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 <eniojeli@gmail.com>
Ivan Bezerra Allaman <ivanalam@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
**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 inferential 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 aproach of R.

So, the main aim of this package is make available in R environment the conventional aproach 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**


Examples

```r
## 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, r1=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) - balanced
tk2u <- with(CRD2, TukeyC(x=dfm[-1,], model='y ~ x', which='x'))
summary(tk2u)
```
## From: aov - balanced
```r
av <- with(CRD2,
aov(y ~ x,
data=dfm))
summary(av)
```
save as `tk3` and ```
```summary(tk3)```

## From: aov - unbalanced
```r
avu <- with(CRD2,
aov(y ~ x,
data=dfm[-1,]))
summary(avu)
```
save as `tk3u` and ```
```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)
```r
tk1 <- with(RCBD,
  TukeyC(x=dm,
y=y,
  model='y ~ blk + tra',
  which='tra'))
```
save as `tk1` and ```summary(tk1)```plot(tk1)

## From: data.frame (dfm), which='tra'
```r
tk2 <- with(RCBD,
  TukeyC(x=dfm,
  model='y ~ blk + tra',
  which='tra'))
```
save as `tk2` and ```summary(tk2)```plot(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 ~ N*P*K + blk',  
    which='N'))
summary(ntk1)
plot(ntk1)
TukeyC-package  

```r
summary(ntk1)

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

```r
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

```r
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

```r
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)
```

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

```r
data(SSPE)

## From: design matrix (dm) and response variable (y)
## Main factor: P

```r
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

```r
tk2 <- with(SSPE, 
  TukeyC(dm,

```
TukeyC-package

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)
pot(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',
   fl1=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',
   fl1=1,
   fl2=1)

summary(tkn6)
pot(tkn6)

## Nested: p2/sp1/SSP
tkn7 <- TukeyC.nest(av,
   which='P:SP:SSP',
   error='Within',
   fl1=2,
   fl2=1)

summary(tkn7)
pot(tkn7)
**CRD1**

*Completely Randomized Design (CRD)*

**Description**

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

**Usage**

```r
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**

```r
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 lm, aov or aovlist.

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))
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.

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.
make.TukeyC.groups

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 values 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>)

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 tested 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**

```r
def 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 repicates 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 whether the experiment is or not balanced.
- `mt` The model table.
- `round` Integer indicating the number of decimal places.
plot.TukeyC

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

<table>
<thead>
<tr>
<th>Slot</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Table of means of the factors</td>
</tr>
<tr>
<td>Means</td>
<td>The means of the factors</td>
</tr>
<tr>
<td>Result</td>
<td>The result of the Tukey test</td>
</tr>
<tr>
<td>Sig.Level</td>
<td>The significance of the test</td>
</tr>
<tr>
<td>Diff_Prob</td>
<td>A matrix with the observed means differences (upper.tri) and respective</td>
</tr>
<tr>
<td>MSD</td>
<td>The minimum significant difference</td>
</tr>
<tr>
<td>Replicates</td>
<td>The replicate number for each factor</td>
</tr>
</tbody>
</table>

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
```r
## S3 method for class 'TukeyC'
pplot(x,
       result=TRUE,
       replicates=TRUE,
       pch=19,
       col=NULL,
       xlab=NULL,
       ylab=NULL,
       xlim=NULL,
       ylim=NULL,
       id.lab=NULL,
       id.las=1,
       r1=TRUE,
       r1.lty=3,
       ...)```

rl.col='gray',
mn=TRUE,
mn.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’.
rl.col
Line color of ‘rl’.
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 serie of points (the means) and a vertical line showing the minimum e 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

See Also

plot

Examples

```r
## Examples: Completely Randomized Design (CRD)
## More details: demo(package="TukeyC")
##
library(TukeyC)
data(CRD)

## From: vectors x and y
tk1 <- with(CRD,  
  TukeyC(x=x,  
        y=y,  
        model='y ~ x',  
        which='x'))
plot(tk1,  
     id.las=2,  
     r1=FALSE)

## From: design matrix (dm) and response variable (y)
tk2 <- with(CRD,  
  TukeyC(x=dm,  
        y=y,  
        model='y ~ x',  
        which='x'))
plot(tk2,  
     mm.1ty=3,  
     id.las=2,  
     r1=FALSE)

## From: data.frame (dfm)
tk3 <- with(CRD,  
  TukeyC(x=dfm,  
        model='y ~ x',  
        which='x'))
plot(tk3,  
     id.las=2,  
     r1=FALSE)

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

tk4 <- with(CRD,  
  TukeyC(x=av,  
         which='x'))
```
Randomized Complete Block Design (RCBD)

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

Usage
```r
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.

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
```r
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 tree 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).
**Description**

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

**Usage**

```r
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.

**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(SSPE)

Source

SSPE

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

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

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>A given object of the class TukeyC.</td>
</tr>
<tr>
<td>complete</td>
<td>A logical value indicating if the summary is complete (mean difference and p-value) or only the groups.</td>
</tr>
<tr>
<td>...</td>
<td>Potential further arguments (required by generic).</td>
</tr>
</tbody>
</table>
Author(s)
José Cláudio Faria (<joseclaudio.faria@gmail.com>)
Enio Jelihovschi (<eniojelihovs@gmail.com>)
Ivan Bezerra Allaman (<ivanalaman@gmail.com>)

References

See Also
TukeyC

Examples
```r
## 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)
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)
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",
    f1L = 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",
    f1L = 1,
### Nested: k2/n2/P

```r
tk3 <- with(FE,
  TukeyC.nest(x=dm,
    y=y,
    model='y ~ blk + N*P*K',
    which='K:N:P',
    f1=2,
    f2=2))
summary(tk3)
```

### Nested: p1/n1/K

```r
tk4 <- with(FE,
  TukeyC.nest(x=dm,
    y=y,
    model='y ~ blk + N*P*K',
    which='P:N:K',
    f1=1,
    f2=1))
summary(tk4)
```

---

### Example: Split-plot Experiment (SPE)

```r
data(SPE)
```

#### Design Parameters:
- **Main Factor:** P

```r
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

```r
tkn1 <- with(SPE,
  TukeyC.nest(x=dm,
    y=y,
    model='y ~ blk + P*SP + Error(blk/P)',
    which='P:SP',
    error='Within',
    f1=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,
  Tukey.C.nest(dm, 
  y, 
  model='y ~ blk + P*SP*SSP + Error(blk/P/SP)',
  which='P:SSP',
  error='blk:P:SP',
  f1=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',
  f1=1,
  f2=1)
summary(tkn2)
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

```r
# Default S3 method:
TukeyC(x,
y=NULL,
model1,
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.
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.

error The error to be considered.

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 values are: 'mm' = 'minimum and maximum', 's' = 'standart deviation', 'se' = 'standart deviation of the mean'.

... Potential further arguments (required by generic).

Details

The function TukeyC returns an object of class TukeyC respectively 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:

av A list storing the result of aov.

groups A vector of length equal the number of factor levels marking the groups generated.

nms A vector of the labels of the factor levels.

ord A vector which keeps the position of the means of the factor levels in decreasing order.

m.inf A matrix which keeps the means and dispersion of the factor levels in decreasing order.

sig.level 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


Examples

```r
## 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)
TukeyC.nest

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*K',
  which='N'))
summary(tk1)

## Nested: p1/N
ntk1 <- with(FE,
  TukeyC.nest(x=dm,
y=y,
  model='y ~ blk + N*K',
  which='P:N',
  fl1=1))
summary(ntk1)

tukeycNest

The TukeyC test for Factorial, Split-plot and Split-split-plot Experi-
ments

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,
fl1,
fl2=0,
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.

f1  A vector of length 1 giving the level of the first factor in nesting order tested.

f2  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 values are: c('mm', 's', 'se') = 'minimum and maximum', 's' = 'standard deviation', 'se' = 'standard 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:

- `av`: A list storing the result of `aov`.
- `groups`: A vector of length equal the number of factor levels marking the groups generated.
- `nms`: A vector of the labels of the factor levels.
- `ord`: A vector which keeps the position of the means of the factor levels in decreasing order.
- `m.inf`: A matrix which keeps the means and dispersion of the factor levels in decreasing order.
- `sig.level`: A vector of length 1 giving the level of significance of the test.
- `r`: A vector of length 1 giving the number of replicates.
- `which`: The name of the factor whose levels were tested.
- `tab`: 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.
- `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.

**Author(s)**

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

**References**


**Examples**

```r
## Example: Split-split-plot Experiment (SSPE)
## More details: demo(package='TukeyC')
##
data(SSPE)
```
## From: design matrix (dm) and response variable (y)

## Main factor: P

```r
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

```r
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

```r
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

```r
tkn1 <- with(SSPE.nest(dm,
  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

```r
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

```r
tkn2 <- TukeyC.nest(av,
  which='P:SP:SSP',
  ...)
TukeyC.nest

error='Within',
fl1=1,
fl2=1)
summary(tkn2)

## Nested: p2/sp1/SSP

tkn3 <- TukeyC.nest(av,
which='P:SP:SSP',
error='Within',
fl1=2,
fl2=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
  SPE; 19
  SSPE, 20

*Topic m.inf
  m.inf, 12

*Topic package
  make.TukeyC.groups, 13
  make.TukeyC.test, 14
  plot, 17
  plot.TukeyC, 15
  RCB, 18
  sorghum, 18
  SPE, 19
  SPE; 19
  SSPE, 20
  summary, 20
  TukeyC, 21, 25
  TukeyC-package, 2
  TukeyC.nest, 29
  TukeyHSD, 2

*Topic univar
  cv, 10
  cld, 2
  CRD1, 9
  CRD2, 9
  cv, 10
  FE, 11
  glht, 2
  HSD.test, 2
  LSD, 11
  m.inf, 12