Package 'lhs'

July 13, 2012

Version 0.10

Date 2012-07-12

Title Latin Hypercube Samples

Depends R (>= 2.14.2)

Description This package provides a number of methods for creating and augmenting Latin Hypercube Samples

License GPL (>= 2)

Author Rob Carnell [aut, cre]

Maintainer Rob Carnell <carnellr@battelle.org>

Repository CRAN

Repository/R-Forge/Project lhs

Repository/R-Forge/Revision 45

Date/Publication 2012-07-13 16:24:19

R topics documented:

lhs-package *The Latin Hypercube Sample (lhs) Package*

Description

This package provides a number of methods for creating and augmenting Latin Hypercube Samples

For a complete list of functions, use library(help="lhs").

Author(s)

Rob Carnell

See Also

randomLHS, geneticLHS, improvedLHS, maximinLHS, and optimumLHS to generate Latin Hypercube Samples. optAugmentLHS, optSeededLHS, and augmentLHS to modify and augment existing designs.

augmentLHS *Augment a Latin Hypercube Design*

Description

Augments an existing Latin Hypercube Sample, adding points to the design, while maintaining the *latin* properties of the design.

Usage

```
augmentLHS(lhs, m=1)
```

Arguments

lhs	The Latin Hypercube Design to which points are to be added
m	The number of additional points to add to matrix lhs

Details

Augments an existing Latin Hypercube Sample, adding points to the design, while maintaining the *latin* properties of the design. Augmentation is perfomed in a random manner.

The algorithm used by this function has the following steps. First, create a new matrix to hold the candidate points after the design has been re-partitioned into $(n + m)^2$ cells, where n is number of points in the original lhs matrix. Then randomly sweep through each column (1...k) in the repartitioned design to find the missing cells. For each column (variable), randomly search for an empty row, generate a random value that fits in that row, record the value in the new matrix. The new matrix can contain more filled cells than m unles $m = 2n$, in which case the new matrix will

contain exactly m filled cells. Finally, keep only the first m rows of the new matrix. It is guaranteed to have m full rows in the new matrix. The deleted rows are partially full. The additional candidate points are selected randomly due to the random search for empty cells.

Value

An n by k Latin Hypercube Sample matrix with values uniformly distributed on [0,1]

Author(s)

Rob Carnell

References

Stein, M. (1987) Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*. **29**, 143–151.

See Also

randomLHS, geneticLHS, improvedLHS, maximinLHS, and optimumLHS to generate Latin Hypercube Samples. optAugmentLHS and optSeededLHS to modify and augment existing designs.

Examples

```
a <- randomLHS(4,3)
a
augmentLHS(a, 2)
```

---

geneticLHS *Latin Hypercube Sampling with a Genetic Algorithm*

Description

Draws a Latin Hypercube Sample from a set of uniform distributions for use in creating a Latin Hypercube Design. This function attempts to optimize the sample with respect to the S optimality criterion through a genetic type algorithm.

Usage

```
geneticLHS(n=10, k=2, pop=100, gen=4, pMut=.1, verbose=FALSE)
```

Arguments

n	The number of partitions (simulations or design points)
k	The number of replications (variables)
pop	The number of designs in the initial population
gen	The number of generations over which the algorithm is applied
pMut	The probability with which a mutation occurs in a column of the progeny
verbose	Print informational messages

Details

Latin hypercube sampling (LHS) was developed to generate a distribution of collections of parameter values from a multidimensional distribution. A square grid containing possible sample points is a Latin square iff there is only one sample in each row and each column. A Latin hypercube is the generalisation of this concept to an arbitrary number of dimensions. When sampling a function of k variables, the range of each variable is divided into n equally probable intervals. n sample points are then drawn such that a Latin Hypercube is created. Latin Hypercube sampling generates more efficient estimates of desired parameters than simple Monte Carlo sampling.

This program generates a Latin Hypercube Sample by creating random permutations of the first n integers in each of k columns and then transforming those integers into n sections of a standard uniform distribution. Random values are then sampled from within each of the n sections. Once the sample is generated, the uniform sample from a column can be transformed to any distribution by using the quantile functions, e.g. qnorm(). Different columns can have different distributions.

S-optimality seeks to maximize the mean distance from each design point to all the other points in the design, so the points are as spread out as possible.

Genetic Algorithm:

1. Generate pop random latin hypercube designs of size n by k
2. Calculate the S optimality measure of each design
3. Keep the best design in the first position and throw away half of the rest of the population
4. Take a random column out of the best matrix and place it in a random column of each of the other matricies, and take a random column out of each of the other matricies and put it in copies of the best matrix thereby causing the progeny
5. For each of the progeny, cause a genetic mutation pMut percent of the time. The mutation is accomplished by swtching two elements in a column

Value

An n by k Latin Hypercube Sample matrix with values uniformly distributed on [0,1]

Author(s)

Rob Carnell

References

Stocki, R. (2005) A method to improve design reliability using optimal Latin hypercube sampling *Computer Assisted Mechanics and Engineering Sciences* **12**, 87–105.

Stein, M. (1987) Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*. **29**, 143–151.

See Also

randomLHS, improvedLHS, maximinLHS, and optimumLHS to generate Latin Hypercube Samples. optAugmentLHS, optSeededLHS, and augmentLHS to modify and augment existing designs.

Examples

```
geneticLHS(4, 3, 50, 5, .25)
```

`improvedLHS` *Improved Latin Hypercube Sample*

Description

Draws a Latin Hypercube Sample from a set of uniform distributions for use in creating a Latin Hypercube Design. This function attempts to optimize the sample with respect to an optimum euclidean distance between design points.

Usage

```
improvedLHS(n, k, dup=1)
```

Arguments

n	The number of partitions (simulations or design points)
k	The number of replications (variables)
dup	A factor that determines the number of candidate points used in the search. A multiple of the number of remaining points than can be added.

Details

Latin hypercube sampling (LHS) was developed to generate a distribution of collections of parameter values from a multidimensional distribution. A square grid containing possible sample points is a Latin square iff there is only one sample in each row and each column. A Latin hypercube is the generalisation of this concept to an arbitrary number of dimensions. When sampling a function of k variables, the range of each variable is divided into n equally probable intervals. n sample points are then drawn such that a Latin Hypercube is created. Latin Hypercube sampling generates more efficient estimates of desired parameters than simple Monte Carlo sampling.

This program generates a Latin Hypercube Sample by creating random permutations of the first n integers in each of k columns and then transforming those integers into n sections of a standard uniform distribution. Random values are then sampled from within each of the n sections. Once the sample is generated, the uniform sample from a column can be transformed to any distribution by using the quantile functions, e.g. qnorm(). Different columns can have different distributions.

This function attempts to optimize the sample with respect to an optimum euclidean distance between design points.

$$Optimumdistance = fracnn^{\frac{1,0}{k}}$$

Value

An n by k Latin Hypercube Sample matrix with values uniformly distributed on [0,1]

Author(s)

Rob Carnell

References

Beachkofski, B., Grandhi, R. (2002) Improved Distributed Hypercube Sampling *American Institute of Aeronautics and Astronautics Paper* **1274**.

This function is based on the MATLAB program written by John Burkardt and modified 16 Feb 2005 http://www.csit.fsu.edu/~burkardt/m_src/ihs/ihs.m

See Also

randomLHS, geneticLHS, maximinLHS, and optimumLHS to generate Latin Hypercube Samples. optAugmentLHS, optSeededLHS, and augmentLHS to modify and augment existing designs.

Examples

```
improvedLHS(4, 3, 2)
```

`maximinLHS` *Maximin Latin Hypercube Sample*

Description

Draws a Latin Hypercube Sample from a set of uniform distributions for use in creating a Latin Hypercube Design. This function attempts to optimize the sample by maximizing the minium distance between design points (maximin criteria).

Usage

```
maximinLHS(n, k, dup=1)
```

Arguments

n	The number of partitions (simulations or design points)
k	The number of replications (variables)
dup	A factor that determines the number of candidate points used in the search. A multiple of the number of remaining points than can be added.

Details

Latin hypercube sampling (LHS) was developed to generate a distribution of collections of parameter values from a multidimensional distribution. A square grid containing possible sample points is a Latin square iff there is only one sample in each row and each column. A Latin hypercube is the generalisation of this concept to an arbitrary number of dimensions. When sampling a function of k variables, the range of each variable is divided into n equally probable intervals. n sample points are then drawn such that a Latin Hypercube is created. Latin Hypercube sampling generates more efficient estimates of desired parameters than simple Monte Carlo sampling.

This program generates a Latin Hypercube Sample by creating random permutations of the first n integers in each of k columns and then transforming those integers into n sections of a standard uniform distribution. Random values are then sampled from within each of the n sections. Once the sample is generated, the uniform sample from a column can be transformed to any distribution by using the quantile functions, e.g. qnorm(). Different columns can have different distributions.

Here, values are added to the design one by one such that the maximin criteria is satisfied.

Value

An n by k Latin Hypercube Sample matrix with values uniformly distributed on [0,1]

Author(s)

Rob Carnell

References

Stein, M. (1987) Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*. **29**, 143–151.

This function is motivated by the MATLAB program written by John Burkardt and modified 16 Feb 2005 http://www.csit.fsu.edu/~burkardt/m_src/ihs/ihs.m

See Also

randomLHS, geneticLHS, improvedLHS and optimumLHS to generate Latin Hypercube Samples. optAugmentLHS, optSeededLHS, and augmentLHS to modify and augment existing designs.

Examples

```
maximinLHS(4, 3, 2)
```

optAugmentLHS *Optimal Augmented Latin Hypercube Sample*

Description

Augments an existing Latin Hypercube Sample, adding points to the design, while maintaining the *latin* properties of the design. This function attempts to add the points to the design in an optimal way.

Usage

```
optAugmentLHS(lhs, m=1, mult=2)
```

Arguments

lhs	The Latin Hypercube Design to which points are to be added
m	The number of additional points to add to matrix lhs
mult	m*mult random candidate points will be created.

Details

Augments an existing Latin Hypercube Sample, adding points to the design, while maintaining the *latin* properties of the design. This function attempts to add the points to the design in a way that maximizes S optimality.

S-optimality seeks to maximize the mean distance from each design point to all the other points in the design, so the points are as spread out as possible.

Value

An n by k Latin Hypercube Sample matrix with values uniformly distributed on [0,1]

Author(s)

Rob Carnell

References

Stein, M. (1987) Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*. **29**, 143–151.

See Also

randomLHS, geneticLHS, improvedLHS, maximinLHS, and optimumLHS to generate Latin Hypercube Samples. optSeededLHS and augmentLHS to modify and augment existing designs.

Examples

```
a <- randomLHS(4,3)
a
optAugmentLHS(a, 2, 3)
```

---

optimumLHS *Optimum Latin Hypercube Sample*

Description

Draws a Latin Hypercube Sample from a set of uniform distributions for use in creating a Latin Hypercube Design. This function uses the Columnwise Pairwise (CP) algorithm to generate an optimal design with respect to the S optimality criterion.

Usage

```
optimumLHS(n=10, k=2, maxSweeps=2, eps=.1, verbose=FALSE)
```

Arguments

n	The number of partitions (simulations or design points)
k	The number of replications (variables)
maxSweeps	The maximum number of times the CP algorithm is applied to all the columns.
eps	The optimal stopping criterion
verbose	Print informational messages

Details

Latin hypercube sampling (LHS) was developed to generate a distribution of collections of parameter values from a multidimensional distribution. A square grid containing possible sample points is a Latin square iff there is only one sample in each row and each column. A Latin hypercube is the generalisation of this concept to an arbitrary number of dimensions. When sampling a function of k variables, the range of each variable is divided into n equally probable intervals. n sample points are then drawn such that a Latin Hypercube is created. Latin Hypercube sampling generates more efficient estimates of desired parameters than simple Monte Carlo sampling.

This program generates a Latin Hypercube Sample by creating random permutations of the first n integers in each of k columns and then transforming those integers into n sections of a standard uniform distribution. Random values are then sampled from within each of the n sections. Once the sample is generated, the uniform sample from a column can be transformed to any distribution by using the quantile functions, e.g. qnorm(). Different columns can have different distributions.

S-optimality seeks to maximize the mean distance from each design point to all the other points in the design, so the points are as spread out as possible.

This function uses the CP algorithm to generate an optimal design with respect to the S optimality criterion.

Value

An n by k Latin Hypercube Sample matrix with values uniformly distributed on [0,1]

Author(s)

Rob Carnell

References

Stocki, R. (2005) A method to improve design reliability using optimal Latin hypercube sampling *Computer Assisted Mechanics and Engineering Sciences* **12**, 87–105.

See Also

randomLHS, geneticLHS, improvedLHS and maximinLHS to generate Latin Hypercube Samples. optAugmentLHS, optSeededLHS, and augmentLHS to modify and augment existing designs.

Examples

```
optimumLHS(4, 3, 5, .05)
```

---

optSeededLHS *Optimum Seeded Latin Hypercube Sample*

Description

Augments an existing Latin Hypercube Sample, adding points to the design, while maintaining the *latin* properties of the design. This function then uses the columnwise pairwise (CP) algoritm to optimize the design. The original design is not necessarily maintained.

Usage

```
optSeededLHS(seed, m=1, maxSweeps=2, eps=.1, verbose=FALSE)
```

Arguments

seed	The number of partitions (simulations or design points)
m	The number of additional points to add to matrix seed
maxSweeps	The maximum number of times the CP algorithm is applied to all the columns.
eps	The optimal stopping criterion
verbose	Print informational messages

Details

Augments an existing Latin Hypercube Sample, adding points to the design, while maintaining the *latin* properties of the design. This function then uses the CP algoritm to optimize the design. The original design is not necessarily maintained.

Value

An n by k Latin Hypercube Sample matrix with values uniformly distributed on [0,1]

Author(s)

Rob Carnell

References

Stein, M. (1987) Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*. **29**, 143–151.

See Also

randomLHS, geneticLHS, improvedLHS, maximinLHS, and optimumLHS to generate Latin Hypercube Samples. optAugmentLHS and augmentLHS to modify and augment existing designs.

Examples

```
a <- randomLHS(4,3)
a
optSeededLHS(a, 2, 2, .1)
```

---

randomLHS *Random Latin Hypercube*

Description

Draws a Latin Hypercube Sample from a set of uniform distributions for use in creating a Latin Hypercube Design. This sample is taken in a random manner without regard to optimization.

Usage

```
randomLHS(n, k, preserveDraw)
```

Arguments

n	The number of partitions (simulations or design points)
k	The number of replications (variables)
preserveDraw	Default:FALSE. Ensures that two subsequent draws with the same n, but one with k and one with m variables (k<m), will have the same first k columns if the seed is the same.

Details

Latin hypercube sampling (LHS) was developed to generate a distribution of collections of parameter values from a multidimensional distribution. A square grid containing possible sample points is a Latin square iff there is only one sample in each row and each column. A Latin hypercube is the generalisation of this concept to an arbitrary number of dimensions. When sampling a function of k variables, the range of each variable is divided into n equally probable intervals. n sample points are then drawn such that a Latin Hypercube is created. Latin Hypercube sampling generates more efficient estimates of desired parameters than simple Monte Carlo sampling.

This program generates a Latin Hypercube Sample by creating random permutations of the first n integers in each of k columns and then transforming those integers into n sections of a standard uniform distribution. Random values are then sampled from within each of the n sections. Once the sample is generated, the uniform sample from a column can be transformed to any distribution by using the quantile functions, e.g. qnorm(). Different columns can have different distributions.

Value

An n by k Latin Hypercube Sample matrix with values uniformly distributed on [0,1]

Author(s)

Rob Carnell and D. Mooney

References

Stein, M. (1987) Large Sample Properties of Simulations Using Latin Hypercube Sampling. *Technometrics*. **29**, 143–151.

See Also

geneticLHS, improvedLHS, maximinLHS, and optimumLHS to generate Latin Hypercube Samples. optAugmentLHS, optSeededLHS, and augmentLHS to modify and augment existing designs.

Examples

```
# draw a Latin hypercube
randomLHS(4, 3)

# transform a Latin hypercube
X <- randomLHS(5, 2)
Y <- matrix(0, nrow=5, ncol=2)
Y[,1] <- qnorm(X[,1], mean=3, sd=0.1)
Y[,2] <- qbeta(X[,2], shape1=2, shape2=3)

# check the preserveDraw option
set.seed(1976)
X <- randomLHS(6,3,preserveDraw=TRUE)
set.seed(1976)
Y <- randomLHS(6,5,preserveDraw=TRUE)
all(abs(X - Y[,1:3]) < 1E-12) # TRUE
```

Index