).

Value

This function invisibly returns a list with one component for each component of plotList containing the output of the appropriate "plot.multcomp" call plus the output of "compFn".

Author(s)

Spencer Graves

See Also

boxplot multcompTs multcompLetters plot.multcomp TukeyHSD par gpar

Examples

```
# Example from help("TukeyHSD")
multcompBoxplot(breaks~tension, data=warpbreaks)
# 'sortFn' can be either a function or a function name
# default order is 'decreasing=TRUE'
multcompBoxplot(breaks~tension, data=warpbreaks,
        sortFn=median, decreasing=FALSE)


##################
library(multcomp)
data(recovery)
# Horizontal boxplots with both
# multcomp Ts and Letters on the right
# Using recovery{multcomp} data set
multcompBoxplot(minutes~blanket, recovery)


# Plotting boxes rather than letters and Ts

multcompBoxplot(minutes~blanket, recovery,
                plotList=list(
                    boxplot=list(fig=c(0, 0.75, 0, 1), las=1,
                        cex.axis=1.5),
                    multcompTs=list(fig=c(0.7, 0.85, 0, 1),
                        type='boxes'),
                    multcompLetters=list(
                        fig=c(0.87, 0.97, 0.03, 0.98),
                        type='boxes') ) )


###################

# Vertical boxplots with both
# multcomp Ts and Letters on the top
multcompBoxplot(minutes~blanket, recovery,
                        horizontal=FALSE)

# Horizontal boxplots with 2 different
# displays of the "Ts" on the left
multcompBoxplot(minutes~blanket, recovery,
  plotList=list(
      boxplot=list(fig=c(0.3, 1, 0, 1)),
      multcompTs=list(fig=c(0, 0.15, 0, 1),
        orientation="reverse"),
      multcompTs=list(fig=c(0.15, 0.3, 0, 1),
        type="boxes", orientation="reverse",
        mar=c(5,2, 4, 0)+.1) ) )

library(MASS)
anorx <-
```

```
multcompBoxplot(Postwt~Treat, data=anorexia)

## Not run:
# Confirm than sortFn=NULL or NA
# leaves the order unchanged
library(multcomp)
data(cholesterol)
cholesterol$trt3 <- with(cholesterol, factor(
  as.character(trt), levels=levels(trt)[c(5:4,1:3)]))
multcompBoxplot(response ~ trt3, cholesterol,
          sortFn=NULL)
multcompBoxplot(response ~ trt3, cholesterol,
          sortFn=NA)

## End(Not run)
```

---

multcompLetters | *Letter summary of similarities and differences*

Description

Convert a logical vector or a vector of p-values or a correlation or distance matrix into a character-based display in which common characters identify levels or groups that are not significantly different. Designed for use with the output of functions like TukeyHSD, diststats, simint, simtest, csimint, csimtestmultcomp, friedmanmc, kruskalmcpgirmess.

Usage

```
multcompLetters(
  x,
  compare = "<",
  threshold = 0.05,
  Letters = c(letters, LETTERS, "."),
  reversed = FALSE
)

multcompLetters2(formula, x, data, ...)

multcompLetters3(z, y, x, data, ...)

multcompLetters4(object, comp, ...)
```

Arguments

x   One of the following: (1) A square, symmetric matrix with row names. (2) A vector with hyphenated names, which identify individual items or factor levels after "strsplit". (3) An object of class "dist". If x (or x[1]) is not already of

class "logical", it is replaced with do.call(compare, list(x, threshold)), which by default converts numbers (typically p-values) less than 0.05 to TRUE and everything else to FALSE. If x is a matrix, its diagonal must be or must convert to FALSE.

compare	function or binary operator; not used if class(x) is "logical".
threshold	Second (reference) argument to "compare".
Letters	Vector of distinct characters (or character strings) used to connect levels that are not significantly different. They should be recognizable when concatenated. The last element of "Letters" is used as a prefix for a reuse of "Letters" if more are needed than are provided. For example, with the default "Letters", if 53 distinct connection columns are required, they will be "a", ..., "z", "A", ..., "Z", and ".a". If 54 are required, the last one will be ".b". If 105 are required, the last one will be "..a", etc. (If the algorithm generates that many distinct groups, the display may be too busy to be useful, but the algorithm shouldn't break.)
reversed	A logical value indicating whether the order of the letters should be reversed. Defaults to FALSE.
formula	The formula used to make the test (lm, aov, glm, etc.). Like y ~ x.
data	Data used to make the test.
...	Extra arguments passed to multcompLetters.
z	Categorical variables used in the test.
y	Value of the response variable.
object	An object of class aov or lm for the time being.
comp	A object with multiple comparison or a function name to perform a multiple comparison.

Details

Produces a "Letter-Based Representation of All- Pairwise Comparisons" as described by Piepho (2004). (The present algorithm does NOT perform his "sweeping" step.) multcompLettersx are wrapper of multcompLetters that will reorder the levels of the data so that the letters appear in a descending order of the mean. mulcompletters3 is similar to multcompletters2 except that it uses vector names to separate and the later has an formula interface. multcompLetters4 will take a aov or lm object and a comparison test and will produce all the letters for the terms and interactions.

Value

An object of class 'multcompLetters', which is a list with the following components:

Letters	character vector with names = the names of the levels or groups compared and with values = character strings in which common values of the function argument "Letters" identify levels or groups that are not significantly different (or more precisely for which the corresponding element of "x" was FALSE or was converted to FALSE by "compare").
monospacedLetters	
	Same as "Letters" but with spaces so the individual grouping letters will line up with a monospaced type font.

LetterMatrix   Logical matrix with one row for each level compared and one column for each "Letter" in the "letter-based representation". The output component "Letters" is obtained by concatenating the column names of all columns with TRUE in that row.

multcompLetters4 will return a named list with the terms containing a object of class 'multcompLetters' as produced by `multcompLetters`.

Functions

- `multcompLetters2()`:
- `multcompLetters3()`: create a compact letters display and order the letters
- `multcompLetters4()`: create a compact letters display using a aov object

Author(s)

Spencer Graves, Hans-Peter Piepho and Luciano Selzer

References

Piepho, Hans-Peter (2004) "An Algorithm for a Letter-Based Representation of All-Pairwise Comparisons", Journal of Computational and Graphical Statistics, 13(2)456-466.

See Also

[multcompBoxplot](#) [plot.multcompLetters](#) [print.multcompLetters](#) [multcompTs](#) [vec2mat](#)

Examples

```
##
## 1.  a logical vector indicating signficant differences
##
dif3 <- c(FALSE, FALSE, TRUE)
names(dif3) <- c("A-B", "A-C", "B-C")
dif3L <- multcompLetters(dif3)
dif3L
print(dif3L)
print(dif3L, TRUE)

##
## 2.  numeric vector indicating statistical significance
##
dif4 <- c(.01, .02, .03, 1)
names(dif4) <- c("a-b", "a-c", "b-d", "a-d")
(diff4.T <- multcompLetters(dif4))

(dif4.L1 <- multcompLetters(dif4,
      Letters=c("*", ".")))
# "Letters" can be any character strings,
# but they should be recognizable when
# concatenated.
```

```
##
## 3.  distance matrix
##
dJudge <- dist(USJudgeRatings)
dJl <- multcompLetters(dJudge, compare='>', threshold = median(dJudge))
# comparison of 43 judges;   compact but undecipherable:
dJl

x <- array(1:9, dim=c(3,3),
   dimnames=list(LETTERS[1:3], NULL) )
d3 <- dist(x)
dxLtrs <- multcompLetters(d3, compare=">", threshold=2)

d3d <- dist(x, diag=TRUE)
dxdLtrs <- multcompLetters(d3d, compare=">", threshold=2)


all.equal(dxLtrs, dxdLtrs)


d3u <- dist(x, upper=TRUE)
dxuLtrs <- multcompLetters(d3d, compare=">", threshold=2)


all.equal(dxLtrs, dxuLtrs)


##
## 4.  cor matrix
##
set.seed(4)
x100 <- matrix(rnorm(100), ncol=5,
              dimnames=list(NULL, LETTERS[1:5]) )
cx <- cor(x100)
cxLtrs <- multcompLetters(abs(cx), threshold=.3)


##
##5. reversed
##

dif3 <- c(FALSE, FALSE, TRUE)
names(dif3) <- c("A-B", "A-C", "B-C")
dif3L <- multcompLetters(dif3)
dif3L.R <- multcompLetters(dif3, rev = TRUE)
dif3L
dif3L.R


##
##6. multcompletters2 usage
```

```
experiment <- data.frame(treatments = gl(11, 20, labels = c("dtl", "ctrl", "treat1",
                  "treat2", "treatA2", "treatB", "treatB2",
                  "treatC", "treatD", "treatA1", "treatX")),
                y = c(rnorm(20, 10, 5), rnorm(20, 20, 5), rnorm(20, 22, 5), rnorm(20, 24, 5),
                  rnorm(20, 35, 5), rnorm(20, 37, 5), rnorm(20, 40, 5), rnorm(20, 43, 5),
                  rnorm(20, 45, 5), rnorm(20, 60, 5), rnorm(20, 60, 5)))
exp_tukey <- TukeyHSD(exp_aov <- aov(y  ~ treatments, data = experiment))
exp_letters1 <- multcompLetters(exp_tukey$treatments[,4])
exp_letters1
#Notice lowest mean treatments gets a "e"
#Ordered letters
multcompLetters2(y ~ treatments, exp_tukey$treatments[,"p adj"], experiment)
multcompLetters2(y ~ treatments, exp_tukey$treatments[,"p adj"], experiment, reversed = TRUE)

##7. multcompletters3 usage

multcompLetters3("treatments", "y", exp_tukey$treatments[,"p adj"], experiment)

##8. multcompletters4 usage


multcompLetters4(exp_aov, exp_tukey)
```

---

multcompTs	*"T" depiction of undifferentiated classes*

Description

Convert a logical vector or a vector of p-values or a correlation or distance matrix into a matrix with an associated visual display to show whether the differences between items exceed a threshold. Designed for use with the output of functions like TukeyHSD, diststats, simint, simtest, csimint, csimtestmultcomp, friedmanmc, kruskalmcpgirmess.

Usage

```
multcompTs(x, compare = "<", threshold = 0.05, sep = ".")
```

Arguments

x	One of the following: (1) A square, symmetric matrix with row names. (2) A vector with hyphenated names, which identify individual items or factor levels after "strsplit(..., '-')". (3) An object of class "dist".
	If x (or x[1]) is not already of class "logical", it is replaced with do.call(compare, list(x, threshold)), which by default converts numbers (typically p-values) less than 0.05 to TRUE and everything else to FALSE. If x is a matrix, its diagonal must be or must convert to FALSE.
compare	function or binary operator; not used if class(x) is "logical".

threshold   Second (reference) argument to "compare".

sep   Concatenation character for names of objects with identical similarity / dissim-
                 ilarity patterns. The output of multcompTs is matrix for which the number of
                 rows = (number of columns + number of uses of the "sep" character).

Details

Produces a matrix of class "multcompTs", describing the "undifferentiated classes" that identify the
other factor levels or items that are not distinct or not significantly different from the "base" of the
"T"; if two or more levels have the same pattern of significant differences, the two are combined into
one "T" with two "bases". The resulting T's are similar to the "undifferentiated classes" discussed
by Donaghue (2004).

Value

An object of class "multcompTs", which is a matrix of values -1, 0, 1, with one row for each level
compared and one column for each "T", read as follows: 1 = base of the "T" represented by that
column, 0 = level(s) not significantly different, and (-1) = levels(s) significantly different. If two
or more levels have the same pattern of significant and insignificant differences, they are combined
into a single column that can be represented by a "T" with multiple bases. The column name will
be a character string concatenating all row names with "1" in that column separated by the "sep"
character. Thus, the matrix should have as many 1's as it has rows. Also, the lower triangular
portion should have as many "-1's" as there are "TRUE" (e.g., significant) differences among the
comparisons.

Author(s)

Spencer Graves and Hans-Peter Piepho

References

John R. Donaghue (2004) "Implementing Shaffer's multiple comparison procedure for a large num-
ber of groups", pp. 1-23 in Benjamini, Bretz and Sarkar (eds) Recent Developments in Multiple
Comparison Procedures (Institute of Mathematical Statistics Lecture Notes-Monograph Series vol.
47)

Spencer Graves and Hans-Peter Piepho (2006) "Simple Visualizations of Paired Comparisons",
vignette("VisualizingPairedComparisons", package = "multcompView")

See Also

multcompBoxplot multcompLetters plot.multcompTs vec2mat dist

Examples

```
##
## 0.  Conference presentation comparing Ts and Letters
##
dir(system.file('doc', package='multcompView'),
    pattern='\\.pdf$', full.name=TRUE)
```

```
##
## 1.  logical vector indicating different pairs
##
dif3 <- c(FALSE, FALSE, TRUE)
names(dif) <- c("a-b", "a-c", "b-c")
multcompTs(dif3)


##
## 2.  numeric vector indicating statistical significance
##
dif4 <- c(.01, .02, .03, 1)
names(dif4) <- c("a-b", "a-c", "b-d", "a-d")
(diff4.T <- multcompTs(dif4))
plot(diff4.T)


##
## 3.  Distance matrix
##
dJudge <- dist(USJudgeRatings)
dJt <- multcompTs(dJudge, compare='>', threshold = median(dJudge))
# comparison of 43 judges;  compact but undecipherable:
plot(dJt, cex.axis=.5)

x <- array(1:9, dim=c(3,3),
    dimnames=list(LETTERS[1:3], NULL) )
d3 <- dist(x)
dxTs <- multcompTs(d3, compare=">", threshold=2)
plot(dxTs)

d3d <- dist(x, diag=TRUE)
dxdTs <- multcompTs(d3d, compare=">", threshold=2)


all.equal(dxTs, dxdTs)


d3u <- dist(x, upper=TRUE)
dxuTs <- multcompTs(d3d, compare=">", threshold=2)


all.equal(dxTs, dxuTs)


##
## 4.  cor matrix
##
set.seed(4)
x100 <- matrix(rnorm(100), ncol=5,
               dimnames=list(NULL, LETTERS[1:5]) )
cx <- cor(x100)
cxTs <- multcompTs(abs(cx), threshold=.3)
plot(cxTs)
```

plot.multcompLetters   *plot multcomp graphics*

Description

Plot graphic(s) for multcompTs or multcompLetters objects

Usage

```
## S3 method for class 'multcompLetters'
plot(
  x,
  horizontal = FALSE,
  col = 1:6,
  type = c("Letters", "boxes"),
  add = FALSE,
  at,
  width,
  fig = c(0, 1, 0, 1),
  label.levels = if (add) NA else 0.05,
  label.groups = NA,
  ...
)
```

Arguments

x	an object of class 'multcompTs' or 'multcompLetters'.
horizontal	A logical scalar indicating whether the list of items compared reads left to right (horizontal = TRUE) or top to bottom (horizontal = FALSE). If this multcomp graphic accompanies boxplots for different levels or groups compared, the 'boxplot' argument 'horizontal' is the negation of the multcomp plot 'horizontal' argument.
col	The color for each group of items or factor levels. The colors will cross the different items or factor levels and will therefore have the orientation specified via 'horizontal'. If the number of columns exceeds length(col), col is recycled. For alternative choices for col, see "Color Specification" in the par help page.
type	An alternative display for either multcompTs or multcompLetters is 'boxes' (or rectangles). If type="boxes" with "multcompTs", the "base(s)" of each "T" will be indicated by a triangle.
add	TRUE to add to an existing plot; FALSE to start a new plot. The names of the factor levels or items compared will be plotted only if add=FALSE.

at	A numeric vector or matrix or a list with components "at" and "width". If a list, both components must be either a numeric vector or matrix. The numeric vector "at" (whether the function argument or "at" component of the "at" list) must be either a numeric vector or matrix giving the locations where the "Ts" or "Letters" graphics should be drawn. length(at) is 1, 2 or 3 times the number of the number of factor levels or items compared. If length(at) is twice the number of levels or items compared, it gives the range of the display for that level; the base of a "T" will be at the midpoint. If length(at) is three times the number of items compared, the intermediate number will be the center of the base of the "T".
width	A numeric vector or matrix with as many rows as "Ts" or "groups" and with up to three columns. With one column, it will be the "center" of the plot range for that group. With two columns, they will delimit the range. With three, they will provide "bottom", "center", and "top" of the range for that set of grouping indicators. If "at" is a list, the argument "width" is ignored and is taken from the list "at".
fig	A numerical vector of the form 'c(x1, x2, y1, y2)' giving the coordinates of the figure region in the display region of the plot device, as described on the [par](#) help page.
label.levels	NA for no labels or distance away from the plot for the labels as a proportion of the plot range.
label.groups	NA for no labels or distance away from the plot for the labels as a proportion of the plot range.
...	graphical parameters can be given as described on the [plot](#) help page or for plot.multcompLetters as describe on the [gpar](#) help page.

Details

The requested graphic is either plotted by itself or added to an existing plot as specified by the arguments. The placement can be controlled by 'fig' and 'at'.

The fontsize and fontface of a plot of a multcompLetters object with 'type = "Letters"' can be adjusted as describe on the [gpar](#) help page.

Value

A list with two components:

at	A matrix with three columns giving the middle and extremes of the display for each of the factor levels or items compared.
width	A matrix with as many rows as "Ts" or comparator levels and with two columns giving the plot range for that comparator level.

Author(s)

Spencer Graves

See Also

multcompTs multcompLetters multcompBoxplot gpar

Examples

```
##
## plot(multcompTs(...))
##
dif4 <- c(.1, .02, .03, 1)
names(dif4) <- c("A-B", "A-C", "B-C", "A-D")
(mcT4 <- multcompTs(dif4))
# Standard plot, base of "Ts" point left
mcT4.1 <- plot(mcT4, label.groups=0.05)
# Redo using "at" = list
plot(mcT4, label.groups=0.05, at=mcT4.1)
# Same plot with group labels closer to the figure
plot(mcT4, label.groups=0.02)

# Base of "Ts" point right
plot(mcT4, label.groups=TRUE, orientation="r")
# Base of "Ts" point down
plot(mcT4, horizontal=TRUE, label.groups=0.05)
# Base of "Ts" point up
plot(mcT4, horizontal=TRUE, label.groups=0.05,
     orientation="r")

# Same 4 plots but with boxes & triangles, not Ts
plot(mcT4, label.groups=0.05, type="b")
plot(mcT4, label.groups=0.05, orientation="r",
     type="b")
plot(mcT4, horizontal=TRUE, label.groups=0.05,
     type="b")
plot(mcT4, horizontal=TRUE, label.groups=0.05,
     orientation="r", type="b")

##
## plot(multcompLetters(...))
##
# ... using dif4 from above
(mcL4 <- multcompLetters(dif4, Letters=LETTERS))
# Standard plot
## Not run:
# Requires (grid)
mcL4.1 <- plot(mcL4, label.groups=0.05)
# Redo using "at" = list
plot(mcL4, label.groups=0.05, at=mcL4.1)

# With bold face and larger font
plot(mcL4, label.groups=0.05,
     fontsize=28, fontface="bold")

# Horizontal rather than vertical
```

```
plot(mcL4, horizontal=TRUE, label.groups=0.05)

## End(Not run)

# Same as boxes rather than letters
plot(mcL4, label.groups=0.05, type="b")
plot(mcL4, horizontal=TRUE, label.groups=0.05,
     type="b")
```

---

plot.multcompTs   *plot multcomp graphics*

Description

Plot graphic(s) for multcompTs or multcompLetters objects

Usage

```
## S3 method for class 'multcompTs'
plot(
  x,
  horizontal = FALSE,
  col = 1:6,
  type = c("Ts", "boxes"),
  orientation = c("standard", "reverse"),
  add = FALSE,
  at,
  width,
  fig = c(0, 1, 0, 1),
  lwd = 3,
  label.levels = if (add) NA else 0.05,
  label.groups = NA,
  T.base = 0.4,
  ...
)
```

Arguments

x	an object of class 'multcompTs' or 'multcompLetters'.
horizontal	A logical scalar indicating whether the list of items compared reads left to right (horizontal = TRUE) or top to bottom (horizontal = FALSE). If this multcomp graphic accompanies boxplots for different levels or groups compared, the 'box-plot' argument 'horizontal' is the negation of the multcomp plot 'horizontal' argument.

col	The color for each group of items or factor levels. The colors will cross the different items or factor levels and will therefore have the orientation specified via 'horizontal'. If the number of columns exceeds length(col), col is recycled. For alternative choices for col, see "Color Specification" in the [par](#) help page.
type	An alternative display for either multcompTs or multcompLetters is 'boxes' (or rectangles). If type="boxes" with "multcompTs", the "base(s)" of each "T" will be indicated by a triangle.
orientation	The "standard" orientation has the 'multcompTs' pointing towards the names of the items or factor levels; with the "reverse" orientation, the bases of the "Ts" point away. By default, the names are on the left or below unless the mean of the relevant 'fig' range is less than 0.5.
add	TRUE to add to an existing plot; FALSE to start a new plot. The names of the factor levels or items compared will be plotted only if add=FALSE.
at	A numeric vector or matrix or a list with components "at" and "width". If a list, both components must be either a numeric vector or matrix. The numeric vector "at" (whether the function argument or "at" component of the "at" list) must be either a numeric vector or matrix giving the locations where the "Ts" or "Letters" graphics should be drawn. length(at) is 1, 2 or 3 times the number of the number of factor levels or items compared. If length(at) is twice the number of levels or items compared, it gives the range of the display for that level; the base of a "T" will be at the midpoint. If length(at) is three times the number of items compared, the intermediate number will be the center of the base of the "T".
width	A numeric vector or matrix with as many rows as "Ts" or "groups" and with up to three columns. With one column, it will be the "center" of the plot range for that group. With two columns, they will delimit the range. With three, they will provide "bottom", "center", and "top" of the range for that set of grouping indicators. If "at" is a list, the argument "width" is ignored and is taken from the list "at".
fig	A numerical vector of the form 'c(x1, x2, y1, y2)' giving the coordinates of the figure region in the display region of the plot device, as described on the [par](#) help page.
lwd	width of line to connect elements of "T" graphics that might not otherwise be connected.
label.levels	NA for no labels or distance away from the plot for the labels as a proportion of the plot range.
label.groups	NA for no labels or distance away from the plot for the labels as a proportion of the plot range.
T.base	A numeric scalar giving the proportion of the available space devoted to the base of the Ts; used only when type="Ts".
...	graphical parameters can be given as described on the [plot](#) help page or for plot.multcompLetters as describe on the [gpar](#) help page.

Details

The requested graphic is either plotted by itself or added to an existing plot as specified by the arguments. The placement can be controlled by 'fig' and 'at'.

The fontsize and fontface of a plot of a multcompLetters object with 'type = "Letters"' can be adjusted as describe on the [gpar](#) help page.

Value

A list with two components:

at	A matrix with three columns giving the middle and extremes of the display for each of the factor levels or items compared.
width	A matrix with as many rows as "Ts" or comparator levels and with two columns giving the plot range for that comparator level.

Author(s)

Spencer Graves

See Also

[multcompTs](#) [multcompLetters](#) [multcompBoxplot](#) [gpar](#)

Examples

```
##
## plot(multcompTs(...))
##
dif4 <- c(.1, .02, .03, 1)
names(dif4) <- c("A-B", "A-C", "B-C", "A-D")
(mcT4 <- multcompTs(dif4))
# Standard plot, base of "Ts" point left
mcT4.1 <- plot(mcT4, label.groups=0.05)
# Redo using "at" = list
plot(mcT4, label.groups=0.05, at=mcT4.1)
# Same plot with group labels closer to the figure
plot(mcT4, label.groups=0.02)

# Base of "Ts" point right
plot(mcT4, label.groups=TRUE, orientation="r")
# Base of "Ts" point down
plot(mcT4, horizontal=TRUE, label.groups=0.05)
# Base of "Ts" point up
plot(mcT4, horizontal=TRUE, label.groups=0.05,
     orientation="r")

# Same 4 plots but with boxes & triangles, not Ts
plot(mcT4, label.groups=0.05, type="b")
plot(mcT4, label.groups=0.05, orientation="r",
     type="b")
plot(mcT4, horizontal=TRUE, label.groups=0.05,
     type="b")
plot(mcT4, horizontal=TRUE, label.groups=0.05,
     orientation="r", type="b")
```

```
##
## plot(multcompLetters(...))
##
# ... using dif4 from above
(mcL4 <- multcompLetters(dif4, Letters=LETTERS))
# Standard plot
## Not run:
# Requires (grid)
mcL4.1 <- plot(mcL4, label.groups=0.05)
# Redo using "at" = list
plot(mcL4, label.groups=0.05, at=mcL4.1)

# With bold face and larger font
plot(mcL4, label.groups=0.05,
     fontsize=28, fontface="bold")

# Horizontal rather than vertical
plot(mcL4, horizontal=TRUE, label.groups=0.05)

## End(Not run)

# Same as boxes rather than letters
plot(mcL4, label.groups=0.05, type="b")
plot(mcL4, horizontal=TRUE, label.groups=0.05,
     type="b")
```

---

print.multcompLetters   *print a multcompLetters object*

Description

print method for an object of class 'multcompLetters'.

Usage

```
## S3 method for class 'multcompLetters'
print(x, all = FALSE, ...)
```

Arguments

x	an object of class 'multcompLetters'
all	FALSE to print only the character vector representations of the 'multcompLetters' comparison summary; TRUE to print also the matrix representation.
...	Other optional print parameters as described on the [print](#) help page.

Details

Prints only the Letters component of the 'multcompLetters' list unless all=TRUE.

Value

the named, character vector representation of the 'multcompLetters' evaluation of the distance relationships

Author(s)

Spencer Graves

See Also

[multcompLetters](#)

Examples

```
dif3 <- c(FALSE, FALSE, TRUE)
names(dif3) <- c("A-B", "A-C", "B-C")
dif3L <- multcompLetters(dif3)
dif3L
print(dif3L)
print(dif3L, TRUE)
```

---

vec2mat	*Convert a vector with hyphenated names into a matrix.*

Description

Convert a vector with hypehnated names into a symmetric matrix with names

Usage

```
vec2mat(x, sep = "-")
```

Arguments

x   Either (1) a vector with hyphenated names indicating pairs of factor levels, groups or items that are and are not significantly different or (2) a matrix indicating same. If x is already a matrix, it is checked for symmetry. NAs are not allowed.

sep   "strsplit" character to apply to names(x).

Details

x must have names each of which contains exactly one hyphen; if not, vec2mat issues an error message. If the same comparison is present multiple times, the last value is used; no check is made for duplicates.

Value

A symmetric matrix of the same class as the input with names obtained from unique(strsplit(names(x))).
All nonspecified elements will be 1 if class(x) is numeric, FALSE if logical and "" if character. Used
by the functions 'multcompLetters' and 'multcompTs'.

Author(s)

Spencer Graves

See Also

[multcompLetters](#) [multcompTs](#)

Examples

```
dif3 <- c(FALSE, FALSE, TRUE)
names(dif3) <- c("a-b", "a-c", "b-c")
vec2mat(dif3)

dif3. <- 1:3
names(dif3.) <- c("a-b", "a-c", "b-c")
vec2mat(dif3.)

dif.ch <- c("this",'is','it')
names(dif.ch) <- c("a-b", "a-c", "b-c")
vec2mat(dif.ch)

vec2mat(array(1, dim=c(2,2)))
```

---

vec2mat2   *Convert a vector of hyphenated names into a character matrix.*

Description

Convert a vector of hyphenated names into a character matrix with 2 columns containing the names
split in each row.

Usage

```
vec2mat2(x, sep = "-")
```

Arguments

x	Vector of hyphenated names
sep	"strsplit" character to apply to names(x).

Details

If each element of x does not contain exactly 1 "sep" character, an error is issued.

Value

A character matrix with rownames = x and with the character string preceding the "sep" character in the first column and the character string following the "sep" character in the second column.

Author(s)

Spencer Graves

See Also

[vec2mat](#) [multcompLetters](#)

Examples

```
vec2mat2(c("a-b", "a-c", "b-c"))

vec2mat2(c("a-b", "b-a"))
```

Index