Inverse Transform method

Rationale

Otherwise known as inverse CDF method. The continuous case:

Suppose that we have a continuous random variable X having Cummulative Density Function (CDF) F_X. Then, the random variable

$$U = F_X(X)$$

has a Uniform distribution $U \sim U_{[0,1]}$. So to generate a random variate x from the distribution of X, we can use the following transformation

$$F_X^{-1}(U) = x$$

where $F_X^{-1}(.)$ is the inverse CDF or quantile function.

Weibull example

We want to generate a sample of $10,000$ random realizations from a Weibull distribution $W(5,2)$ using the Inverse CDF method.

If $X \sim \mathcal{W}ei(\alpha, \beta)$ having PDF and CDF defined respectively as

$$\begin{aligned} &\textbf{PDF} \quad f_X(x) = \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} exp\left(-\frac{x}{\beta}\right)^{\alpha} \\ &\textbf{CDF} \quad F_X(x) = 1 - exp\left(-\frac{x}{\beta}\right)^{\alpha} \end{aligned}$$

Weibull example solve

Then by the Inverse CDF method, we can generate realizations of X by equating $U = F_X(x)$ and solving for x. We have

$$\begin{aligned} U &= 1 - exp\left(-\frac{x}{2}\right)^5 \\ 1 - U &= exp\left(-\frac{x}{2}\right)^5 \\ -ln(1-U) &= \left(\frac{x}{2}\right)^5 \\ \left(-ln(1-U)\right)^{1/5} &= \frac{x}{2} \\ 2\left(-ln(1-U)\right)^{1/5} &= x \end{aligned}$$

Next up: Python program.

Entrée [5]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
import seaborn as sns

# 1. Define Inverse CDF function
def Inverse_CDF_Weibull(n, alpha, beta) :
        u = np.random.uniform(low=0.0, high=1.0, size=n)          # generate uniform numb
        data = beta*((-np.log(1-u))**(1/alpha))                    # forumla derived

        return pd.DataFrame(data = data, columns = ['data'])      # return a data frame i

# 2. Generate realizations of the desired distribution
np.random.seed(2023)
dataset = Inverse_CDF_Weibull(n = 1000, alpha = 5, beta = 2)
dataset.head()
```

Out[5]:

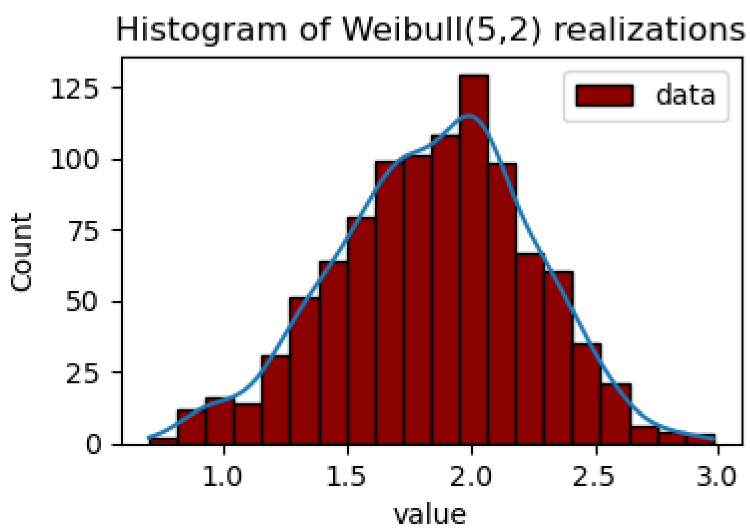
	data
0	1.655497
1	2.343973
2	1.952544
3	1.340683
4	1.372833

Entrée [7]:

```
# 3. Plot with seabortn
plt.figure(figsize=(4,2.5))
plot = sns.histplot(dataset, kde = True, bins = 20, facecolor="darkred", edgecolor='black
plot.set(title='Histogram of Weibull(5,2) realizations')
plot.set(xlabel="value")
```

Out[7]:

```
[Text(0.5, 0, 'value')]
```



References

Rizzo, M.L. (2019). Statistical Computing with R, Second Edition (2nd ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9780429192760 (https://doi.org/10.1201/9780429192760)

Python: {https://www.python.org/ (https://www.python.org/)