

Package ‘TukeyC’

April 10, 2025

Type Package

Title Conventional Tukey Test

Version 1.3-43

Date 2025-04-09

Maintainer Ivan Bezerra Allaman <ivanalaman@gmail.com>

Depends R (>= 2.6.0)

Imports doBy, xtable

Suggests pbkrtest (>= 0.4-6), lme4

Description

Provides tools to perform multiple comparison analyses, based on the well-known Tukey's ``Honestly Significant Difference" (HSD) test. In models involving interactions, "TukeyC" stands out from other R packages by implementing intuitive and easy-to-use functions. In addition to accommodating traditional R methods such as `lm()` and `aov()`, it has also been extended to objects of the `lmer()` class, that is, mixed models with fixed effects. For more details see Tukey (1949) <[doi:10.2307/3001913](https://doi.org/10.2307/3001913)>.

License GPL (>= 2)

URL <https://github.com/jcfaria/TukeyC>

Encoding UTF-8

LazyLoad yes

NeedsCompilation no

Author Jose Claudio Faria [aut],
Enio G. Jelihovschi [aut],
Ivan Bezerra Allaman [aut, cre]

Repository CRAN

Date/Publication 2025-04-10 08:50:08 UTC

Contents

boxplot.TukeyC	2
CRD1	4

CRD2	5
cv	5
FE	6
LSD	7
make.TukeyC.groups	7
make.TukeyC.test	8
plot.TukeyC	9
print.TukeyC	12
RCBD	13
sorghum	14
SPE	15
SPET	15
SSPE	16
summary	16
Tukey test	20
TukeyC	21
xtable.TukeyC	30
Index	32

boxplot.TukeyC	<i>Boxplot TukeyC Objects</i>
----------------	-------------------------------

Description

S3 method to plot TukeyC objects.

Usage

```
## S3 method for class 'TukeyC'
boxplot(x,
        mean.type = c('line', 'point', 'none'),
        xlab      = NULL,
        mean.col  = 'gray',
        mean.pch  = 1,
        mean.lwd  = 1,
        mean.lty  = 1,
        args.legend = NULL, ...)
```

Arguments

x	A TukeyC object.
mean.type	The type of mean must be plotted. O default is "line".
xlab	A label for the 'x' axis.
mean.col	A vector of colors for the means representation.
mean.pch	A vector of plotting symbols or characters. Only if type are "point".

mean.lwd	Line width of mean.
mean.lty	Line type of mean. Only if type are "line".
args.legend	List of additional arguments to be passed to legend; The default is NULL.
...	Optional plotting parameters.

Details

The boxplot.TukeyC function is a S3 method to plot 'TukeyC' objects. The difference to generic function is the Tukey inference under frame and the plot of means within box.

Value

'NULL' (invisibly). The main purpose of this function is to produce a plot.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
Enio Jelihovschi (<eniojelihovs@gmail.com>
Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

References

Murrell, P. (2005) *R Graphics*. Chapman and Hall/CRC Press.

See Also

[boxplot](#)

Examples

```
##  
## Examples: Completely Randomized Design (CRD)  
## More details: demo(package='TukeyC')  
##  
  
library(TukeyC)  
data(CRD1)  
  
## From: formula  
# Simple!  
tk1 <- TukeyC(y ~ x,  
              data=CRD1$dfm,  
              which='x')  
boxplot(tk1)  
  
# A little more elaborate!  
boxplot(tk1,  
        mean.lwd=1.3,  
        mean.col='red')  
  
# A little more!
```

```
boxplot(tk1,
        mean.lwd=1.3,
        mean.lty=2,
        mean.col='red',
        args.legend=list(x='bottomleft'))

# With point type!
boxplot(tk1,
        mean.type='point')

boxplot(tk1,
        mean.type='point',
        mean.pch=19,
        cex=1.5,
        mean.col='red')

# With other point
boxplot(tk1,
        mean.type='point',
        mean.pch='+',
        cex=2,
        mean.col='blue',
        args.legend=list(x='bottomleft'))
```

CRD1

Completely Randomized Design (CRD)

Description

A list illustrating the resources of TukeyC package related to Completely Randomized Design ('CRD').

Usage

```
data(CRD1)
```

Details

A simulated data to model a Completely Randomized Design ('CRD') of 4 factor levels and 6 repetitions.

CRD2	<i>Completely Randomized Design ('CRD')</i>
------	---

Description

A list illustrating the resources of TukeyC package related to Completely Randomized Design ('CRD').

Usage

```
data(CRD2)
```

Details

A simulated data to model a Completely Randomized Design ('CRD') of 45 factor levels and 4 repetitions.

cv	<i>Coefficient of the experiment variation</i>
----	--

Description

It obtains the coefficient of variation of the experiment obtained by models `lm`, `aov` and `aovlist`.

Usage

```
cv(x,
   round=2)
```

Arguments

<code>x</code>	A object of the class <code>lm</code> , <code>aov</code> or <code>aovlist</code> .
<code>round</code>	An integer value indicating the number of decimal places to be used. The default value is 2.

Details

```
sqrt(MSError)*100/mean(x)
```

Value

<code>x</code>	named numeric vector
----------------	----------------------

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
Enio Jelihovschi (<eniojelihovs@gmail.com>
Ivan Bezerra Allaman (<ivanalaman@gmail.com>

Examples

```
library(TukeyC)

## Completely Randomized Design (CRD - aov)
data(CRD1)
av1 <- with(CRD1,
            aov(y ~ x,
                data=dfm))
summary(av1)
cv(av1)

## Randomized Complete Block Design (RCBD - aov)
data(RCBD)
av2 <- with(RCBD,
            aov(y ~ blk + tra,
                data=dfm))
summary(av2)
cv(av2)

## Split-plot experiment (SPE - aovlist)
data(SPE)
av3 <- with(SPE,
            aov(y ~ blk + P*SP + Error(blk/P),
                data=dfm))
summary(av3)
cv(av3)

## Split-split-plot experiment (SSPE - aovlist)
data(SSPE)
av4 <- with(SSPE,
            aov(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                data=dfm))
summary(av4)
cv(av4)

## storing
res <- cv(av4)
res[2:3]
```

Description

A list illustrating the resources of TukeyC package related to Factorial Experiment ('FE').

Usage

```
data(FE)
```

Details

A simulated data to model a Factorial Experiment ('FE') with 3 factors, 2 levels per factor and 4 blocks.

LSD

*Latin Squares Design (LSD)***Description**

A list illustrating the resources of TukeyC package related to Latin Squares Design ('LSD').

Usage

```
data(LSD)
```

Details

A simulated data to model a Latin Squares Design ('LSD') with 5 factor levels 5 rows and 5 columns.

make.TukeyC.groups

*Make Tukey Groups***Description**

Builds groups of means, according to the method of Tukey.

Usage

```
make.TukeyC.groups(x)
```

Arguments

x A square matrix where the lines and columns are all levels of the factor being test sorted in decreasing order. The matrix content are TRUE or FALSE, i.e, indicating whether or not equality between the levels of factor.

Details

This function is an algorithm used to build up the groups of means, allowing for overlapping.

Value

The object returned by this function is a character matrix and the rownames are the levels of the factor being tested sorted in decreasing order. The matrix has the same number of columns as the number of groups generated.

Note

This function is mainly for internal use in the TukeyC package.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
Enio Jelihovschi (<eniojelihovs@gmail.com>
Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

make.TukeyC.test	<i>Make Tukey Test</i>
------------------	------------------------

Description

This function implements the Tukey test for balanced or unbalanced designs and schemes.

Usage

```
make.TukeyC.test(obj,  
  MSE,  
  sig.level,  
  dfr,  
  round,  
  adjusted.pvalue)
```

Arguments

obj	A data.frame with the means and replicate of the factors.
MSE	A vector of length 1 giving the mean squared error.
sig.level	A vector of length 1 giving the level of significance of the test.
dfr	A vector of length 1 giving the degrees of freedom of 'MSE'.
round	Integer indicating the number of decimal places.
adjusted.pvalue	A vector of pvalues obtained of the ptukey function.

Value

A list with 5 slots containing the most important results of the test performed:

Result	The result of the Tukey test
Sig.Level	The significance of the test
Diff_Prob	A matrix with the observed means differences (<code>upper.tri</code>) and respective probabilities (<code>lower.tri</code>)
MSD	The minimum significant difference
Replicates	The replicate number for each factor

Note

This function is mainly for internal use in the **TukeyC** package.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
 Enio Jelihovschi (<eniojelihovs@gmail.com>
 Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

 plot.TukeyC

Plot TukeyC and TukeyC.nest Objects

Description

S3 method to plot TukeyC and TukeyC.nest objects.

Usage

```
## S3 method for class 'TukeyC'
plot(x,
      result      = TRUE,
      replicates  = TRUE,
      pch         = 19,
      col         = NULL,
      xlab        = NULL,
      ylab        = NULL,
      xlim        = NULL,
      ylim        = NULL,
      axisx       = TRUE,
      axisy       = TRUE,
      id.lab      = NULL,
      id.las      = 1,
      yl          = TRUE,
      yl.lty      = 3,
      yl.col      = 'gray',
```

```

dispersion = c('none', 'mm', 'sd', 'ci', 'cip'),
d.lty      = 1,
d.col      = 'black',
title      = '', ...)

```

Arguments

<code>x</code>	A TukeyC object.
<code>result</code>	The result of the test (letters) should be visible.
<code>replicates</code>	The number of replicates should be visible.
<code>pch</code>	A vector of plotting symbols or characters.
<code>col</code>	A vector of colors for the means representation.
<code>xlab</code>	A label for the 'x' axis.
<code>ylab</code>	A label for the 'y' axis.
<code>xlim</code>	The 'x' limits of the plot.
<code>ylim</code>	The 'y' limits of the plot.
<code>axisx</code>	Axis x? If 'TRUE' you must accept the default, otherwise, you must customize.
<code>axisy</code>	Axis y? If 'TRUE' you must accept the default, otherwise, you must customize.
<code>id.lab</code>	Factor level names at 'x' axis.
<code>id.las</code>	Factor level names written either horizontally or vertically.
<code>yl</code>	Horizontal (reference) line connecting the circle to the 'y' axis.
<code>yl.lty</code>	Line type of 'yl'.
<code>yl.col</code>	Line color of 'yl'.
<code>dispersion</code>	Vertical line through the circle (mean value) linking the minimum to the maximum of the factor level values corresponding to that mean value. Other options are: sd (standard deviation), ci (confidence interval), cip (pooled confidence interval) and none.
<code>d.lty</code>	Line type of dispersion.
<code>d.col</code>	A vector of colors for the line type of dispersion.
<code>title</code>	A title for the plot.
<code>...</code>	Optional plotting parameters.

Details

The `plot.TukeyC` function is a S3 method to plot 'Tukey' and `TukeyC.nest` objects. It generates a series of points (the means) and a vertical line showing the dispersion of the values corresponding to each group mean. The `ci` option is calculated utilizing each treatment variance as estimating of population variance. The `cip` option is calculated utilizing the means square error (MSE) as estimating of population variance.

Value

'NULL' (invisibly). The main purpose of this function is to produce a plot.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
Enio Jelihovschi (<eniojelihovs@gmail.com>
Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

References

Murrell, P. (2005) *R Graphics*. Chapman and Hall/CRC Press.

See Also

[plot](#)

Examples

```
##
## Examples: Completely Randomized Design (CRD)
## More details: demo(package='TukeyC')
##

library(TukeyC)
data(CRD2)

## From: formula
tk1 <- with(CRD2,
            TukeyC(y ~ x,
                  data=dfm,
                  which='x'))

old.par <- par(mar=c(6, 3, 6, 2))
plot(tk1,
     id.las=2)

plot(tk1,
     y1=FALSE,
     disp='sd',
     id.las=2)

## From: aov
av <- with(CRD2,
           aov(y ~ x,
              data=dfm))
summary(av)

tk2 <- TukeyC(x=av,
              which='x')
plot(tk2,
     disp='sd',
     y1=FALSE,
     id.las=2)

# From: lm
```

```
av_lm <- with(CRD2,
              lm(y ~ x,
                 data=dfm))

tk3 <- TukeyC(x=av_lm,
              which='x')

par(mfrow=c(2, 1))
plot(tk3,
      disp='ci',
      id.las=2,
      yl=FALSE)

plot(tk3,
      disp='cip',
      id.las=2,
      yl=FALSE)

par(mfrow=c(1, 1))
par(old.par)
```

print.TukeyC

Print Method for TukeyC objects.

Description

Returns (and prints) a list for objects of class TukeyC.

Usage

```
## S3 method for class 'TukeyC'
print(x, ...)
```

Arguments

x A given object of the class TukeyC.
... Further arguments (require by generic).

Value

A list with the following elements:

a list of length 5

In the first position of the list there is a data.frame with the means and the groupings. In the second position of the list there is a scalar with the significance level. In the third position there is a matrix with the p-values obtained in each mean comparison. In the fourth position there is another matrix with the values obtained from the minimum significance difference. In the fifth position there is a vector with the number of repetitions of the experiment.

a list of length 5

In the first position there is a data.frame with the names of the treatments and the means. In the second position there is another data.frame with the means, minimum and maximum of the data. In the third position there are the means with the lower and upper limits of the confidence interval using the standard deviation to calculate the margin of error. In the fourth position there is also a data.frame with the means and the lower and upper limits of a confidence interval using the standard error of the mean of each treatment to calculate the margin of error. In the fifth position there is also a data.frame with the means and the lower and upper limits of the confidence interval using the standard error of the experimental error to calculate the margin of error.

a list of length 1

An object of the type "call".

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>)

Enio G. Jelihovschi (<eniojelihovs@gmail.com>)

Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

See Also

[TukeyC](#)

Examples

```
data(RCBD)

tk <- with(RCBD,
  TukeyC(y ~ blk + tra,
  data=dfm,
  which='tra'))
tk
```

RCBD

Randomized Complete Block Design (RCBD)

Description

A list illustrating the resources of TukeyC package related to Randomized Complete Block Design ('RCBD').

Usage

```
data(RCBD)
```

Details

A simulated data to model a Randomized Complete Block Design ('RCBD') of 5 factor levels, 4 blocks and 4 factor levels repetitions one in each block.

sorghum	<i>Completely Randomized Design (CRD)</i>
---------	---

Description

The experiment consists of 16 treatments (cultivars) of sorghum conducted in a balanced squared lattice design and the yield by plot (kg/plot).

Usage

```
data(sorghum)
```

Format

An incomplete balanced block design with 4 blocks, 16 treatments, and 5 repetitions, that is, the yield of each treatment is measured 5 times. `sorghum` is a list with 4 elements. The first 'x' is a factor of length 80 with 16 levels describing the treatments. The second 'dm' is data.frame describing the design matrix. Its columns are 'x', 'bl' (blocks) and 'r' repetitions. The third 'y' is a numeric vector the yields. The fourth 'dfm' is a data frame with four columns. The first three columns are the design matrix and the fourth is 'y'.

Details

The experiment was conducted at EMBRAPA Milho e Sorgo (The Brazilian Agricultural Research Corporation, Corn and Sorghum section).

Source

Ramalho, M.A.P. and Ferreira and D.F. and Oliveira, A.C. (2000) *Experimentacao em Genetica e Melhoramento de Plantas*. Editora UFLA, Lavras, Brazil, page 167.

Examples

```
library(TukeyC)

data(sorghum)

av <- aov(y ~ r/bl + x,
          data=sorghum$dfm)

tk <- TukeyC(av,
             which='x',
             sig.level=0.05)
```

```
summary(tk)
```

```
plot(tk)
```

SPE

Split-plot Experiment (SPE)

Description

A list to illustrate the resources of TukeyC package related to Split-plot Experiment ('SPE').

Usage

```
data(SPE)
```

Details

A simulated data to model a Split-plot Experiment ('SPE') with 3 plots, each one split 4 times and 6 repetitions per split.

SPET

Split-plot Experiment in Time (SPET)

Description

The experiment consists of 8 treatments (7 leguminous cover crops and maize) in a Randomized Complete Block Design ('RCBD') and the yield by plot (kg/plot).

Usage

```
data(SPET)
```

Source

Gomes, F.P. (1990). Curso de Estatística Experimental. 13 ed. Editora NOBEL, Piracicaba, Brazil, page 157.

SSPE	<i>Split-split-plot Experiment (SSPE)</i>
------	---

Description

A list to illustrate the resources of TukeyC package related to Split-split-plot Experiment ('SSPE').

Usage

```
data(SSPE)
```

Details

A simulated data to model a Split-split-plot Experiment ('SSPE') with 3 plots, each one split 3 times, each split, split again 5 times and 4 repetitions per split-split.

summary	<i>Summary Method for TukeyC and TukeyC.nest Objects</i>
---------	--

Description

Returns (and prints) a summary list for TukeyC objects.

Usage

```
## S3 method for class 'TukeyC'
summary(object,
        complete=TRUE, ...)
```

Arguments

object	A given object of the class TukeyC.
complete	A logical value indicating if the summary is complete (mean difference and p-value) or only the groups.
...	Potential further arguments (required by generic).

Value

A matrix if complete argument equal TRUE or data.frame if complete argument equal FALSE.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
 Enio Jelihovschi (<eniojelihovs@gmail.com>
 Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

References

Chambers, J.M. and Hastie, T.J. (1992) *Statistical Models in S*. Wadsworth and Brooks/Cole.

See Also

[TukeyC](#)

Examples

```
##
## Examples: Completely Randomized Design (CRD)
## More details: demo(package='TukeyC')
##

## The parameters can be: formula, aov, lm and aovlist
data(CRD2)

## From: formula
tk1 <- with(CRD2,
            TukeyC(y ~ x,
                  data=dfm,
                  which='x',
                  id.trim=5))
summary(tk1)

##
## Example: Randomized Complete Block Design (RCBD)
## More details: demo(package='TukeyC')
##

## The parameters can be: formula, aov, lm and aovlist
data(RCBD)

## From: formula
tk2 <- with(RCBD,
            TukeyC(y ~ blk + tra,
                  data=dfm,
                  which='tra'))
summary(tk2)

##
## Example: Latin Squares Design (LSD)
## More details: demo(package='TukeyC')
##

## The parameters can be: design matrix and the response variable,
## data.frame or aov

data(LSD)

## From: formula
tk3 <- with(LSD,
```

```

      TukeyC(y ~ rows + cols + tra,
            data=dfm,
            which='tra'))
summary(tk3)

##
## Example: Factorial Experiment (FE)
## More details: demo(package='TukeyC')
##

## The parameters can be: design matrix and the response variable,
## data.frame or aov

data(FE)
## From: design matrix (dm) and response variable (y)
## Main factor: N
tk4 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
                  data=dfm,
                  which='N'))
summary(tk4)

## Nested: p1/N
## Testing N inside of level one of P
ntk1 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
                  data=dfm,
                  which='P:N',
                  f11=1))
summary(ntk1)

## Nested: k1/p1/N
## Testing N inside of level one of K and level one of P
ntk2 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
                  data=dfm,
                  which='K:P:N',
                  f11=1,
                  f12=1))
summary(ntk2)

## Nested: k2/n2/P
ntk3 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
                  data=dfm,
                  which='K:N:P',
                  f11=2,
                  f12=2))
summary(ntk3)

## Nested: p1/n1/K
ntk4 <- with(FE,
            TukeyC(y ~ blk + N*P*K,

```

```

                                data=dfm,
                                which='P:N:K',
                                f11=1,
                                f12=1))
summary(ntk4)

##
## Example: Split-plot Experiment (SPE)
## More details: demo(package='TukeyC')
##

data(SPE)
## From: formula
## Main factor: P
## It is necessary to inform the appropriate error for the test
tk1 <- with(SPE,
            TukeyC(y ~ blk + P*SP + Error(blk/P),
                  data=dfm,
                  which='P',
                  error='blk:P'))
summary(tk1)

## Nested: p1/SP
tkn1 <- with(SPE,
            TukeyC(y ~ blk + P*SP + Error(blk/P),
                  data=dfm,
                  which='P:SP',
                  f11=1 ))
summary(tkn1)

## From: formula
## Main factor: P
## It is necessary to inform the appropriate error for the test
data(SSPE)

tk1 <- with(SSPE,
            TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                  data=dfm,
                  which='P',
                  error='blk:P'))
summary(tk1)

## Main factor: SP
## It is necessary to inform the appropriate error for the test
tk2 <- with(SSPE,
            TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                  data=dfm,
                  which='SP',
                  error='blk:P:SP'))
summary(tk2)

## Main factor: SSP
tk3 <- with(SSPE,

```

```

      TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
            data=dfm,
            which='SSP'))
summary(tkn3)

## Nested: p1/SSP
tkn1 <- with(SSPE,
  TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
        data=dfm,
        which='P:SSP',
        f11=1))
summary(tkn1)

## From: aovlist
av <- with(SSPE,
  aov(y ~ blk + P*SP*SSP + Error(blk/P/SP),
      data=dfm))
summary(av)

## Nested: P1/SP1/SSP
tkn2 <- TukeyC(av,
  which='P:SP:SSP',
  f11=1,
  f12=1)
summary(tkn2)

## Nested: P2/SP1/SSP
tkn3 <- TukeyC(av,
  which='P:SP:SSP',
  f11=2,
  f12=1)
summary(tkn3)

## Nested: SSP2/P1/SP - it is necessary to inform how to combine the errors
tkn4 <- TukeyC(av,
  which='SSP:P:SP',
  f11=2,
  f12=1,
  error='Within/blk:P/blk:P:SP')
summary(tkn4)

```

 Tukey test

Conventional Tukey Test

Description

This package performs what is known as the Tukey HSD test in the conventional way. It also uses an algorithm which divides the set of all means in groups and assigns letters to the different groups, allowing for overlapping. This is done for simple experimental designs and schemes. The most

usual designs are: Completely Randomized Design ('CRD'), Randomized Complete Block Design ('RCBD') and Latin Squares Design ('LSD'). The most usual schemes are: Factorial Experiment ('FE'), Split-Plot Experiment ('SPE') and Split-Split-Plot Experiment ('SPE').

The package can be used for both balanced or unbalanced (when possible), experiments.

R has some functions (`TukeyHSD` provided by `stats`, `glht` provided by `multcomp`, `HSD.test` provided by `agricolae` and `cld` provided by `multcomp`) which also performs the Tukey test. The `TukeyHSD` returns intervals based on the range of the sample means rather than the individual differences. Those intervals are based on Studentized range statistics and are, in essence, confidence intervals. This approach has two advantages: the p-value is showed allowing the user to flexibilize the inferencial decision and also make it possible to plot the result of the test. However, it has one disadvantage, since the final result is more difficult to understand and summarize. Others (`glht`, `cld`) are also useful but difficult to manage. Additionally, most of users of other statistical softwares are very used with letters grouping the means of the factor tested, making unattractive or difficult to adapt to the current approach of R.

So, the main aim of this package is make available in R environment the conventional approach of Tukey test with a set of flexible functions and S3 methods.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>)

Enio Jelihovschi (<eniojelihovs@gmail.com>)

Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

References

Miller, R.G. (1981) *Simultaneous Statistical Inference*. Springer.

Ramalho M.A.P, Ferreira D.F and Oliveira A.C. (2000) *Experimentacao em Genetica e Melhoramento de Plantas*. Editora UFLA.

Steel, R.G., Torrie, J.H and Dickey D.A. (1997) *Principles and procedures of statistics: a biometrical approach*. Third Edition.

Yandell, B.S. (1997) *Practical Data Analysis for Designed Experiments*. Chapman & Hall.

TukeyC

The TukeyC Test for Single Experiments

Description

These are methods for objects of class `formula`, `lm`, `aov`, `aovlist` and `lmerMod` for single, factorial, split-plot and split-split-plot experiments.

Usage

```
TukeyC(x,...)

## S3 method for class 'formula'
TukeyC(formula,
        data          = NULL,
        which         = NULL,
        f11           = NULL,
        f12           = NULL,
        error         = NULL,
        sig.level     = .05,
        round         = 2,
        adjusted.pvalue = 'none',
        ...)

## S3 method for class 'lm'
TukeyC(x,
        which         = NULL,
        f11           = NULL,
        f12           = NULL,
        error         = NULL,
        sig.level     = .05,
        round         = 2,
        adjusted.pvalue = 'none',
        ...)

## S3 method for class 'aovlist'
TukeyC(x,
        which         = NULL,
        f11           = NULL,
        f12           = NULL,
        error         = NULL,
        sig.level     = .05,
        round         = 2,
        adjusted.pvalue = 'none',
        ...)

## S3 method for class 'lmerMod'
TukeyC(x,
        which         = NULL,
        f11           = NULL,
        f12           = NULL,
        error         = NULL,
        sig.level     = .05,
        round         = 2,
        adjusted.pvalue = 'none',
        ...)
```

Arguments

x, formula	A formula, lm, aov, aovlist and lmerMod class object. Objects of the formula class follow “response variable ~ predicted variable.
data	A object of the data.frame class. Use only objects of formula class.
which	The name of the treatment to be used in the comparison. The name must be inside quoting marks.
f11	A vector of length 1 giving the level of the first factor in nesting order tested.
f12	A vector of length 1 giving the level of the second factor in nesting order tested.
error	The error to be considered. If from experiment at split plot or split-split plot pay attention! See details!
sig.level	Level of Significance used in the TukeyC algorithm to create the groups of means. The default value is 0.05.
round	Integer indicating the number of decimal places.
adjusted.pvalue	Method for adjusting p values (see p.adjust to more details). The possible values are: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" and "none". The default is "none".
...	Potential further arguments (required by generic).

Details

The function TukeyC returns an object of class TukeyC containing the groups of means plus other necessary variables for summary and plot.

The generic functions summary and plot are used to obtain and print a summary and a plot of the results.

The error arguments may be used whenever the user want a specific error other than the experimental error. At the split plot and split-split plot experiment, combination of error may be specified with "/" in the sequence of the which argument. For example, a object of aovlist class, a possible combination would be error = 'Within/blk:plot' at case block split plot experiment with which = 'subplot:plot' argument.

Value

The function TukeyC returns a list of the class TukeyC with the slots:

Result	A data.frame storing the result of Tukey test.
Sig.level	A scalar giving the level of significance of the test.
Diff_Prob	A matrix at the lower diagonal with p-values and upper diagonal with means differences.
MSD	A matrix with minimum significance differences by Tukey methodology. If balanced data, then all values are equal.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>
 Enio Jelihovschi (<eniojelihovs@gmail.com>
 Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

References

Miller, R.G. (1981) *Simultaneous Statistical Inference*. Springer.
 Ramalho M.A.P, Ferreira D.F and Oliveira A.C. (2000) *Experimentacao em Genetica e Melhoramento de Plantas*. Editora UFPA.
 Steel, R.G., Torrie, J.H and Dickey D.A. (1997) *Principles and procedures of statistics: a biometrical approach*. Third Edition.
 Yandell, B.S. (1997) *Practical Data Analysis for Designed Experiments*. Chapman and Hall.

Examples

```
##
## Examples: Randomized Complete Block Design (RCBD)
## More details: demo(package='TukeyC')
##

## The parameters can be: formula, aov, lm, aovlist and lmerMod

data(RCBD)

## From: formula
tk1 <- with(RCBD,
            TukeyC(y ~ blk + tra,
                  data=dfm,
                  which='tra'))
summary(tk1)

## From: merMod
## This class is specific of the lme4 package.

if(require(lme4)){
  lmer1 <- with(RCBD,
               lmer(y ~ (1|blk) + tra,
                   data=dfm))

  tk2 <- TukeyC(lmer1,
               which='tra')
  summary(tk2)
}

##
## Example: Latin Squares Design (LSD)
## More details: demo(package='TukeyC')
##
```



```
data(LSD)

## From: formula
tk3 <- with(LSD,
            TukeyC(y ~ rows + cols + tra,
                  data=dfm,
                  which='tra'))
summary(tk3)

## From: aov
av1 <- with(LSD,
            aov(y ~ rows + cols + tra,
               data=dfm))

tk4 <- TukeyC(av1,
              which='tra')
summary(tk4)

## From: lm
lm1 <- with(LSD,
            lm(y ~ rows + cols + tra,
               data=dfm))

tk5 <- TukeyC(lm1,
              which='tra')
summary(tk5)

##
## Example: Factorial Experiment (FE)
## More details: demo(package='TukeyC')
##

data(FE)
## From: formula
## Main factor: N
tk6 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
                  data=dfm,
                  which='N'))
summary(tk6)

## Nested: p1/N
# From: formula
n_tk1 <- with(FE,
              TukeyC(y ~ blk + N*P*K,
                    data=dfm,
                    which='P:N',
                    f11=1))
summary(n_tk1)

## Nested: p2/N
# From: lm
lm2 <- with(FE,
```

```

lm(y ~ blk + N*P*K,
   dfm)

n_tk2 <- with(FE,
              TukeyC(lm2,
                     which='P:N',
                     f11=2))
summary(n_tk2)

## Nested: n1/P
# From: aov
av2 <- with(FE,
            aov(y ~ blk + N*P*K,
               dfm))

n_tk3 <- with(FE,
              TukeyC(av2,
                     which='N:P',
                     f11=1))
summary(n_tk3)

# From: merMod

if(require(lme4)){
  lmer2 <- with(FE,
               lmer(y ~ (1|blk) + N*P*K,
                   dfm))

  n_tk4 <- with(FE,
               TukeyC(lmer2,
                     which='N:P',
                     f11=1))
  summary(n_tk4)
}

##
## Example: Split-plot Experiment (SPET)
## More details: demo(package='TukeyC')
##
data(SPET)

## From lm
lm3 <- with(SPET,
            lm(y ~ blk*tra + tra*year,
              dfm))

# crotgrantiana/year
sp_tk1 <- TukeyC(lm3,
                which='tra:year',
                f11=1)
summary(sp_tk1)

```

```

# year1/tra
# It is necessary to set year error with trat error in the order of the "which" argument.
# It is necessary to inform how to combine the errors
sp_tk2 <- TukeyC(lm3,
                which='year:tra',
                error='Residuals/blk:tra',
                f11=1)
summary(sp_tk2)

# From merMod
# Only tra

if(require(lme4)){
  lmer3 <- with(SPET,
               lmer(y ~ blk + (1|blk:tra) + tra*year,
                    dfm))

  # comparison only tra
  sp_tk3 <- TukeyC(lmer3,
                  which = 'tra',
                  error = 'blk:tra')
  summary(sp_tk3)

  # year1/tra
  sp_tk4 <- TukeyC(lmer3,
                  which='year:tra',
                  error='Residual/blk:tra',
                  f11=1)
  summary(sp_tk4)
}

## Example: Split-split-plot Experiment (SSPE)
## More details: demo(package='TukeyC')
##

data(SSPE)
## From: formula
## Main factor: P
## It is necessary to inform the appropriate error for the test
ssp_tk1 <- with(SSPE,
               TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                     data=dfm,
                     which='P',
                     error='blk:P'))
summary(ssp_tk1)

## Main factor: SP
## It is necessary to inform the appropriate error for the test
ssp_tk2 <- with(SSPE,
               TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                     data=dfm,
                     which='SP',

```

```

                                error='blk:P:SP'))
summary(ssp_tk2)

## Main factor: SSP
ssp_tk3 <- with(SSPE,
               TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                      data=dfm,
                      which='SSP'))
summary(ssp_tk3)

## From: aov
## Main factor: SSP
av3 <- with(SSPE,
            aov(y ~ blk + P*SP*SSP + Error(blk/P/SP),
               data=dfm))

ssp_tk4 <- TukeyC(av3,
                  which='SSP')
summary(ssp_tk4)

## Nested: p1/SP
## It is necessary to inform the appropriate error for the test
ssp_tk5 <- TukeyC(av3,
                  which='P:SP',
                  error='blk:P:SP',
                  f11=1)
summary(ssp_tk5)

## Nested: p1/SSP
ssp_tk6 <- TukeyC(av3,
                  which='P:SSP',
                  f11=1)
summary(ssp_tk6)

## Nested: p1/sp1/SSP
## Testing SSP inside of level one of P and level one of SP
ssp_tk7 <- TukeyC(av3,
                  which='P:SP:SSP',
                  f11=1,
                  f12=1)
summary(ssp_tk7)

## Nested: p2/sp1/SSP
ssp_tk8 <- TukeyC(av3,
                  which='P:SP:SSP',
                  f11=2,
                  f12=1)
summary(ssp_tk8)

## Nested: sp1/P
## It is necessary to inform the appropriate error for the test
ssp_tk9 <- TukeyC(av3,
                  which='SP:P',

```

```

        error='blk:P:SP/blk:P',
        f11=1)

summary(ssp_tk9)

## Nested: ssp1/SP
ssp_tk10 <- TukeyC(av3,
                  which='SSP:SP',
                  error='Within/blk:P:SP',
                  f11=1)
summary(ssp_tk10)

## Nested: ssp1/sp1/P
## It is necessary to inform the appropriate error for the test
ssp_tk11 <- TukeyC(av3,
                  which='SSP:SP:P',
                  error='Within/blk:P:SP/blk:P',
                  f11=1,
                  f12=1)
summary(ssp_tk11)

## UNBALANCED DATA
## The average are adjusted by "Least-Square-Means" methodology.
## From: formula
data(CRD2)

uCRD2 <- CRD2$dfm
uCRD2[c(3, 5, 10, 44, 45), 3] <- NA

utk1 <- TukeyC(y ~ x,
              data=uCRD2,
              which='x')
summary(utk1)

## From: lm
u1m1 <- lm(y ~ x,
          data=uCRD2)

utk2 <- TukeyC(u1m1,
              which='x')
summary(utk2)

## Factorial Experiments
## Nested: p1/N
# From: lm

uFE <- FE$dfm
uFE[c(3, 6, 7, 20, 31, 32), 5] <- NA

u1m2 <- lm(y ~ blk + N*P*K,
          uFE)

```

```
## Nested: p1/N
utk3 <- TukeyC(ulm2,
               data=uFE,
               which='P:N',
               f11=1)
summary(utk3)

## Nested: p2/n2/K
utk4 <- TukeyC(ulm2,
               data=uFE,
               which='P:N:K',
               f11=2,
               f12=2)
summary(utk4)
```

xtable.TukeyC	<i>xtable method for TukeyC objects.</i>
---------------	--

Description

Convert an TukeyC object to an xtable.TukeyC object, which can then be printed as a LaTeX or HTML table. This function is an additional method to xtable function of xtable package.

Usage

```
## S3 method for class 'TukeyC'
xtable(x, ...)
```

Arguments

x	A given object of the class TukeyC.
...	Further arguments (require by xtable::xtable).

Value

For most xtable methods, an object of class "xtable.TukeyC", "xtable" which inherits the data.frame class and contains several additional attributes specifying the table formatting options.

Author(s)

Jose Claudio Faria (<joseclaudio.faria@gmail.com>)
Enio G. Jelihovschi (<eniojelihovs@gmail.com>)
Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

See Also

[xtable](#)

Examples

```
data(RCBD)

if(require(xtable)){
  lm1 <- with(RCBD,
             lm(y ~ blk + tra,
                data=dfm))

  tk1 <- TukeyC(lm1,
                which='tra')
  tb <- xtable(tk1)
  print(tb)
}
```

Index

- * **TukeyC**
 - print.TukeyC, [12](#)
 - xtable.TukeyC, [30](#)
- * **aov**
 - cv, [5](#)
- * **coefficient of variation**
 - cv, [5](#)
- * **datasets**
 - CRD1, [4](#)
 - CRD2, [5](#)
 - FE, [6](#)
 - LSD, [7](#)
 - RCBD, [13](#)
 - sorghum, [14](#)
 - SPE, [15](#)
 - SPET, [15](#)
 - SSPE, [16](#)
- * **package**
 - boxplot.TukeyC, [2](#)
 - make.TukeyC.groups, [7](#)
 - make.TukeyC.test, [8](#)
 - plot.TukeyC, [9](#)
 - print.TukeyC, [12](#)
 - summary, [16](#)
 - Tukey test, [20](#)
 - TukeyC, [21](#)
 - xtable.TukeyC, [30](#)
- * **univar**
 - cv, [5](#)
- boxplot, [3](#)
- boxplot.TukeyC, [2](#)
- cld, [21](#)
- CRD1, [4](#)
- CRD2, [5](#)
- cv, [5](#)
- FE, [6](#)
- glht, [21](#)
- HSD.test, [21](#)
- LSD, [7](#)
- make.TukeyC.groups, [7](#)
- make.TukeyC.test, [8](#)
- plot, [11](#)
- plot.TukeyC, [9](#)
- print.TukeyC, [12](#)
- RCBD, [13](#)
- sorghum, [14](#)
- SPE, [15](#)
- SPET, [15](#)
- SSPE, [16](#)
- summary, [16](#)
- Tukey test, [20](#)
- TukeyC, [13](#), [17](#), [21](#)
- TukeyC-package (Tukey test), [20](#)
- TukeyHSD, [21](#)
- xtable, [30](#)
- xtable.TukeyC, [30](#)