Package ‘ScottKnott’

March 15, 2025

Type Package

Title The ScottKnott Clustering Algorithm

Version 1.3-3

Date 2025-03-14

Maintainer Ivan Bezerra Allaman <ivanalaman@gmail.com>

Depends R (>= 4.0.0), base

Imports doBy, xtable

Suggests lme4

Description Perform the balanced (Scott and Knott, 1974) and unbalanced <doi:10.1590/1984-70332017v17n1a1> Scott & Knott algorithm.

License GPL (>= 2)

URL https://github.com/ivanalaman/ScottKnott, https://lec.pro.br/software/pac-r/scottknott

Encoding UTF-8

LazyLoad yes

LazyData true

NeedsCompilation no

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

Repository CRAN

Date/Publication 2025-03-15 15:30:02 UTC

Contents

`ScottKnott-package` *The ScottKnott Clustering Algorithm*

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), Dilson 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 increse 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 abtained'.

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`, `aovlist`, `lm` and `lmer` resulting object of previous analysis of variance. The results are given in the usual way as well as in graphical way using thermometers with diferent group colors.

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

As of version 1.2-8, the ScottKnott package is able to analyze unbalanced data based on the article 'Adjusting the Scott-Knott cluster analyzes for unbalanced designs' by Conrado et al.

Author(s)

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

References

Bony S., Pichon N., Ravel C., Durix A., Balfourier F., Guillaumin J.J. 2001. The Relationship be-tween Mycotoxin Synthesis and Isolate Morphology in Fungal Endophytes of Lolium perenne. *New Phytologist*, **1521**, 125-137.

Borges L.C., FERREIRA D.F. 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 L.C.A. 1985. Clustering Means in ANOVA by Simultaneous Testing. *Biometrics*, **411**, 39-48.

Da Silva E.C, Ferreira D.F, Bearzoti E. 1999. Evaluation of power and type I error rates of Scott-Knotts test by the method of Monte Carlo. *Cienc. agrotec.*, Lavras, **23**, 687-696.

Dilson A.B, David S.D., Kazimierz J., William W.K. 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 C.E., Bilbro J.D. 1978. Illustration of a Cluster Analysis Method for Mean Separation. *Agron J*, **70**, 462-465.

Wilkinson, G.N, Rogers, C.E. 1973. Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 22, No. 3, pp. 392-399.

Jyotsna S., Zettler L.W., van Sambeek J.W., Ellersieck M.R., Starbuck C.J. 2003. Symbiotic Seed Germination and Mycorrhizae of Federally Threatened Platanthera PraeclaraOrchidaceae. *American Midland Naturalist*, **1491**, 104-120.

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

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

Conrado, T. V., Ferreira, D. F., Scapim, C. A., and Maluf, W. R. "Adjusting the Scott-Knott cluster analyses for unbalanced designs." *Crop Breeding and Applied Biotechnology* **17**.1 (2017): 1-9.

boxplot.SK *Boxplot SK Objects*

Description

S3 method to plot SK objects.

Usage

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

Arguments

x	A SK object.
mean.type	The type of mean must be ploted. O default is "line".
xlab	A label for the 'x' axis.
mean.col	A vector of colors for the means representation.
mean.pch	A vector of plotting symbols or characters. Only if type are "point".
mean.lwd	Line width of mean.
mean.lty	Line type of mean. Only if type are "line".
args.legend	List of additional arguments to be passed to legend; The default is NULL.
...	Optional plotting parameters.

Details

The boxplot.SK function is a S3 method to plot 'SK' objetcs. The difference to generic function is the Scott & Knott inference under frame and the plot of means within box.

Author(s)

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

References

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

See Also

boxplot

Examples

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

library(ScottKnott)
data(CRD1)

## From: formula
# Simple!
sk1 <- SK(y ~ x,
          data=CRD1$dfm,
          which='x')
boxplot(sk1)

# A little more elaborate!
boxplot(sk1,
        mean.lwd=1.3,
        mean.col='red')

# A little more!
boxplot(sk1,
        mean.lwd=1.3,
        mean.lty=2,
        mean.col='red',
        args.legend=list(x='bottomleft'))

# With point type!
boxplot(sk1,
        mean.type='point')

boxplot(sk1,
        mean.type='point',
        mean.pch=19,
        cex=1.5,
        mean.col='red')

# With other point
boxplot(sk1,
        mean.type='point',
        mean.pch='+',
        cex=2,
        mean.col='blue',
        args.legend=list(x='bottomleft'))
```

CRD1 *Completely Randomized Design (CRD)*

Description

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

Usage

```
data(CRD1)
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 **ScottKnott** package related to Completely Randomized Design (‘`CRD`’).

Usage

```
data(CRD2)
CRD2
```

Details

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

`FE` *Factorial Experiment (FE)*

Description

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

Usage

```
data(FE)
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

```
data(LSD)
LSD
```

Details

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

`plot.SK` *Plot SK objects*

Description

S3 method to plot `SK` objects.

Usage

```
## S3 method for class 'SK'
plot(x,
     result         = TRUE,
     replicates     = TRUE,
     pch            = 19,
     col            = NULL,
     xlab           = NULL,
     ylab           = NULL,
     xlim           = NULL,
     ylim           = NULL,
     axisx          = TRUE,
     axisy          = TRUE,
     id.lab         = NULL,
     id.las         = 1,
     yl             = TRUE,
     yl.lty         = 3,
     yl.col         = 'gray',
     dispersion     = c('none','mm','sd','ci','cip'),
     d.lty          = 1,
     d.col          = 'black',
     title          = '', ...)
```

Arguments

`x` A `SK` 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.

`axisx` Axis x? If ‘`TRUE`’ you must accept the default, otherwise, you must customize.

`axisy` Axis y? If ‘`TRUE`’ you must accept the default, otherwise, you must customize.

id.lab Factor level names at ‘x’ axis.

id.las Factor level names written either horizontally or vertically.

yl Horizontal (reference) line connecting the circle to the ‘y’ axis.

yl.lty Line type of ‘yl’.

yl.col Line color of ‘yl’.

dispersion Vertical line through the circle (mean value) linking the minimum to the maximum of the factor level values corresponding to that mean value. Other options are: sd (standard deviation), ci (confidence interval), cip (pooled confidence interval) and none.

d.lty Line type of dispersion.

d.col A vector of colors for the line type of dispersion.

title A title for the plot.

... Optional plotting parameters.

Details

The plot.SK function is a S3 method to plot ‘Scott and Knott’ objetcs. It generates a serie of points (the means) and a vertical line showing the dispersion of the values corresponding to each group mean. The ci options is calculed utilizing each treatment variance as estimating of population variance. The cip options is calculed utilizing the means square error (MSE) as estimating of population variance.

Author(s)

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

References

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

See Also

plot

Examples

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

library(ScottKnott)
data(CRD2)

## From: formula
sk1 <- with(CRD2,
            SK(y ~ x,
               data=dfm,
               which='x'))

old.par <- par(mar=c(6, 3, 6, 2))
plot(sk1,
     id.las=2)

plot(sk1,
     yl=FALSE,
     disp='sd',
     id.las=2)

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

sk2 <- SK(x=av,
          which='x')
plot(sk2,
     disp='sd',
     yl=FALSE,
     id.las=2)

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

sk3 <- SK(x=av_lm,
          which='x')

par(mfrow=c(2, 1))
plot(sk3,
     disp='ci',
     id.las=2,
     yl=FALSE)

plot(sk3,
     disp='cip',
     id.las=2,
     yl=FALSE)

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

print.SK *Print Method for* SK *objects.*

Description

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

Usage

```
## S3 method for class 'SK'
print(x, digits = 2L,...)
```

Arguments

x	A given object of the class SK.
digits	Minimal number of _significant_ digits. The default is 2.
...	Further arguments (require by generic).

Author(s)

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

See Also

SK

Examples

```
data(RCBD)

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

RCBD *Randomized Complete Block Design (RCBD)*

Description

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

Usage

```
data(RCBD)
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 SK Test for Single Experiments*

Description

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

Usage

```
SK(x,...)

## S3 method for class 'formula'
SK(formula,
       data            = NULL,
       which           = NULL,
       fl1             = NULL,
       fl2             = NULL,
       error           = NULL,
       sig.level       = .05,
       round           = 2,
       ...)

## S3 method for class 'lm'
SK(x,
       which           = NULL,
       fl1             = NULL,
       fl2             = NULL,
       error           = NULL,
       sig.level       = .05,
       round           = 2,
       ...)

## S3 method for class 'aovlist'
SK(x,
       which           = NULL,
       fl1             = NULL,
       fl2             = NULL,
       error          = NULL,
       sig.level      = .05,
       round          = 2,
       ...)

## S3 method for class 'lmerMod'
SK(x,
       which          = NULL,
       fl1            = NULL,
       fl2            = NULL,
       error          = NULL,
       sig.level      = .05,
       round          = 2,
       ...)
```

Arguments

`x, formula` A `formula`, `lm`, `aov`, `aovlist` and `lmerMod` class object. Objects of the `formula` class follow “response variable ~ predicted variable.

`data` A object of the `data.frame` class. Use only objects of `formula` class.

`which` The name of the treatment to be used in the comparison. The name must be inside quoting marks.

`fl1` A vector of length 1 giving the level of the first factor in nesting order tested.

`fl2` A vector of length 1 giving the level of the second factor in nesting order tested.

`error` The error to be considered. If from experiment at split plot or split-split plot pay attention! See details!

`sig.level` Level of Significance used in the SK algorithm to create the groups of means. The default value is 0.05.

`round` Integer indicating the number of decimal places.

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

Details

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

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

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

Value

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

`out` A `list` storing the result of Scott & Knott test.

`info` A `list` storing the descriptive statistics.

`stat` A `matrix` with the statistics of each clustering process.

`clus` A `list` with the groups formed in each clustering process.

Author(s)

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

References

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

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

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

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

Examples

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

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

data(RCBD)

## From: formula
sk1 <- with(RCBD,
            SK(y ~ blk + tra,
               data=dfm,
               which='tra'))
summary(sk1)

## From: merMod
## This class is specific of the lme4 package.
## Not run:
  if(require(lme4)){
    lmer1 <- with(RCBD,
                  lmer(y ~ (1|blk) + tra,
                       data=dfm))
    sk2 <-  SK(lmer1,
               which='tra')
    summary(sk2)
  }

## End(Not run)
##
## Example: Latin Squares Design (LSD)
## More details: demo(package='SK')
##

data(LSD)

## From: formula
sk3 <- with(LSD,
            SK(y ~ rows + cols + tra,
               data=dfm,
               which='tra'))
summary(sk3)

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

sk4 <- SK(av1,
              which='tra')
summary(sk4)

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

sk5 <- SK(lm1,
          which='tra')
summary(sk5)

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

data(FE)
## From: formula
## Main factor: N
sk6 <- with(FE,
            SK(y ~ blk + N*P*K,
               data=dfm,
               which='N'))
summary(sk6)

## Nested: p1/N
```

```
# From: formula
n_sk1 <- with(FE,
              SK(y ~ blk + N*P*K,
                 data=dfm,
                 which='P:N',
                 fl1=1))
summary(n_sk1)

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

n_sk2 <- with(FE,
              SK(lm2,
                 which='P:N',
                 fl1=2))
summary(n_sk2)

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

n_sk3 <- with(FE,
              SK(av2,
                 which='N:P',
                 fl1=1))
summary(n_sk3)

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

    n_sk4 <- with(FE,
                  SK(lmer2,
                     which='N:P',
                     fl1=1))
    summary(n_sk4)
  }

## End(Not run)

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

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

# crotgrantiana/year
sp_sk1 <- SK(lm3,
             which='tra:year',
             fl1=1)
summary(sp_sk1)

# year1/tra
# It is necessary to set year error with trat error in the order of the "which" argument.
# It is necessary to inform how to combinate the errors
sp_sk2 <-  SK(lm3,
              which='year:tra',
              error='Residuals/blk:tra',
              fl1=1)
summary(sp_sk2)

# From merMod
# Onty tra
## Not run:
  if(require(lme4)){
    lmer3 <- with(SPET,
                  lmer(y ~ blk + (1|blk:tra) + tra*year,
                       dfm))

    # comparison only tra
    sp_sk3 <- SK(lmer3,
                 which = 'tra',
                 error = 'blk:tra')
    summary(sp_sk3)

    # year1/tra
    sp_sk4 <- SK(lmer3,
                 which='year:tra',
                 error='Residual/blk:tra',
                 fl1=1)
    summary(sp_sk4)
  }

## End(Not run)

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

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

```
ssp_sk1 <- with(SSPE,
                SK(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                   data=dfm,
                   which='P',
                   error='blk:P'))
summary(ssp_sk1)

## Main factor: SP
## It is necessary to inform the appropriate error for the test
ssp_sk2 <- with(SSPE,
                SK(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                   data=dfm,
                   which='SP',
                   error='blk:P:SP'))
summary(ssp_sk2)

## Main factor: SSP
ssp_sk3 <- with(SSPE,
                SK(y ~ blk + P*SP*SSP + Error(blk/P/SP),
                   data=dfm,
                   which='SSP'))
summary(ssp_sk3)

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

ssp_sk4 <- SK(av3,
              which='SSP')
summary(ssp_sk4)

## Nested: p1/SP
## It is necessary to inform the appropriate error for the test
ssp_sk5 <- SK(av3,
              which='P:SP',
              error='blk:P:SP',
              fl1=1)
summary(ssp_sk5)

## Nested: p1/SSP
ssp_sk6 <- SK(av3,
              which='P:SSP',
              fl1=1)
summary(ssp_sk6)

## Nested: p1/sp1/SSP
## Testing SSP inside of level one of P and level one of SP
ssp_sk7 <- SK(av3,
              which='P:SP:SSP',
              fl1=1,
              fl2=1)
```

```
summary(ssp_sk7)

## Nested: p2/sp1/SSP
ssp_sk8 <- SK(av3,
              which='P:SP:SSP',
              fl1=2,
              fl2=1)
summary(ssp_sk8)

## Nested: sp1/P
## It is necessary to inform the appropriate error for the test
ssp_sk9 <- SK(av3,
              which='SP:P',
              error='blk:P:SP/blk:P',
              fl1=1)

summary(ssp_sk9)

## Nested: ssp1/SP
ssp_sk10 <- SK(av3,
               which='SSP:SP',
               error='Within/blk:P:SP',
               fl1=1)
summary(ssp_sk10)

## Nested: ssp1/sp1/P
## It is necessary to inform the appropriate error for the test
ssp_sk11 <- SK(av3,
               which='SSP:SP:P',
               error='Within/blk:P:SP/blk:P',
               fl1=1,
               fl2=1)
summary(ssp_sk11)

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

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

usk1 <-  SK(y ~ x,
            data=uCRD2,
            which='x')
summary(usk1)

## From: lm
ulm1 <- lm(y ~ x,
           data=uCRD2)

usk2 <- SK(ulm1,
           which='x')
```

```
summary(usk2)

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

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

ulm2 <- lm(y ~ blk + N*P*K,
           uFE)

## Nested: p1/N
usk3 <- SK(ulm2,
           data=uFE,
           which='P:N',
           fl1=1)
summary(usk3)

## Nested: p2/n2/K
usk4 <- SK(ulm2,
           data=uFE,
           which='P:N:K',
           fl1=2,
           fl2=2)
summary(usk4)
```

`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 ‘`tr`’ is a factor of lenght 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).

Source

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

Examples

```
library(ScottKnott)

data(sorghum)

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

sk <- SK(av,
         which='x',
         sig.level=0.05)

summary(sk)

plot(sk)
```

SPE *Split-plot Experiment (SPE)*

Description

A `list` to ilustrate the resources of **ScottKnott** package related to Split-plot Experiment (‘`SPE`’).

Usage

```
data(SPE)
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 Estatistica Experimental. 13 ed. Editora NOBEL, Piracicaba, Brazil, page 157.

SSPE *Split-split-plot Experiment (SSPE)*

Description

A `list` to ilustrate the resources of **ScottKnott** package related to Split-split-plot Experiment (‘SSPE’).

Usage

```
data(SSPE)
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 Objects*

Description

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

Usage

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

Arguments

object	A given object of the class SK.
...	Potential further arguments (required by generic).

Author(s)

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

References

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

See Also

SK

Examples

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

data(CRD2)
## From: formula
sk1 <- with(CRD2,
            SK(y ~ x,
               data=dfm,
               which='x',
               id.trim=5))
summary(sk1)
```

xtable.SK *xtable method for* SK *objects.*

Description

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

Usage

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

Arguments

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

Author(s)

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

See Also

xtable

Examples

```
data(RCBD)
## Not run:
  if(require(xtable)){
    lm1 <- with(RCBD,
                  lm(y ~ blk + tra,
                     data=dfm))

    sk1 <-  SK(lm1,
               which='tra')
    tb <- xtable(sk1)
    print(tb)
  }

## End(Not run)
```

Index