Name: Somya Khandelwal

Roll Number: 200123056

MA323 - Lab 08

Question 1

In [3]:

```
import numpy as np
import cmath
import random
import math
def lower_bound(arr, val, n):
    st=0
    en=n
    while(st<en):
        mid=math.floor((st+en)/2)
        if(arr[mid]>val):
            en=mid
        else:
            st=mid+1
    return st

def eval(n):
    arr=np.random.normal(0,1,n)
    arr.sort()
    arr1=[]
    for i in range(0,n):
        u = random.random()
        y = -1*math.log(u)
        gen=lower_bound(arr,(1-y)/2,n)
        gen/=n
        arr1.append(gen)
    print("Expected value is {} and variance is {}".format(np.mean(arr1),np.var(ar
r1)))
    return np.var(arr1)
def eval_antithetic(n):
    arr=np.random.normal(0,1,n)
    arr.sort()
    arr1=[]
    for i in range(0,n):
        u = random.random()
        y = -1*math.log(u)
        y1 = -1*math.log(1-u)
        gen=lower_bound(arr,(1-y)/2,n)
        gen+=lower_bound(arr,(1-y1)/2,n)
        gen/=(2*n)
        arr1.append(gen)
    print("For antithetic, Expected value is {} and variance is {}".format(np.mean
(arr1),np.var(arr1)))
    return np.var(arr1)
x1 = eval(100000)
```

```
x2 = eval_antithetic(100000)
print("Percentage varinace reduction is {}".format((1-x2/x1)*100))
```

```
Expected value is 0.5098441149999998 and variance is 0.026899054896106778
For antithetic, Expected value is 0.5099523178500001 and variance is 0.00307779858
24803214
Percentage varinace reduction is 88.55796757779105
```

Question 2

In [2]:

```
import numpy as np
import pandas as pd
from scipy.stats import norm
N = 1000
h = 0.5
def genInvTrans():
    seq = []
    for i in range(N):
        u = np.random.uniform(0,1)
        seq.append(-np.log(u))
    return seq
print("Estimate generated using Inverse Transform method = ", np.mean(genInvTrans(
)))
print("Variance of estimator generated using Inverse Transform method = ", np.var(
genInvTrans()))
print()
def genEstimator():
    seq = []
    for i in range(N):
        u = np.random.uniform(0,1)
        seq.append(np.log(u)/(h-1))
    return np.exp(np.multiply(-h,seq))/(1-h)
print("Estimate generated using Importance Sampling method = ", np.mean(genEstimat
or()))
print("Variance of estimator generated using Importance Sampling method = ", np.va
r(genEstimator()))
print()
var1 = np.var(genEstimator())
var2 = np.var(genInvTrans())
print("Amount of reduction in variance = ", (var2-var1)/var2 * 100)
```

```
Estimate generated using Inverse Transform method =  0.9648583054492895
Variance of estimator generated using Inverse Transform method =
1.133811339780228

Estimate generated using Importance Sampling method =  0.9975965974881018
Variance of estimator generated using Importance Sampling method =
0.3217935004224239

Amount of reduction in variance =  70.69169546311798
```