Package 'aclhs'

November 5, 2025

Title Autocorrelated Conditioned Latin Hypercube Sampling

Version 1.0.1

Description Implementation of the autocorrelated conditioned Latin Hypercube Sampling (acLHS) algorithm for 1D (time-series) and 2D (spatial) data. The acLHS algorithm is an extension of the conditioned Latin Hypercube Sampling (cLHS) algorithm that allows sampled data to have similar correlative and statistical features of the original data. Only a properly formatted dataframe needs to be provided to yield subsample indices from the primary function. For more details about the cLHS algorithm, see Minasny and McBratney (2006), <doi:10.1016/j.cageo.2005.12.009>. For acLHS, see Le and Vargas (2024) <doi:10.1016/j.cageo.2024.105539>.

License MIT + file LICENSE

URL https://github.com/vargaslab/acLHS

BugReports https://github.com/vargaslab/acLHS/issues

Depends R (>= 3.5)

Imports DEoptim (>= 2.2.8), geoR (>= 1.9.6), graphics (>= 4.5.1), stats (>= 4.5.1), utils (>= 4.5.1)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

Author Van Huong Le [aut, ctb] (ORCID: <https://orcid.org/0000-0002-3700-9987>), Rodrigo Vargas [aut] (ORCID: <https://orcid.org/0000-0001-6829-5333>), Gabriel Laboy [ctb, cre] (ORCID: <https://orcid.org/0009-0004-1538-5035>)

Maintainer Gabriel Laboy <glaboy1@asu.edu>

Repository CRAN

Date/Publication 2025-11-05 20:10:02 UTC

Contents

aclhs 2
aclhs.get_correlations 3
aclhs.plot_params 4
aclhs.plot_sampling_distribution 5
aclhs.plot_scatterplot 6
aclhs.plot_univariate_pdf 7
aclhs.plot_variogram_comparison 8
aclhs.vario_params 9
ex_data_1D 9
ex_data_2D 10
score_samples 11

Index **12**

`aclhs` *Get subsample indices using the acLHS algorithm.*

Description

This function extracts a desired number of subsample indices from a dataframe using the acLHS algorithm. The function works for either 1D or 2D data, where it is assumed the last two columns of data are the independent and dependent variables, respectively. Determining the optimal subsamples is done using the DEoptim package, which introduces elements of nondeterminism through randomization. If you desire consistent results, ensure to set a seed before running the function.

Usage

```
aclhs(
  df,
  num_samples,
  weights,
  iter = 1000,
  vario_params = aclhs.vario_params(),
  export_file = NULL
)
```

Arguments

`df`	A dataframe with three columns of data
`num_samples`	The number of desired subsamples
`weights`	A vector of three weights for each objective function
`iter`	The max number of iterations to perform to find optimized indices
`vario_params`	A list of parameters to use when computing Variograms
`export_file`	The name of a CSV to export subsampled rows to

Value

A numeric vector of subsample indices of the original data

Examples

```
## acLHS sampling example
data(ex_data_2D)
input2D <- ex_data_2D

# Set Variogram parameters
v_params <- aclhs.vario_params(num_lags=10, dir=0, tol=90, min_pairs=1)

## Set weights for each objective function, respectively
w <- c(10, 1000, 0.001)

## Run the sampling algorithm
aclhs_samples <- aclhs(df=input2D, num_samples=50, weights=w, iter=100,
                       vario_params=v_params,
                       export_file=tempfile(fileext=".csv"))

## Subsample original data
df_sampled <- input2D[aclhs_samples,]
```

---

aclhs.get_correlations

Computes correlations between the original and aclhs-sampled data.

Description

Computes the Pearson, Spearman, and Kendall correlations of the independent and dependent variable of the original and aclhs-sampled data. Correlation values are rounded to the third decimal place.

Usage

```
aclhs.get_correlations(df, aclhs_samples)
```

Arguments

df	The original data in dataframe format
aclhs_samples	The acLHS-derived sample indices

Value

A dataframe with the original and aclhs-sampled correlation values

Examples

```
## Get the data of interest and get the acLHS sample indices
data(ex_data_2D)
input2D <- ex_data_2D
aclhs_sam <- aclhs(df=input2D, num_samples=50, weights=c(1,1,1), iter=100)

## Compute the correlations
correlations <- aclhs.get_correlations(df=input2D, aclhs_samples=aclhs_sam)
```

`aclhs.plot_params` *Set parameters for plotting.*

Description

Sets various parameters for plotting including plot title, axis labels, plot dimensions and resolution, and whether to add a legend to the plot. By default, a plot will not be created, and the location of where the legend should be placed on the plot should be passed (e.g., "topright").

Usage

```
aclhs.plot_params(
  file_name,
  plot_title = "",
  xlab = "",
  ylab = "",
  width = 1000,
  height = 1000,
  res = 150,
  legend = NULL
)
```

Arguments

`file_name`	The name of the file to store the plot in (should end with '.png')
`plot_title`	The title of the plot (default is blank)
`xlab`	The label for the x axis of the plot (default is blank)
`ylab`	The label for the y axis of the plot (default is blank)
`width`	The width of the plot (default is 1000)
`height`	The height of the plot (default is 1000)
`res`	The resolution of the plot (default is 150)
`legend`	The location of the legend on the plot (default is NULL)

Value

A list of the set plotting parameters

Examples

```
## Set the parameters
p_params <- aclhs.plot_params(file_name=tempfile(fileext=".png"),
                              plot_title=expression(bold("Sample Distribution")),
                              xlab=expression(bold("X [km]")),
                              ylab=expression(bold("Y [km]")),
                              legend="topright")

## Access one of the the set parameters
p_params$plot_title
```

`aclhs.plot_sampling_distribution`

Plots the acLHS samples distribution.

Description

Plots the acLHS sample distribution for either 1D or 2D data. acLHS samples will be overlayed over the original data points in blue.

Usage

```
aclhs.plot_sampling_distribution(df, aclhs_samples, plot_params)
```

Arguments

`df`	The original data in dataframe format
`aclhs_samples`	The acLHS-derived sample indices
`plot_params`	The plotting parameters to use

Value

No return value, called for side effects

Examples

```
## Get the data of interest and get the acLHS sample indices
data(ex_data_2D)
input2D <- ex_data_2D
aclhs_sam <- aclhs(df=input2D, num_samples=50, weights=c(1,1,1), iter=100)

## Set plotting parameters
p_params <- aclhs.plot_params(file_name=tempfile(fileext=".png"),
                              xlab=expression(bold("X [km]")),
                              ylab=expression(bold("Y [km]")))

## Create plot
aclhs.plot_sampling_distribution(df=input2D, aclhs_samples=aclhs_sam,
                                 plot_params=p_params)
```

`aclhs.plot_scatterplot` *Plot the scatterplot of the acLHS subsamples.*

Description

Plots the acLHS-sampled points of independent and dependent variables of the data as a scatterplot over the original points.

Usage

```
aclhs.plot_scatterplot(df, aclhs_samples, plot_params)
```

Arguments

`df`	The original data in dataframe format
`aclhs_samples`	The acLHS-derived sample indices
`plot_params`	The plotting parameters to use

Value

No return value, called for side effects

Examples

```
#' ## Get the data of interest and get the acLHS sample indices
data(ex_data_2D)
input2D <- ex_data_2D
aclhs_sam <- aclhs(df=input2D, num_samples=50, weights=c(1,1,1), iter=100)

## Set plotting parameters
p_params <- aclhs.plot_params(file_name=tempfile(fileext=".png"),
                              xlab=expression(bold("Temperature")),
                              ylab=expression(bold("CO2 Efflux")))

## Create plot
aclhs.plot_scatterplot(df=input2D, aclhs_samples=aclhs_sam,
                       plot_params=p_params)
```

`aclhs.plot_univariate_pdf` *Plot the univariate PDF for a column of acLHS-derived samples.*

Description

Plots the univariate PDF of acLHS-sampled points over the original univariate PDF data. The PDF can be plotted for either the dependent or independent variable of the original data.

Usage

```
aclhs.plot_univariate_pdf(df, aclhs_samples, col, plot_params)
```

Arguments

`df`	The original data in dataframe format
`aclhs_samples`	The acLHS-derived sample indices
`col`	The column of data to plot
`plot_params`	The plotting parameters to use

Value

No return value, called for side effects

Examples

```
## Get the data of interest and get the acLHS sample indices
data(ex_data_2D)
input2D <- ex_data_2D
aclhs_sam <- aclhs(df=input2D, num_samples=50, weights=c(1,1,1), iter=100)

## Set plotting parameters
p_params <- aclhs.plot_params(file_name=tempfile(fileext=".png"),
                              xlab=expression(bold("Temperature [Celsius]")),
                              ylab=expression(bold("Fn(Temperature)")))

## Create plot
aclhs.plot_univariate_pdf(df=input2D, aclhs_samples=aclhs_sam, col=3,
                          plot_params=p_params)
```

`aclhs.plot_variogram_comparison` *Plot the Variogram comparison of the acLHS subsamples.*

Description

Plots the acLHS-sampled Variogram against the Variogram of the original data. A best-fit curve of the original Variogram is added for clearer comparison.

Usage

```
aclhs.plot_variogram_comparison(df, aclhs_samples, vario_params, plot_params)
```

Arguments

`df`	The original dataframe
`aclhs_samples`	The acLHS-derived sample indices
`vario_params`	The parameters to set for computing a Variogram
`plot_params`	The plotting parameters to use

Value

No return value, called for side effects

Examples

```
#' ## Get the data of interest and get the acLHS sample indices
data(ex_data_2D)
input2D <- ex_data_2D
v_params <- aclhs.vario_params(num_lags=10, dir=0, tol=90, min_pairs=1)
aclhs_sam <- aclhs(df=input2D, num_samples=50, weights=c(1,1,1),
                   iter=100, vario_params=v_params)

## Set plotting parameters
p_params <- aclhs.plot_params(file_name=tempfile(fileext=".png"),
                              xlab=expression(bold("Distance [km]")),
                              ylab=expression(bold("Semivariance")))

## Create plot
aclhs.plot_variogram_comparison(df=input2D, aclhs_samples=aclhs_sam,
                                vario_params=v_params, plot_params=p_params)
```

aclhs.vario_params *Set parameters for computing a Variogram.*

Description

Sets specific parameters for computing Variograms within the acLHS 1D or 2D function calls. Note that the lag value computed for Variograms will always be the 'minimum' of the independent data (i.e., for 1D minimum time between points and for 2D minimum distance between points).

Usage

```
aclhs.vario_params(num_lags = 8, dir = 0, tol = 90, min_pairs = 1)
```

Arguments

num_lags	The number of lags
dir	The direction
tol	The tolerance
min_pairs	The minimum number of pairs

Value

A list of the set Variogram parameters

Examples

```
## Store the parameters into a variable
v_params <- aclhs.vario_params(num_lags=10, dir=0, tol=90, min_pairs=1)

## Access one of the the set parameters
v_params$num_lags
```

ex_data_1D *Daily CO2 Efflux Measurements within a Temperate Forest*

Description

A dataset containing daily CO2 efflux levels as a dependent variable and temperature as an independent variables within a temperate forest for a full year.

Usage

```
ex_data_1D
```

Format

`ex_data_1D`**:**

A data frame with 365 rows and 3 columns:

Time The day of the year

Temp The temperature

CO2 The carbon dioxide efflux

doi:10.1007/s1110401735064

Examples

```
data(ex_data_1D)
```

`ex_data_2D` *Spatial Distribution of Soil CO2 Efflux for CONUS*

Description

A dataset containing spatially distributed CO2 efflux levels as a dependent variable and temperature as an independent variables within CONUS.

Usage

```
ex_data_2D
```

Format

`ex_data_2D`**:**

A data frame with 903 rows and 4 columns:

X The longitude

Y The latitude

rTemp The temperature

rRS The carbon dioxide efflux

doi:10.1111/gcb.15666

Examples

```
data(ex_data_2D)
```

score_samples *Computes a score from three objective functions.*

Description

Computes a score from the sum of three objective functions multiplied by their respective weights. The score is used to determine the best set of indices subsampled by the acLHS algorithm, where lower is better.

Usage

```
score_samples(
  var_samples,
  df,
  num_samples,
  quantile_ind,
  corrs,
  min_val,
  vario_dep,
  vario_params,
  weights
)
```

Arguments

var_samples	Subsampled indices to test
df	A dataframe with three columns of data
num_samples	The number of subsamples
quantile_ind	The quantile of the independent variable in df
corrs	A vector of three correlations of the two variables in df
min_val	The minimum time or distance between two points in df
vario_dep	The computed Variogram of the data
vario_params	The parameters to set for computing a Variogram
weights	A vector of three weights for each objective function

Value

Returns the summed score of the weighted objective functions

Index

∗ **datasets**
ex_data_1D, 9
ex_data_2D, 10

aclhs, 2
aclhs.get_correlations, 3
aclhs.plot_params, 4
aclhs.plot_sampling_distribution, 5
aclhs.plot_scatterplot, 6
aclhs.plot_univariate_pdf, 7
aclhs.plot_variogram_comparison, 8
aclhs.vario_params, 9

ex_data_1D, 9
ex_data_2D, 10

score_samples, 11