

Package ‘TukeyC’

January 13, 2018

Type Package

Title Conventional Tukey Test

Version 1.3-0

Date 2018-01-12

Author José Cláudio Faria <joseclaudio.faria@gmail.com>,
Enio G. Jelihovschi <eniojelihovs@gmail.com>,
Ivan Bezerra Allaman <ivanalaman@gmail.com>

Maintainer José Cláudio Faria <joseclaudio.faria@gmail.com>

Depends R (>= 2.6.0), doBy

Suggests pbkrtest (>= 0.4-6), lme4

Description Perform the conventional Tukey test from formula, lm, aov, aovlist and lmer objects.

License GPL (>= 2)

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

Encoding latin1

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2018-01-13 17:13:13 UTC

R topics documented:

CRD1	2
CRD2	2
cv	3
FE	4
LSD	4
make.TukeyC.groups	5
make.TukeyC.test	6
plot.TukeyC	7
print.TukeyC	9

RCBD	10
sorghum	11
SPE	12
SPET	12
SSPE	13
summary	13
Tukey test	17
TukeyC	18

Index	28
--------------	-----------

CRD1	<i>Completely Randomized Design (CRD)</i>
------	---

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 *Coefficient of the experiment variation*

Description

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

Usage

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

Arguments

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

Details

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

Value

x	named numeric vector
---	----------------------

Author(s)

José Cláudio 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]

```

FE	<i>Factorial Experiment (FE)</i>
----	----------------------------------

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	<i>Latin Squares Design (LSD)</i>
-----	-----------------------------------

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 cols 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)

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

make.TukeyC.test *Make Tukey Test*

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 (upper.tri) and respective probabilities (lower.tri)
MSD	The minimum significative difference
Replicates	The replicate number for each factor

Note

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

Author(s)

José Cláudio 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,
      id.lab          = NULL,
      id.las          = 1,
      rl              = TRUE,
      rl.lty          = 3,
      rl.col          = 'gray',
      dispersion      = c('none', 'mm', 'sd', 'ci', 'cip'),
      dispersion.lty  = 1,
      title           = '', ...)
```

Arguments

x	A TukeyC object.
result	The result of the test (letters) should be visible.
replicates	The number of replicates should be visible.
pch	A vector of plotting symbols or characters.
col	A vector of colors for the means representation.
xlab	A label for the 'x' axis.
ylab	A label for the 'y' axis.
xlim	The 'x' limits of the plot.
ylim	The 'y' limits of the plot.
id.lab	Factor level names at 'x' axis.
id.las	Factor level names written either horizontally or vertically.
rl	Horizontal line connecting the circle to the 'y' axis.
rl.lty	Line type of 'rl'.

r1.col	Line color of 'r1'.
dispersion	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.
dispersion.lty	Line type of dispersion.
title	A title for the plot.
...	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` options is calculated utilizing each treatment variance as estimating of population variance. The `cip` options is calculated utilizing the means square error (MSE) as estimating of population variance.

Author(s)

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

References

Murrell, P. (2005) *R Graphics*. Chapman & 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,
     dispersion='sd',
     mm.lty=3,
     id.las=2,
     r1=FALSE)

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

tk2 <- TukeyC(x=av,
              which='x')
plot(tk2,
     r1=FALSE,
     id.las=2)

# From: lm
av_lm <- with(CRD2,
              lm(y ~ x,
                 data=dfm))

tk3 <- TukeyC(x=av_lm,
              which='x')
plot(tk3,
     dispersion='ci',
     id.las=2,
     r1=FALSE)

plot(tk3,
     dispersion='cip',
     id.las=2,
     r1=FALSE)

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).

Author(s)

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

Completely Randomized Design (CRD)

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)
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 & D.F. and Oliveira, A.C. (2000) *Experimentação em Genética e Melhoramento de Plantas*. Editora UFPA, 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	<i>Split-plot Experiment (SPE)</i>
-----	------------------------------------

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	<i>Split-plot Experiment in Time (SPET)</i>
------	---

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)  
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).

Author(s)

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

References

Chambers, J.M. & Hastie, T.J. (1992) *Statistical Models in S*. Wadsworth & 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',
                    f1=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(tk3)

```



```

## 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 funtions and S3 methods.

Author(s)

José Cláudio 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 & Oliveira A.C. (2000) *Experimentação em Genética e Melhora-mento de Plantas*. Editora UFLA.
- Steel, R.G., Torrie, J.H & 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.

<code>fl2</code>	A vector of length 1 giving the level of the second factor in nesting order tested.
<code>error</code>	The error to be considered. If from experiment at split plot or split-split plot pay attention! See details!
<code>sig.level</code>	Level of Significance used in the TukeyC algorithm to create the groups of means. The default value is 0.05.
<code>round</code>	Integer indicating the number of decimal places.
<code>adjusted.pvalue</code>	Method for adjusting p values (see <code>p.adjust</code> to more details). The possible values are: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr" and "none". The default is "none".
<code>...</code>	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:

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

Author(s)

José Cláudio 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 & Oliveira A.C. (2000) *Experimentação em Genética e Melhora-mento de Plantas*. Editora UFLA.

Steel, R.G., Torrie, J.H & 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.

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
ntk1 <- with(FE,
            TukeyC(y ~ blk + N*P*K,
                  data=dfm,
                  which='P:N',
                  f11=1))
summary(ntk1)

## Nested: p2/N
# From: lm
lm2 <- with(FE,
            lm(y ~ blk + N*P*K,
              dfm))

ntk2 <- with(FE,
            TukeyC(lm2,
                  which='P:N',
                  f11=2))
summary(ntk2)

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

```

```

ntk3 <- with(FE,
             TukeyC(av2,
                    which='N:P',
                    f11=1))
summary(ntk3)

# From: merMod
## Not run:
if(require(lme4)){
lmer2 <- with(FE,
              lmer(y ~ (1|blk) + N*P*K,
                  dfm))

ntk3 <- with(FE,
             TukeyC(lmer2,
                    which='N:P',
                    f11=1))
summary(ntk3)
}

## End(Not run)
##
## 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
sptk1 <- TukeyC(lm3,
               which='tra:year',
               f11=1)
summary(sptk1)

# 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
sptk2 <- TukeyC(lm3,
               which='year:tra',
               error='Residuals/blk:tra',
               f11=1)
summary(sptk2)

# From merMod
# Only tra
## Not run:
if(require(lme4)){
lmer3 <- with(SPET,

```

```

      lmer(y ~ blk + (1|blk:tra) + tra*year,
          dfm))

# comparison only tra
sptk3 <- TukeyC(lmer3,
               which = 'tra',
               error = 'blk:tra')
summary(sptk3)

# year1/tra
sptk4 <- TukeyC(lmer3,
               which='year:tra',
               error='Residual/blk:tra',
               fl1=1)
summary(sptk4)
}

## End(Not run)
## 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
ssptk1 <- with(SSPE,
              TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                    data=dfm,
                    which='P',
                    error='blk:P'))
summary(ssptk1)

## Main factor: SP
## It is necessary to inform the appropriate error for the test
ssptk2 <- with(SSPE,
              TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                    data=dfm,
                    which='SP',
                    error='blk:P:SP'))
summary(ssptk2)

## Main factor: SSP
ssptk3 <- with(SSPE,
              TukeyC(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                    data=dfm,
                    which='SSP'))
summary(ssptk3)

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

```



```

        data=dfm))

ssptk4 <- TukeyC(av3,
                 which='SSP')
summary(ssptk4)

## Nested: p1/SP
## It is necessary to inform the appropriate error for the test
ssptk5 <- TukeyC(av3,
                 which='P:SP',
                 error='blk:P:SP',
                 f11=1)
summary(ssptk5)

## Nested: p1/SSP
ssptk6 <- TukeyC(av3,
                 which='P:SSP',
                 f11=1)
summary(ssptk6)

## Nested: p1/sp1/SSP
## Testing SSP inside of level one of P and level one of SP
ssptk7 <- TukeyC(av3,
                 which='P:SP:SSP',
                 f11=1,
                 f12=1)
summary(ssptk7)

## Nested: p2/sp1/SSP
ssptk8 <- TukeyC(av3,
                 which='P:SP:SSP',
                 f11=2,
                 f12=1)
summary(ssptk8)

## Nested: sp1/P
## It is necessary to inform the appropriate error for the test
ssptk9 <- TukeyC(av3,
                 which='SP:P',
                 error='blk:P:SP/blk:P',
                 f11=1)

summary(ssptk9)

## Nested: ssp1/SP
ssptk10 <- TukeyC(av3,
                  which='SSP:SP',
                  error='Within/blk:P:SP',
                  f11=1)
summary(ssptk10)

## Nested: ssp1/sp1/P
## It is necessary to inform the appropriate error for the test

```

```

ssptk11 <- TukeyC(av3,
                  which='SSP:SP:P',
                  error='Within/blk:P:SP/blk:P',
                  f11=1,
                  f12=1)
summary(ssptk11)

## 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(u1m2,
               data=uFE,
               which='P:N',
               f11=1)
summary(utk3)

## Nested: p2/n2/K
utk4 <- TukeyC(u1m2,
               data=uFE,
               which='P:N:K',
               f11=2,
               f12=2)
summary(utk4)

```


Index

- *Topic **TukeyC**
 - print.TukeyC, 9
 - *Topic **aoV**
 - cv, 3
 - *Topic **coefficient of variation**
 - cv, 3
 - *Topic **datasets**
 - CRD1, 2
 - CRD2, 2
 - FE, 4
 - LSD, 4
 - RCBD, 10
 - sorghum, 11
 - SPE, 12
 - SPET, 12
 - SSPE, 13
 - *Topic **package**
 - make.TukeyC.groups, 5
 - make.TukeyC.test, 6
 - plot.TukeyC, 7
 - print.TukeyC, 9
 - summary, 13
 - Tukey test, 17
 - TukeyC, 18
 - *Topic **univar**
 - cv, 3
- c1d, 18
- CRD1, 2
- CRD2, 2
- cv, 3
- FE, 4
- glht, 18
- HSD.test, 18
- LSD, 4
- make.TukeyC.groups, 5
- make.TukeyC.test, 6
- plot, 8
- plot.TukeyC, 7
- print.TukeyC, 9
- RCBD, 10
- sorghum, 11
- SPE, 12
- SPET, 12
- SSPE, 13
- summary, 13
- Tukey test, 17
- TukeyC, 10, 14, 18
- TukeyC-package (Tukey test), 17
- TukeyHSD, 18