

# Package ‘TukeyC’

February 19, 2015

**Type** Package

**Title** Conventional Tukey Test

**Version** 1.1-5

**Date** 2014-08-16

**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), base

**Description** Perform the conventional Tukey test from aov and aovlist objects

**License** GPL (>= 2)

**Encoding** latin1

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2014-08-19 00:42:19

## R topics documented:

TukeyC-package	2
CRD1	9
CRD2	9
cv	10
FE	11
LSD	11
m.inf	12
make.TukeyC.groups	13
make.TukeyC.test	14
plot.TukeyC	15
RCBD	18

sorghum . . . . .	18
SPE . . . . .	19
SPET . . . . .	19
SSPE . . . . .	20
summary . . . . .	20
TukeyC . . . . .	25
TukeyC.nest . . . . .	29

<b>Index</b>	<b>34</b>
--------------	-----------

---

TukeyC-package	<i>Conventional Tukey Test</i>
----------------	--------------------------------

---

## 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.

## Examples

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

## The parameters can be: vectors, design matrix and the response variable,
## data.frame or aov
data(CRD2)

## From: design matrix (dm) and response variable (y) - balanced
tk1 <- with(CRD2,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ x',
                  which='x'))

summary(tk1)
plot(tk1,
     id.las=2,
     rl=FALSE)

## From: design matrix (dm) and response variable (y) - unbalanced
tk1u <- with(CRD2,
            TukeyC(x=dm[-1,],
                  y=y[-1],
                  model='y ~ x',
                  which='x',
                  dispersion='s'))

summary(tk1u)
plot(tk1u)

## From: data.frame (dfm) - balanced
tk2 <- with(CRD2,
            TukeyC(x=dfm,
                  model='y ~ x',
                  which='x',
                  dispersion='se'))

summary(tk2)
plot(tk2)

## From: data.frame (dfm) - unbalanced
tk2u <- with(CRD2,
            TukeyC(x=dfm[-1,],
                  model='y ~ x',
                  which='x'))

summary(tk2u)
```

```

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

tk3 <- TukeyC(x=av,
             which='x')
summary(tk3)

## From: aov - unbalanced
avu <- with(CRD2,
           aov(y ~ x,
              data=dfm[-1,]))
summary(avu)

tk3u <- TukeyC(x=avu,
              which='x')
summary(tk3u)

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

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

data(RCBD)

## Design matrix (dm) and response variable (y)
tk1 <- with(RCBD,
           TukeyC(x=dm,
                 y=y,
                 model='y ~ blk + tra',
                 which='tra'))
summary(tk1)
plot(tk1)

## From: data.frame (dfm), which='tra'
tk2 <- with(RCBD,
           TukeyC(x=dfm,
                 model='y ~ blk + tra',
                 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: design matrix (dm) and response variable (y)
tk1 <- with(LSD,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ rows + cols + tra',
                  which='tra'))

summary(tk1)
plot(tk1)

## From: data.frame
tk2 <- with(LSD,
            TukeyC(x=dfm,
                  model='y ~ rows + cols + tra',
                  which='tra'))

summary(tk2)

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

tk3 <- TukeyC(av,
              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
tk1 <- with(FE,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ blk + N*P*K',
                  which='N'))

summary(tk1)
plot(tk1)

## Nested: p1/N
## Testing N inside of level one of P
ntk1 <- with(FE,
             TukeyC.nest(x=dm,
```

```

                                y=y,
                                model='y ~ blk + N*P*K',
                                which='P:N',
                                f11=1))
summary(ntk1)

## Nested: k1/P
ntk2 <- with(FE,
             TukeyC.nest(x=dm,
                         y=y,
                         model='y ~ blk + N*P*K',
                         which='K:P',
                         f11=1))
summary(ntk2)

## Nested: k1/p1/N
## Testing N inside of level one of K and level one of P
ntk3 <- with(FE,
             TukeyC.nest(x=dm,
                         y=y,
                         model='y ~ blk + N*P*K',
                         which='K:P:N',
                         f11=1,
                         f12=1))
summary(ntk3)

## Nested: k2/n2/P
ntk4 <- with(FE,
             TukeyC.nest(x=dm,
                         y=y,
                         model='y ~ blk + N*P*K',
                         which='K:N:P',
                         f11=2,
                         f12=2))
summary(ntk4)

## Nested: p1/n1/K
ntk5 <- with(FE,
             TukeyC.nest(x=dm,
                         y=y,
                         model='y ~ blk + N*P*K',
                         which='P:N:K',
                         f11=1,
                         f12=1))
summary(ntk5)

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

data(SPE)

```

```

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

## From: design matrix (dm) and response variable (y)
## Main factor: P
tk1 <- with(SPE,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ blk + P*SP + Error(blk/P)',
                  which='P',
                  error='blk:P'))

summary(tk1)

## Main factor: SP
tk2 <- with(SPE,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ blk + P*SP + Error(blk/P)',
                  which='SP',
                  error='Within'))

summary(tk2)
plot(tk2)

## Nested: p=1/sp
tkn1 <- with(SPE,
            TukeyC.nest(x=dm,
                       y=y,
                       model='y ~ blk + P*SP + Error(blk/P)',
                       which='P:SP',
                       error='Within',
                       fl1=1 ))

summary(tkn1)

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

data(SSPE)

## From: design matrix (dm) and response variable (y)
## Main factor: P
tk1 <- with(SSPE,
            TukeyC(dm,
                  y,
                  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                  which='P',
                  error='blk:P'))

summary(tk1)

# Main factor: SP
tk2 <- with(SSPE,
            TukeyC(dm,

```

```

        y,
        model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
        which='SP',
        error='blk:P:SP'))
summary(tk2)

# Main factor: SSP
tk3 <- with(SSPE,
            TukeyC(dm,
                  y,
                  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                  which='SSP',
                  error='Within'))
summary(tk3)
plot(tk3)

## Nested: p1/SP
tkn1 <- with(SSPE,
             TukeyC.nest(dm,
                        y,
                        model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                        which='P:SP',
                        error='blk:P:SP',
                        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
## Testing SSP inside of level one of P and level one of SP
tkn6 <- TukeyC.nest(av,
                   which='P:SP:SSP',
                   error='Within',
                   f11=1,
                   f12=1)
summary(tkn6)
plot(tkn6)

## Nested: p2/sp1/SSP
tkn7 <- TukeyC.nest(av,
                   which='P:SP:SSP',
                   error='Within',
                   f11=2,
                   f12=1)
summary(tkn7)
plot(tkn7)

```



---

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

*Completely Randomized Design ('CRD')*

---

**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

<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)

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

*Factorial Experiment (FE)*

---

### **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.

---

m.inf	<i>Calculates the Means and Dispersion</i>
-------	--

---

**Description**

Calculates the means and dispersion for one, two and three factor(s) by `model` and `model.frame` functions.

**Usage**

```
m.inf.1a(x,
        which,
        dispersion=c('mm', 's', 'se'))
m.inf.1b(x,
        which,
        dispersion=c('mm', 's', 'se'))
m.inf.2a(x,
        which1,
        which2,
        dispersion=c('mm', 's', 'se'))
m.inf.2b(x,
        which1,
        which2,
        dispersion=c('mm', 's', 'se'))
m.inf.3a(x,
        which1,
        which2,
        which3,
        dispersion=c('mm', 's', 'se'))
m.inf.3b(x,
        which1,
        which2,
        which3,
        dispersion=c('mm', 's', 'se'))
```

**Arguments**

x	A SK object.
which	The name of the treatment to be used in the comparison. For all the value is determined internally by the package.

which1	The name of the treatment to be used in the comparison. For all the value is determined internally by the package.
which2	The name of the treatment to be used in the comparison. For all the value is determined internally by the package.
which3	The name of the treatment to be used in the comparison. For all the value is determined internally by the package.
dispersion	The dispersion to be considered to the means. The possible vaues are: 'mm' = 'minimum and maximum', 's' = 'standart deviation', 'se' = 'standart deviation of the mean'.

**Note**

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

**Author(s)**

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

---

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)**

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(r=r,  
                 MSE=MSE,  
                 m.inf=m.inf,  
                 ord=ord,  
                 sig.level=sig.level,  
                 dfr=dfr,  
                 bal=bal,  
                 mt=mt,  
                 round=round)
```

**Arguments**

r	A vector of the number of replicates of each level of the factor being tested.
MSE	A vector of length 1 giving the mean squared error.
m.inf	A matrix of the levels of the factor being tested in decreasing order.
ord	A vector of ordered levels of the factor being tested.
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'.
bal	A vector of length 1 giving the information wheter the experiment is or not balanced.
mt	The model table.
round	Integer indicating the number of decimal places.

**Value**

A list with 7 slots containing the most important results of the test performed:

Table	Table of means of the factors
Means	The means of the factors
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,
```

```

r1.col='gray',
mm=TRUE,
mm.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.
r1	Horizontal line connecting the circle to the 'y' axis.
r1.lty	Line type of 'r1'.
r1.col	Line color of 'r1'.
mm	Vertical line through the circle (mean value) linking the minimum to the maximum of the factor level values corresponding to that mean value.
mm.lty	Line type of mm.
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 minimum and maximum of the values corresponding to each group mean.

### 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: vectors x and y
tk1 <- with(CRD2,
            TukeyC(x=x,
                  y=y,
                  model='y ~ x',
                  which='x'))

plot(tk1,
     id.las=2,
     rl=FALSE)

## From: design matrix (dm) and response variable (y)
tk2 <- with(CRD2,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ x',
                  which='x'))

plot(tk2,
     mm.lty=3,
     id.las=2,
     rl=FALSE)

## From: data.frame (dfm)
tk3 <- with(CRD2,
            TukeyC(x=dfm,
                  model='y ~ x',
                  which='x'))

plot(tk3,
     id.las=2,
     rl=FALSE)

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

tk4 <- with(CRD2,
            TukeyC(x=av,
                  which='x'))
```

```
plot(tk4,
      rl=FALSE,
      id.las=2)
```

---

 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 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)
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: vectors, design matrix and the response variable,
## data.frame or aov
data(CRD2)

## From: design matrix (dm) and response variable (y)
tk1 <- with(CRD2,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ x',
                  which='x',
                  id.trim=5))

summary(tk1)

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

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

data(RCBD)

## Design matrix (dm) and response variable (y)
tk1 <- with(RCBD,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ blk + tra',
                  which='tra'))

summary(tk1)

##
```

```

## 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: design matrix (dm) and response variable (y)
tk1 <- with(LSD,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ rows + cols + tra',
                  which='tra'))

summary(tk1)

##
## 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
tk1 <- with(FE,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ blk + N*P*K',
                  which='N'))

summary(tk1)

## Nested: p1/N
## Testing N inside of level one of P
ntk1 <- with(FE,
            TukeyC.nest(x=dm,
                       y=y,
                       model='y ~ blk + N*P*K',
                       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.nest(x=dm,
                       y=y,
                       model='y ~ blk + N*P*K',
                       which='K:P:N',
                       f11=1,

```

```

                                f12=1))
summary(ntk2)

## Nested: k2/n2/P
ntk3 <- with(FE,
             TukeyC.nest(x=dm,
                        y=y,
                        model='y ~ blk + N*P*K',
                        which='K:N:P',
                        f11=2,
                        f12=2))

summary(ntk3)

## Nested: p1/n1/K
ntk4 <- with(FE,
             TukeyC.nest(x=dm,
                        y=y,
                        model='y ~ blk + N*P*K',
                        which='P:N:K',
                        f11=1,
                        f12=1))

summary(ntk4)

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

data(SPE)

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

## From: design matrix (dm) and response variable (y)
## Main factor: P
tk1 <- with(SPE,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ blk + P*SP + Error(blk/P)',
                  which='P',
                  error='blk:P'))

summary(tk1)

## Nested: p1/SP
tkn1 <- with(SPE,
            TukeyC.nest(x=dm,
                       y=y,
                       model='y ~ blk + P*SP + Error(blk/P)',
                       which='P:SP',
                       error='Within',
                       f11=1 ))

summary(tkn1)

```

```

data(SSPE)

## From: design matrix (dm) and response variable (y)
## Main factor: P
tk1 <- with(SSPE,
            TukeyC(dm,
                  y,
                  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                  which='P',
                  error='blk:P'))

summary(tk1)

# Main factor: SP
tk2 <- with(SSPE,
            TukeyC(dm,
                  y,
                  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                  which='SP',
                  error='blk:P:SP'))

summary(tk2)

# Main factor: SSP
tk3 <- with(SSPE,
            TukeyC(dm,
                  y,
                  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                  which='SSP',
                  error='Within'))

summary(tk3)

## Nested: p1/SSP
tkn1 <- with(SSPE,
            TukeyC.nest(dm,
                       y,
                       model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                       which='P:SSP',
                       error='blk:P:SP',
                       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.nest(av,
                   which='P:SP:SSP',
                   error='Within',
                   f11=1,
                   f12=1)

summary(tkn2)

```



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

summary(tkn3)
```

---

TukeyC

*The TukeyC Test for Single Experiments*

---

### Description

These are methods for objects of class vector, matrix or data.frame joined as default, aov and aovlist for single experiments.

### Usage

```
## Default S3 method:
TukeyC(x,
       y=NULL,
       model,
       which,
       error,
       sig.level=.05,
       round=2,
       dispersion=c('mm', 's', 'se'), ...)

## S3 method for class 'aov'
TukeyC(x,
       which=NULL,
       sig.level=.05,
       round=2,
       dispersion=c('mm', 's', 'se'), ...)

## S3 method for class 'aovlist'
TukeyC(x,
       which,
       error,
       sig.level=.05,
       round=2,
       dispersion=c('mm', 's', 'se'), ...)
```

### Arguments

x                    A design matrix, data.frame or an aov object.

<code>y</code>	A vector of response variable. It is necessary to inform this parameter only if 'x' represent the design matrix.
<code>which</code>	The name of the treatment to be used in the comparison. The name must be inside quoting marks.
<code>model</code>	If 'x' is a <code>data.frame</code> object, the model to be used in the aov must be specified.
<code>error</code>	The error to be considered.
<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>dispersion</code>	The dispersion to be considered to the means. The possible vaues are: 'mm' = 'minimum and maximum', 's' = 'standart deviation', 'se' = 'standart deviation of the mean'.
<code>...</code>	Potential further arguments (required by generic).

### Details

The function `TukeyC` returns an object of class `TukeyC` respectivally 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.

### Value

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

<code>av</code>	A list storing the result of aov.
<code>groups</code>	A vector of length equal the number of factor levels marking the groups generated.
<code>nms</code>	A vector of the labels of the factor levels.
<code>ord</code>	A vector which keeps the position of the means of the factor levels in decreasing order.
<code>m.inf</code>	A matrix which keeps the means and dispersion of the factor levels in decreasing order.
<code>sig.level</code>	A vector of length 1 giving the level of significance of the test.

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

## The parameters can be: vectors, design matrix and the response variable,
## data.frame or aov
data(CRD2)

## From: design matrix (dm) and response variable (y)
tk1 <- with(CRD2,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ x',
                  which='x',
                  id.trim=5))
summary(tk1)

## From: data.frame (dfm)
tk2 <- with(CRD2,
            TukeyC(x=dfm,
                  model='y ~ x',
                  which='x',
                  id.trim=5))
summary(tk2)

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

tk3 <- with(CRD2,
            TukeyC(x=av,
                  which='x',
                  id.trim=5))
summary(tk3)

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

```
##

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

data(RCBD)

## Design matrix (dm) and response variable (y)
tk1 <- with(RCBD,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ blk + tra',
                  which='tra'))
summary(tk1)

## From: data.frame (dfm), which='tra'
tk2 <- with(RCBD,
            TukeyC(x=dfm,
                  model='y ~ blk + tra',
                  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: design matrix (dm) and response variable (y)
tk1 <- with(LSD,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ rows + cols + tra',
                  which='tra'))
summary(tk1)

## From: data.frame
tk2 <- with(LSD,
            TukeyC(x=dfm,
                  model='y ~ rows + cols + tra',
                  which='tra'))
summary(tk2)

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

```

tk3 <- TukeyC(av,
              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
tk1 <- with(FE,
            TukeyC(x=dm,
                  y=y,
                  model='y ~ blk + N*P*K',
                  which='N'))
summary(tk1)

## Nested: p1/N
ntk1 <- with(FE,
             TukeyC.nest(x=dm,
                        y=y,
                        model='y ~ blk + N*P*K',
                        which='P:N',
                        f11=1))

summary(ntk1)

```

---

TukeyC.nest

*The TukeyC test for Factorial, Split-plot and Split-Split plot Experiments*


---

### Description

These are methods for objects of class vector, matrix or data.frame joined as default, aov and aovlist for factorial, split-plot and split-split-plot experiments.

### Usage

```

## Default S3 method:
TukeyC.nest(x,
            y=NULL,
            model,
            which,
            error,
            f11,
            f12=0,

```

```

        sig.level=.05,
        round=2,
        dispersion=c('mm', 's', 'se'), ...)

## S3 method for class 'aov'
TukeyC.nest(x,
            which,
            f11,
            f12=0,
            sig.level=.05,
            round=2,
            dispersion=c('mm', 's', 'se'), ...)

## S3 method for class 'aovlist'
TukeyC.nest(x,
            which,
            error,
            f11,
            f12=0,
            sig.level=.05,
            round=2,
            dispersion=c('mm', 's', 'se'), ...)

```

### Arguments

x	A design matrix, data.frame or an aov object.
y	A vector of response variable. It is necessary to inform this parameter only if x represent the design matrix.
which	The name of the treatment to be used in the comparison. The name must be inside quoting marks.
model	If x is a data.frame object, the model to be used in the aov must be specified.
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, only in case of split-plots experiments.
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.
dispersion	The dispersion to be considered to the means. The possible vaues are: 'mm' = 'minimum and maximum', 's' = 'standart deviation', 'se' = 'standart deviation of the mean'.
...	Potential further arguments (required by generic).

### Details

The function `TukeyC.nest` returns an object of class `TukeyC.nest` 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.

### Value

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

<code>av</code>	A list storing the result of <code>aov</code> .
<code>groups</code>	A vector of length equal the number of factor levels marking the groups generated.
<code>nms</code>	A vector of the labels of the factor levels.
<code>ord</code>	A vector which keeps the position of the means of the factor levels in decreasing order.
<code>m.inf</code>	A matrix which keeps the means and dispersion of the factor levels in decreasing order.
<code>sig.level</code>	A vector of length 1 giving the level of significance of the test.
<code>r</code>	A vector of length 1 giving the number of replicates.
<code>which</code>	The name of the factor whose levels were tested.
<code>tab</code>	An array keeping the names of the factors and factor levels and also the mean value of the repetitions for every combination of factor levels.
<code>f1</code>	A vector of length 1 giving the level of the first factor in nesting order tested.
<code>f12</code>	A vector of length 1 giving the level of the second factor in nesting order tested.

### 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 Melhoria 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

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

data(SSPE)
```

```

## From: design matrix (dm) and response variable (y)
## Main factor: P
tk1 <- with(SSPE,
            TukeyC(dm,
                  y,
                  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                  which='P',
                  error='blk:P'))

summary(tk1)
plot(tk1)

# Main factor: SP
tk2 <- with(SSPE,
            TukeyC(dm,
                  y,
                  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                  which='SP',
                  error='blk:P:SP',
                  dispersion='s'))

summary(tk2)
plot(tk2)

# Main factor: SSP
tk3 <- with(SSPE,
            TukeyC(dm,
                  y,
                  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                  which='SSP',
                  error='Within',
                  dispersion='se'))

summary(tk3)
plot(tk3)

## Nested: p1/SP
tkn1 <- with(SSPE,
            TukeyC.nest(dm,
                       y,
                       model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
                       which='P:SP',
                       error='blk:P:SP',
                       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
## Testing SSP inside of level one of P and level one of SP
tkn2 <- TukeyC.nest(av,
                   which='P:SP:SSP',

```



```
                                error='Within',
                                f11=1,
                                f12=1)
summary(tkn2)

## Nested: p2/sp1/SSP
tkn3 <- TukeyC.nest(av,
                    which='P:SP:SSP',
                    error='Within',
                    f11=2,
                    f12=1)
summary(tkn3)
```

# Index

- \*Topic **aov**
    - cv, [10](#)
  - \*Topic **coefficient of variation**
    - cv, [10](#)
  - \*Topic **datasets**
    - CRD1, [9](#)
    - CRD2, [9](#)
    - FE, [11](#)
    - LSD, [11](#)
    - RCBD, [18](#)
    - sorghum, [18](#)
    - SPE, [19](#)
    - SPET, [19](#)
    - SSPE, [20](#)
  - \*Topic **m.inf**
    - m.inf, [12](#)
  - \*Topic **package**
    - make.TukeyC.groups, [13](#)
    - make.TukeyC.test, [14](#)
    - plot.TukeyC, [15](#)
    - summary, [20](#)
    - TukeyC, [25](#)
    - TukeyC-package, [2](#)
    - TukeyC.nest, [29](#)
  - \*Topic **univar**
    - cv, [10](#)
- 
- c1d, [2](#)
  - CRD1, [9](#)
  - CRD2, [9](#)
  - cv, [10](#)
  
  - FE, [11](#)
  
  - glht, [2](#)
  
  - HSD.test, [2](#)
  
  - LSD, [11](#)
  
  - m.inf, [12](#)
  
  - make.TukeyC.groups, [13](#)
  - make.TukeyC.test, [14](#)
  
  - plot, [17](#)
  - plot.TukeyC, [15](#)
  
  - RCBD, [18](#)
  
  - sorghum, [18](#)
  - SPE, [19](#)
  - SPET, [19](#)
  - SSPE, [20](#)
  - summary, [20](#)
  
  - TukeyC, [21](#), [25](#)
  - TukeyC-package, [2](#)
  - TukeyC.nest, [29](#)
  - TukeyHSD, [2](#)