Package ‘daewr’

May 10, 2020

Type Package

Title Design and Analysis of Experiments with R

Version 1.2-5

Date 2020-5-7

Maintainer John Lawson <lawsonjsl7net@gmail.com>

Description Contains Data frames and functions used in the book ``Design and Analysis of Experiments with R''.

License GPL-2

Imports lattice, FrF2, graphics, grDevices, stats, stringi

LazyLoad yes

LazyData yes

Suggests R.rsp

VignetteBuilder R.rsp

Author John Lawson [aut, cre],
Gerhard Krennrich [aut]

Repository CRAN

Repository/R-Forge/Project daewr

Repository/R-Forge/Revision 205

Repository/R-Forge/DateTimeStamp 2020-05-09 15:51:42

Date/Publication 2020-05-10 21:40:06 UTC

NeedsCompilation no

R topics documented:

daewr-package 4
Altscreen 4
antifungal 5
Apo 6
apple 7
arso 7

augm 8
Bdish 9
Bff 10
bha 11
BIBsize 11
bioequiv 13
bioeqv 13
blood 14
BoxM 15
BPmonitor 15
bread 16
cake 17
cement 17
chem 18
chipman 19
COdata 19
colormap 20
cont 21
cpipe 22
culture 23
dairy 24
DefScreen 24
drug 25
EEw1s1 26
EEw1s2 27
EEw1s3 28
EEw2s1 29
EEw2s2 30
EEw2s3 31
EEw3 32
eptaxr 33
eptaxs2 34
eptaxyb 35
Fcrit 36
fhstep 36
fhstepDS 37
firstm 38
FitDefSc 39
fnextrm 39
fntrmDS 40
Fpower 41
Fpower1 42
Fpower2 43
fullnormal 44
gagerr 46
gapstat 47
Gaptest 48
gear 50

halfnorm 51
hardwood 52
HierAFS 53
ihstep 54
inject 55
LenthPlot 55
LGB 57
LGBc 58
mod 60
ModelRobust 61
MPV 62
Naph 62
OptPB 63
pastry 64
PBDes 64
pest 65
pesticide 66
plasma 67
polvdat 67
polymer 68
prodstd 69
qsar 70
rcb 70
residue 71
rubber 72
sausage 72
Smotor 73
soup 74
soupmx 75
splitPdes 75
SPMPV 76
strung 77
strungtile 77
sugarbeet 78
taste 79
teach 80
Tet 80
tile 81
Treb 82
Tukey1df 82
vci 84
virus 85
volt 86
web 87

Index **88**

`daewr-package` *Data frames and functions for Design and Analysis of Experiments with R*

Description

This package contains the data sets and functions from the book Design and Analysis of Experiments with R published by CRC in 2013.

Details

Package:	daewr
Type:	Package
Version:	1.2-5
Date:	2012-05-10
License:	GPL-2
LazyLoad:	yes

Author(s)

John Lawson

Maintainer: John Lawson <lawsonjsl7net@gmail.com>

References

J. Lawson, Design and Analysis of Experiments with R, CRC 2013.

Examples

```
Fcrit(.05,2,15)
Fpower1(alpha=.05,nlev=3,nreps=4,Delta=3,sigma=sqrt(2.1))
BIBsize(6,3)
```

`Altscreen` *Alternate 16 run screening designs*

Description

Recalls Jones and Montgomery's 16 run screening designs from data frames

Usage

```
Altscreen(nfac, randomize=FALSE)
```

Arguments

`nfac`	input- an integer
`randomize`	input - logical

Value

a data frame containing the alternate screening design

Author(s)

John Lawson

References

Jones, B. and Montgomery, D. C. (2010) "Alternatives to resolution IV screening designs in 16 runs", Int. J. Experimental Design and Process Optimization, Vol 1, No. 4, 2010.

Examples

```
Altscreen(6)
Altscreen(6, randomize=TRUE)
```

`antifungal` *Two-period crossover study of antifungal agent*

Description

Data from the Two-period crossover study of an antifungal agent in chapter 9 of Design and Analysis of Experiments with R

Usage

```
data(antifungal)
```

Format

A data frame with 34 observations on the following 5 variables.

`Group` a factor with levels `1` `2`

`Subject` a factor with levels `1` `2` `3` `4` `5` `6` `7` `8` `9` `10` `11` `12` `14` `15` `16` `17` `18`

`Period` a factor with levels `1` `2`

`Treat` a factor with levels `A` `B`

`pl` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(antifungal)
```

`Apo` *apolipoprotein survey varaince component study*

Description

Data from the apolipoprotein survey variance component study of Chapter 5 in Design and Analysis of Experiments with R

Usage

```
data(Apo)
```

Format

A data frame with 30 observations on the following 2 variables.

`lab` a factor with levels `A` `B` `C` `D`

`conc` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(Apo)
```

`apple` *Confounded apple slice browning experiment*

Description

Data from the confounded apple slice browning experiment in chapter 7 of Design and Analysis of Experiments with R

Usage

```
data(apple)
```

Format

A data frame with 24 observations on the following 4 variables.

`Block` a factor with levels `1` `2` `3` `4`

`A` a factor with levels `0` `1` `2` `3`

`B` a factor with levels `0` `1` `2`

`rating` a numeric vector containing the response

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(apple)
```

`arso` $2\hat{}(7-3)$ *arsenic removal experiment*

Description

Data from the $2^{(7-3)}$ arsenic removal experiment in chapter 6 of Design and Analysis of Experiments with R

Usage

```
data(arso)
```

Format

A data frame with 8 observations on the following 8 variables.

- `A` a factor with levels `-1` `1`
- `B` a factor with levels `-1` `1`
- `C` a factor with levels `-1` `1`
- `D` a factor with levels `-1` `1`
- `E` a factor with levels `-1` `1`
- `F` a factor with levels `-1` `1`
- `G` a factor with levels `-1` `1`
- `y1` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(arso)
```

`augm` — $2\hat{}(7-3)$ *arsenic removal experiment augmented with mirror image*

Description

Data from the $2^{(7-3)}$ arsenic removal experiment augmented with mirror image in chapter 6 of Design and Analysis of Experiments with R

Usage

```
data(augm)
```

Format

A data frame with 8 observations on the following 8 variables.

- `A` a factor with levels `-1` `1`
- `B` a factor with levels `-1` `1`
- `C` a factor with levels `-1` `1`
- `fold` a factor with levels `original` `mirror`
- `D` a factor with levels `-1` `1`
- `E` a factor with levels `-1` `1`
- `F` a factor with levels `-1` `1`
- `G` a factor with levels `-1` `1`
- `y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(augm)
```

Bdish *Confounded Block Dishwashing Experiment*

Description

Data from the Confounded Block Dishwashing Experiment in chapter 7 of Design and Analysis of Experiments with R

Usage

```
data(Bdish)
```

Format

A data frame with 16 observations on the following 5 variables.

`Blocks` a factor with levels `1` `2` `3` `4`

`A` a factor with levels `-1` `1`

`B` a factor with levels `-1` `1`

`C` a factor with levels `-1` `1`

`D` a factor with levels `-1` `1`

`y` a numeric vector containing the response

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(Bdish)
```

`Bff` *Confounded block fractional mouse growth experiment*

Description

Data from the Confounded block fractional factorial mouse growth experiment in chapter 7 of Design and Analysis of Experiments with R

Usage

```
data(Bff)
```

Format

A data frame with 16 observations on the following 5 variables.

`Blocks` a factor with levels `1` `2` `3` `4` `5` `6` `7` `8`

`A` a factor with levels `-1` `1`

`B` a factor with levels `-1` `1`

`C` a factor with levels `-1` `1`

`D` a factor with levels `-1` `1`

`E` a factor with levels `-1` `1`

`F` a factor with levels `-1` `1`

`G` a factor with levels `-1` `1`

`H` a factor with levels `-1` `1`

`weight` a numeric vector containing the response

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(Bff)
```

bha *mouse liver enzyme experiment*

Description

Data from the mouse liver enzyme experiment in chapter 4 of Design and Analysis of Experiments with R

Usage

```
data(bha)
```

Format

A data frame with 16 observations on the following 4 variables.

`block` a factor with levels `1` `2`

`strain` a factor with levels `A/J` `129Ola` `NIH` `BALB/c`

`treat` a factor with levels `treated` `control`

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(bha)
```

BIBsize *Balanced incomplete blocksize*

Description

This function computes the number of blocks, treatment frequency and lambda for a potential BIB design

Usage

```
BIBsize(t,k)
```

Arguments

`t`	input - number of levels of the treatment factor
`k`	input - blocksize or number of experimental units per block

Value

a list containing the b=number of blocks, r=number of treatment replicates and lambda for a potential BIB design with t levels of treatment factor and blocksize k.

Author(s)

John Lawson

Examples

```
BIBsize(6,3)
## The function is currently defined as
BIBsize<-function(t,k)
{
  b<-t
  r<-0
  lambda<-0
  check<-0
  while (check==0) {
   while (r==0) {
     #cat("r=",r)
     testr<-(b*k)/t
     #cat("testr=",testr,"b=",b)
     if (testr==floor(testr)) {
       r<-testr
       } else {
       b<-b+1
       }
     }
      #cat("b=",b, "r=",r)
      testl<-(r*(k-1))/(t-1)
      #cat("testl=",testl,"b=",b)
      if (testl==floor(testl)) {
       lambda<-testl
       check=1
       } else {
       r<-0
       b<-b+1
      #cat("b=",b, "r=",r)
        }

    #cat("lambda=",lambda)
    }
  cat("Posible BIB design with b=",b," and r=",r," lambda=",lambda,"\n")
}
```

`bioequiv` *Extra-period crossover bioequivalence study*

Description

Data from the extra-period crossover bioequivalence study in chapter 9 of Design and Analysis of Experiments with R

Usage

```
data(bioequiv)
```

Format

A data frame with 108 observations on the following 5 variables.

`Group` a factor with levels `1` `2`

`Subject` a factor with levels `1` `2` `3` `4` `5` `6` `7` `8` `9` `10` `11` `12` `13` `14` `15` `16` `17` `18` `19` `21` `23` `24` `25` `26` `27` `28` `30` `31` `32` `33` `34` `35` `36` `120` `122` `129`

`Period` a factor with levels `1` `2` `3`

`Treat` a factor with levels `A` `B`

`Carry` a factor with levels `none` `A` `B`

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(bioequiv)
```

`bioeqv` *Latin Square bioequivalence experiment*

Description

Data from the Latin Square bioequivalence experiment in chapter 4 of Design and Analysis of Experiments with R

Usage

```
data(bioeqv)
```

Format

A data frame with 9 observations on the following 4 variables.

`Period` a factor with levels `1` `2` `3`

`Subject` a factor with levels `1` `2` `3`

`Treat` a factor with levels `A` `B` `C`

`AUC` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(bioeqv)
```

`blood` *Variance component study of calcium in blood serum*

Description

Data from the Variance component study of calcium in blood serum in chapter 5 of Design and Analysis of Experiments with R

Usage

```
data(blood)
```

Format

A data frame with 27 observations on the following 3 variables.

`sol` a factor with levels `1` `2` `3` `4`

`lab` a factor with levels `A` `B` `C`

`calcium` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(blood)
```

BoxM *Box and Meyer's unreplicated 2^4 from Chapter 3*

Description

Data from Box and Meyer's unreplicated 2^4 in chapter 3 of Design and Analysis of Experiments with R

Usage

```
data(BoxM)
```

Format

A data frame with 16 observations on the following 4 variables.

- `A` a numeric vector containing the coded (-1,1) levels of factor A
- `B` a numeric vector containing the coded (-1,1) levels of factor B
- `C` a numeric vector containing the coded (-1,1) levels of factor C
- `D` a numeric vector containing the coded (-1,1) levels of factor D
- `y` a numeric vector containing the response

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

References

Box, G. E. P. "George's Column", *Quality Engineering*, Vol. 3, pp. 405-410.

Examples

```
data(BoxM)
```

BPmonitor *blood pressure monitor experiment*

Description

Data from the blood pressure monitor experiment experiment in Chapter 7 of Design and Analysis of Experiments with R

Usage

```
data(BPmonitor)
```

Format

A data frame with 12 observations on the following 3 variables.

`Block` a factor with levels `1` `2` `3` `4` `5` `6`

`Treatment` a factor with levels `"P"` `"A"` `"B"` `"C"`

`pressure` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(BPmonitor)
```

`bread` *Bread rise experiment data from Chapter 2*

Description

Data from the bread rise experiment in chapter 2 of Design and Analysis of Experiments with R

Usage

```
data(bread)
```

Format

A data frame with 12 observations on the following 3 variables.

`loaf` a numeric vector

`time` a numeric vector

`height` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(bread)
```

`cake` *Split-Plot response surface for cake baking experiment*

Description

Data from the Split-Plot response surface for cake baking experiment in chapter 10 of Design and Analysis of Experiments with R

Usage

```
data(cake)
```

Format

A data frame with 11 observations on the following 6 variables.

`Ovenrun` a factor with levels `1` `2` `3` `4`

`x1` a numeric vector

`x2` a numeric vector

`y` a numeric vector

`x1sq` a numeric vector

`x2sq` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(cake)
```

`cement` *CCD design for cement workability experiment*

Description

Data from the CCD design for cement workability experiment in chapter 10 of Design and Analysis of Experiments with R

Usage

```
data(cement)
```

Format

A data frame with 20 observations on the following 4 variables.

`Block` a factor with levels `1` `2`

`x1` a numeric vector

`x2` a numeric vector

`x3` a numeric vector

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(cement)
```

`chem` *Chemical process experiment data from Chapter 3*

Description

Data from the Chemical process experiment in chapter 3 of Design and Analysis of Experiments with R

Usage

```
data(chem)
```

Format

A data frame with 16 observations on the following 4 variables.

`A` a numeric vector containing the coded (-1,1) levels of factor A

`B` a numeric vector containing the coded (-1,1) levels of factor B

`C` a numeric vector containing the coded (-1,1) levels of factor C

`D` a numeric vector containing the coded (-1,1) levels of factor D

`y` a numeric vector containing the response

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(chem)
```

chipman *Williams' crossover design for sprinting experiment*

Description

Data from the Williams' crossover design for sprinting experiment in chapter 9 of Design and Analysis of Experiments with R

Usage

```
data(chipman)
```

Format

A data frame with 36 observations on the following 5 variables.

`Square` a factor with levels `1 2`

`Group` a factor with levels `1 2 3`

`Subject` a factor with levels `1 2 3 4 5 6 7 8 9 10 11 12`

`Period` a factor with levels `1 2 3`

`Treat` a factor with levels `1 2 3`

`Carry` a factor with levels `0 1 2 3`

`Time` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(chipman)
```

COdata *CO emmisions experiment data from Chapter 3*

Description

Data from the CO emissions experiment in chapter 3 of Design and Analysis of Experiments with R

Usage

```
data(COdata)
```

Format

A data frame with 18 observations on the following 3 variables.

`Eth` a factor with levels `0.1` `0.2` `0.3`

`Ratio` a factor with levels `14` `15` `16`

`CO` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(COdata)
```

`colormap` *This function makes a colormap of correlations in a design matrix*

Description

This function makes a colormap of the correlations of a design matrix stored in the data frame design

Usage

```
colormap(design, mod)
```

Arguments

`design`	input - a data frame containing columns of the numeric factor levels
`mod`	input - a number indicationg the model for the colormap 1 = linear model containing only the terms in the dataframe 2 = linear model plus two factor interactions 3 = linear model plus 2 and 3 factor interactions 4 = linear model plus 2, 3, and 4 factor interactions

Author(s)

John Lawson

Examples

```
# color map of 2^(4-1) design
library(FrF2)
design <- FrF2(8, 4, randomize = FALSE)
colormap(design, mod=3)

# Makes color map for saturated 2^(7-4) design in Figure 6.14 p. 197
library(FrF2)
design <-FrF2( 8, 7)
colormap(design, mod=2)

# Makes colormap of an Alternate Screening Design
library(daewr)
ascr<-Altscreen(7)
colormap(ascr, mod=2)

# Makes colormap of a Model Robust Design
library(daewr)
MR16 <- ModelRobust('MR16m7g5', randomize = FALSE)
colormap(MR16, mod=2)

## The function is currently defined as
function(design,mod) {
#################### Inputs ########################################
# design - a data frame containing columns of the numeric factor levels
# mod - the model for the color plot of correlations
#    1 = Linear model containing only the terms in the data frame
#    2 = Linear model plus two factor interactions
#    3 = Linear model plus 2 and 3 factor interactions
#    4 = Linear model plus 2, 3 and 4 factor interactions
####################################################################
y<-runif(nrow(design),0,1)
if(mod==1) {test <- model.matrix(lm(y~(.),data=design))}
if(mod==2) {test <- model.matrix(lm(y~(.)^2,data=design))}
if(mod==3) {test <- model.matrix(lm(y~(.)^3,data=design))}
if(mod==4) {test <- model.matrix(lm(y~(.)^4,data=design))}
names<-colnames(test)
names<-gsub(':','',names)
names<-gsub('1','',names)
colnames(test)<-names
cmas<-abs(cor(test[,ncol(test):2]))
cmas<-cmas[c((ncol(cmas)):1), ]
rgb.palette <- colorRampPalette(c("white", "black"), space = "rgb")
levelplot(cmas, main="Map of absolute correlations", xlab="", ylab="", col.regions=rgb.palette(120),
            cuts=100, at=seq(0,1,0.01),scales=list(x=list(rot=90))) }
```

---

cont	*Control factor array and summary statistics for controller circuit design experiment*

Description

Data from the control factor array and summary statistics for controller circuit design experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(cont)
```

Format

A data frame with 18 observations on the following 6 variables.

`A` a numeric vector

`B` a numeric vector

`C` a numeric vector

`D` a numeric vector

`F` a numeric vector

`lns2` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(cont)
```

`cpipe` *Split-plot response surface for ceramic pipe experiment*

Description

Data from the Split-plot response surface for ceramic pipe experiment in chapter 10 of Design and Analysis of Experiments with R

Usage

```
data(cpipe)
```

Format

A data frame with 48 observations on the following 6 variables.

`WP` a factor with levels `1` `2` `3` `4` `5` `6` `7` `8` `9` `10` `11` `12`

`A` a numeric vector

`B` a numeric vector

`P` a numeric vector

`Q` a numeric vector

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(cpipe)
```

`culture` *paecilomyces variotii culture experiment*

Description

Data from the paecilomyces variotii culture experiment experiment in chapter 6 of Design and Analysis of Experiments with R

Usage

```
data(culture)
```

Format

A data frame with 16 observations on the following 9 variables.

`A` a factor with levels `-1 1`

`B` a factor with levels `-1 1`

`C` a factor with levels `-1 1`

`D` a factor with levels `-1 1`

`E` a factor with levels `-1 1`

`F` a factor with levels `-1 1`

`G` a factor with levels `-1 1`

`H` a factor with levels `-1 1`

`y1` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(culture)
```

`dairy` *Repeated measures study with dairy cow diets*

Description

Data from the Repeated measures study with dairy cow diets in chapter 9 of Design and Analysis of Experiments with R (compact format)

Usage

```
data(dairy)
```

Format

A data frame with 120 observations on the following 5 variables.

`Diet` a factor with levels `"Barley"` `"Mixed"` `"Lupins"`

`pr1` a numeric vector

`pr2` a numeric vector

`pr3` a numeric vector

`pr4` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(dairy)
```

`DefScreen` *Definitive Screening Designs*

Description

Recalls Jones and Nachtsheim's Definitive screening designs for 3-level factors and 3-level factors with added 2-level categorical factors.

Usage

```
DefScreen(m, c=0, center=0, randomize=FALSE)
```

Arguments

`m`	input- an integer, the m=number of 3-level factors
`c`	input- an integer, the m=number of 2-level categorical factors, default is zero if not supplied
`center`	input- an integer, the number of extra center points. This must be zero when c>0
`randomize`	input - logical

Value

a data frame containing the definitive screening design with 3-level factors first followed by 2-level factors.

Author(s)

John Lawson

References

Jones, B. and Nachtsheim, C. J. (2011) "A Class of Three Level Designs for Definitive Screening in the Presence of Second-Order Effects", Journal of Quality Technology, Vol 43, No. 1, 2011, pp 1-15. Jones, B. and Nachtsheim, C. J. (2013) "Definitive Screening Designs with Added Two-Level Categorical Factors", Journal of Quality Technology, Vol 44, No. 2, 2013, pp. 121-129.

Examples

```
DefScreen(m=8,c=2)
DefScreen(12)
DefScreen(m=4,c=4, randomize=TRUE)
```

`drug` *Data from rat behavior experiment in Chapter 4*

Description

Data from rat behavior experiment in Chapter 4 of Design and Analysis of Experiments with R

Usage

```
data(drug)
```

Format

A data frame with 50 observations on the following 3 variables.

`rat` a factor with levels `1` `2` `3` `4` `5` `6` `7` `8` `9` `10`

`dose` a factor with levels `0.0` `0.5` `1.0` `1.5` `2.0`

`rate` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(drug)
```

EEw1s1 *D-efficient Estimation Equivalent Response Surface Designs*

Description

Recalls Jones and Goos JQT Estimation Equivalent Response Surface Designs for 1 whole plot factor and 1 sub-plot factor from a catalog

Usage

```
EEw1s1(des, randomize=FALSE)
```

Arguments

`des`	input- a character variable containing the name of a design in the catalog. If left blank, the function prints a table showing all the design names in the catalog
`randomize`	input- a logical

Value

design

Author(s)

John Lawson

References

Jones, B. and Goos, P.(2012) "An Algorithm for Finding D-Efficient Equivalent-Estimation Second-Order Split Plot Designs", Journal of Quality Technology, Vol 44, No. 4, pp281-303, 2012.

Examples

```
EEw1s1()
EEw1s1('EE8R4WP')
EEw1s1('EE8R4WP', randomize=TRUE)
```

EEw1s2 *D-efficient Estimation Equivalent Response Surface Designs*

Description

Recalls Jones and Goos JQT Estimation Equivalent Response Surface Designs for 1 whole plot factor and 2 sub-plot factors from a catalog

Usage

```
EEw1s2(des, randomize=FALSE)
```

Arguments

`des`	input- a character variable containing the name of a design in the catalog. If left blank, the function prints a table showing all the design names in the catalog
`randomize`	input- a logical

Value

design

Author(s)

John Lawson

References

Jones, B. and Goos, P.(2012) "An Algorithm for Finding D-Efficient Equivalent-Estimation Second-Order Split Plot Designs", Journal of Quality Technology, Vol 44, No. 4, pp281-303, 2012.

Examples

```
EEw1s2( )
EEw1s2('EE12R4WP')
EEw1s2('EE12R4WP', randomize=TRUE)
EEw1s2('EE12R6WP')
EEw1s2('EE12R6WP', randomize=TRUE)
EEw1s2('EE14R7WP')
EEw1s2('EE14R7WP', randomize=TRUE)
EEw1s2('EE15R5WP')
EEw1s2('EE15R5WP', randomize=TRUE)
EEw1s2('EE16R4WP')
EEw1s2('EE16R4WP', randomize=TRUE)
EEw1s2('EE18R6WP')
EEw1s2('EE18R6WP', randomize=TRUE)
EEw1s2('EE20R4WP')
EEw1s2('EE20R4WP', randomize=TRUE)
EEw1s2('EE20R5WP')
```

```
EEw1s2('EE20R5WP', randomize=TRUE)
EEw1s2('EE21R7WP')
EEw1s2('EE21R7WP', randomize=TRUE)
EEw1s2('EE24R4WP')
EEw1s2('EE24R4WP', randomize=TRUE)
EEw1s2('EE24R6WP')
EEw1s2('EE24R6WP', randomize=TRUE)
EEw1s2('EE25R5WP')
EEw1s2('EE25R5WP', randomize=TRUE)
EEw1s2('EE28R7WP')
EEw1s2('EE28R7WP', randomize=TRUE)
EEw1s2('EE30R6WP')
EEw1s2('EE30R6WP', randomize=TRUE)
EEw1s2('EE30R5WP')
EEw1s2('EE30R5WP', randomize=TRUE)
EEw1s2('EE35R7WP')
EEw1s2('EE35R7WP', randomize=TRUE)
EEw1s2('EE36R6WP')
EEw1s2('EE36R6WP', randomize=TRUE)
EEw1s2('EE42R7WP')
EEw1s2('EE42R7WP', randomize=TRUE)
```

`EEw1s3` *D-efficient Estimation Equivalent Response Surface Designs*

Description

Recalls Jones and Goos JQT Estimation Equivalent Response Surface Designs for 1 whole plot factor and 3 sub-plot factors from a catalog

Usage

```
EEw1s3(des, randomize=FALSE)
```

Arguments

`des`	input- a character variable containing the name of a design in the catalog. If left blank, the function prints a table showing all the design names in the catalog
`randomize`	input- a logical

Value

design

Author(s)

John Lawson

References

Jones, B. and Goos, P.(2012) "An Algorithm for Finding D-Efficient Equivalent-Estimation Second-Order Split Plot Designs", Journal of Quality Technology, Vol 44, No. 4, pp281-303, 2012.

Examples

```
EEw1s3()
EEw1s3('EE16R4WP')
EEw1s3('EE16R4WP', randomize=TRUE)
```

EEw2s1 *D-efficient Estimation Equivalent Response Surface Designs*

Description

Recalls Jones and Goos JQT Estimation Equivalent Response Surface Designs for 2 whole plot factors and 1 sub-plot factor from a catalog

Usage

```
EEw2s1(des, randomize=FALSE)
```

Arguments

des	input- a character variable containing the name of a design in the catalog. If left blank, the function prints a table showing all the design names in the catalog
randomize	input- a logical

Value

design

Author(s)

John Lawson

References

Jones, B. and Goos, P.(2012) "An Algorithm for Finding D-Efficient Equivalent-Estimation Second-Order Split Plot Designs", Journal of Quality Technology, Vol 44, No. 4, pp281-303, 2012.

Examples

```
EEw2s1()
EEw2s1('EE21R7WP')
EEw1s1('EE21R7WP', randomize=TRUE)
```

EEw2s2 *D-efficient Estimation Equivalent Response Surface Designs*

Description

Recalls Jones and Goos JQT Estimation Equivalent Response Surface Designs for 2 whole plot factors and 1 sub-plot factor from a catalog

Usage

```
EEw2s2(des, randomize=FALSE)
```

Arguments

`des`	input- a character variable containing the name of a design in the catalog. If left blank, the function prints a table showing all the design names in the catalog
`randomize`	input- a logical

Value

design

Author(s)

John Lawson

References

Jones, B. and Goos, P.(2012) "An Algorithm for Finding D-Efficient Equivalent-Estimation Second-Order Split Plot Designs", Journal of Quality Technology, Vol 44, No. 4, pp281-303, 2012.

Examples

```
EEw2s2()
EEw2s2('EE21R7WP')
EEw1s2('EE21R7WP', randomize=TRUE)
```

EEw2s3 *D-efficient Estimation Equivalent Response Surface Designs*

Description

Recalls Jones and Goos JQT Estimation Equivalent Response Surface Designs for 2 whole plot factors and 1 sub-plot factor from a catalog

Usage

```
EEw2s3(des, randomize=FALSE)
```

Arguments

`des`	input- a character variable containing the name of a design in the catalog. If left blank, the function prints a table showing all the design names in the catalog
`randomize`	input- a logical

Value

design

Author(s)

John Lawson

References

Jones, B. and Goos, P.(2012) "An Algorithm for Finding D-Efficient Equivalent-Estimation Second-Order Split Plot Designs", Journal of Quality Technology, Vol 44, No. 4, pp281-303, 2012.

Examples

```
EEw2s3()
EEw2s3('EE24R8WP')
EEw1s3('EE24R8WP', randomize=TRUE)
```

`EEw3` *D-efficient Estimation Equivalent Response Surface Designs*

Description

Recalls Jones and Goos JQT Estimation Equivalent Response Surface Designs for 3 whole plot factors and 1-2 sub-plot factors from a catalog

Usage

```
EEw3(des, randomize=FALSE)
```

Arguments

`des`	input- a character variable containing the name of a design in the catalog. If left blank, the function prints a table showing all the design names in the catalog
`randomize`	input- a logical

Value

design

Author(s)

John Lawson

References

Jones, B. and Goos, P.(2012) "An Algorithm for Finding D-Efficient Equivalent-Estimation Second-Order Split Plot Designs", Journal of Quality Technology, Vol 44, No. 4, pp281-303, 2012.

Examples

```
EEw3()
EEw3('EE22R11WP')
EEw3('EE22R11WP', randomize=TRUE)
EEw3('EE48R12WP')
EEw3('EE48R12WP', randomize=TRUE)
```

`eptaxr` *Single array and raw response for silicon layer growth experiment*

Description

Data from the single array and raw response for silicon layer growth experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(eptaxr)
```

Format

A data frame with 64 observations on the following 9 variables.

`A` a numeric vector

`B` a numeric vector

`C` a numeric vector

`D` a numeric vector

`E` a numeric vector

`F` a numeric vector

`G` a numeric vector

`H` a numeric vector

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(eptaxr)
```

eptaxs2 — *Control array and variance of response for silicon layer growth experiment*

Description

Data from the control array and variance of response for silicon layer growth experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(eptaxs2)
```

Format

A data frame with 16 observations on the following 9 variables.

`A` a numeric vector

`B` a numeric vector

`C` a numeric vector

`D` a numeric vector

`E` a numeric vector

`F` a numeric vector

`G` a numeric vector

`H` a numeric vector

`s2` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(eptaxs2)
```

`eptaxyb` *Control array and mean response for silicon layer growth experiment*

Description

Data from the control array and mean response for silicon layer growth experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(eptaxyb)
```

Format

A data frame with 16 observations on the following 9 variables.

- `A` a numeric vector
- `B` a numeric vector
- `C` a numeric vector
- `D` a numeric vector
- `E` a numeric vector
- `F` a numeric vector
- `G` a numeric vector
- `H` a numeric vector
- `ybar` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(eptaxyb)
```

Fcrit *F-Distribution critical values*

Description

Gets F-distribution critical values

Usage

```
Fcrit(alpha, nu1, nu2)
```

Arguments

alpha	input- right tail area beyond critical value
nu1	input - numerator degrees of freedom for F-distribution
nu2	input - denominator degrees of freedom for F-distribution

Value

returned critical value

Author(s)

John Lawson

Examples

```
Fcrit(.05,2,15)
## The function is currently defined as
function(alpha,nu1,nu2) qf(1-alpha,nu1,nu2)
```

fhstep *Subsequent steps in a forward stepwise regression that preserves model hierarchy*

Description

This function performs a single step of a hierarchical forward stepwise regression by entering additional term(s) to a model already created by ihstep or fhstep. If an interaction or quadratic term is entered first, the parent main effects are also entered into the model. This function is called by HierAFS.R

Usage

```
fhstep(y,des,m,c,prvm)
```

Arguments

`y`	input - this is a data frame containing a single numeric column of response data.
`des`	input - this is a data frame containing the numeric columns of the candidate independent variables. The column names of des are of length 1 i.e., letters of the alphabet. The m three-level factors always preceed the c two-level factors in the design.
`m`	input - this is an integer equal to the number of three-level factors in the design
`c`	input - this is an integer equal to the number of two-level factors in the design. Note m+c must be equal to the number of columns of des.
`prvm`	input - this is a vector of text names of the terms in the model. This is created as the value resulting from running ihstep or fhstep.

Value

returned vector of terms entered in the model at this step.

Author(s)

John Lawson

Examples

```
library(daewr)
des <- DefScreen( m = 8 )
pd<-c(5.35,4.4,12.91,3.79,4.15,14.05,11.4,4.29,3.56,11.4,10.09,5.9,9.54,4.53,3.919,8.1,5.35)
trm<-ihstep(pd,des,m=8,c=0)
trm<-fhstep(pd,des,m=8,c=0,trm)
trm<-fhstep(pd,des,m=8,c=0,trm)
trm<-fhstep(pd,des,m=8,c=0,trm)
```

`fhstepDS` *Forward Stepwise modeling taking into account special structure of Definitive Screening Design*

Description

This function performs a single step of a forward stepwise regression by entering an additional 2nd order term to a model already created by FitDefSc.R or fhstepDS.R This function is called by FitDefSc.R

Usage

```
fhstepDS(y,des,m,c,prvm)
```

Arguments

`y`	input - this is a data frame containing a single numeric column of response data.
`des`	input - this is a data frame containing the numeric columns of the candidate independent variables. The column names of des are of length 1 i.e., letters of the alphabet. The m three-level factors always preceed the c two-level factors in the design.
`m`	input - this is an integer equal to the number of three-level factors in the design
`c`	input - this is an integer equal to the number of two-level factors in the design. Note m+c must be equal to the number of columns of des.
`prvm`	input - this is a vector of text names of the terms in the model. This is created as the value resulting from running ihstep or fhstep.

Value

returned vector of terms entered in the model at this step.

Author(s)

John Lawson

`firstm` *Find first term to enter forward stepwise regression that preserves model hierarchy*

Description

This function finds the first term to enter a hierarchical forward stepwise regression. If the term is an interaction or quadratic term, the parent main effects are also included. This function is called by ihstep.R

Usage

```
firstm(y,des)
```

Arguments

`y`	input - this is a data frame containing a single numeric column of response data.
`des`	input - this is a data frame containing the numeric columns of the candidate independent variables. The column names of des are of length 1. The m three-level factors always preceed the c two-level factors in the design.

Value

returned vector of terms to be entered in the model at the first step.

Author(s)

John Lawson

`FitDefSc` *An Effective Design Based Model Fitting Method for Definitive Screening Designs*

Description

This function performs fits a model to a Definitive Screeing Design by first restricting main effects to the smallest main effects and those significant at at least the .20 level in a main effects model. Next forward stepwise selection is used to enter 2 factor interactions and quadratic effects.

Usage

```
FitDefSc(y,design,alpha=.05)
```

Arguments

`y`	input - this is a vector containing a single numeric column of response data.
`design`	input - this is a data frame containing the numeric columns of the candidate independent variables created by the DefScreen function with only numerical factors i.e. c=0. The factor names or colnames(design) should always be of length 1 (for example letters of the alphabet "A", "B", etc.)
`alpha`	input - alpha to enter in the forward stepwise regression with second order candidates should be between 0.05 and 0.20

Author(s)

John Lawson

Examples

```
design<-DefScreen(m=5,c=0,randomize=FALSE)
Smeso<-c(241,295,260,338,320,265,275,248,66,383,313)
FitDefSc(Smeso,design,alpha=.12)
```

`fnextrm` *Find first term to enter forward stepwise regression that preserves model hierarchy*

Description

This function finds the first term to enter a hierarchical forward stepwise regression. If the term is an interaction or quadratic term, the parent main effects are also included. This function is called by ihstep.R

Usage

```
fnextrm(y,des,prvm)
```

Arguments

`y`	input - this is a data frame containing a single numeric column of response data.
`des`	input - this is a data frame containing the numeric columns of the candidate independent variables. The column names of des are of length 1. The m three-level factors always preceed the c two-level factors in the design.
`prvm`	input - this is a vector of text names of the terms in the model. This is created as the value resulting from running ihstep or fhstep.

Value

returned vector of terms to be entered in the model at the next step.

Author(s)

John Lawson

`fntrmDS` *Find first term to enter forward stepwise regression that preserves model hierarchy*

Description

This function finds the first term to enter a hierarchical forward stepwise regression. If the term is an interaction or quadratic term, the parent main effects are also included. This function is called by ihstep.R

Usage

```
fntrmDS(y,des,prvm)
```

Arguments

`y`	input - this is a data frame containing a single numeric column of response data.
`des`	input - this is a data frame containing the numeric columns of the candidate independent variables. The column names of des are of length 1. The m three-level factors always preceed the c two-level factors in the design.
`prvm`	input - this is a vector of text names of the terms in the model. This is created as the value resulting from running ihstep or fhstep.

Value

returned vector of terms to be entered in the model at the next step.

Author(s)

John Lawson

`Fpower` *F-Distribution Power Calculation*

Description

Calculates the power for the non-central F-distribution

Usage

```
Fpower(alpha, nu1, nu2, nc)
```

Arguments

`alpha`	input - critical value alpha
`nu1`	input - degrees of freedom for numerator
`nu2`	input - degrees of freedom for denominator
`nc`	input - noncentrality parameter

Value

probability of exceeding fcrit(alpha, nu1,nu2) with the non-central F-distribution with nu1 and nu2 degrees of freedom and noncentrality parameter nc

Author(s)

John Lawson

Examples

```
Fpower(0.05,2,15,6.428)

## The function is currently defined as
function(alpha,nu1,nu2,nc) 1-pf(Fcrit(alpha,nu1,nu2),nu1,nu2,nc)
```

Fpower1 *F-Distribution Power Calculation*

Description

Calculates the power for one-way ANOVA

Usage

```
Fpower1(alpha,nlev,nreps,Delta,sigma)
```

Arguments

alpha	input - significance level of the F-test.
nlev	input - the number of levels of the factor
nreps	input - the number of replicates in each level of the factor.
Delta	input - the size of a practical difference in two cell means.
sigma	input - the standard deviation of the experimental error.

Value

probability of exceeding fcrit(alpha, nu1,nu2) with the non-central F-distribution with nu1 and nu2 degrees of freedom and noncentrality parameter nc

Author(s)

John Lawson

Examples

```
Fpower1(alpha=.05,nlev=3,nreps=4,Delta=3,sigma=sqrt(2.1))


rmin <-2 #smallest number of replicates considered
rmax <-6 # largest number of replicates considered
alpha <- rep(0.05, rmax - rmin +1)
sigma <-rep(sqrt(2.1), rmax - rmin +1)
nreps <-c(rmin:rmax)
nlev <- rep(3,rmax - rmin +1)
nreps <- rmin:rmax
Delta <- rep(3,rmax - rmin +1)
power <- Fpower1(alpha,nlev,nreps,Delta,sigma)
data.frame(r=nreps,Power=power)


## The function is currently defined as
Fpower1<-function(alpha=NULL, nlev=NULL,nreps=NULL, Delta=NULL, sigma=NULL)
{
```

```
##### Power Calculation for one way ANOVA ###########
# Argument list
# alpha the significance level of the test
# nlev the number of levels of the factor
# nreps the number of replicates in each level of the factor
# Delta the size of a practical difference in two cell means
# sigma the standard deviation of the experimental error
############################################################
if (is.null(alpha)|is.null(nlev)|is.null(nreps)|is.null(Delta)|is.null(sigma))
  stop("you must supply alpha, nlev, nreps, Delta and sigma")
css<-(Delta^2)/2
nc<- (nreps*css)/(sigma^2)
df1<-nlev-1
df2<-(nreps-1)*nlev
power <- 1-pf(Fcrit(alpha,df1,df2),df1,df2,nc)
return(power)
}
```

Fpower2 *F-Distribution Power Calculation*

Description

Calculates the power for a two-way ANOVA

Usage

```
Fpower2(alpha,nlev,nreps,Delta,sigma)
```

Arguments

`alpha`	input - significance level of the F-test.
`nlev`	input - vector of length two containing the number of levels of the factors.
`nreps`	input - the the number of replicates in each combination of factor levels.
`Delta`	input - the size of a practical difference in two marginal factor level means.
`sigma`	input - the standard deviation of the experimental error.

Value

probability of exceeding fcrit(alpha, nu1,nu2) with the non-central F-distribution with nu1 and nu2 degrees of freedom and noncentrality parameter nc

Author(s)

John Lawson

Examples

```
power <- Fpower2(.05, nlev = c(4,4), nreps=2, Delta= 1, sigma=.32)

rmin <- 2 # smallest number of replicates
rmax <- 4 # largest number of replicates
alpha <- .05
sigma <- .32
Delta <- 1.0
nlev <- c(4,4)
nreps <- c(rmin:rmax)
result <- Fpower2(alpha, nlev, nreps, Delta, sigma)
options(digits = 5)
result


## The function is currently defined as
Fpower2<-function(alpha=NULL, nlev=NULL,nreps=NULL, Delta=NULL, sigma=NULL)
{
##### Power Calculation for two way ANOVA ###########
# Argument list
# alpha the significance level of the test.
# nlev vector containing the number of levels of the factors.
# nreps the number of replicates in each combination of factor levels.
# Delta the size of a practical difference in two marginal factor level means.
# sigma the standard deviation of the experimental error.
############################################################
if (is.null(alpha)|is.null(nlev)|is.null(nreps)|is.null(Delta)|is.null(sigma))
  stop("you must supply alpha, nlev, nreps, Delta and sigma")
if(length(nlev)<2)
  stop ("nlev must be a two component vecto containing levels of the 1st and 2nd factors")
a <- nlev[1]
b <- nlev[2]
cssb <- (Delta^2)/2
ncb <- a*(nreps*cssb)/(sigma^2)
cssa<-(Delta^2)/2
nca<- b*(nreps*cssa)/(sigma^2)
dfa<- a-1
dfb<- b-1
df2<-(nreps-1)*b*a
powera <- 1-pf(Fcrit(alpha,dfa,df2),dfa,df2,nca)
powerb <- 1-pf(Fcrit(alpha,dfb,df2),dfa,df2,nca)
result <-cbind(nreps,df2,powera,powerb)
}
```

---

fullnormal *This function makes a full normal plot of the elements of the vector called effects*

Description

This function makes a full normal plot of the elements of the vector called effects

Usage

```
fullnormal(effects, labs, alpha = 0.05, refline = "TRUE")
```

Arguments

effects	input - vector of effects to be plotted
labs	input - vector of labels of the effects to be plotted
alpha	input - alpha level for labeling of significant effects using Lenth statistic
refline	input - logical variable that indicates whether a reference line is added to the plot (default is "TRUE")

Author(s)

John Lawson

Examples

```
# Example Separate Normal plots of whole and split plot effects from an unreplicated split-plot
data(plasma)
sol<-lm(y~A*B*C*D*E,data=plasma)
summary(sol)
# get whole plot effects and split plot effects
effects<-coef(sol)
effects<-effects[c(2:32)]
Wpeffects<-effects[c(1:4, 6:11, 16:19, 26)]
Speffects<-effects[c(5,12:15,20:25,27:31)]

#make separate normal plots
fullnormal(Wpeffects,names(Wpeffects),alpha=.10)
fullnormal(Speffects,names(Speffects),alpha=.05)


## The function is currently defined as
function (effects, labs, alpha = 0.05, refline = "TRUE")
{
    crit <- LenthPlot(effects, alpha = alpha, plt = FALSE)["ME"]
    names <- names(effects)
    names <- gsub(":", "", names)
    names <- gsub("1", "", names)
    le <- length(effects)
    for (i in 1:le) {
        logc <- (abs(effects[i]) <= crit)
        if (logc) {
            names[i] <- " "
        }
    }
    qqnorm(effects, ylab = "Estimated Effects", xlab = "Normal Scores")
    x <- qqnorm(effects, plot = FALSE)
    zscr <- (x$x)
    effp <- effects[zscr > 0]
```

```
    zp <- zscr[zscr > 0]
    namep <- names[zscr > 0]
    effn <- effects[zscr < 0]
    zn <- zscr[zscr < 0]
    namen <- names[zscr < 0]
    text(zp, effp, namep, pos = 1)
    text(zn, effn, namen, pos = 3)
    ahe <- abs(effects)
    s0 <- 1.5 * median(ahe)
    selhe <- ahe < (2.5 * s0)
    pse = 1.5 * median(ahe[selhe])
    if (refline) {
        abline(0, pse)
    }
  }
```

---

gagerr *Gauge R&R Study*

Description

Data from the Gauge R&R Study in chapter 5 of Design and Analysis of Experiments with R

Usage

```
data(gagerr)
```

Format

A data frame with 60 observations on the following 3 variables.

`part` a factor with levels `1 2 3 4 5 6 7 8 9 10`

`oper` a factor with levels `1 2 3`

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(gagerr)
```

gapstat — *This function computes the gap statistic which is used to test for an outlier using Daniels method*

Description

This function computes the gap statistic which is used to test for an outlier using Daniels method

Usage

```
gapstat(beta, pse)
```

Arguments

`beta`	input - vector of coefficients from saturated model fit to the data
`pse`	input - Lenth's PSE statistic calculated from the elements of beta

Value

returned gap statistic

Author(s)

John Lawson

Examples

```
## The function is currently defined as
function (beta, pse)
{
    p <- length(beta)
    psehe <- pse
    sel <- beta >= 0
    betap <- beta[sel]
    betap <- sort(betap)
    betas <- betap[1]
    sel <- beta < 0
    betan <- beta[sel]
    nn <- length(betan)
    betan <- sort(betan)
    betal <- betan[nn]
    zl <- qnorm((nn - 0.375)/(p + 0.25))
    zs <- qnorm((nn + 1 - 0.375)/(p + 0.25))
    gap <- ((betas - betal)/psehe)/(zs - zl)
    return(gap)
  }
```

Gaptest *This function uses Daniel's Method to find an outlier in an unreplicated $2\hat{}(k-p)$ design.*

Description

This function uses Daniel's Method to find an outlier in an unreplicated $2^{(k-p)}$ design.

Usage

```
Gaptest(DesY)
```

Arguments

DesY	input this is a data frame containing an unreplicated $2^{(k-p)}$ design. The last variable in the data frame should be the numeric response.

Author(s)

John Lawson

References

Box, G.E.P. (1991) "George's column: Finding bad values in factorial designs", Quality Engineering, 3, 249-254.

Examples

```
# Example from Box(1991)
data(BoxM)
Gaptest(BoxM)


## The function is currently defined as
function (DesY)
{
    ncheck <- dim(DesY)
    ncheck <- ncheck[1]
    tcnd = TRUE
    if (ncheck == 8) {
        tcnd = FALSE
    }
    if (ncheck == 16) {
        tcnd = FALSE
    }
    if (ncheck == 32) {
        tcnd = FALSE
    }
    if (tcnd) {
```

```
        stop("This function only works for 8, 16, or 32 run designs",
            "\n")
    }
    else {
        if (ncheck == 8)
            ncheck = 16
        critg16 <- c(1.7884, 5.1009)
        critg32 <- c(1.7297, 5.8758)
        modf <- lm(y ~ (.)^4, x = TRUE, data = DesY)
        nbeta <- dim(DesY)
        nbeta <- nbeta[1]
        he <- modf$coef
        selcol <- which(!is.na(he))
        he <- he[selcol]
        he <- he[-1]
        p <- length(he)
        n <- p + 1
        cn1 <- names(he)
        ccn1 <- gsub("[^A-Z]", "", cn1)
        names(he) <- ccn1
        ahe <- abs(he)
        s0 <- 1.5 * median(ahe)
        selhe <- ahe < (2.5 * s0)
        pse = 1.5 * median(ahe[selhe])
        gap <- gapstat(he, pse)
        if (ncheck == 16) {
            test = (gap > critg16[1])
        }
        else {
            test = (gap > critg32[1])
        }
        if (test) {
            X <- modf$x
            X <- X[, selcol]
            X <- X[, -1]
            se <- as.matrix(sign(he), nrow = 1)
            sigef <- LGB(he, rpt = FALSE, plt = FALSE)
            for (i in 1:length(he)) {
                if (sigef[i] == "yes") {
                  se[i] = 0
                }
            }
            sp <- X %*% se
            asp <- abs(sp)
            oo <- max.col(t(asp))
            ae <- abs(he)
            sae <- sort(ae)
            nsmall <- round(length(he)/2)
            bias <- 2 * sum(sae[1:nsmall])
            y <- DesY$y
            ycorr <- DesY$y
            ycorr[oo] <- ycorr[oo] + (-1 * sign(sp[oo])) * bias
            detect <- c(rep("no", n))
```

```
            detect[oo] <- "yes"
            cat("Initial Outlier Report", "\n")
            cat("Standardized-Gap = ", gap, "Significant at 50th percentile",
                "\n")
            DesYc <- cbind(DesY[, 1:(dim(DesY)[2] - 1)], ycorr)
            modf <- lm(ycorr ~ (.)^4, x = TRUE, data = DesYc)
            che <- modf$coef
            che <- che[!is.na(che)]
            che <- che[-1]
            p <- length(che)
            n <- p + 1
            cn <- names(che)
            ccn <- gsub("[^A-Z]", "", cn)
            names(che) <- ccn
            ache <- abs(che)
            s0 <- 1.5 * median(ache)
            selche <- ache < (2.5 * s0)
            psec = 1.5 * median(ache[selche])
            gap <- gapstat(he, psec)
            if (ncheck == 16)
                test2 = (gap > critg16[2])
            else test2 = (gap > critg32[2])
            if (test2) {
                cat("Final Outlier Report", "\n")
                cat("Standardized-Gap = ", gap, "Significant at 99th percentile",
                  "\n")
                cat("   ", "\n")
                cat("    Corrrected Data Report  ", "\n")
                cat("Response  Corrected Response   Detect Outlier",
                  "\n")
                cat(paste(format(DesY$y, width = 8), format(DesYc$ycorr,
                  width = 13), "           ", format(detect,
                  width = 10), "\n"), sep = "")
                tce <- LGB(che)
            }
            else {
                cat("Final Outlier Report", "\n")
                cat("No significant outlier detected in second pass",
                  "\n")
                LGB(he)
                cat("    ", "\n")
            }
        }
    }
}
```

gear — *Unreplicated split-plot fractional-factorial experiment on geometric distortion of drive gears*

Description

Data from the unreplicated split-plot fractional-factorial experiment on geometric distortion of drive gears in chapter 8 of Design and Analysis of Experiments with R

Usage

```
data(gear)
```

Format

A data frame with 16 observations on the following 6 variables.

`A` a factor with levels `-1` `1`

`B` a factor with levels `-1` `1`

`C` a factor with levels `-1` `1`

`P` a factor with levels `-1` `1`

`Q` a factor with levels `-1` `1`

`y` a numeric vector containing the response

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(gear)
```

`halfnorm` *This function makes a half normal plot of the elements of the vector called effects*

Description

This function makes a half normal plot of the elements of the vector called effects

Usage

```
halfnorm(effects, labs, alpha = 0.05, refline = "TRUE")
```

Arguments

`effects`	input - vector of effects to be plotted
`labs`	input - vector of labels of the effects to be plotted
`alpha`	input - alpha level for labeling of significant effects using Lenth statistic
`refline`	input - logical variable that indicates whether a reference line is added to the plot (default is "TRUE")

Author(s)

John Lawson

Examples

```
# Example Separate Normal plots of whole and split plot effects from an unreplicated split-plot
data(plasma)
sol<-lm(y~A*B*C*D*E,data=plasma)
# get whole plot effects and split plot effects
effects<-coef(sol)
effects<-effects[c(2:32)]
Wpeffects<-effects[c(1:4, 6:11, 16:19, 26)]
Speffects<-effects[c(5,12:15,20:25,27:31)]

#make separate half normal plots
halfnorm(Wpeffects,names(Wpeffects),alpha=.10)
halfnorm(Speffects,names(Speffects),alpha=.05)
```

---

hardwood *low grade hardwood conjoint study*

Description

Data from the low grade hardwood conjoint study in chapter 6 of Design and Analysis of Experiments with R

Usage

```
data(hardwood)
```

Format

A data frame with 12 observations on the following 5 variables.

`Design` a factor with levels "RC" "AC" "OCI" "OCII"

`Price` a numeric variable

`Density` a factor with levels "Clear" "Heavy" "Medium"

`Guarantee` a factor with levels "1y" "Un"

`Rating` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(hardwood)
```

HierAFS *RSM forward regression keeping model hierarchy*

Description

This function performs a hierarchical forward stepwise regression. If an interaction or quadratic term is entered in the model, the parent main effects are also entered into the model.

Usage

```
HierAFS(y,x,m,c,step)
```

Arguments

y	input - this is a vector containing a single numeric column of response data.
x	input - this is a data frame containing the numeric columns of the candidate independent variables. The m three-level factors always preceed the c two-level factors in the design. The factor names or colnames(x) should always be of length (for example letters of the alphabet "A", "B", etc.)
m	input - this is an integer equal to the number of three-level factors in the design
c	input - this is an integer equal to the number of two-level factors in the design. Note m+c must be equal to the number of columns of des.
step	input - this is a single numeric value containing the n umber of steps requested.

Value

returned data frame the first column is a factor variable containing the formula for the model fit at each step, the second numeric column is the R-square statistic for the model fit with each formula.

Author(s)

Gerhard Krennrich, and modified by John Lawson

Examples

```
#Definitive Screening Design
library(daewr)
set.seed(1234)
x <- DefScreen(m=5,c=0)
x$y <- 3*x$A + 2*x$B + 3*x$A*x$B + 2*x$A^2 + 2*x$C+rnorm(nrow(x),0,.5)
HierAFS(x$y,x[,-6],m=5,c=0,step=3)
# Alternate Screening Example
library(daewr)
Design<-Altscreen(nfac=6,randomize=FALSE)
Thickness<-c(4494,4592,4357,4489,4513,4483,4288,4448,4691,4671,4219,4271,4530,4632,4337,4391)
HierAFS(Thickness,Design,m=0,c=6,step=3)
```

ihstep — *First step in a forward stepwise regression that preserves model hierarchy*

Description

This function performs the first step of a hierarchical forward stepwise regression. If an interaction or quadratic term is entered first, the parent main effects are also entered into the model. This function is called by HierAFS.R

Usage

```
ihstep(y,des,m,c)
```

Arguments

`y`	input - this is a data frame containing a single numeric column of response data.
`des`	input - this is a data frame containing the numeric columns of the candidate independent variables. The column names of des are of length 1 i.e., letters of the alphabet. The m three-level factors always preceed the c two-level factors in the design.
`m`	input - this is an integer equal to the number of three level factors in the design
`c`	input - this is an integer equal to the number of two level factors in the design. Note m+c must be equal to the number of columns of des.

Value

returned vector of terms entered in the model at this step.

Author(s)

John Lawson

Examples

```
library(daewr)
des <- DefScreen( m = 8 )
pd<-c(5.35,4.4,12.91,3.79,4.15,14.05,11.4,4.29,3.56,11.4,10.09,5.9,9.54,4.53,3.919,8.1,5.35)
trm<-ihstep(pd,des,m=8,c=0)
```

`inject` *Single array for injection molding experiment*

Description

Data from the single array for injection molding experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(inject)
```

Format

A data frame with 20 observations on the following 8 variables.

`A` a numeric vector

`B` a numeric vector

`C` a numeric vector

`D` a numeric vector

`E` a numeric vector

`F` a numeric vector

`G` a numeric vector

`shrinkage` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(inject)
```

`LenthPlot` *Lenth's Plot of Effects*

Description

Plot of the factor effects with significance levels based on robust estimation of contrast standard errors.

Usage

```
LenthPlot(obj, alpha = 0.05, plt = TRUE, limits = TRUE,
    xlab = "factors", ylab = "effects", faclab = NULL, cex.fac = par("cex.lab"),
    cex.axis=par("cex.axis"), adj = 1, ...)
```

Arguments

`obj`	object of class `lm` or vector with the factor effects.
`alpha`	numeric. Significance level used for the *margin of error* (ME) and *simultaneous margin of error* (SME). See Lenth(1989).
`plt`	logical. If `TRUE`, a spikes plot with the factor effects is displayed. Otherwise, no plot is produced.
`limits`	logical. If `TRUE` ME and SME limits are displayed and labeled.
`xlab`	character string. Used to label the x-axis. "factors" as default.
`ylab`	character string. Used to label the y-axis. "effects" as default.
`faclab`	list with components `idx` (numeric vector) and `lab` (character vector). The `idx` entries of effects vector (taken from `obj`) are labelled as `lab`. The rest of the effect names are blanked. If `NULL` all factors are labelled using the coefficients' name.
`cex.fac`	numeric. Character size used for the factor labels.
`cex.axis`	numeric. Character size used for the axis.
`adj`	numeric between 0 and 1. Determines where to place the "ME" (margin of error) and the "SME" (simultaneous margin of error) labels (character size of 0.9*`cex.axis`). 0 for extreme left hand side, 1 for extreme right hand side.
`...`	extra parameters passed to `plot`.

Details

If `obj` is of class `lm`, `2*coef(obj)` is used as factor effect with the intercept term removed. Otherwise, `obj` should be a vector with the factor effects. Robust estimate of the contrasts standard error is used to calculate *marginal* (ME) and *simultaneous margin of error* (SME) for the provided significance (`1 -alpha`) level. See Lenth(1989). Spikes are used to display the factor effects. If `faclab` is `NULL`, factors are labelled with the effects or coefficient names. Otherwise, those `faclab$idx` factors are labelled as `faclab$lab`. The rest of the factors are blanked.

Value

The function is called mainly for its side effect. It returns a vector with the value of alpha used, the estimated PSE, ME and SME.

Author(s)

Ernesto Barrios. Extension provided by Kjetil Kjernsmo (2013).

References

Lenth, R. V. (1989). "Quick and Easy Analysis of Unreplicated Factorials". *Technometrics* Vol. 31, No. 4. pp. 469–473.

Examples

```
# Example Separate Normal plots of whole and split plot effects from an unreplicated split-plot
data(plasma)
sol<-lm(y~A*B*C*D*E,data=plasma)
summary(sol)
# get whole plot effects and split plot effects
effects<-coef(sol)
LenthPlot(effects,alpha=.05)
```

`LGB` *This function uses the LGB Method to detect significant effects in unreplicated fractional factorials.*

Description

This function uses the LGB Method to detect significant effects in unreplicated fractional factorials.

Usage

```
LGB(Beta, alpha = 0.05, rpt = TRUE, plt = TRUE, pltl = TRUE)
```

Arguments

`Beta`	input - this is the numeric vector of effects or coefficients to be tested
`alpha`	input - This is the significance level of the test
`rpt`	input - this is a logical variable that controls whether the report is written (default is TRUE)
`plt`	input - this is a logical variable that controls whether a half-normal plot is made (default is TRUE)
`pltl`	input - this is a logical variable that controls whether the significance limit line is drawn on the half-normal plot (default is TRUE)

Author(s)

John Lawson

References

Lawson, J., Grimshaw, S., Burt, J. (1998) "A quantitative method for identifying active contrasts in unreplicated factorial experiments based on the half-normal plot", Computational Statistics and Data Analysis, 26, 425-436.

Examples

```
data(chem)
modf<-lm(y~A*B*C*D,data=chem)
LGB(coef(modf)[-1],rpt=FALSE)


## The function is currently defined as
LGB <- function(Beta, alpha=.05,rpt=TRUE, plt=TRUE, pltl=TRUE) {
sigLGB<-LGBc(Beta,alpha,rpt,plt,pltl)
}
```

`LGBc` *This function does the calculations for the LGB Method to detect significant effects in unreplicated fractional factorials.*

Description

This function uses the LGB Method to detect significant effects in unreplicated fractional factorials.

Usage

```
LGBc(Beta, alpha = 0.05, rpt = TRUE, plt = TRUE, pltl = TRUE)
```

Arguments

`Beta`	input - this is the numeric vector of effects or coefficients to be tested
`alpha`	input - This is the significance level of the test
`rpt`	input - this is a logical variable that controls whether the report is written (default is TRUE)
`plt`	input - this is a logical variable that controls whether a half-normal plot is made (default is TRUE)
`pltl`	input - this is a logical variable that controls whether the significance limit line is drawn on the half-normal plot (default is TRUE)

Author(s)

John Lawson

References

Lawson, J., Grimshaw, S., Burt, J. (1998) "A quantitative method for identifying active contrasts in unreplicated factorial experiments based on the half-normal plot", Computational Statistics and Data Analysis, 26, 425-436.

Examples

```
data(chem)
modf<-lm(y~A*B*C*D,data=chem)
sig<-LGBc(coef(modf)[-1],rpt=FALSE)


## The function is currently defined as
function (Beta, alpha = 0.05, rpt = TRUE, plt = TRUE, pltl = TRUE)
{
    siglev <- c(0.1, 0.05, 0.025, 0.01)
    df <- c(7, 8, 11, 15, 16, 17, 26, 31, 32, 35, 63, 127)
    crittab <- matrix(c(1.265, 1.196, 1.161, 1.122, 1.11, 1.106,
        1.072, 1.063, 1.06, 1.059, 1.037, 1.023, 1.534, 1.385,
        1.291, 1.201, 1.186, 1.178, 1.115, 1.099, 1.093, 1.091,
        1.056, 1.034, 1.889, 1.606, 1.449, 1.297, 1.274, 1.26,
        1.165, 1.14, 1.13, 1.127, 1.074, 1.043, 2.506, 2.026,
        1.74, 1.447, 1.421, 1.377, 1.232, 1.197, 1.185, 1.178,
        1.096, 1.058), ncol = 4, byrow = FALSE)
    colind <- which(siglev == alpha, arr.ind = TRUE)
    if (length(colind) == 0) {
        stop("this function works only when alpha= .1, .05, .025 or .01")
    }
    rowind <- which(df == length(Beta), arr.ind = TRUE)
    if (length(rowind) == 0) {
        stop("this function works only for coefficent vectors of
length 7,8,11,15,16,26,31,32,35,63,or 127")
    }
    critL <- crittab[rowind, colind]
    acj <- abs(Beta)
    ranks <- rank(acj, ties.method = "first")
    s0 <- 1.5 * median(acj)
    p <- (ranks - 0.5)/length(Beta)
    z <- qnorm((p + 1)/2)
    moda <- lm(acj ~ -1 + z)
    beta1 <- moda$coef
    sel <- acj < 2.5 * s0
    modi <- lm(acj[sel] ~ -1 + z[sel])
    beta2 <- modi$coef
    Rn <- beta1/beta2
    pred <- beta2 * z
    n <- length(acj[sel])
    df <- n - 1
    sig <- sqrt(sum(modi$residuals^2)/df)
    se.pred <- sig * (1 + 1/n + (z^2)/sum(z[sel]^2))^0.5
    pred.lim <- pred + qt(0.975, df) * se.pred
    sigi <- c(rep("no", length(Beta)))
    sel2 <- acj > pred.lim
    sigi[sel2] <- "yes"
    if (plt) {
        plot(z, acj, xlab = "Half Normal Scores", ylab = "Absoulute Effects")
        lines(sort(z), sort(pred), lty = 1)
        for (i in 1:length(Beta)) {
```

```
            if (sigi[i] == "yes")
                text(z[i], acj[i], names(Beta)[i], pos = 1)
        }
        if (pltl) {
            lines(sort(z), sort(pred.lim), lty = 3)
        }
    }
    if (rpt) {
        cat("Effect Report", "\n")
        cat("  ", "\n")
        cat("Label     Half Effect    Sig(.05)", "\n")
        cat(paste(format(names(Beta), width = 8), format(Beta,
            width = 8), "      ", format(sigi, width = 10), "\n"),
            sep = "")
        cat("  ", "\n")
        cat("Lawson, Grimshaw & Burt Rn Statistic = ", Rn, "\n")
        cat("95th percentile of Rn = ", critL, "\n")
    }
    return(sigi)
  }
```

---

mod *Mod function*

Description

Gets mod of a to base b

Usage

```
mod(a,b)
```

Arguments

a	input- an integer
b	input - an integer

Value

remainder of a/b or mod(a,b)

Author(s)

John Lawson

Examples

```
mod(5,3)
## The function is currently defined as
mod<-function(a,b)
{a-b*floor(a/b)}
```

`ModelRobust` *Model Robust Factorial Designs*

Description

Recalls Li and Nachtsheim's model robust factorial designs from a catalog of data frames

Usage

```
ModelRobust(des, randomize=FALSE)
```

Arguments

`des`	input- a character variable containing the name of a design in the catalog. If left blank, the function prints a table showing all the design names in the catalog
`randomize`	input- a logical

Value

design

Author(s)

John Lawson

References

Li, W. and Nachtsheim, C. J. (2000) "Model Robust factorial Designs", Technometrics, Vol 42, No. 4, pp345-352, 2000.

Examples

```
ModelRobust()
ModelRobust('MR8m4g3')
ModelRobust('MR8m4g3', randomize=TRUE)
```

`MPV` *mixture process variable experiment with mayonnaise*

Description

Data from the mixture process variable experiment with mayonnaise in chapter 11 of Design and Analysis of Experiments with R

Usage

```
data(MPV)
```

Format

A data frame with 35 observations on the following 4 variables.

`x1` a numeric vector

`x2` a numeric vector

`x3` a numeric vector

`z1` a numeric vector

`z2` a numeric vector

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(MPV)
```

`Naph` *Yields of naphthalene black*

Description

Data from the Yields of naphthalene black of Chapter 5 in Design and Analysis of Experiments with R

Usage

```
data(Naph)
```

Format

A data frame with 30 observations on the following 2 variables.

`sample` a factor with levels `1 2 3 4 5 6`

`yield` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(Naph)
```

`OptPB` *Optimum Plackett-Burman Designs*

Description

Selects the columns from a Plackett-Burman Design Produced by FrF2 that will minimize model dependence for main effects and two factor interactions and returns the design in a data frame

Usage

```
OptPB(nruns, nfactors, randomize=FALSE)
```

Arguments

`nruns`	input- an integer representing the number of runs in the design
`nfactors`	input - in integer representing the number of factors in the design
`randomize`	input - logical

Value

design

Author(s)

John Lawson

References

Fairchild, K. (2011) "Screening Designs that Minimize Model Dependence", MS Project Department of Statistics Brigham Young University, Dec. 2011.

Examples

```
OptPB(12,8)
```

`pastry` *Blocked response surface design for pastry dough experiment*

Description

Data from the Blocked response surface design for pastry dough experiment in chapter 10 of Design and Analysis of Experiments with R

Usage

```
data(pastry)
```

Format

A data frame with 28 observations on the following 5 variables.

`Block` a factor with levels `1 2 3 4 5 6 7`

`x1` a numeric vector

`x2` a numeric vector

`x3` a numeric vector

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(pastry)
```

`PBDes` *Plackett-Burman Designs*

Description

Creates a 12, 20, or 24 run Plackett-Burman design in a data frame with numeric factor levels by cyclically rotating the factor leves in the first row

Usage

```
PBDes(nruns, nfactors, randomize=FALSE)
```

Arguments

`nruns`	input- an integer representing the number of runs in the design
`nfactors`	input - in integer representing the number of factors in the design
`randomize`	input - logical

Value

design

Author(s)

John Lawson

References

Lawson, J. (2015) "Design and Analysis of Experiments with R page 229",CRC Press, Boca Raton, 2015.

Examples

```
PBDes(nruns=12,nfactors=11)
PBDes(nruns=12,nfactors=11,randomize=TRUE)
PBDes(nruns=12,nfactors=9)
PBDes(nruns=20,nfactors=19)
PBDes(nruns=24,nfact=16)
PBDes(nruns=24,nfactors=23)
```

`pest` *Pesticide formulation experiment*

Description

Data from the Pesticide formulation experiment in chapter 11 of Design and Analysis of Experiments with R

Usage

```
data(pest)
```

Format

A data frame with 13 observations on the following 4 variables.

`x1` a numeric vector

`x2` a numeric vector

`x3` a numeric vector

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(pest)
```

`pesticide` *pesticide application experiment*

Description

Data from the pesticide application experiment in chapter 5 of Design and Analysis of Experiments with R

Usage

```
data(pesticide)
```

Format

A data frame with 16 observations on the following 4 variables.

`form` a factor with levels `A` `B`

`tech` a factor with levels `1` `2`

`plot` a factor with levels `1` `2`

`residue` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(pesticide)
```

plasma *Unreplicated split-plot 2^5 experiment on plasma treatment of paper*

Description

Data from the unreplicated split-plot 2^5 experiment on plasma treatment of paper in chapter 8 of Design and Analysis of Experiments with R

Usage

```
data(plasma)
```

Format

A data frame with 32 observations on the following 6 variables.

- `A` a factor with levels `-1` `1`
- `B` a factor with levels `-1` `1`
- `C` a factor with levels `-1` `1`
- `D` a factor with levels `-1` `1`
- `E` a factor with levels `-1` `1`
- `y` a numeric vector containing the response

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(plasma)
```

polvdat *Polvoron mixture experiment*

Description

Data from the Polvoron mixture experiment in chapter 11 of Design and Analysis of Experiments with R

Usage

```
data(polvdat)
```

Format

A data frame with 12 observations on the following 4 variables.

`x1` a numeric vector

`x2` a numeric vector

`x3` a numeric vector

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(polvdat)
```

`polymer` *polymerization strength variability study*

Description

Data from the polymerization strength variability study in chapter 5 of Design and Analysis of Experiments with R

Usage

```
data(polymer)
```

Format

A data frame with 120 observations on the following 5 variables.

`lot` a factor with levels `1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30`

`box` a factor with levels `1 2`

`prep` a factor with levels `1 2`

`test` a factor with levels `1 2`

`strength` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(polymer)
```

prodstd — *Complete control factor array and noise factor array for connector experiment*

Description

Data from the complete control factor array and noise factor array for connector experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(prodstd)
```

Format

A data frame with 16 observations on the following 16 variables.

`A` a numeric vector

`B` a numeric vector

`C` a numeric vector

`D` a numeric vector

`E` a numeric vector

`F` a numeric vector

`G` a numeric vector

`Pof` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(prodstd)
```

qsar *Library of substituted hydroxyphenylurea compounds*

Description

Data from the Library of substituted hydroxyphenylurea compounds in chapter 10 of Design and Analysis of Experiments with R (compact format)

Usage

```
data(qsar)
```

Format

A data frame with 36 observations on the following 4 variables.

`Compound` a numeric vector

`HE` a numeric vector

`DMz` a numeric vector

`S0K` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(qsar)
```

rcb *generalized RCB golf driving experiment*

Description

Data from the generalized RCB golf driving experiment in chapter 4 of Design and Analysis of Experiments with R

Usage

```
data(rcb)
```

Format

A data frame with 135 observations on the following 3 variables.

`id` a factor with levels `1 2 3 4 5 6 7 8 9`

`teehgt` a factor with levels `1 2 3`

`cdistance` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(rcb)
```

`residue` *Herbicide degradation experiment*

Description

Data from the Herbicide degradation experiment in chapter 9 of Design and Analysis of Experiments with R

Usage

```
data(residue)
```

Format

A data frame with 16 observations on the following 3 variables.

`soil` a factor with levels `"C" "P"`

`moisture` a factor with levels `"L" "H"`

`temp` a factor with levels `10 30`

`X1` a numeric vector

`X2` a numeric vector

`X3` a numeric vector

`X4` a numeric vector

`X5` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(residue)
```

`rubber` *Rubber Elasticity data*

Description

Data from the Rubber Elasticity Study in chapter 5 of Design and Analysis of Experiments with R

Usage

```
data(rubber)
```

Format

A data frame with 96 observations on the following 4 variables.

`supplier` a factor with levels `A` `B` `C` `D`

`batch` a factor with levels `I` `II` `III` `IV`

`sample` a factor with levels `1` `2`

`elasticity` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(rubber)
```

`sausage` *Split-plot experiment on sausage casing with RCB in whole plot*

Description

Data from the Split-plot experiment on sausage casing with RCB in whole plot in chapter 7 of Design and Analysis of Experiments with R

Usage

```
data(sausage)
```

Format

A data frame with 32 observations on the following 5 variables.

`Block` a factor with levels `1 2`

`Gbatch` a factor with levels `1 2 3 4`

`A` a factor with levels `-1 1`

`B` a factor with levels `-1 1`

`C` a factor with levels `-1 1`

`D` a factor with levels `-1 1`

`ys` a numeric vector containing the response

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(sausage)
```

`Smotor` *Single array for starting motor experiment*

Description

Data from the single array for starting motor experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(Smotor)
```

Format

A data frame with 18 observations on the following 6 variables.

`A` a factor with levels `1 2`

`B` a factor with levels `1 2 3`

`C` a factor with levels `1 2 3`

`D` a factor with levels `1 2 3`

`E` a factor with levels `1 2`

`torque` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(Smotor)
```

---

soup *dry mix soup experiment*

Description

Data from the dry mix soup experiment in chapter 6 of Design and Analysis of Experiments with R

Usage

```
data(soup)
```

Format

A data frame with 16 observations on the following 6 variables.

- `A` a factor with levels `-1` `1`
- `B` a factor with levels `-1` `1`
- `C` a factor with levels `-1` `1`
- `D` a factor with levels `-1` `1`
- `E` a factor with levels `-1` `1`
- `y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(soup)
```

`soupmx` *dry soup mix variance component study*

Description

Data from the dry soup mix variance component study of Chapter 5 in Design and Analysis of Experiments with R

Usage

```
data(soupmx)
```

Format

A data frame with 12 observations on the following 2 variables.

`batch` a factor with levels `1 2 3 4`

`weight` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(soupmx)
```

`splitPdes` *Split-plot cookie baking experiment*

Description

Data from the Split-plot cookie baking experiment in chapter 8 of Design and Analysis of Experiments with R

Usage

```
data(splitPdes)
```

Format

A data frame with 24 observations on the following 5 variables.

`short` a factor with levels `100 80`

`trayT` a factor with levels `RoomT Hot`

`bakeT` a factor with levels `low mid high`

`batch` a factor with levels `1 2`

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(splitPdes)
```

`SPMPV` *Split-plot mixture process variable experiment with vinyl*

Description

Data from the Split-plot mixture process variable experiment with vinyl in chapter 10 of Design and Analysis of Experiments with R

Usage

```
data(SPMPV)
```

Format

A data frame with 28 observations on the following 7 variables.

- `wp` a factor with levels `1` `2` `3` `4` `5` `6` `7`
- `z1` a numeric vector
- `z2` a numeric vector
- `x1` a numeric vector
- `x2` a numeric vector
- `x3` a numeric vector
- `y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(SPMPV)
```

strung *Repeated measures study with dairy cow diets*

Description

Data from the Repeated measures study with dairy cow diets in chapter 9 of Design and Analysis of Experiments with R (strung out format)

Usage

```
data(strung)
```

Format

A data frame with 120 observations on the following 5 variables.

`Diet` a factor with levels `"Barley"` `"Mixed"` `"Lupins"`

`Cow` a factor with levels `1 2 3 4 5 6 7 8 9 10`

`week` a factor with levels `1 2 3 4`

`protein` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(strung)
```

strungtile *Strung out control factor array and raw response data for Ina tile experiment*

Description

Data from the strung out control factor array and raw response data for Ina tile experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(strungtile)
```

Format

A data frame with 16 observations on the following 16 variables.

`A` a numeric vector
`B` a numeric vector
`C` a numeric vector
`D` a numeric vector
`E` a numeric vector
`F` a numeric vector
`G` a numeric vector
`H` a numeric vector
`AH` a numeric vector
`BH` a numeric vector
`CH` a numeric vector
`DH` a numeric vector
`EH` a numeric vector
`FH` a numeric vector
`GH` a numeric vector
`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(strungtile)
```

`sugarbeet` *Sugarbeet data from Chapter 2*

Description

Sugarbeet data from chapter 2 of Design and Analysis of Experiments with R

Usage

```
data(sugarbeet)
```

Format

A data frame with 18 observations on the following 2 variables.

`treat` a factor with levels `A` `B` `C` `D`
`yield` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(sugarbeet)
```

`taste` *taste test panel experiment*

Description

Data from the taste test panel experiment in Chapter 7 of Design and Analysis of Experiments with R

Usage

```
data(taste)
```

Format

A data frame with 24 observations on the following 3 variables.

`panelist` a factor with levels `1 2 3 4 5 6 7 8 9 10 11 12`

`recipe` a factor with levels `"A" "B" "C" "D"`

`score` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(taste)
```

`teach` *Teaching experiment data from Chapter 2*

Description

Data from the teaching experiment in chapter 2 of Design and Analysis of Experiments with R

Usage

```
data(teach)
```

Format

A data frame with 30 observations on the following 4 variables.

`class` a numeric vector

`method` a factor with levels `1 2 3`

`score` a factor with levels `1 2 3 4 5`

`count` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(teach)
```

`Tet` *Tetracycline concentration in plasma*

Description

Data from the Tetracycline concentration in plasma study in chapter 10 of Design and Analysis of Experiments with R (compact format)

Usage

```
data(Tet)
```

Format

A data frame with 9 observations on the following 2 variables.

`Time` a numeric vector

`Conc` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(Tet)
```

`tile` *Control factor array and summary statistics for Ina tile experiment*

Description

Data from the control factor array and summary statistics for Ina tile experiment in chapter 12 of Design and Analysis of Experiments with R

Usage

```
data(tile)
```

Format

A data frame with 8 observations on the following 11 variables.

`A` a numeric vector

`B` a numeric vector

`C` a numeric vector

`D` a numeric vector

`E` a numeric vector

`F` a numeric vector

`G` a numeric vector

`y1` a numeric vector

`y2` a numeric vector

`ybar` a numeric vector

`lns2` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(tile)
```

Treb *Box-Behnken design for trebuchet experiment*

Description

Data from the Box-Behnken design for trebuchet experiment in chapter 10 of Design and Analysis of Experiments with R

Usage

```
data(Treb)
```

Format

A data frame with 15 observations on the following 4 variables.

`x1` a numeric vector

`x2` a numeric vector

`x3` a numeric vector

`y` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(Treb)
```

Tukey1df *This function performs Tukey's single degree of freedom test for interaction in an unreplicated two-factor design*

Description

This function performs Tukey's single degree of freedom test for interaction in an unreplicated two-factor design

Usage

```
Tukey1df(data)
```

Arguments

`data` input - this is a data frame with three variables, the first variable is a numeric response and next two variables are factors. There should be ab lines in the data frame where a is the number of levels of the first factor, and b is the number of levels of the second factor.

Author(s)

John Lawson

Examples

```
library(daewr)
data(virus)
Tukey1df(virus)

## The function is currently defined as
function (data)
{
    y <- data[, 1]
    Afactor <- data[, 2]
    Bfactor <- data[, 3]
    tst1 <- is.factor(Afactor)
    tst2 <- is.factor(Bfactor)
    tst3 <- is.numeric(y)
    if (tst1 & tst2 & tst3) {
        a <- nlevels(Afactor)
        b <- nlevels(Bfactor)
    }
    else {
        stop("The first column of the data frame is the numeric response,
the 2nd and 3rd columns should be coded as factors")
    }
    tst4 <- max(a, b) > 2
    tst5 <- length(y) == a * b
    if (tst4 & tst5) {
        ybb <- with(data, tapply(y, Bfactor, mean))
        yba <- with(data, tapply(y, Afactor, mean))
        sbb <- with(data, tapply(y, Bfactor, sum))
        sba <- with(data, tapply(y, Afactor, sum))
        ybardd <- mean(y)
        CT <- (sum(y)^2)/(a * b)
        ssA <- sum(sba^2/b) - CT
        ssB <- sum(sbb^2/a) - CT
        ssE <- sum(y^2) - CT - ssA - ssB
        ybdj <- rep(ybb, 6)
        prody <- y * ybdj
        sumprod <- tapply(prody, Afactor, sum)
        leftsum <- sum(sumprod * yba)
        ssAB <- (a * b * (leftsum - (ssA + ssB + a * b * ybardd^2) *
            ybardd)^2/(ssA * ssB))
        ssR <- ssE - ssAB
```

```
        F <- ssAB/(ssR/((a - 1) * (b - 1) - 1))
        Pval <- 1 - pf(F, 1, ((a - 1) * (b - 1) - 1))
        cat("Source           df     SS        MS        F     Pr>F",
            "\n")
        cat("A              ", paste(format(a - 1, width = 6),
            " ", format(round(ssA, 4), justify = "right"), "  ",
            format(round(ssA/(a - 1), 4), justify = "right"),
            "\n"), sep = "")
        cat("B              ", paste(format(b - 1, width = 6),
            " ", format(round(ssB, 4), justify = "right"), "  ",
            format(round(ssB/(b - 1), 4), justify = "right"),
            "\n"), sep = "")
        cat("Error          ", paste(format((b - 1) * (a - 1),
            width = 6), " ", format(round(ssE, 4), justify = "right"),
            "  ", format(round(ssE/(a - 1) * (b - 1), 4), justify = "right"),
            "\n"), sep = "")
        cat("NonAdditivity", paste(format(1, width = 6), " ",
            format(round(ssAB, 4), justify = "right"), "  ",
            format(round(ssAB, 4), justify = "right"), "  ",
            format(round(F, 2), justify = "right"), "  ", format(round(Pval,
                4), justify = "right"), "\n"), sep = "")
        cat("Residual     ", paste(format((b - 1) * (a - 1) -
            1, width = 6), " ", format(round(ssR, 4), justify = "right"),
            "  ", format(round(ssR/((a - 1) * (b - 1) - 1), 4),
                justify = "right"), "\n"), sep = "")
    }
    else {
        stop("This function only works for unreplicated 2-factor
factorials with >2 levels for one of the factors")
    }
  }
```

---

vci — *confidence limits for method of moments estimators of variance components*

Description

function for getting confidence intervals on variance components estimated by the method of moments

Usage

```
vci(confl,c1,ms1,nu1,c2,ms2,nu2)
```

Arguments

`confl`	input- confidence level
`c1`	input - linear combination coefficient of ms1 in the estimated variance component

`ms1`	input - Anova mean square 1
`nu1`	input - Anova degrees of freedom for mean square 1
`c2`	input - linear combination coefficient of ms2 in the estimated variance component
`ms2`	input - Anova mean square 2
`nu2`	input - Anova degrees of freedom for mean square 2

Value

returned delta, Lower and Upper limits

Author(s)

John Lawson

Examples

```
vci(.90,.05,.014852,2,.05,.026885,18)
## The function is currently defined as
vci<-function(confl,c1,ms1,nu1,c2,ms2,nu2){
  delta<-c1*ms1-c2*ms2
  alpha<-1-confl
  Falpha1<-qf(confl,nu1,10000000)
  Falpha12<-qf(confl,nu1,nu2)
  Fconf2<-qf(alpha,nu2,10000000)
  Fconf12<-qf(alpha,nu1,nu2)
  Falpha2<-qf(confl,nu2,10000000)
  Fconf1<-qf(alpha,nu1,10000000)
  Fconf12<-qf(alpha,nu1,nu2)
  G1<-1-(1/Falpha1)
  H2<-(1/Fconf2)-1
  G12<-((Falpha12-1)**2-G1**2*Falpha12**2-H2**2)/Falpha12
  VL<-G1**2*c1**2*ms1**2+H2**2*c2**2*ms2**2+G12*c1*c2*ms1*ms2
  H1<-(1/Fconf1)-1
  G2<-1-(1/Falpha2)
  H12<-((1-Fconf12)**2-H1**2*Fconf12**2-G2**2)/Fconf12
  VU<-H1**2*c1**2*ms1**2+G2**2*c2**2*ms2**2
  L<-delta-sqrt(VL)
  U<-delta+sqrt(VU)
  cat("delta=",delta," Lower Limit=",L," Upper Limit=",U,"\n")
}
```

`virus` *Assay of Viral Contamination experiment data from Chapter 3*

Description

Data from the Assay of Viral Contamination experiment in chapter 3 of Design and Analysis of Experiments with R

Usage

```
data(virus)
```

Format

A data frame with 18 observations on the following 3 variables.

`y` a numeric vector

`Sample` a factor with levels `1 2 3 4 5 6`

`Dilution` a factor with levels `3 4 5`

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(virus)
```

---

volt — *Volt meter experiment data from Chapter 3*

Description

Data from the Volt meter experiment in chapter 3 of Design and Analysis of Experiments with R

Usage

```
data(volt)
```

Format

A data frame with 16 observations on the following 3 variables.

`y` a numeric vector

`A` a factor containing the levels (22, 32) of factor A

`B` a factor containing the levels (0.5, 5.0) of factor B

`C` a factor containing the levels (0.5, 5.0) of factor C

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(volt)
```

`web` *Web page design experiment data from Chapter 3*

Description

Data from the web page design experiment in chapter 3 of Design and Analysis of Experiments with R

Usage

```
data(web)
```

Format

A data frame with 36 observations on the following 6 variables.

`A` a factor with levels `1` `2`

`B` a factor with levels `1` `2`

`C` a factor with levels `1` `2`

`D` a factor with levels `1` `2`

`visitors` a numeric vector

`signup` a numeric vector

Source

Design and Analysis of Experiments with R, by John Lawson, CRC/Chapman Hall

Examples

```
data(web)
```

Index

∗Topic **datagen**
Altscreen, 4
BIBsize, 11
DefScreen, 24
EEw1s1, 26
EEw1s2, 27
EEw1s3, 28
EEw2s1, 29
EEw2s2, 30
EEw2s3, 31
EEw3, 32
Fcrit, 36
Fpower1, 42
Fpower2, 43
mod, 60
ModelRobust, 61
OptPB, 63
PBDes, 64
∗Topic **datasets**
antifungal, 5
Apo, 6
apple, 7
arso, 7
augm, 8
Bdish, 9
Bff, 10
bha, 11
bioequiv, 13
bioeqv, 13
blood, 14
BoxM, 15
BPmonitor, 15
bread, 16
cake, 17
cement, 17
chem, 18
chipman, 19
COdata, 19
cont, 21
cpipe, 22
culture, 23
dairy, 24
drug, 25
eptaxr, 33
eptaxs2, 34
eptaxyb, 35
gagerr, 46
gear, 50
hardwood, 52
inject, 55
MPV, 62
Naph, 62
pastry, 64
pest, 65
pesticide, 66
plasma, 67
polvdat, 67
polymer, 68
prodstd, 69
qsar, 70
rcb, 70
residue, 71
rubber, 72
sausage, 72
Smotor, 73
soup, 74
soupmx, 75
splitPdes, 75
SPMPV, 76
strung, 77
strungtile, 77
sugarbeet, 78
taste, 79
teach, 80
Tet, 80
tile, 81
Treb, 82
virus, 85

volt, 86
web, 87
∗Topic **design**
LenthPlot, 55
∗Topic **hplot**
colormap, 20
fullnormal, 44
halfnorm, 51
∗Topic **htest**
fhstep, 36
fhstepDS, 37
firstm, 38
FitDefSc, 39
fnextrm, 39
fntrmDS, 40
gapstat, 47
Gaptest, 48
HierAFS, 53
ihstep, 54
LGB, 57
LGBc, 58
Tukey1df, 82
vci, 84
∗Topic **package**
daewr-package, 4

Altscreen, 4
antifungal, 5
Apo, 6
apple, 7
arso, 7
augm, 8

Bdish, 9
Bff, 10
bha, 11
BIBsize, 11
bioequiv, 13
bioeqv, 13
blood, 14
BoxM, 15
BPmonitor, 15
bread, 16

cake, 17
cement, 17
chem, 18
chipman, 19
COdata, 19
colormap, 20
cont, 21
cpipe, 22
culture, 23

daewr *(*daewr-package*)*, 4
daewr-package, 4
dairy, 24
DefScreen, 24
drug, 25

EEw1s1, 26
EEw1s2, 27
EEw1s3, 28
EEw2s1, 29
EEw2s2, 30
EEw2s3, 31
EEw3, 32
eptaxr, 33
eptaxs2, 34
eptaxyb, 35

Fcrit, 36
fhstep, 36
fhstepDS, 37
firstm, 38
FitDefSc, 39
fnextrm, 39
fntrmDS, 40
Fpower, 41
Fpower1, 42
Fpower2, 43
fullnormal, 44

gagerr, 46
gapstat, 47
Gaptest, 48
gear, 50

halfnorm, 51
hardwood, 52
HierAFS, 53

ihstep, 54
inject, 55

LenthPlot, 55
LGB, 57
LGBc, 58

mod, 60
ModelRobust, 61
MPV, 62

Naph, 62

OptPB, 63

pastry, 64
PBDes, 64
pest, 65
pesticide, 66
plasma, 67
polvdat, 67
polymer, 68
prodstd, 69

qsar, 70

rcb, 70
residue, 71
rubber, 72

sausage, 72
Smotor, 73
soup, 74
soupmx, 75
splitPdes, 75
SPMPV, 76
strung, 77
strungtile, 77
sugarbeet, 78

taste, 79
teach, 80
Tet, 80
tile, 81
Treb, 82
Tukey1df, 82

vci, 84
virus, 85
volt, 86

web, 87