Package ‘mixexp’

June 2, 2011

Type Package

Title Design and analysis of mixture experiments

Version 0.5-1

Date 2011-5-24

Author John Lawson <lawson@byu.edu>

Maintainer John Lawson <lawson@byu.edu>

Description This package contains functions for creating designs for mixture experiments, making ternary contour plots, and making mixture effect plots.

License GPL2.0

Depends gdata, lattice, grid

LazyLoad yes

R topics documented:

`mixexp-package` *This package contains functions for creating designs for mixture experiments and making graphical display of results of mixture experiments.*

Description

The **mixexp** package provides functions for creating mixture experiment designs in an unconstrained simplex or constrained mixture space. Functions are also provided for making ternary contour plots, pictures of constrained regions, design points, and mixture effect plots.

Details

Package:	mixexp
Type:	Package
Version:	1.0-1
Date:	2011-05-31
License:	GPL2.0
Dependencies: gdata, lattice, grid	
LazyLoad:	yes
Packaged:	2011-05-31 19:54:07 UTC; Lawson
Built:	R 2.12.1; i386-pc-mingw32; 2011-05-31 19:54:08 UTC; windows

Index:

```
conmx                   example constraint matrix used as input to function
                        crvtave
crvtave                 function for creating extreme vertices designs and
                        centroids; this function calls Eflags, Nrows, and
Vertcen
DesignPoints            function for plotting design points and or mixture
                        constraint in the simplex
Eflags                  function for calling Pieple's fortran code cnvrt
                        to create extreme vertices designs and prints any
                        error messages
Effplot                 function for making mixture effect plots
MixturePlot             function for making contour plots in simplex region
Nrows                   function for calling Pieple's fortran code cnvrt to
                        create extreme vertices designs and returns the
                        number of rows in the resulting design
SCD                     function for creating Simplex Centroid Designs
SLD                     function for creating Simplex Lattice Designs
Vertcen                 function for calling Pieple's fortran code cnvrt
                        to create extreme vertices designs and returns the
                        resulting design
Xvert                   function for creating extreme vertices design
                        and centroids, this function calls crvtave
```

Author(s)

John Lawson <lawson@byu.edu>

Maintainer: John Lawson <lawson@byu.edu>

`conmx` *Example constraint matrix from Pieple 1988*

Description

This is an .rda file containing the constraint matrix.

Usage

```
data(conmx)
```

Format

An 8 x 4 matrix

Source

source

References

Pieple, G. F. (1988) Programs for Generating Extreme Vertices and Centroids of Linearly Constrained Experimental Regions, *Journal of Quality Technology*, Vol. 20, No. 2.

`crvtave` *This function creates an extreme vertices design*

Description

This function calls the function Vertcen which uses Pieple's (1988) fortran code (cnvrt) for generating extreme vertices and centroids of linearly constrained mixture experimental regions.

Usage

```
crvtave(ndm,conmx)
```

Arguments

`ndm`	is an integer representing the highest order of centroids requested. An overall centroid is always included, 0 indicates no other centroids will be created, 1 indicates edge centroids are requested, 2 indicates face centroids, etc.
`conmx`	This is the matrix of constraints.

Value

vtcn	This is a data frame containing the extreme vertices design. The columns are labeled x1, x2 ...xn, where n is the number of mixture variables. The last column is labeled dimen and it indicates the order of centroid where 0 is an extreme vertex, 1 is an edge centroid, 2 is a face centroid, and n is the overall centroid.

Note

This function calls the function Eflags to get error messages from cnvrt, the function Vertcen to get the extreme vertices and centroids from cnvrt, and the function Nrows to get the number of vertices and centroids from cnvrt.

Author(s)

John S. Lawson <lawson@byu.edu>

References

1. Pieple, G. F. "Programs for Generating Extreme Vertices and Centroids of Linearly Constrained Experimental Regions" *Journal of Quality Technology*, Vol 20, No. 2, pp. 125-139, 1988.

Examples

```
data(conmx)
crvtave(1,conmx)
```

DesignPoints — *This function plots design points and or constraints in the simplex mixture space.*

Description

This function plots design points and or constraints in the simplex mixture space. It calls the function MixturePlot that does the actual plotting.

Usage

```
DesignPoints(des,x,y,z,x1lower,x1upper,x2lower,x2upper,x3lower,x3upper)
```

Arguments

des	data frame containing x1 x2 and x3 coordinates of data points to be plotted
x	vector of x3 coordinates of design points to be plotted
y	vector of x2 coordinates of design points to be plotted
z	vector of x1 coordinates of design points to be plotted
x1lower	This is the lower constraint on x1
x1upper	This is the upper constraint on x1

x2lower	This is the lower constraint on x2
x2upper	This is the upper constraint on x2
x3lower	This is the lower constraint on x3
x3upper	This is the upper constraint on x3

Note

This function calls MixturePlot. If either des and x,y,z are missing no design points will be plotted, and if x1lower, x1upper, etc. are all zero no constraints will be plotted.

Author(s)

John S. Lawson <lawson@byu.edu>

References

1. Pieple, G. F. "Programs for Generating Extreme Vertices and Centroids of Linearly Consrtained Experimental Regions" *Journal of Quality Technology*, Vol 20, No. 2, pp. 125-139, 1988.

Examples

```
dat<-SCD(3)
DesignPoints(des=dat)

x1<-c(1,0,0,.5,.5, 0,.33333)
x2<-c(0,1,0,.5,0,.5,.33333)
x3<-c(0,0,1,0,.5,.5,.33333)
DesignPoints(x=x3,y=x2,z=x1)

dat<-data.frame(x1,x2,x3)
DesignPoints(des=dat)

DesignPoints(x1lower=0,x1upper=.8,x2lower=.10,x2upper=.95,x3lower=.05,x3upper=.50)
```

Effplot   *This function creates mixture effect plots*

Description

This function makes effect plots using the Cox or Pieple directions in constrained mixture space.

Usage

```
Effplot(des,mod,dir)
```

Arguments

`des` data frame containing the design points and response data for a mixture experiment. The data frame must contain the variables x1, x2 ...xn for the mixture variables, and y for the response. n must be between 2 and 12. Only effect plots for linear models can be made when the number of factors is greater than 6.

`mod` an interger representing the model to be traced: 1 for a linear model, 2 for a quadratic model, and 3 for a special cubic model

`dir` an interger representing the direction for which the effect plot is made: 1 for Cox direction, 2 for Pieple direction.

Value

`PX` This is a matrix containing the coordinates of the effect plot traces that are plotted.

Note

This function calls the function crvtave to get the design centroid from cnvrt.

Author(s)

John S. Lawson `<lawson@byu.edu>`

References

1. Pieple, G. F. "Measuring Component Effects in Constrained Mixture Experiments" *Technometrics*, Vol 25, pp. 97-105, 1982.

Examples

```
#Example from Li, Tolley, Lee(2010) response is perm
x1<-c(.572,.358,.286,.286,.286,.143,.357)
x2<-c(.214,.428,.500,.357,.214,.500,.500)
x3<-c(.214,.214,.214,.357,.500,.357,.143)
y<-c(7.7,18.4,24.2,9.8,5.9,23.0,19.4)
des<-data.frame(x1,x2,x3,y)
Effplot(des,2,2)


#Example from Snee, Marquart(1976)
x1<-c(.1,.1,.1,.15,.1,.1,.1,.4,.35,.30,.1,.45,.45,.45,.45,.45,.259,.259,.259,.259)
x2<-c(.5,.05,.5,.05,.05,.5,.05,.05,.05,.5,.5,.05,.2,.15,.25,.1,.222,.222,.222,.222)
x3<-c(0,0,0,0,.1,.1,.1,.1,.1,0,.1,0,0,0,.1,.1,.05,.05,.05,.05)
x4<-c(0,0,.1,.1,0,.1,.1,.1,.1,0,0,0,.1,.1,0,0,.05,.05,.05,.05)
x5<-c(.1,.55,.1,.6,.55,.1,.55,.1,.1,.1,.2,.45,.1,.1,.1,.1,.244,.244,.244,.244)
x6<-c(.2,.2,.2,.05,.2,.05,.05,.2,.2,.05,.05,.05,.05,.2,.05,.2,.125,.125,.125,.125)
x7<-c(.05,.05,0,.05,0,0,0,.05,.05,0,.05,0,.05,0,.05,0,.025,.025,.025,.025)
x8<-c(.05,.05,0,0,0,.05,.05,0,.05,.05,0,0,.05,0,0,.05,.025,.025,.025,.025)
y<-c(30,113,17,94,89,18,90,20,21,15,28,48,18,7,16,19,38,30,35,40)
des<-data.frame(x1,x2,x3,x4,x5,x6,x7,x8,y)
```

```
Effplot(des,1,2)


# Weed control example from Lawson & Erjavec
x1<-c(1,0,0,.5,.5,0,.33333,.33333,.33333)
x2<-c(0,1,0,.5,0,.5,.33333,.33333,.33333)
x3<-c(0,0,1,0,.5,.5,.33333,.33333,.33333)
y<-c(73,68,80,77,86,75,92,93,88)
des<-data.frame(x1,x2,x3,y)
Effplot(des,3)



# Polvoron Example from Lawson
des<-Xvert(x1=c(0,.8),x2=c(.10,.95),x3=c(.05,.50),ndm=1)
dat<-as.matrix(des)
# remove the edge centroid at the top
dat<-dat[c(1:6,8:11), ]
# add two more centroids
dat<-rbind(dat,dat[10, ],dat[10,])
# response vector
y<-c(5.75,3.69,5.33,5.68,3.85,3.83,5.88,5.87,5.23,6.54,6.82,6.41)
# make the data frame for plotting
des<-data.frame(dat[,1:3],y)
Effplot(des,3)
```

`Eflags` *Loads compiled fortran in shared file cnvrt and returs the error messages*

Description

This function loads and runs the compiled fortran code cnvrt and prints error messages. cnvrt is Pieple's 1988 JQT fortran code for extreme vertices designs.

Usage

```
Eflags(ndm,nvrr,ncon2,rtheta2)
```

Arguments

`ndm`	This is the order of centroids desired (0=none, 1=edge centroids, 2=face centroids etc.)
`nvrr`	This is the number of mixture variables (maximum is 12)
`ncon2`	This is the number of constraints (maximum is 45)
`rtheta2`	This is the constraint matrix stored as a vector of columns.

Value

`ifa`	This is the vector of error flags. A negative value for flag 1 indicates that there are inconsistent constraints. A negative value for flag2 indicates there are two many vertices and centroids, this program only works when # vertices + # centroids <=1000. A negative value for flag 3 indicates an error encountered when calling subroutine allnr.

Note

This function is called by the function crtave.

Author(s)

John S. Lawson `<lawson@byu.edu>`

References

1. Pieple, G. F. "Programs for Generating Extreme Vertices and Centroids of Linearly Consrtained Experimental Regions" *Journal of Quality Technology*, Vol 20, No. 2, pp. 125-139, 1988.

`MixturePlot` *This function makes contour plots in the simplex mixture space.*

Description

This function makes contour plots in the simplex mixture space, it also can draw constraint lines and show design points.

Usage

```
MixturePlot(x,y,z,w,des,n.breaks,res,lims,color.palette,constrts,
            contrs,cols,despts,mod,x3lab,x2lab,x1lab,corner.labs)
```

Arguments

`x`	x3 locations for known points
`y`	x2 locations for known points
`z`	x1 locations for known points
`w`	y locations for known points
`des`	data frame with x1,x2,x3, and y locations for known points
`n.breaks`	number of breaks between levels
`res`	number of color blocks between 0 and 1 of x
`lims`	vector of lower and upper constraints for x1,x2,x3
`color.palette`	is the color palette to use
`constrts`	if TRUE constraints found in lines will be added to the graph
`contrs`	if TRUE contour lines will be added to the graph
`cols`	if TRUE regions between contour lines will be colored

`despts`	if TRUE plots the design points in data frame des
`mod`	is an indicator for the model 1=linear, 2=quadratic, 3=special cubic
`x3lab`	label for the x3 axis
`x2lab`	label for the x2 axis
`x1lab`	label for the x1 axis
`corner.labs`	labels for x3, x2 and x1 vertices

Author(s)

John S. Lawson <lawson@byu.edu>

References

1. Cornell, J. A. *Experiments with Mixtures: Models and Analysis of Mixture Data*, John Wiley & Sons, New York, third edition, 2002.

2. See R Ternary Level Plot Function http://www.siftp.net/index.shtml

Examples

```
##Usage and Examples - Example from page 458 DAE with SAS
dat = data.frame(
          "x1"=c(1,.8,.6,.5,.5,.33333,.3,.3,.1,.1,0,0,0),
          "x2"=c(0,.1,.2,0,.5,.33333,.2,.5,.1,.8,0,.5,1),
          "x3"=c(0,.1,.2,.5,0,.33333,.5,.2,.8,.1,1.0,.5,0),
          "y"=c(48.7,49.5,50.2,52.8,49.3,51.1,52.7,50.3,60.7,49.9,64.9,53.5,50.6)
                 )
MixturePlot(dat$x3,dat$x2,dat$x1,dat$y, x3lab="Fraction x3",
  x2lab="Fraction x2", x1lab="Fraction x1", corner.labs=c("x3","x2","x1"),
  constrts=FALSE,contrs=TRUE,cols=TRUE, mod=2,n.breaks=9)

# Weed control example from Lawson & Erjavec
x1<-c(1,0,0,.5,.5,0,.33333,.33333,.33333)
x2<-c(0,1,0,.5,0,.5,.33333,.33333,.33333)
x3<-c(0,0,1,0,.5,.5,.33333,.33333,.33333)
y<-c(73,68,80,77,86,75,92,93,88)
des<-data.frame(x1,x2,x3,y)
MixturePlot(des=des,x3lab="Fraction C",x2lab="Fraction B",
   x1lab="Fraction A",corner.labs=c("C","B","A"),mod=3,n.breaks=5,cols=TRUE)
```

`Nrows` *Loads compiled fortran in shared file cnvrt and returns the number of rows in the resulting design*

Description

This function loads and runs the compiled fortran code cnvrt. cnvrt is Pieple's 1988 JQT fortran code for extreme vertices designs.

Usage

```
Nrows(ndm,nvrr,ncon2,rtheta2)
```

Arguments

`ndm`	This is the order of centroids desired (0=none, 1=edge centroids, 2=face centroids etc.)
`nvrr`	This is the number of mixture variables (maximum is 12)
`ncon2`	This is the number of constraints (maximum is 45)
`rtheta2`	This is the constraint matrix stored as a vector of columns.

Value

nvrtr

`nvrtr`	This is the number of rows in rxvt the matrix of extreme vertices and centroids

Note

This function is called by the function crtave.

Author(s)

John S. Lawson `<lawson@byu.edu>`

References

1. Pieple, G. F. "Programs for Generating Extreme Vertices and Centroids of Linearly Consrtained Experimental Regions" *Journal of Quality Technology*, Vol 20, No. 2, pp. 125-139, 1988.

`SCD` *This function creates simplex centroid mixture designs*

Description

This function creates simplex centroid designs in unconstrained mixture experiment space.

Usage

```
SCD(fac)
```

Arguments

`fac`	This is the number of factors

Value

`SC`	This is a data frame containing the simplex centroid design. The columns are labeled x1, x2 ...xn, where n is the number of mixture variables.

Author(s)

John S. Lawson <lawson@byu.edu>

References

1. Cornell, J. A. *Experiments with Mixtures: Models and Analysis of Mixture Data*, John Wiley & Sons, New York, third edition, 2002.

Examples

```
SCD(3)

des<-SCD(5)

des<-SCD(12)
```

SLD *This function creates simplex lattice mixture designs*

Description

This function creates simplex lattice designs in unconstrained mixture experiment space.

Usage

```
SLD(fac, lev)
```

Arguments

fac	This is the number of factors, this must be between 2 and 12
lev	This is the number of levels

Value

SL	This is a data frame containing the simplex lattice design. The columns are labeled x1, x2 ...xn, where n is the number of mixture variables.

Author(s)

John S. Lawson <lawson@byu.edu>

References

1. Cornell, J. A. *Experiments with Mixtures: Models and Analysis of Mixture Data*, John Wiley \& Sons, New York, third edition, 2002.

Examples

```
des<-SLD(3,2)

des<-SLD(4,3)
```

`Vertcen` *Loads compiled fortran in shared file cnvrt*

Description

This function loads and runs the compiled fortran code cnvrt. cnvrt is Pieple's 1988 JQT fortran code for extreme vertices designs.

Usage

```
Vertcen(ndm,nvrr,ncon2,rtheta2)
```

Arguments

`ndm`	This is the order of centroids desired (0=none, 1=edge centroids, 2=face centroids etc.)
`nvrr`	This is the number of mixture variables (maximum is 12)
`ncon2`	This is the number of constraints (maximum is 45)
`rtheta2`	This is the constraint matrix stored as a vector of columns.

Value

`rxvt`	This is the matrix of vertices and centroids stored as a vector of columns.

Note

This function is called by the function crtave.

Author(s)

John S. Lawson `<lawson@byu.edu>`

References

1. Pieple, G. F. "Programs for Generating Extreme Vertices and Centroids of Linearly Consrtained Experimental Regions" *Journal of Quality Technology*, Vol 20, No. 2, pp. 125-139, 1988.

`Xvert` *This function creates an extreme vertices design in a constrained mixture space.*

Description

This function calls the function crvtave to create an extreme vertices design in a constrained mixture space. If there are only three factors the function DesignPoints is called to plot the results.

Usage

```
Xvert(x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,ndm)
```

Arguments

`x1`	a vector containing lower and upper constraints on x1
`x2`	a vector containing lower and upper constraints on x2
`x3`	a vector containing lower and upper constraints on x3
`x4`	a vector containing lower and upper constraints on x4
`x5`	a vector containing lower and upper constraints on x5
`x6`	a vector containing lower and upper constraints on x6
`x7`	a vector containing lower and upper constraints on x7
`x8`	a vector containing lower and upper constraints on x8
`x9`	a vector containing lower and upper constraints on x9
`x10`	a vector containing lower and upper constraints on x10
`x11`	a vector containing lower and upper constraints on x11
`x12`	a vector containing lower and upper constraints on x12
`ndm`	is an integer representing the highest order of centroids requested. An overall centroid is always included, 0 indicates no other centroids will be created, 1 indicates edge centroids are requested, etc.

Note

This function calls crvtave. If the number of factors is 3, the function DesignPoints is called to graph the results.

Author(s)

John S. Lawson `<lawson@byu.edu>`

References

1. Pieple, G. F. "Programs for Generating Extreme Vertices and Centroids of Linearly Consrtained Experimental Regions" *Journal of Quality Technology*, Vol 20, No. 2, pp. 125-139, 1988.

Examples

```
# Polvoron Example from Lawson
des<-Xvert(x1=c(0,.8),x2=c(.10,.95),x3=c(.05,.50),ndm=1)

# Exercise 11.3 DAE with SAS
Xvert(x1=c(.2,.8),x2=c(.1,.4),x3=c(.1,.5))

#Snee Marquardt(1976) example
Xvert(x1=c(.1,.45),x2=c(.05,.50),x3=c(0,.10),x4=c(0,.4),x5=c(.1,.6),x6=c(.05,.2),
      x7=c(0,.05),x8=c(0,.05),ndm=0)
```

Index