Package ‘CompExpDes’

November 4, 2024

Type Package

Title Computer Experiment Designs

Version 1.0.5

Maintainer Ashutosh Dalal <ashutosh.dalal97@gmail.com>

Description In computer experiments space-filling designs are having great impact. Most popularly used space-filling designs are Uniform designs (UDs), Latin hypercube designs (LHDs) etc. For further references one can see Mckay (1979) <DOI:10.1080/00401706.1979.10489755> and Fang (1980) <https://cir.nii.ac.jp/crid/1570291225616774784>. In this package, we have provided algorithms for generate efficient LHDs and UDs. Here, generated LHDs are efficient as they possess lower value of Maxpro measure, Phi_p value and Maximum Absolute Correlation (MAC) value based on the weightage given to each criterion. On the other hand, the produced UDs are having good space-filling property as they always attain the lower bound of Discrete Discrepancy measure. Further, some useful functions added in this package for adding more value to this package.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Ashutosh Dalal [aut, cre],
Cini Varghese [aut, ctb],
Rajender Parsad [aut, ctb],
Mohd Harun [aut, ctb]

Repository CRAN

Date/Publication 2024-11-04 12:50:02 UTC

Contents

`Best_Model` *Find Best Model*

Description

This function will try to find out a significant model for each combinations based on adjusted R^2. Then user need to select which model they want to use.

Usage

```
Best_Model(model, data)
```

Arguments

`model`	Provide a vector that contains all the individual terms present in a full model
`data`	Provide data in a matrix or data frame format where you want to fit the model

Value

Generate a list of significant models for various combinations of factors.

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

Examples

```
## Not run:
library(CompExpDes)
# Sample data
data <- data.frame(
  x1 = c(1.0, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8, 4.2, 4.6, 5.0, 5.4),
  x2 = c(50, 25, 5, 30, 55, 45, 20, 10, 35, 60, 40, 15),
  x3 = c(2.5, 6.0, 4.0, 1.0, 5.5, 4.5, 3.0, 2.0, 6.5, 3.5, 1.5, 5.0),
  x4 = c(45, 25, 55, 35, 65, 15, 70, 20, 50, 30, 60, 40),
  y = c(0.0795, 0.0118, 0.0109, 0.0991, 0.1266, 0.0717, 0.1319, 0.0900, 0.1739,
  0.1176, 0.1836, 0.1424)
)
# List of terms in the polynomial model
model <- list('x1', 'x2', 'x3', 'x4', 'x1:x2', 'x1:x3', 'x1:x4',
```

```
                    'x2:x3', 'x2:x4', 'x3:x4', 'I(x1^2)',
                    'I(x2^2)', 'I(x3^2)', 'I(x4^2)')

Best_Model(model,data)

## End(Not run)
```

Discrete_Discrepancy *Measure of Discrete Discrepancy*

Description

Discrete Discrepancy is a measure of uniformity for any uniform design. Lesser the value of Discrete Discrepancy measure, better is the uniform design.

Usage

```
Discrete_Discrepancy(Design,a,b)
```

Arguments

Design	A matrix
a	Any value a>b>0. By default it is set to 1.
b	Any value a>b>0. By default it is set to 0.5.

Value

The function calculates the value of Discrete Discrepancy measure and its lower bound for a given design.

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

References

Qin H, Fang KT (2004)<DOI:10.1007/s001840300296> Discrete discrepancy in factorial designs. Metrika, 60, 59-72.

Examples

```
library(CompExpDes)
lhd1<-matrix(c(1,5,7,3,4,2,6,2,1,4,5,3,7,6,4,5,6,1,2,3,7),nrow=7,ncol=3,byrow=FALSE)
lhd2<-cbind(lhd1[,3],lhd1[,2],lhd1[,1])
ud<-rbind(lhd1,lhd2)
Discrete_Discrepancy(ud, 1, 0.5)
```

LHDs_I *Latin Hypercube Designs (LHDs) for Prime Numbers*

Description

For prime number of factors, F (>2), this method will generate LHDs with levels, L ranges from F+2 to F^2. Maxpro criterion measure, Phi_p measure also provided as a measure of space-filling and also as an orthogonality measure maximum absolute correlation (MAC) value also provided.

Usage

```
LHDs_I(levels,factors,weight,iterations)
```

Arguments

`levels`	L,ranges from (F+2) to F^2
`factors`	A prime number, F (>2)
`weight`	Weight should be given to Maxpro, Phi_p and MAC such that sum is 1. Default it is 0.3, 0.3 and 0.4
`iterations`	Number of iterations. By default it is 400.

Value

This function will provide a series of LHDs along with space-filling and orthogonality measures for the generated LHDs.

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

References

McKay, M.D., Beckman, R.J. and Conover, W.J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2), 239-245.

Examples

```
## Not run:
library(CompExpDes)
LHDs_I(9,3,c(0.6,0,0.4))

## End(Not run)
```

LHDs_II	*Latin Hypercube Designs (LHDs) for Any Numbers of Factors*

Description

For any even number of factors, F (>=4), this method will generate LHDs with levels, L ranges from F+2 to sC2, where s=2*F+1. Maxpro criterion measure, Phi_p measure also provided as a measure of space-filling and as an orthogonality measure, maximum absolute correlation (MAC) value also provided.

Usage

```
LHDs_II(levels,factors,weight,iterations)
```

Arguments

`levels`	Ranges from (F+2) to sC2, where s=2*F+1
`factors`	F, any even number (>=4)
`weight`	Weight should be given to Maxpro, Phi_p and MAC such that sum is 1. Default it is 0.3, 0.3 and 0.4
`iterations`	Number of iterations. By default it is 400.

Value

This function will provide a series of LHDs along with space-filling and orthogonality measures for the generated LHDs.

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

References

McKay, M.D., Beckman, R.J. and Conover, W.J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2), 239-245.

Examples

```
## Not run:
library(CompExpDes)
LHDs_II(20,3,c(0.4,0.2,0.4))

## End(Not run)
```

MAC — *Maximum Absolute Correlation*

Description

Maximum Absolute Correlation (MAC) is the maximum absolute value among off diagonal values of a correlation matrix.

Usage

```
MAC(matrix)
```

Arguments

`matrix`	Input a matrix

Value

It returns a maximum absolute correlation value for a given matrix.

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

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, 43(1), 1-15.

Examples

```
library(CompExpDes)
lhd<-matrix(c(1,5,7,3,4,2,6,2,1,4,5,3,7,6,4,5,6,1,2,3,7),nrow=7,ncol=3,byrow=FALSE)
MAC(lhd)
```

Maxpro_Measure — *Measure of Maxpro criterion*

Description

This function calculates Maxpro criterion for a given space-filling design. Lesser the value of it better the design, in the sense that the design has maximum spread in higher dimensional spaces.

Usage

```
Maxpro_Measure(Design)
```

Arguments

`Design` Provide a design in a matrix format

Value

Provides Maxpro criterion value given by Joseph et al. (2015).

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

References

Joseph, V.R., Gul, E. and Ba, S. (2015). Maximum projection designs for computer experiments. Biometrika, 102 (2), 371-380.

Examples

```
library(CompExpDes)
lhd<-matrix(c(1,5,7,3,4,2,6,2,1,4,5,3,7,6,4,5,6,1,2,3,7),nrow=7,ncol=3,byrow=FALSE)
Maxpro_Measure(lhd)
```

`Meeting_Number` *Maximum Coincidence (or Meeting) numbers between rows*

Description

Finding out Maximum coincidence (or Meeting) number between unique pair of rows.

Usage

```
Meeting_Number(matrix)
```

Arguments

`matrix` Provide any matrix

Value

This function provides the maximum coincidence number between any pair of rows of for a given matrix.

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

Examples

```
library(CompExpDes)
  mat<-matrix(c(1,2,3,3,2,1,4,2,1),nrow=3,byrow=TRUE)
  Meeting_Number(mat)
```

PhipMeasure *Phi_p criterion*

Description

For a given design Phi_p criterion (Morris and Mitchell, 1995) is calculated using this function. Lesser the value of Phi_p criterion better the design in terms of space-filling.

Usage

```
PhipMeasure(design,p=15,q=2)
```

Arguments

`design`	A design matrix is needed
`p`	Any positive integer. Default value of p = 15.
`q`	Any positive integer. Default value of q = 2. This implies that we are considering here Euclidean distance.

Value

Generates Phi_p criterion value

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

References

Morris, M.D. and Mitchell, T.J. (1995). Exploratory designs for computer experiments. Journal of Statistical Planning and Inference, 43, 38-402.

Examples

```
library(CompExpDes)
lhd<-matrix(c(1,5,7,3,4,2,6,2,1,4,5,3,7,6,4,5,6,1,2,3,7),nrow=7,ncol=3,byrow=FALSE)
PhipMeasure(lhd,p=15,q=2)
```

UDesigns_I *Orthogonal Uniform Designs with Two Factors*

Description

This series of UDs can be obtained for a composite number of levels, L with always two factors, F. Further, "Excellent" type UDs are Excellent in space-filling with larger number of runs available for L >= 6. On the other hand, "Good" type UDs are good in space-filling and lesser the number of runs, available for L >= 9. Generated designs are UDs under discrete discrepancy measure, as all designs will attain the lower bound value of discrete discrepancy.

Usage

```
UDesigns_I(levels, type)
```

Arguments

`levels`	Any composite number >=6 (if "Excellent") or >=9 (if "Good")
`type`	"Excellent" or "Good"

Details

Type "Excellent" or type "Good" both can exist for a same parameter range. For type "Excellent" it will require more runs than designs generated by type "Good". But type "Excellent" provides designs which are having more spread than type "Good" series designs.

Value

Returns a uniform designs along with number of factors, levels, runs, maximum absolute correlation (MAC) value and discrete discrepancy measure along with its lower bound value.

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

References

Fang, K.T. (1980). The uniform design: application of number-theoretic methods in experimental design. Acta Math Appl Sin, 3, 363-372.

Examples

```
library(CompExpDes)
UDesigns_I(levels=6, type="Excellent")
```

UDesigns_II *Uniform Designs with Multiple Factors with Minimal Runs*

Description

In this series, the Uniform Designs (UDs) are high dimensional with lesser number of runs will always attain lower bound of discrete discrepancy. They are available for any even number of factors, F (>= 4) with F(F+1) levels each.

Usage

```
UDesigns_II(factors)
```

Arguments

`factors`	any even number >=4

Value

Returns a series of high dimensional UDs along with number of factors, levels, runs, MAC value and discrete discrepancy measure along with its lower bound value.

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

References

Fang, K.T. (1980). The uniform design: application of number-theoretic methods in experimental design. Acta Math Appl Sin, 3, 363-372.

Examples

```
library(CompExpDes)
UDesigns_II(4)
```

UDesigns_III *Nearly Orthogonal Uniform Designs for Two and Four Factors*

Description

This function will provide nearly orthogonal uniform designs (UDs) for number of factors, F = 2 and 4 but a flexible number of levels, L >=3.

Usage

```
UDesigns_III(levels,factors)
```

Arguments

`levels`	Number of levels, L >=3
`factors`	Number of factors 2 or 4

Value

This function will generate 3 Uniform Designs along with the number of levels,factors, runs, MAC value and discrete discrepancy value along with its lower bound value.

Author(s)

Ashutosh Dalal, Cini varghese, Rajender Parsad and Mohd Harun

References

Fang, K.T. (1980). The uniform design: application of number-theoretic methods in experimental design. Acta Math Appl Sin, 3, 363-372.

Examples

```
## Not run:
library(CompExpDes)
UDesigns_III(3)

## End(Not run)
```

Index