

Package ‘ScottKnott’

March 29, 2010

Type Package

Title The ScottKnott Clustering Algoritm

Version 1.1.0

Date 2010-03-28

Author Enio Jelihovschi <eniojelihovs@gmail.com>, Jose Claudio Faria
<joseclaudio.faria@gmail.com> and Sergio Oliveira <solive@uesc.br>

Maintainer Enio Jelihovschi <eniojelihovs@gmail.com>

Depends R (>= 2.6.0), stats, base

Description Division of an ANOVA experiment treatment means into homogeneous distinct groups
using the clustering method of Scott & Knott

License GPL (>= 2)

R topics documented:

ScottKnott-package	2
CRD1	7
CRD2	7
FE	7
LSD	8
MaxValue	8
plot.SK	10
RCBD	12
SK	12
SK.nest	15
SPE	18
SSPE	18
summary	18

Index	22
--------------	-----------

Description

The Scott & Knott clustering algorithm is a very useful clustering algorithm widely used as a multiple comparison method in the Analysis of Variance context, as for example Gates and Bilbro (1978), Bony et al. (2001), Dilon et al. (2002) and Jyotsna et al. (2003).

It was developed by Scott, A.J. and Knott, M (Scott and Knott, 1974). All methods used up to that date as, for example, the t-test, Tukey, Duncan, Newman-Keuls procedures, have overlapping problems. By overlapping we mean the possibility of one or more treatments to be classified in more than one group, in fact, as the number of treatments reach a number of twenty or more, the number of overlappings could increase as reaching 5 or greater what makes almost impossible to the experimenter to really distinguish the real groups to which the means should belong. The Scott & Knott method does not have this problem, what is often cited as a very good quality of this procedure.

The Scott & Knott method make use of a clever algorithm of cluster analysis, where, starting from the the whole group of observed mean effects, it divides, and keep dividing the sub-groups in such a way that the intersection of any two groups formed in that manner is empty.

Using their own words 'we study the consequences of using a well-known method of cluster analysis to partition the sample treatment means in a balanced design and show how a corresponding likelihood ratio test gives a method of judging the significance of difference among groups attained'.

Many studies, using the method of Monte Carlo, suggest that the Scott Knott method performs very well compared to other methods due to fact that it has high power and type I error rate almost always in accordance with the nominal levels. The ScottKnott package performs this algorithm starting either from `vectors`, `matrices` or `data.frames` joined as default, a `aov` or `aovlist` resulting object of previous analysis of variance. The results are given in the usual way as well as in graphical way using thermometers with different group colors.

In a few words, the test of Scott & Knott is a clustering algorithm used as an one of the alternatives where multiple comparison procedures are applied with a very important and almost unique characteristic: it does not present overlapping in the results.

Details

Package:	ScottKnott
Type:	Package
Version:	1.0.0
Date:	2008-08-19
License:	GPL (>= 2)

Author(s)

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

Sergio Oliveira (<solive@uesc.br>)

References

- Bony S, Pichon N, Ravel C, Durix A, Balfourier F, Guillaumin JJ 2001. The Relationship between Mycotoxin Synthesis and Isolate Morphology in Fungal Endophytes of *Lolium perenne*. *New Phytologist*, **1521**, 125-137.
- Borges LC, FERREIRA DF 2003. Poder e taxas de erro tipo I dos testes Scott-Knott, Tukey e Student-Newman-Keuls sob distribuicoes normal e nao normais dos residuos. Power and type I errors rate of Scott-Knott, Tukey and Student-Newman-Keuls tests under normal and no-normal distributions of the residues. *Rev. Mat. Estat.*, Sao Paulo, **211**: 67-83.
- Calinski T, Corsten LCA 1985. Clustering Means in ANOVA by Simultaneous Testing. *Biometrics*, **411**, 39-48.
- Da Silva EC, Ferreira DF, Bearzoti E 1999. Evaluation of power and type I error rates of Scott-Knott's test by the method of Monte Carlo. *Cienc. agrotec.*, Lavras, **23**, 687-696.
- Dilson AB, David SD, Kazimierz J, William WK 2002. Half-sib progeny evaluation and selection of potatoes resistant to the US8 genotype of *Phytophthora infestans* from crosses between resistant and susceptible parents. *Euphytica*, **125**, 129-138.
- Gates CE, Bilbro JD 1978. Illustration of a Cluster Analysis Method for Mean Separation. *Agron J*, **70**, 462-465.
- Jyotsna S, Zettler LW, van Sambeek JW, Ellersieck MR, Starbuck CJ 2003. Symbiotic Seed Germination and Mycorrhizae of Federally Threatened *Platanthera Praeclara* Orchidaceae. *American Midland Naturalist*, **1491**, 104-120.
- Ramalho MAP, Ferreira DF, Oliveira AC 2000. *Experimentacao em Genetica e Melhoramento de Plantas*. Editora UFLA.
- Scott RJ, Knott M 1974. A cluster analysis method for grouping mans in the analysis of variance. *Biometrics*, **30**, 507-512.

Examples

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

## 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)
sk1 <- with(CRD2, SK(x=dm, y=y, model='y ~ x', which='x', sig.level=0.005,
  id.trim=5))
summary(sk1)
plot(sk1, col=rainbow(max(sk1$groups)), mm.lty=3, id.las=2, rl=FALSE,
  title='Factor levels, sig.level=0.005', )

## From: data.frame (dfm)
sk2 <- with(CRD2, SK(x=dfm, model='y ~ x', which='x', id.trim=5))
summary(sk2)
plot(sk2, col=rainbow(max(sk2$groups)), id.las=2, rl=FALSE)
```

```

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

sk3 <- with(CRD2, SK(x=av, which='x', id.trim=5))
summary(sk3)
plot(sk3, col=rainbow(max(sk3$groups)), rl=FALSE, id.las=2, title=NULL)

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

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

data(RCBD)

## Design matrix (dm) and response variable (y)
sk1 <- with(RCBD, SK(x=dm, y=y, model='y ~ blk + tra', which = 'tra'))
summary(sk1)
plot(sk1)

## From: data.frame (dfm), which='tra'
sk2 <- with(RCBD, SK(x=dfm, model='y ~ blk + tra', which='tra'))
summary(sk2)
plot(sk2, mm.lty=3, title='Factor levels')

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

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

data(LSD)

## From: design matrix (dm) and response variable (y)
sk1 <- with(LSD, SK(x=dm, y=y, model='y ~ rows + cols + tra', which='tra'))
summary(sk1)
plot(sk1)

## From: data.frame
sk2 <- with(LSD, SK(x=dfm, model='y ~ rows + cols + tra', which='tra'))
summary(sk2)
plot(sk2, title='Factor levels')

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

sk3 <- SK(av, which='tra')
summary(sk3)
plot(sk3, title='Factor levels')

##

```

```

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

## 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
sk1 <- with(FE, SK(x=dm, y=y, model='y ~ blk + N * P * K',
  which='N'))
summary(sk1)
plot(sk1, title='Main effect: N')

## Nested: N/P=1
nsk1 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + N * P * K',
  which='N:P', fl2=1))
summary(nsk1)
plot(nsk1, title='Effect: N/P=1')

## Nested: P/K=1
nsk2 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + N * P * K',
  which='P:K', fl2=1))
summary(nsk2)
plot(nsk2, title='Effect: P/K=1')

## Nested: N/P=2/K=2
nsk3 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + N * P * K',
  which='N:P:K', fl2=2, fl3=2))
summary(nsk3)
plot(nsk3, title='Effect: N/P=2/K=2')

## Nested: P/N=1/K=1
nsk4 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + P * N * K',
  which='P:N:K', fl2=1, fl3=1))
summary(nsk4)
plot(nsk4, title='Effect: P/N=1/K=1')

## Nested: K/N=1/P=1
nsk5 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + K * N * P',
  which='K:N:P', fl2=1, fl3=1))
summary(nsk5)
plot(nsk5, title='Effect: K/N=1/P=1')

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

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
sk1 <- with(SPE, SK(x=dm, y=y, model='y ~ blk + sp*p + Error(blk/p)',
  which='p', error='blk:p'))
summary(sk1)
plot(sk1)

## Main factor: sp
sk2 <- with(SPE, SK(x=dm, y=y, model='y ~ blk + sp*p + Error(blk/p)',
  which='sp', error='Within', sig.level=0.025 ))
summary(sk2)
plot(sk2, xlab='Groups', ylab='Main effect: sp',
  title='Main effect: sp, sig.level=0.025')

## Nested: sp/p=1
skn1 <- with(SPE, SK.nest(x=dm, y=y, model='y ~ blk + sp*p + Error(blk/p)',
  which='sp:p', error='Within', fl2=1 ))
summary(skn1)
plot(skn1, title='Effect: sp/p=1')

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

data(SSPE)

## From: design matrix (dm) and response variable (y)
## Main factor: p
sk1 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='p', error='blk:p'))
summary(sk1)

# Main factor: sp
sk2 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='sp', error='blk:p:sp', sig.level=0.025))
summary(sk2)

# Main factor: ssp
sk3 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='ssp', error='Within', sig.level=0.1))
summary(sk3)

## Nested: sp/p=1
skn1 <- with(SSPE, SK.nest(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='sp:p', error='blk:p:sp', fl2=1))
summary(skn1)

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

## Nested: ssp/sp/p (at various levels of sp and p)
skn6 <- SK.nest(av, which='ssp:sp:p', error='Within', fl2=1, fl3=1)
summary(skn6)

skn7 <- SK.nest(av, which='ssp:sp:p', error='Within', fl2=2, fl3=1)
summary(skn7)

```

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

Description

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

Usage

```
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 `ScottKnott` package related to Completely Randomized Design (CRD).

Usage

```
CRD2
```

Details

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

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

Description

A `list` illustrating the resources of `ScottKnott` package related to Factorial Experiment (FE).

Usage

```
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 `ScottKnott` package related to Latin Squares Design (LSD).

Usage

LSD

Details

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

MaxValue

*Algorithm for Pre-order Path in Binary Decision Tree***Description**

Builds groups of means, according to the method of Scott & Knott.

Usage

```
MaxValue(g, means, mMSE, dfr, sig.level, k, group, ngroup, markg, g1=g,
         sqsum=rep(0, g1))
```

Arguments

<code>g</code>	A vector of length 1 giving the upper limit of the possible groups.
<code>means</code>	A vector of the group of treatments means in decreasing order.
<code>mMSE</code>	A vector of length 1 giving the MSE divided by the number of replications.
<code>dfr</code>	A vector of length 1 giving the degrees of freedom of MSE.
<code>sig.level</code>	A vector of length 1 giving the level of significance of the test.
<code>k</code>	A vector of length 1 giving the lower limit of the possible groups.
<code>group</code>	A vector of the same length as <code>means</code> marking the groups generated.
<code>ngroup</code>	A vector of length 1 giving the number of groups generated.
<code>markg</code>	A vector of the same length as <code>means</code> marking the upper limit of the last group generated before the process goes on recursively.
<code>g1</code>	A vector of length 1 which keeps, during the whole process, the value of the initial <code>g</code> .
<code>sqsum</code>	A vector of length 1 giving the sum of the square between groups.

Details

The function MaxValue builds groups of means, according to the method of Scott & Knott.

Basically it is an algorithm for pre-order path in binary decision tree.

Every node of this tree, represents a different group of means and, when the algorithm reaches this node it takes the decision to either split the group in two, or form a group of means.

If the decision is to divide then this node generates two children and the algorithm follows for the node on the left, if, on the other hand, the decision is to form a group, then it returns to the parent node of that node and follows to the right node.

In this way it follows until the last group is formed, the one containing the highest (or the least) mean. In case that the highest (or the least) mean becomes itself a group of one element, the algorithm continues to the former group. In the end, each node without children represents a group of means.

Value

An *vector* with the groups of means.

Note

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

Author(s)

Enio Jelihovschi (<eniojelihovs@gmail.com>)
 Jose Claudio Faria (<joseclaudio.faria@gmail.com>)
 Sergio Oliveira (<solive@uesc.br>)

References

Ramalho MAP, Ferreira DF, Oliveira AC 2000. *Experimentacao em Genetica e Melhoramento de Plantas*. Editora UFLA.

Scott RJ, Knott M 1974. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, **30**, 507-512.

See Also

[SK](#)

Examples

```
##
## Examples: Completely Randomized Design (CRD)
##

data(CRD1)

av      <- with(CRD1, aov(y ~ x, data=dfm)) # Doing an ANOVA
mm      <- model.tables(av, "means")       # summary tables for model fits
tabs    <- mm$tables[-1]                   # all model means
which   <- names(av$model)[2]
tabs    <- tabs[which]                     # specified group means
nn      <- mm$n[names(tabs)]               # repetitions number of specified groups
```

```

MSE      <- sum(resid(av)^2)/av$df.residual
tab       <- tabs[[which]]                # tab=means
means     <- as.vector(tab)
mnumber   <- length(means)                # number of means
nms       <- names(tab)
r         <- nn[[which]]                  # groups and its number of replicates
ord       <- order(means, decreasing=TRUE)
mMSE      <- MSE/r
dfr       <- av$df.residual               # residual degrees of freedom
means     <- means[ord]                   # decreasing ordered means
g         <- mnumber

## Grouping the means: sig.level=.05
groups    <- MaxValue(g, means, mMSE, dfr, sig.level=.05, 1, rep(0, g), 0,
                      rep(0, g))

## The result: sig.level=.05
groups

## Grouping the means: sig.level=.10
groups    <- MaxValue(g, means, mMSE, dfr, sig.level=.10, 1, rep(0, g), 0,
                      rep(0, g))

## The result: sig.level=.10
groups

```

plot.SK

Plot SK and SK.nest Objects

Description

S3 method to plot SK and SK.nest objects.

Usage

```

## S3 method for class 'SK':
plot(x, pch=19, col=NULL,
      xlab=NULL, ylab=NULL, xlim=NULL, ylim=NULL,
      id.lab=NULL, id.las=1, id.col=TRUE,
      rl=TRUE, rl.lty=3, rl.col='gray',
      mm=TRUE, mm.lty=1,
      title="Means grouped by color(s)", ...)

```

Arguments

x	A SK object.
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.

<code>ylim</code>	The y limits of the plot.
<code>id.lab</code>	Factor level names at x-axis.
<code>id.las</code>	Factor level names written either horizontally or vertically.
<code>id.col</code>	A logical value. If TRUE (the default), the <code>col</code> parameter will be used for the x axis.
<code>rl</code>	Horizontal line connecting the circle to the y-axis.
<code>rl.lty</code>	Line type of rl.
<code>rl.col</code>	Line color of rl.
<code>mm</code>	Vertical line through the circle (mean value) linking the minimum to the maximum of the factor level values corresponding to that mean value.
<code>mm.lty</code>	Line type of mm.
<code>title</code>	A title for the plot.
<code>...</code>	Optional plotting parameters.

Details

The `plot.SK` function is a S3 method to plot SK and `SK.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. The groups are differentiated by colors.

Author(s)

Enio Jelihovschi (<eniojelihovs@gmail.com>)
 Jose Claudio Faria (<joseclaudio.faria@gmail.com>)
 Sergio Oliveira (<solive@uesc.br>)

References

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

See Also

[plot](#)

Examples

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

library(ScottKnott)
data(CRD2)

## From: vectors x and y
sk1 <- with(CRD2, SK(x=x, y=y, model='y ~ x', which='x', id.trim=5))
plot(sk1, id.las=2, rl=FALSE, title='Factor levels', sig.level=.05)

## From: design matrix (dm) and response variable (y)
sk2 <- with(CRD2, SK(x=dm, y=y, model='y ~ x', which='x', sig.level=0.005,
  id.trim=5))
```

```

plot(sk2, col=rainbow(max(sk2$groups)), mm.lty=3, id.las=2, rl=FALSE,
     title='Factor levels', sig.level=0.005, )

## From: data.frame (dfm)
sk3 <- with(CRD2, SK(x=dfm, model='y ~ x', which='x', id.trim=5))
plot(sk3, col=rainbow(max(sk3$groups)), id.las=2, id.col=FALSE, rl=FALSE)

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

sk4 <- with(CRD2, SK(x=av, which='x', id.trim=5))
plot(sk4, col=rainbow(max(sk4$groups)), rl=FALSE, id.las=2, id.col=FALSE,
     title=NULL)

```

RCBD

*Randomized Complete Block Design (RCBD)***Description**

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

Usage

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.

SK

*The ScottKnott Clustering Algorithm 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:
SK(x, y=NULL, model, which, id.trim=3, error, sig.level=.05, ...)
## S3 method for class 'aov':
SK(x, which=NULL, id.trim=3, sig.level=.05, ...)
## S3 method for class 'aovlist':
SK(x, which, id.trim=3, error, sig.level=.05, ...)

```

Arguments

<code>x</code>	A design matrix, <code>data.frame</code> or an <code>aov</code> object.
<code>y</code>	A vector of response variable. It is necessary to inform this parameter only if <code>x</code> 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 <code>x</code> is a <code>data.frame</code> object, the model to be used in the <code>aov</code> must be specified.
<code>id.trim</code>	The number of character to trim the id label.
<code>error</code>	The error to be considered.
<code>sig.level</code>	Level of Significance used in the SK algorithm to create the groups of means. The default value is 0.05.
<code>...</code>	Potential further arguments (require by generic).

Details

The function `SK` returns an object of class `SK` 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 `SK` returns a list of the class `SK` 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, minimum and maximum 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)

Enio Jelihovschi (<eniojelihovs@gmail.com>
 Jose Claudio Faria (<joseclaudio.faria@gmail.com>
 Sergio Oliveira (<solive@uesc.br>)

References

Ramalho MAP, Ferreira DF, Oliveira AC 2000. *Experimentacao em Genetica e Melhoramento de Plantas*. Editora UFLA.

Scott RJ, Knott M 1974. A cluster analysis method for grouping mans in the analysis of variance. *Biometrics*, **30**, 507-512.

Examples

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

## 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)
sk1 <- with(CRD2, SK(x=dm, y=y, model='y ~ x', which='x', sig.level=0.005,
  id.trim=5))
summary(sk1)
plot(sk1, col=rainbow(max(sk1$groups)), mm.lty=3, id.las=2, rl=FALSE,
  title='factor levels, sig.level=0.005')

## From: data.frame (dfm)
sk2 <- with(CRD2, SK(x=dfm, model='y ~ x', which='x', id.trim=5))
summary(sk2)
plot(sk2, col=rainbow(max(sk2$groups)), id.las=2, rl=FALSE)

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

sk3 <- with(CRD2, SK(x=av, which='x', id.trim=5))
summary(sk3)
plot(sk3, col=rainbow(max(sk3$groups)), rl=FALSE, id.las=2, title=NULL)

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

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

data(RCBD)

## Design matrix (dm) and response variable (y)
sk1 <- with(RCBD, SK(x=dm, y=y, model='y ~ blk + tra', which = 'tra'))
summary(sk1)
plot(sk1)

## From: data.frame (dfm), which='tra'
sk2 <- with(RCBD, SK(x=dfm, model='y ~ blk + tra', which='tra'))
summary(sk2)
plot(sk2, mm.lty=3, title='Factor levels')

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

## The parameters can be: design matrix and the response variable,
```

```

## data.frame or aov

data(LSD)

## From: design matrix (dm) and response variable (y)
sk1 <- with(LSD, SK(x=dm, y=y, model='y ~ rows + cols + tra', which='tra'))
summary(sk1)
plot(sk1)

## From: data.frame
sk2 <- with(LSD, SK(x=dfm, model='y ~ rows + cols + tra', which='tra'))
summary(sk2)
plot(sk2, title='Factor levels')

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

sk3 <- SK(av, which='tra')
summary(sk3)
plot(sk3, title='Factor levels')

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

## 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
sk1 <- with(FE, SK(x=dm, y=y, model='y ~ blk + N * P * K',
  which='N'))
summary(sk1)
plot(sk1, title='Main effect: N')

## Nested: N/P=1
nsk1 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + N * P * K',
  which='N:P', fl2=1))
summary(nsk1)
plot(nsk1, title='Effect: N/P=1')

```

SK.nest

The ScottKnott Clustering Algorithm 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:
SK.nest(x, y=NULL, model, which, id.trim=3, error, fl2, fl3=0, sig.level=.05, ...)
## S3 method for class 'aov':
SK.nest(x, which, id.trim=3, fl2, fl3=0, sig.level=.05, ...)
## S3 method for class 'aovlist':
SK.nest(x, which, id.trim=3, error, fl2, fl3=0, sig.level=.05, ...)
```

Arguments

<code>x</code>	A design matrix, <code>data.frame</code> or an <code>aov</code> object.
<code>y</code>	A vector of response variable. It is necessary to inform this parameter only if <code>x</code> 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 <code>x</code> is a <code>data.frame</code> object, the model to be used in the aov must be specified.
<code>id.trim</code>	The number of character to trim the id label.
<code>fl2</code>	A vector of length 1 giving the level of the second factor in nesting order tested.
<code>fl3</code>	A vector of length 1 giving the level of the third factor in nesting order tested.
<code>error</code>	The error to be considered, only in case of split-plots experiments.
<code>sig.level</code>	Level of Significance used in the SK algorithm to create the groups of means. The default value is 0.05.
<code>...</code>	Potential further arguments (require by generic).

Details

The function `SK.nest` returns an object of class `SK.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 `SK.nest` returns a list of the class `SK.nest` 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, minimum and maximum 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>fl2</code>	A vector of length 1 giving the level of the second factor in nesting order tested.
<code>fl3</code>	A vector of length 1 giving the level of the third factor in nesting order tested.

Author(s)

Enio Jelihovski (<eniojelihovs@gmail.com>
 Jose Claudio Faria (<joseclaudio.faria@gmail.com>
 Sergio Oliveira (<solive@uesc.br>)

References

Ramalho MAP, Ferreira DF, Oliveira AC 2000. *Experimentacao em Genetica e Melhoramento de Plantas*. Editora UFLA.

Scott RJ, Knott M 1974. A cluster analysis method for grouping mans in the analysis of variance. *Biometrics*, **30**, 507-512.

Examples

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

data(SSPE)
## From: design matrix (dm) and response variable (y)
## Main factor: p
sk1 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='p', error='blk:p'))
summary(sk1)

# Main factor: sp
sk2 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='sp', error='blk:p:sp', sig.level=0.025))
summary(sk2)

# Main factor: ssp
sk3 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='ssp', error='Within', sig.level=0.1))
summary(sk3)

## Nested: sp/p=1
skn1 <- with(SSPE, SK.nest(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='sp:p', error='blk:p:sp', fl2=1))
summary(skn1)

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

## Nested: ssp/sp/p (at various levels of sp and p)
skn2 <- SK.nest(av, which='ssp:sp:p', error='Within', fl2=1, fl3=1)
summary(skn2)

skn3 <- SK.nest(av, which='ssp:sp:p', error='Within', fl2=2, fl3=1)
summary(skn3)
```

SPE

Split-plot Experiment (SPE)

Description

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

Usage

```
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.

SSPE

Split-split-plot Experiment (SSPE)

Description

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

Usage

```
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

Summary Method for SK and SK.nest Objects

Description

Returns (and prints) a summary list for `SK` and `SK.nest` objects.

Usage

```
## S3 method for class 'SK':
summary(object, ...)
## S3 method for class 'SK.nest':
summary(object, ...)
```

Arguments

object A given object of the class SK or SK.nest.
... Potential further arguments (require by generic).

Author(s)

Enio Jelihovschi (<eniojelihovs@gmail.com>)
Jose Claudio Faria (<joseclaudio.faria@gmail.com>)
Sergio Oliveira (<solive@uesc.br>)

References

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

See Also

[ScottKnott](#)

Examples

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

## 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)
sk1 <- with(CRD2, SK(x=dm, y=y, model='y ~ x', which='x', sig.level=0.005,
  id.trim=5))
summary(sk1)

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

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

data(RCBD)

## Design matrix (dm) and response variable (y)
sk1 <- with(RCBD, SK(x=dm, y=y, model='y ~ blk + tra', which = 'tra'))
summary(sk1)

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

## The parameters can be: design matrix and the response variable,
```

```
## data.frame or aov

data(LSD)

## From: design matrix (dm) and response variable (y)
sk1 <- with(LSD, SK(x=dm, y=y, model='y ~ rows + cols + tra', which='tra'))
summary(sk1)

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

## 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
sk1 <- with(FE, SK(x=dm, y=y, model='y ~ blk + N * P * K',
  which='N'))
summary(sk1)

## Nested: N/P=1
nsk1 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + N * P * K',
  which='N:P', fl2=1))
summary(nsk1)

## Nested: N/P=2/K=2
nsk2 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + N * P * K',
  which='N:P:K', fl2=2, fl3=2))
summary(nsk2)

## Nested: P/N=1/K=1
nsk3 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + P * N * K',
  which='P:N:K', fl2=1, fl3=1))
summary(nsk3)

## Nested: K/N=1/P=1
nsk4 <- with(FE, SK.nest(x=dm, y=y, model='y ~ blk + K * N * P',
  which='K:N:P', fl2=1, fl3=1))
summary(nsk4)

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

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
sk1 <- with(SPE, SK(x=dm, y=y, model='y ~ blk + sp*p + Error(blk/p)',
  which='p', error='blk:p'))
```

```

summary(skl)

## Nested: sp/p=1
skn1 <- with(SPE, SK.nest(x=dm, y=y, model='y ~ blk + sp*p + Error(blk/p)',
  which='sp:p', error='Within', fl2=1 ))
summary(skn1)

data(SSPE)

## From: design matrix (dm) and response variable (y)
## Main factor: p
sk1 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='p', error='blk:p'))
summary(skl)

# Main factor: sp
sk2 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='sp', error='blk:p:sp', sig.level=0.025))
summary(sk2)

# Main factor: ssp
sk3 <- with(SSPE, SK(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='ssp', error='Within', sig.level=0.1))
summary(sk3)

## Nested: sp/p=1
skn1 <- with(SSPE, SK.nest(dm, y, model='y ~ blk + ssp*sp*p + Error(blk/p/sp)',
  which='sp:p', error='blk:p:sp', fl2=1))
summary(skn1)

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

## Nested: ssp/sp/p (at various levels of sp and p)
skn2 <- SK.nest(av, which='ssp:sp:p', error='Within', fl2=1, fl3=1)
summary(skn2)

skn3 <- SK.nest(av, which='ssp:sp:p', error='Within', fl2=2, fl3=1)
summary(skn3)

```

Index

*Topic **datasets**

CRD1, [6](#)
CRD2, [7](#)
FE, [7](#)
LSD, [7](#)
RCBD, [12](#)
SPE, [17](#)
SSPE, [18](#)

*Topic **design**

MaxValue, [8](#)
plot.SK, [10](#)
ScottKnott-package, [1](#)
SK, [12](#)
SK.nest, [15](#)
summary, [18](#)

*Topic **htest**

MaxValue, [8](#)
plot.SK, [10](#)
ScottKnott-package, [1](#)
SK, [12](#)
SK.nest, [15](#)
summary, [18](#)

*Topic **package**

MaxValue, [8](#)
plot.SK, [10](#)
ScottKnott-package, [1](#)
SK, [12](#)
SK.nest, [15](#)
summary, [18](#)

*Topic **tree**

MaxValue, [8](#)
plot.SK, [10](#)
ScottKnott-package, [1](#)
SK, [12](#)
SK.nest, [15](#)
summary, [18](#)

*Topic **univar**

MaxValue, [8](#)
plot.SK, [10](#)
ScottKnott-package, [1](#)
SK, [12](#)
SK.nest, [15](#)
summary, [18](#)

CRD1, [6](#)
CRD2, [7](#)

FE, [7](#)

LSD, [7](#)

MaxValue, [8](#)

plot, [11](#)
plot.SK, [10](#)

RCBD, [12](#)

ScottKnott, [18](#)
ScottKnott (*ScottKnott-package*), [1](#)
ScottKnott-package, [1](#)
SK, [9](#), [12](#)
SK.nest, [15](#)
SPE, [17](#)
SSPE, [18](#)
summary, [18](#)