Package: CompExpDes (via r-universe)

September 4, 2024

Type Package

Title Computer Experiment Designs

Version 1.0.3

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 attained the lower bound of Discrete Discrepancy measure.

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]

Date/Publication 2024-09-03 19:50:02 UTC

Repository https://ashutoshdalal97.r-universe.dev

RemoteUrl https://github.com/cran/CompExpDes

RemoteRef HEAD

RemoteSha 56ebaa3c93209fd5ba18a1e0341041d4a3961363

Contents

`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])
lhd<-rbind(lhd1,lhd2)
Discrete_Discrepancy(lhd, 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 runs ranges from F+2 to F^2. Maxpro criterion measure, Phi_p measure also provided as a measure of space-filling and orthogonality measure maximum absolute correlation (MAC) value also provided.

Usage

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

Arguments

`levels`	Ranges from (factors+2) to factors^2
`factors`	A prime number (>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 a prime numbers.

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 number of factors, F (>2), this method will generate LHDs with runs ranges from F+2 to nC2, where n=2*F+1. Maxpro criterion measure, Phi_p measure also provided as a measure of space-filling and orthogonality measure maximum absolute correlation (MAC) value also provided.

Usage

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

Arguments

`levels`	Ranges from (factors+2) to nC2, where n=2*factors+1
`factors`	Any integer (>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 any number.

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 rather than 1 of a correlation matrix.

Usage

```
MAC(matrix)
```

Arguments

matrix	Input a matrix

Value

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

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

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

Based upon a provided design, this function generates the value of maxpro criterion. Lesser the value of it better the design is (if design is not scaled in [0,1]^d).

Usage

```
Maxpro_Measure(Design)
```

Arguments

Design	Provide a design in matrix format

Value

Provides Maxpro criterion value

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)
```

max_coincidence_number *Maximum Coincidence (or Meeting) numbers between rows*

Description

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

Usage

```
max_coincidence_number(matrix)
```

Arguments

`matrix`	Provide any matrix

Value

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

Author(s)

Ashutosh Dalal, Cini Varghese, Rajender Parsad and Mohd Harun

Examples

```
## Not run:
library(CompExpDes)
max_coincidence_number(matrix)

## End(Not run)
```

`PhipMeasure` *Phi_p criterion*

Description

For a given design Phi_p criterion is calculated. It is based on Morris and Mitchell (1995). When the designs are not in [0,1]^d form, lesser the value of Phi_p criterion better it is.

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

For v = pq (p>2,q>=2 and v>=6) these uniform designs are generated. This function provides designs based on two types. It also provides number of factors, number of levels, number of runs along with maximum absolute correlation value and discrete discrepancy measure with its lower bound value.

Usage

```
UDesigns_I(p, q, type)
```

Arguments

p	any integer >2
q	any integer >=2
type	1 or 2

Details

Type 1 or type 2 both can be exist for a same parameter range. For type 1 it will require more runs than designs generated by type 2. But type 1 provides designs which are having more spread than type 2 series designs.

Value

Returns a uniform designs along with number of factors, levels, runs, maximum absolute value and discrete discrepancy measure 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(4,3,1)
```

UDesigns_II *Uniform Designs with multiple factors*

Description

For v = n(n-1)/2, where n (>=5) is any odd number. These are uniform designs interms of discrete discrepancy. It also provides number of factors, number of levels and number of runs and discrete discrepancy measure with its lower bound value.

Usage

```
UDesigns_II(n)
```

Arguments

n	any odd integer >=5

Value

Returns a series of high dimensional uniform designs along with number of factors, levels, runs and discrete discrepancy measure 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(5)
```

`UDesigns_III` *Orthogonal Uniform Designs for Two and Four Factors (Even number v)*

Description

For even number v (>=6) this function will provide total three uniform designs of two factors (2) and four factors (1).

Usage

```
UDesigns_III(v)
```

Arguments

`v`	An even number of factors,v >=6

Details

Uniform Design 1 and Uniform Design 3 are orthogonal/ nearly orthogonal but Uniform Design 2 is always orthogonal design.

Value

This function will generate 3 Uniform Designs along with the number of levels,factors, runs and MAC 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(10)

## End(Not run)
```

Index