Package: midrangeMCP (via r-universe)

Type Package

Title Multiples Comparisons Procedures Based on Studentized Midrange
and Range Distributions

Version 3.1.2

Description Apply tests of multiple comparisons based on studentized
'midrange' and 'range' distributions. The tests are: Tukey
Midrange ('TM' test), Student-Newman-Keuls Midrange ('SNKM'
test), Means Grouping Midrange ('MGM' test) and Means Grouping
Range ('MGR' test). The first two tests were published by
Batista and Ferreira (2020)
<doi:10.1590/1413-7054202044008020>. The last two are being
published.

Imports stats, utils, graphics, grDevices, SMR, writexl, xtable, tcltk

Suggests tkrplot, testthat, knitr, rmarkdown, covr

License GPL (>= 2)

URL https://bendeivide.github.io/midrangeMCP/,
https://github.com/bendeivide/midrangeMCP

BugReports https://github.com/bendeivide/midrangeMCP/issues

LazyData TRUE

RoxygenNote 7.1.1

VignetteBuilder knitr

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

RemoteUrl https://github.com/bendeivide/midrangemcp

RemoteRef HEAD

RemoteSha 18acf4cce6429e9627f62eaeb27322bc9d1cca2d

Contents

---

guimidrangeMCP *Graphical User Interface for midrangeMCP function*

Description

guimidrangeMCP A Graphical User Interface (GUI) for function that returns the MGM, MGR, SNKM and TM tests

Usage

```
guimidrangeMCP(gui = TRUE)
```

Arguments

gui   Logical argument, TRUE or FALSE. The default is TRUE

Value

guimidrangeMCP presents a GUI for the results of the four multiple comparison procedures MGM, MGR, SNKM and TM tests. In addition, the GUI returns a graph of the results, as well as the export of these results to three types of file extension and latex code.

Examples

```
# Loading package
library(midrangeMCP)
if (interactive()) {
  guimidrangeMCP(gui = FALSE)
}
```

MRbarplot *Plotting the result of the multiple comparison procedures*

Description

MRbarplot creates a bar plot with vertical or horizontal bars to compare the mean treatments by the tests: means grouping based on midrange, means grouping based on range, Student-Newman-Keuls and Tukey based on midrange.

Usage

```
MRbarplot(x, MCP = "all", col = grDevices::heat.colors(10), horiz = FALSE, ...)
```

Arguments

x	An object of the MRtest function
MCP	Allows choosing the multiple comparison test. The *defaut* is "all". This option will perform all tests available in the MRtest object.
col	A specification for the plotting color. The *defaut* is heat.colors(10).
horiz	a logical value. If FALSE, the bars are drawn vertically with the first bar to the left. If TRUE, the bars are drawn horizontally with the first at the bottom.
...	Parameters of the [barplot](#) function

Details

The MCP argument allows choosing several tests of multiple comparisons from the MRtest object. For plots in papers, use col = gray.colors(10). For details, see [colors](#) function.

Value

MRbarplot return the bar plot of the tests chosen ("MGM", "MGR", "SNKM" and "TM") to evaluate the treatment means.

Examples

```
# Simulated data (completely randomized design)

rv <- c(100.08, 105.66, 97.64, 100.11, 102.60, 121.29, 100.80,
        99.11, 104.43, 122.18, 119.49, 124.37, 123.19, 134.16,
        125.67, 128.88, 148.07, 134.27, 151.53, 127.31)

# Treatments
treat <- factor(rep(LETTERS[1:5], each = 4))

# Anova
res     <- aov(rv~treat)

# Loading the midrangeMCP package
library(midrangeMCP)

# Choosing tests
results <- MRtest(y = res, trt = "treat", alpha = 0.05,
                  main = "Multiple Comparison Procedures",
                  MCP = c("MGM", "TM"))

MRbarplot(results, MCP = "all") # It will be shown two
                                # graphs. First, for the
                                # results of \code{'MGM'}
                                # and the second for the
                                # results of \code{'TM'}.

MRbarplot(results, MCP = "MGM") # It will be shown
                                  # only the graph
                                  # for the result of
```

```
                              # \code{'MGM'}

  # Plot for papers
  MRbarplot(results, MCP = "all", col = gray.colors(10))
```

MRtest   *Multiple comparison procedures to the means of a factor using the*
                               studentized range and midrange distributions.

Description

MRtest applies the Means grouping based on midrange, Means Grouping based on Range, Student-Newman-Keuls based on midrange and Tukey based on midrange tests. These are new tests for multiple comparisons proposed by the authors (BATISTA, 2016), that are being published.

Usage

```
MRtest(
  y,
  trt = NULL,
  dferror = NULL,
  mserror = NULL,
  replication = NULL,
  alpha = 0.05,
  main = NULL,
  MCP = "all",
  ismean = FALSE
)
```

Arguments

y	Model (aov or lm), numeric vector containing the response variable or the mean of the treatments.
trt	Constant (y = model) or a vector containing the treatments.
dferror	Degrees of freedom of the Mean Square Error.
mserror	Mean Square Error.
replication	Number de repetitions of the treatments in the experiment. For unbalanced data should be informed the harmonic mean of repetitions. This argument should be informed only if ismean = TRUE.
alpha	Significant level. The default is alpha = 0.05.
main	Title of the analysis.
MCP	Allows choosing the multiple comparison test; the *defaut* is "all". This option will go perform all tests. However, the options are: the Means grouping based on midrange test ("MGM"), Means Grouping based on Range test ("MGR"), the Student-Newman-Keuls based on midrange test ("SNKM") and the Tukey based on midrange test ("TM").

ismean	Logic. If FALSE (default), the y argument represents a model (aov or lm) or a numeric vector containing the response variable. If TRUE the y argument represents the mean of treatments.

Details

The MCP argument allows you to choose various tests of multiple comparisons at once. For example, MCP = c("MGM", "MGR"), and so on.

Value

MRtest returns the print of a list of results. First, the summary of y. Second, the statistics of the test chosen. And finally, the mean group results for each test. If MRtest function is stored in an object, the results will be printed and also stored in the object.

Examples

```
# Simulated data (completely randomized design)

# Response variable
rv <- c(100.08, 105.66, 97.64, 100.11, 102.60, 121.29, 100.80,
        99.11, 104.43, 122.18, 119.49, 124.37, 123.19, 134.16,
        125.67, 128.88, 148.07, 134.27, 151.53, 127.31)

# Treatments
treat <- factor(rep(LETTERS[1:5], each = 4))

# Anova
res    <- anova(aov(rv~treat))
DFerror <- res$Df[2]
MSerror <- res$`Mean Sq`[2]

# Loading the midrangeMCP package
library(midrangeMCP)

# applying the tests
results <- MRtest(y = rv,
                  trt = treat,
                  dferror = DFerror,
                  mserror = MSerror,
                  alpha = 0.05,
                  main = "Multiple Comparison Procedure: MGM test",
                  MCP = c("MGM"))

# Other option for the MCP argument is "all". All tests are used.

results$Groups     # Results of the tests
results$Statistics # Main arguments of the tests
results$Summary    # Summary of the response variable

# Using the y argument as aov or lm model
res  <- aov(rv~treat)
```

```
MRtest(y = res, trt = "treat", alpha = 0.05,
       main = "Multiple Comparison Procedure: MGM test",
       MCP = c("MGM"))

# For unbalanced data: It will be used the harmonic mean of
#                         the number of experiment replicates

# Using the previous example
rv <- rv[-1]
treat <- treat[-1]

res  <- lm(rv~treat) # Linear model

# Multiple comparison procedure: MGR test
MRtest(y = res, trt = "treat", alpha = 0.05,
       main = "Multiple Comparison Procedure: MGR test",
       MCP = c("MGR"))

# Assuming that the available data are the averages
#  of the treatments and the analysis of variance

# Analysis of Variance Table

# Response: rv
#           Df Sum Sq Mean Sq F value     Pr(>F)
# treat      4 4135.2 1033.80  14.669 4.562e-05 ***
# Residuals 15 1057.1   70.47

mean.treat <- c(100.87, 105.95, 117.62, 127.97, 140.30)
treat <- factor(LETTERS[1:5])
DFerror <- 15
MSerror <- 70.47488
replic <- 4

MRtest(y = mean.treat,
       trt = treat,
       dferror = DFerror,
       mserror = MSerror,
       replication = replic,
       alpha = 0.05,
       main = "Multiple Comparison Procedure: MGM test",
       MCP = c("MGM"),
       ismean = TRUE)
```

MRwrite   *Export the results of the* MRtest *function.*

Description

The x object from a MRtest is written to file arguments.

Usage

```
MRwrite(x, MCP = "all", extension = "csv", dataMR = "all")
```

Arguments

x	object from the `MRtest` functions.
MCP	Allows choosing the multiple comparison test. The *defaut* is "all". This option will go perform all tests from the `MRtest` object.
extension	Type of format of the file. Four options `"csv"`, `"txt"` `"xlsx"` and `"latex"`. The *default* is `"csv"`.
dataMR	Allows to choose the results to bee written. Three options are available: `"groups"`, `"summary"` or `"all"`. The option `"groups"` writes the treatment mean groups avaluated by the chosen test in the MCP argument. The `"summary"` writes the descriptive statistics of the response variable. The options `"all"` should be chosen for both results.

Details

Note that the choice of the tests in the `MRwrite` function must be in accordance with the tests chosen in the x argument.

Value

`MRwrite` writes the most important results for the chosen tests in the MCP argument.

Examples

```
# Simulated data (completely randomized design)

rv <- c(100.08, 105.66, 97.64, 100.11, 102.60, 121.29, 100.80,
        99.11, 104.43, 122.18, 119.49, 124.37, 123.19, 134.16,
        125.67, 128.88, 148.07, 134.27, 151.53, 127.31)

# Treatments
treat <- factor(rep(LETTERS[1:5], each = 4))

# Anova
res <- aov(rv~treat)

# Loading the midrangeMCP package
library(midrangeMCP)

# Choosing any tests
results <- MRtest(y = res, trt = "treat", alpha = 0.05,
                  main = "Multiple Comparison Procedures",
                  MCP = c("MGM", "TM"))

#Export file in latex (Output in Console)
MRwrite(results, MCP = "all", extension = "latex", dataMR = "all")
```

```
#Observation: The MRwrite function export
#             only one extension at a time
```

Index