# clhs

*Release 1.0.0*

**Apr 27, 2021**

# Overview

# CHAPTER 1

## About conditioned Latin Hypercube Sampling (cLHS) in Python

This code is based on the cLHS method of Minasny & McBratney (2006). It follows some of the code from the R package clhs of Roudier et al.

For cLHS the problem is: given $N$ sites with ancillary variables ($X$), select $x$ a sub-sample of size $n \ll N$ in order that $x$ forms a Latin hypercube, or the multivariate distribution of $X$ is maximally stratified.

In short, this code attempts to create a Latin Hypercube sample by selecting only from input data. It uses simulated annealing to force the sampling to converge more rapidly, and also allows for setting a stopping criterion on the objective function described in Minasny & McBratney (2006).

Credits: Erika Wagoner (wagoner47) and Zhonghua Zheng (zzheng93)

CHAPTER 2

# Installation instructions

## 2.1 Install on local machine with `pip`

```
$ pip install clhs
```

## 2.2 Install on local machine from source

The get the latest verson that is not uploaded to PyPI yet:

1. Clone the github repository

   ```
   $ git clone https://github.com/wagoner47/clhs_py.git
   ```

   Or using SSH clone

   ```
   $ git clone git@github.com:wagoner47/clhs_py.git
   ```

2. Move into the new directory

   ```
   $ cd clhs_py
   ```

3. Run the setup script

   ```
   $ python setup.py install
   ```

You may also supply the *–user* option to install for a single user (which is helpful if you don't have admin/root privledges, for instance)

```
$ python setup.py install --user
```

Other options are also available for the setup script. To see all of them with documentation, use

```
$ python setup.py install --help
```

CHAPTER 3

# Licensing

clhs_py is licensed with the MIT License.

# CHAPTER 4

# Quickstart

This script is created by Zhonghua Zheng (zzheng25@illinois.edu), for the purpose of showing:
- how to use cLHS
- the comparison between cLHS and random sampling

[1]:
```
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import xarray as xr
import clhs as cl
```

## 4.1 Create a Dataset

[2]:
```
ds = xr.tutorial.open_dataset('air_temperature') # use xr.tutorial.load_dataset() for␣
↪xarray<v0.11.0
df=ds["air"][0,:,:].to_dataframe().reset_index()[["lat","lon","air"]]
# set temperature and relative humidity, relative humidty is normal distribution
df["temp"] = df["air"]-273.15
df["rh"] = np.random.normal(50, 12, 1325)
df.shape[0]
```

[2]:
```
1325
```

## 4.2 Implement cLHS

[3]:
```
# set sample number
num_sample=15
# cLHS
sampled=cl.clhs(df[["temp","rh"]], num_sample, max_iterations=1000)
clhs_sample=df.iloc[sampled["sample_indices"]]
# random sample, as a comparison
random_sample=df.sample(num_sample)
```

```
cLHS:100%||1000/1000 [Elapsed time: 6.365708112716675, ETA: 0.0, 157.09it/s]
```

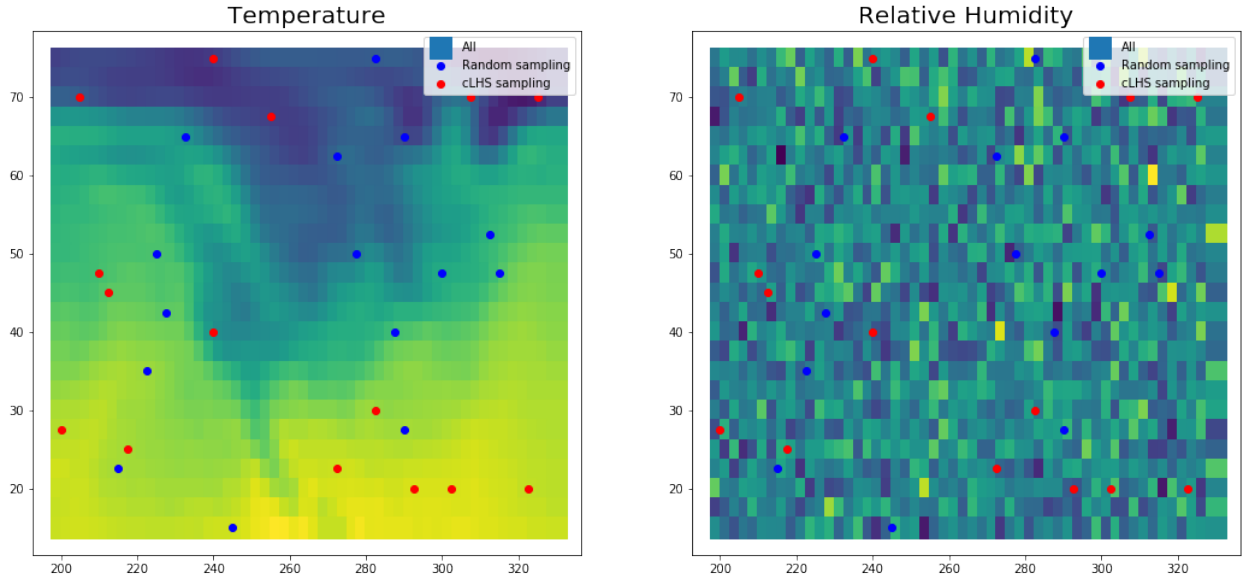
## 4.3 Visualization and Comparison

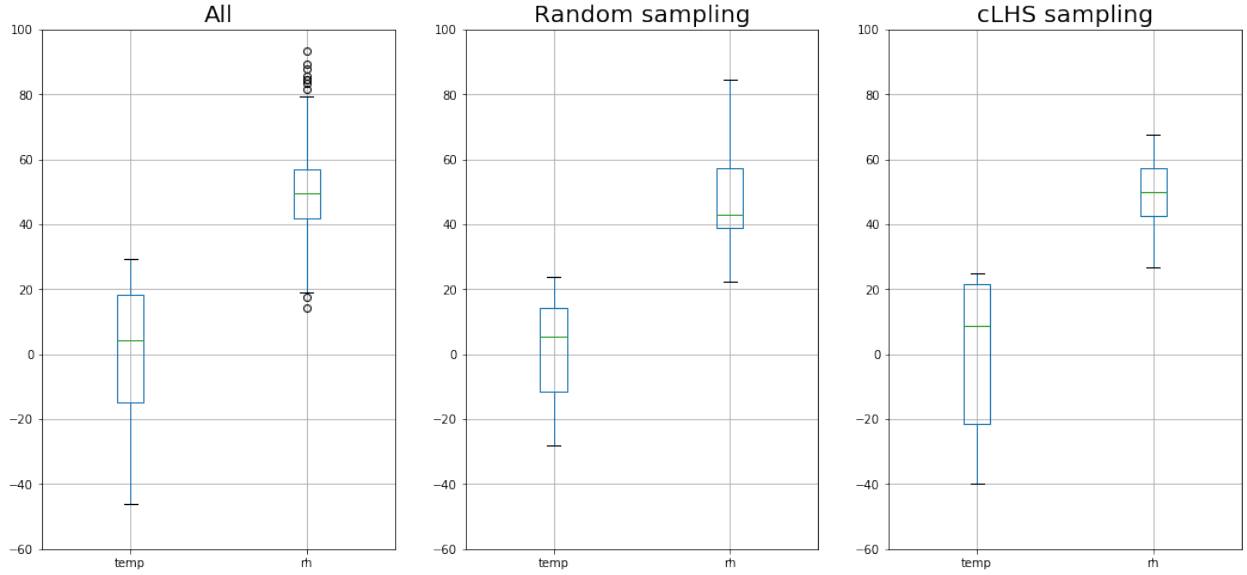
[4]:
```
fig, [ax1,ax2] = plt.subplots(1,2, figsize=(18,8))
ax1.scatter(df["lon"],df["lat"],label="All",c=df["temp"],marker="s",s=300)
ax1.scatter(random_sample["lon"],random_sample["lat"],label="Random sampling",c="blue
↪")
ax1.scatter(clhs_sample["lon"],
            clhs_sample["lat"],
            label="cLHS sampling",c="red")
ax1.legend()
ax1.set_title("Temperature",fontsize=20)

ax2.scatter(df["lon"],df["lat"],label="All",c=df["rh"],marker="s",s=300)
ax2.scatter(random_sample["lon"],random_sample["lat"],label="Random sampling",c="blue
↪")
ax2.scatter(clhs_sample["lon"],
            clhs_sample["lat"],
            label="cLHS sampling",c="red")
ax2.legend()
ax2.set_title("Relative Humidity",fontsize=20)
plt.show()


fig, [ax1, ax2, ax3] = plt.subplots(1,3, figsize=(18,8))
df[["temp","rh"]].boxplot(ax=ax1)
random_sample[["temp","rh"]].boxplot(ax=ax2)
clhs_sample[["temp","rh"]].boxplot(ax=ax3)
ax1.set_ylim([-60,100])
ax1.set_title("All",fontsize=20)
ax2.set_ylim([-60,100])
ax2.set_title("Random sampling",fontsize=20)
ax3.set_ylim([-60,100])
ax3.set_title("cLHS sampling",fontsize=20)
matplotlib.rc('xtick', labelsize=20)
matplotlib.rc('ytick', labelsize=20)
plt.show()

print("Overall")
print(df[["temp","rh"]].describe())
print("\n")
print("Random sampling")
print(random_sample[["temp","rh"]].describe())
print("\n")
```

(continues on next page)

(continued from previous page)

```
print("cLHS sampling")
print(clhs_sample[["temp","rh"]].describe())
print("\n")
```




```
Overall
                temp           rh
count   1325.000000  1325.000000
mean       1.016275    49.783078
std       19.110956    11.866438
min      -46.149994    14.095438
25%      -14.859985    41.674848
50%        4.350006    49.635435
75%       18.250000    57.099548
max       29.450012    93.291254
```

(continues on next page)

(continued from previous page)

```
Random sampling
            temp         rh
count  15.000000  15.000000
mean    0.866668  47.374234
std    17.082689  16.324525
min   -27.949997  22.440250
25%   -11.355003  39.034426
50%     5.350006  43.010765
75%    14.399994  57.418314
max    23.640015  84.635052


cLHS sampling
            temp         rh
count  15.000000  15.000000
mean    1.048006  49.060304
std    24.219866  11.522810
min   -39.949997  26.673582
25%   -21.555000  42.597241
50%     8.850006  50.116637
75%    21.640015  57.448803
max    24.950012  67.569718
```

# CHAPTER 5

# clhs

CHAPTER 6

# Indices and tables

- genindex
- modindex
- search