Package ‘ScottKnott’

May 24, 2026

Version 1.4-0

Date 2026-05-24

Title The ScottKnott Clustering Algorithm

Depends R (>= 4.0.0)

Imports emmeans, xtable

Suggests lme4, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Description Performs the Scott & Knott (1974) clustering algorithm as a multiple comparison method in the Analysis of Variance context, for both balanced and unbalanced <doi:10.1590/1984-70332017v17n1a1> designs. Accepts input from 'formula', 'aov', 'lm', 'aovlist', and 'lmerMod' objects.

License GPL (>= 2)

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

BugReports `https://github.com/ivanalaman/ScottKnott/issues`

Encoding UTF-8

LazyData true

NeedsCompilation no

Author J. C. Faria [aut],
E. G. Jelihovschi [aut],
I. B. Allaman [aut, cre]

Maintainer I. B. Allaman <`ivanalaman@gmail.com`>

Repository CRAN

Date/Publication 2026-05-24 12:30:02 UTC

Contents

ScottKnott-package 2
boxplot.SK 3
CRD1 5
CRD2 6
FE 6
LSD 7
plot.SK 7
print.SK 10
RCBD 11
SK 11
sorghum 20
SPE 21
SPET 21
SSPE 22
summary 22
xtable 23
xtable.SK 23

Index **25**

`ScottKnott-package` *The ScottKnott Clustering Algorithm*

Description

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

Proposed by Scott and Knott (1974), the method overcomes the overlapping problem common to other procedures such as the t-test, Tukey, Duncan, and Newman-Keuls tests. Overlapping occurs when one or more treatments are simultaneously assigned to more than one group; as the number of treatments grows to twenty or more, this ambiguity can make it virtually impossible for the experimenter to distinguish the true group structure. The Scott & Knott method does not have this problem, which is widely regarded as one of its main advantages.

The method uses a cluster analysis algorithm that, starting from the complete set of observed treatment means, recursively partitions them so that any two resulting groups are disjoint.

In 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 obtained”.

Monte Carlo studies suggest that the Scott & Knott method has high power and type I error rates that closely follow the nominal levels. The **ScottKnott** package applies this algorithm to objects of class `formula`, `aov`, `aovlist`, `lm`, and `lmerMod` from a prior analysis of variance, and presents results both numerically and graphically.

As of version 1.2-8, the package handles unbalanced designs using adjusted means, as described in Conrado et al. (2017).

Author(s)

Faria, J. C. (<`joseclaudio.faria@gmail.com`>)
Jelihovschi, E. G. (<`eniojelihovs@gmail.com`>)
Allaman, I. B. (<`ivanalaman@gmail.com`>)

References

Bony S., Pichon N., Ravel C., Durix A., Balfourier F., Guillaumin J.J. 2001. The Relationship between 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 non-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 means 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 representation to be plotted. The 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. Used only when `mean.type = 'point'`.
`mean.lwd`	Line width of mean.
`mean.lty`	Line type of the mean. Used only when `mean.type = '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 an S3 method for plotting `SK` objects. It extends the generic `boxplot` function by overlaying Scott & Knott group labels above the plot frame and drawing the treatment mean within each box.

Author(s)

Faria, J. C. (<`joseclaudio.faria@gmail.com`>)
Jelihovschi, E. G. (<`eniojelihovs@gmail.com`>)
Allaman, I. B. (<`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')

## More customisation
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 a different point symbol
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

Simulated data for a Completely Randomized Design (‘`CRD`’) with 4 treatment levels and 6 replicates per treatment.

`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

Simulated data for a Completely Randomized Design (‘`CRD`’) with 45 treatment levels and 4 replicates per treatment.

`FE` *Factorial Experiment (FE)*

Description

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

Usage

```
data(FE)
FE
```

Details

Simulated data for 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

Simulated data for a Latin Squares Design (‘`LSD`’) with 5 treatment 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('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`	If `TRUE`, the x axis is drawn using defaults; set to `FALSE` to suppress it.
`axisy`	If `TRUE`, the y axis is drawn using defaults; set to `FALSE` to suppress it.
`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`	Type of dispersion bar drawn through each mean point. Options: ‘`mm`’ (min-max range), ‘`sd`’ (standard deviation), ‘`ci`’ (individual confidence interval), ‘`cip`’ (pooled confidence interval). Default is ‘`mm`’.
`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 an S3 method for plotting `SK` objects. It generates a series of points representing the treatment means, optionally with vertical dispersion bars. The ‘`ci`’ option is calculated using each treatment’s own variance as an estimate of the population variance. The ‘`cip`’ option is calculated using the mean square error (MSE) as an estimate of the population variance.

Author(s)

Faria, J. C. (<`joseclaudio.faria@gmail.com`>)
Jelihovschi, E. G. (<`eniojelihovs@gmail.com`>)
Allaman, I. B. (<`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='ScottKnott')
##

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,
     dispersion='sd',
     id.las=2)

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

sk2 <- SK(x=av,
          which='x')

col=c(rep(2, 6),
      rep(3, 36),
      rep(4, 1),
      rep(5, 2))

plot(sk2,
     dispersion='sd',
     yl=FALSE,
     id.las=2,
     col=col,
     d.col=col)

## From: lm
av_lm <- with(CRD2,
           lm(y ~ x,
```

```
              data=dfm))

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

par(mfrow=c(2, 1))
plot(sk2,
     dispersion='ci',
     yl=FALSE,
     id.las=2,
     col=col,
     d.col=col)

plot(sk2,
     dispersion='cip',
     yl=FALSE,
     id.las=2,
     col=col,
     d.col=col)

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 (required by generic).

Author(s)

Faria, J. C. (<`joseclaudio.faria@gmail.com`>)
Jelihovschi, E. G. (<`eniojelihovs@gmail.com`>)
Allaman, I. B. (<`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

Simulated data for a Randomized Complete Block Design (‘`RCBD`’) with 5 treatment levels and 4 blocks (one replicate per 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`	An object of class `formula`, `lm`, `aov`, `aovlist`, or `lmerMod`. Objects of the `formula` class follow “response variable ~ predictor variables”.
`data`	An object of class `data.frame`. Used only when the input is of class `formula`.
`which`	The name of the treatment factor to be compared. Must be quoted.
`fl1`	An integer of length 1 selecting the level of the first nesting factor.
`fl2`	An integer of length 1 selecting the level of the second nesting factor.
`error`	The error term to be used. For split-plot and split-split-plot experiments, see Details.
`sig.level`	Significance level used in the SK algorithm to form the groups of means. Default is `0.05`.
`round`	Integer indicating the number of decimal places. Default is `2`.
`...`	Potential further arguments (required by generic).

Details

The function `SK` returns an object of class `SK` containing the groups of means and the variables needed by `summary` and `plot`.

The `error` argument may be used whenever a specific error term other than the residual is required. In split-plot and split-split-plot experiments, error terms may be combined using `/` in the order implied by the `which` argument. For example, for an `aovlist` object from a block split-plot experiment, a valid combination would be `error = 'Within/blk:plot'` with `which = 'subplot:plot'`.

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)

Faria, J. C. (<`joseclaudio.faria@gmail.com`>)
Jelihovschi, E. G. (<`eniojelihovs@gmail.com`>)
Allaman, I. B. (<`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='ScottKnott')
##

## Input classes accepted: formula, aov, lm, aovlist, and lmerMod.

data(RCBD)

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

## From: lmerMod
## This class requires 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='ScottKnott')
##

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='ScottKnott')
##

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: lmerMod
## 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 in Time (SPET)
## More details: demo(package='ScottKnott')
##
data(SPET)

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

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

## Nested: year1/tra
## It is necessary to set the year error with the treatment error,
## in the order of the which argument.
## It is necessary to specify how to combine the error terms.
sp_sk2 <-  SK(lm3,
              which='year:tra',
              error='Residuals/blk:tra',
              fl1=1)
summary(sp_sk2)

## From: lmerMod
## Main factor: tra
## Not run:
  if(require(lme4)){
    lmer3 <- with(SPET,
```

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

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

    ## Nested: 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='ScottKnott')
##

data(SSPE)
## From: formula
## Main factor: P
## It is necessary to specify the appropriate error term 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 specify the appropriate error term 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 specify the appropriate error term 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 within 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 specify the appropriate error term 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 specify the appropriate error term 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
## Means are adjusted using the Least-Squares Means (LSMeans) 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` *Sorghum Yield: Balanced Squared Lattice Design*

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 length 80 with 16 levels describing the treatments. The second ‘`dm`’ is data.frame describing the design matrix. Its columns are ‘`x`’, ‘`bl`’ (blocks) and ‘`r`’ repetitions. The third ‘`y`’ is a numeric vector the yields. The fourth ‘`dfm`’ is a data frame with four columns. The first three columns are the design matrix and the fourth is ‘`y`’.

Details

The experiment was conducted at EMBRAPA Milho e Sorgo (The Brazilian Agricultural Research Corporation, Corn and Sorghum section).

Source

Ramalho, M.A.P. and Ferreira, D.F. and Oliveira, A.C. (2000). 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` illustrating the resources of **ScottKnott** package related to Split-Plot Experiment (‘`SPE`’).

Usage

```
data(SPE)
SPE
```

Details

Simulated data for a Split-Plot Experiment (‘`SPE`’) with 3 whole plots, 4 sub-plot treatments, and 6 replicates per sub-plot.

`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` illustrating the resources of **ScottKnott** package related to Split-Split-Plot Experiment (‘`SSPE`’).

Usage

```
data(SSPE)
SSPE
```

Details

Simulated data for a Split-Split-Plot Experiment (‘`SSPE`’) with 3 whole plots, 3 sub-plot treatments, 5 sub-sub-plot treatments, and 4 replicates per sub-sub-plot.

`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)

Faria, J. C. (<`joseclaudio.faria@gmail.com`>)
Jelihovschi, E. G. (<`eniojelihovs@gmail.com`>)
Allaman, I. B. (<`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='ScottKnott')
##

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

`xtable` *Create a Table for Export*

Description

This function is re-exported from the **xtable** package so that `xtable()` is available after `library(ScottKnott)` without requiring a separate `library(xtable)` call.

For `SK` objects the S3 method `xtable.SK` is dispatched automatically. For full documentation of the generic see `help("xtable", package = "xtable")`.

See Also

`xtable.SK`, `xtable`

`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 provides an additional method for the `xtable` function from the **xtable** package.

Usage

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

Arguments

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

Author(s)

Faria, J. C. (<`joseclaudio.faria@gmail.com`>)
Jelihovschi, E. G. (<`eniojelihovs@gmail.com`>)
Allaman, I. B. (<`ivanalaman@gmail.com`>)

See Also

`xtable`

Examples

```
data(RCBD)

lm1 <- with(RCBD,
            lm(y ~ blk + tra,
               data=dfm))

sk1 <- SK(lm1,
          which='tra')

tb <- xtable(sk1)
## Not run:
  print(tb)

## End(Not run)
```

Index

∗ **datasets**
  CRD1, 5
  CRD2, 6
  FE, 6
  LSD, 7
  RCBD, 11
  sorghum, 20
  SPE, 21
  SPET, 21
  SSPE, 22
∗ **design**
  ScottKnott-package, 2
∗ **htest**
  ScottKnott-package, 2
∗ **package**
  boxplot.SK, 3
  plot.SK, 7
  print.SK, 10
  ScottKnott-package, 2
  SK, 11
  summary, 22
  xtable.SK, 23
∗ **tree**
  ScottKnott-package, 2
∗ **univar**
  ScottKnott-package, 2
∗ **utilities**
  xtable, 23

boxplot, *4*
boxplot.SK, 3

CRD1, 5
CRD2, 6

FE, 6

LSD, 7

plot, *9*
plot.SK, 7
print.SK, 10

RCBD, 11

ScottKnott *(*ScottKnott-package*)*, 2
ScottKnott-package, 2
SK, *11*, 11, *23*
sorghum, 20
SPE, 21
SPET, 21
SSPE, 22
summary, 22

xtable, *23*, 23, *24*
xtable.SK, *23*, 23