The Box-Müller transform

The Box-Müller transform is a method to generate random standard normal variates from uniform realizations. If U_1 and U_2 are i.i.d from a Uniform distribution $U_{[0,1]}$, then the variables X_1 and X_2 defined by

$$X_1 = \sqrt{-2log(U_1)}\; cos(2\pi U_2)$$
$$X_2 = \sqrt{-2log(U_1)}\; sin(2\pi U_2)$$

are i.i.d $N(0,1)$.

Implementation in the R language

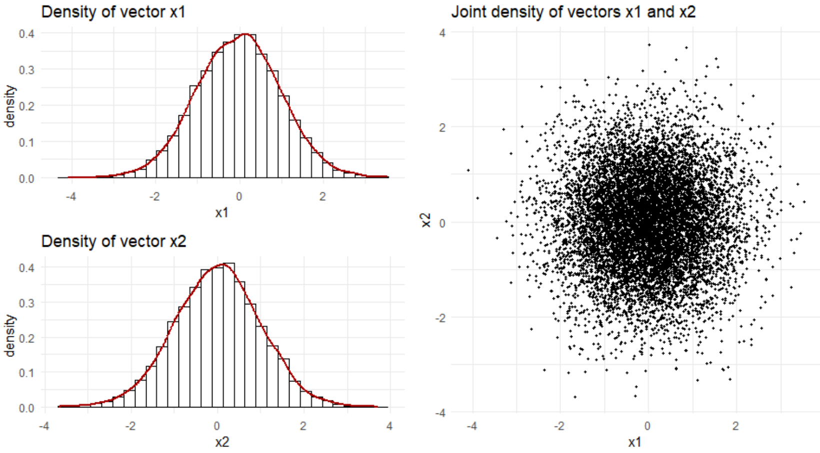
This R code implements the Box-Müller transform to generate $10,000$ $i.i.d.$ realizations of standard Normal random variables X_1 and X_2.

```r
# 1. generate uniform vectors
n = 10000
set.seed(2023)
u1 = runif(n)
u2 = runif(n)

# 2. define Box-M ller transform function
x1 = sqrt(-2*log(u1))*cos(2*pi*u2)
x2 = sqrt(-2*log(u1))*sin(2*pi*u2)
x = data.frame(x1, x2)


# 3. print first lines of dataset
data = x
data[1:6,]
#          x1         x2
# 1  1.139567 -0.4752820
# 2 -1.449969  0.2893102
# 3 -1.791041  0.6499395
# 4 -1.255448  0.5252876
# 5  2.625178 -0.3092095
# 6  1.735641 -1.1015367
```

Visualizing the standard Normals



Implementation in Python

This Python code below implements the Box-Müller transform to generate $10,000$ $i.i.d.$ realizations of standard Normal random variables X_1 and X_2.

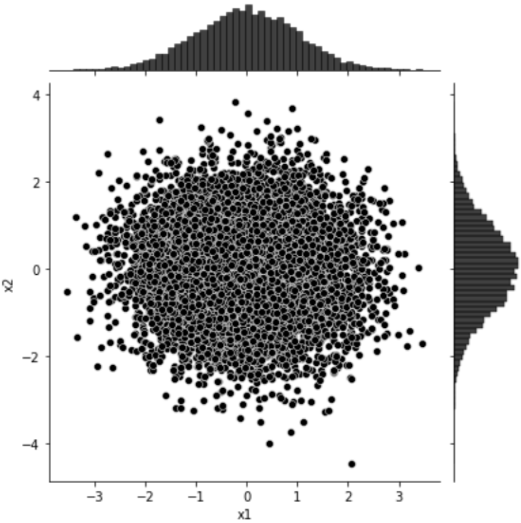
```
import numpy as np
import pandas as pd
import math

# 1. generate uniform vectors
n = 10000
np.random.seed(2023)
u1 = np.random.uniform(low=0.0, high=1.0, size=n)
u2 = np.random.uniform(low=0.0, high=1.0, size=n)

# 2. define Box-M ller transform
x1 = (np.sqrt(-2*np.log(u1))*np.cos(2*math.pi*u2))
x2 = (np.sqrt(-2*np.log(u1))*np.sin(2*math.pi*u2))
x = pd.DataFrame({'x1':x1, 'x2':x2})

# 3. print first lines of dataset
data = x
data[0:6]
         x1        x2
# 0 -0.860024  1.235653
# 1 -0.341486 -0.339862
# 2 -0.577473  0.853466
# 3  1.963577 -0.527138
# 4 -1.889265  0.586374
# 5  1.231823 -0.040409
```

Visualizing the standard Normals


4
2
0
−2
−4
x2
−3
−2
−1
0
1
2
3
x1