SMAI ASSIGNMENT 1

q1

Finite - uniform function

(P(x == xi) = 1/(b-a)) (b>a)

ΣP(x == xi) from a to b => (b-a)/(b-a) = 1

Hence proved

Infinite - geometric function

P(X == x) = (1/(e^x)) (if x != 0)

P(X == 0) = (e-2)/(e-1)

ΣP(x == xi) from 0 to inf

Geometric progression summation = a/1-r = 1

Hence proved

q2

We know that Var = e(x^2)-(e(x))^2 and

E(x) = ∫xp(x)dx(limits from a to b)

Since p(x) = 1/(b-a)

Substituting we get

E(x) = (b^2-a^2)/2(b-a) and E(x^2) = (b3-a3)/3(b-a) this implies

E(x) = (a+b)/2 and E(x^2) = (b^2+ab+a^2)/3

Substituting into variance equation we get

Var = (b-a)^2/12

q3

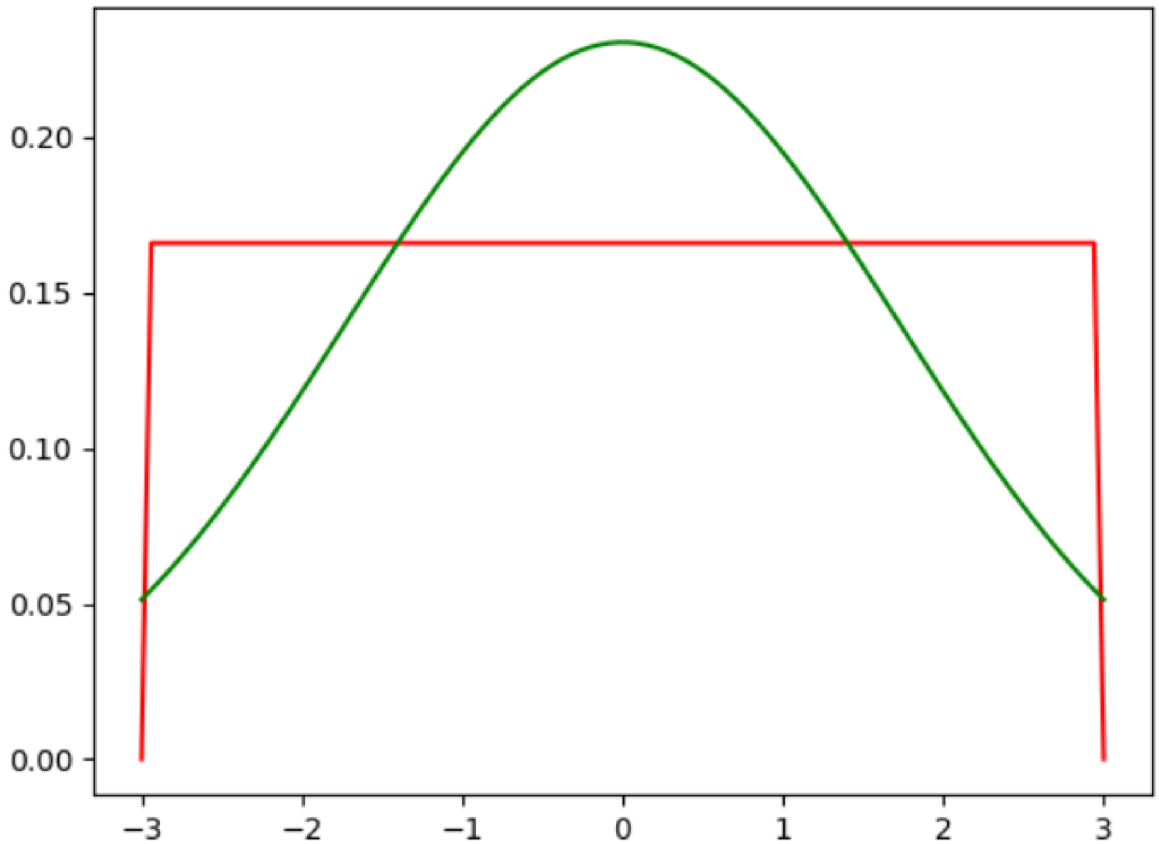
Two different distributions having same mean and variance are

Uniform distribution from [-3 to 3] mean = 0,variance = 3

And Normal distribution from [-3 to 3] with mean = 0 and variance = 3

```
import scipy.stats as stats
import matplotlib.pyplot as plt
import numpy as np
list_x = np.linspace(-3,3,num=100)

list_y = [0.166 if i != 0 and i!=99 else 0 for i in range(0,100) ]
list_norm_pdf = [stats.norm.pdf(i,loc=0,scale=1.73) for i in list_x]

plt.plot(list_x,list_y,color = 'r')

plt.plot(list_x,list_norm_pdf,color = 'g')
plt.show()
```

Running the code will give us the plot



q4

E(x) = ΣxiP(xi) (for discrete variables)

Let E(x) = U

E(x^2) = Σxi^2P(xi) (for discrete variables)

Var = E((x-u)^2)

=> var = E(x^2-2xu+u^2)(using linearity of mean)

=> var = E(x^2) -E(2xu)+E(u^2)

=> var = E(x^2)-U(E(2x)+E(1)(u^2) as constants can be taken out

E(const) = 1 and E(x) = u

=> var = E(x^2) -u(2u) + u^2

=> var = E(x^2) -u^2

q5

We know that E(x) = ∫f(x)xdx

We substitute this with normal distribution

Then

E(x) =1/sqrt(2***Πσ***) ∫(e^-(x-u)2/2*σ*2)xdx[limits from -inf to inf]

Now replace x-u = t in the integral (constants are ignored as they will get canceled)

E(x) = 1/sqrt(2***Πσ***) ∫(e^-(t)2/2*σ*2)(t+u)dt[limits from -inf to inf]

Split into two

1/sqrt(2***Πσ***) ∫(e^-(t)2/2*σ*2)(t)dt+∫(e^-(t)2)u)dt[limits from -inf to inf]

E(x) = 1/sqrt(2***Πσ***) ∫(e^-(t)2)(t)dt+∫(e^-(t)2)u)dt[limits from -inf to inf]

Since first split is odd function the integral will be zero and take u outside

We get E(x) = u

Variance = E(x)^2 -E(x^2)

E(x^2) = 1/sqrt(2***Πσ***) ∫(e^-(x-u)2/2*σ*2)x^2dx

Now substitute t = (x-u)/sqrt(2)***Π***

Then

E(x^2) = 1/sqrt(***Π***) ∫(e^-(t^2)(sqrt(2)t+***Π***)^2dt

Expanding the terms we get

E(x^2) = 2***σ***^2/sqrt(***Π***)∫(e^-(t^2)(t^2)dt +2***σ***^2/sqrt(***Π***)∫((e^-(t^2)(t^2))utdt —--> this term will be zero because it is odd function + u^2(∫e^(-t^2)dt)

Now ∫(e^-(t^2)(t^2)dt limits from [-inf to inf] is sqrt(***Π***)/2

Substituting it we get

E(x^2) = ***σ***^2 + u^2 and E(x) = u then from variance formula

Var = E(x^2) - (E(x))^2

Var = $\boldsymbol{\sigma}$^2 and hence proved

q6

From the plots i infer that histogram is representing the pdf of given distribution

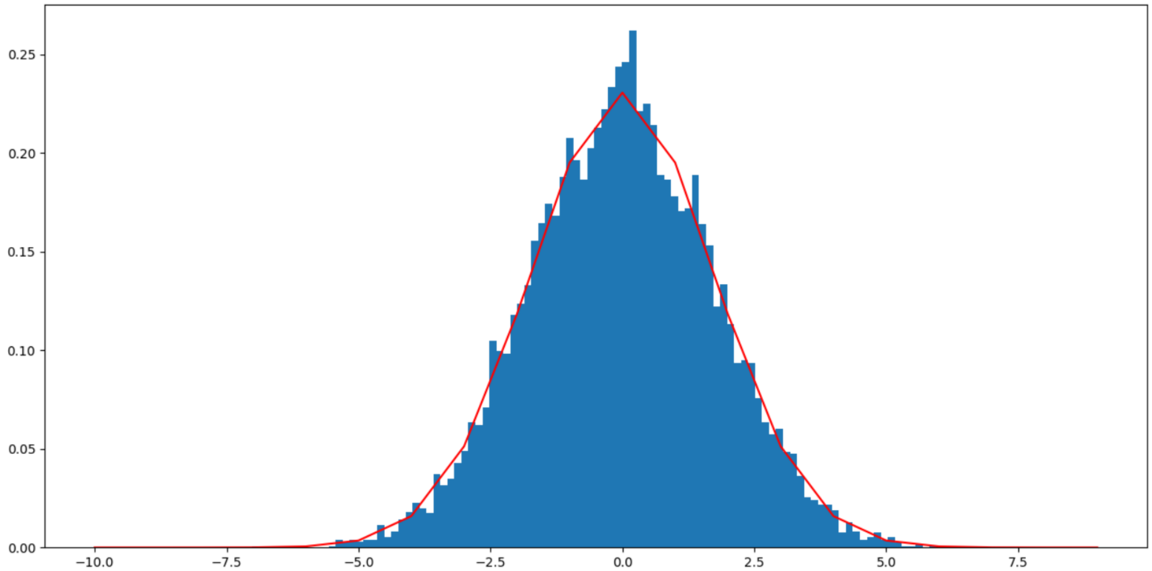
uncomment other plt.hist() and plt.plots to generate other histograms and pdfs

```
import numpy as np
from numpy import random,array
import scipy.stats as stats
import matplotlib.pyplot as plt
list_norm = np.array([])
list_ray = np.array([])
list_expon = np.array([])

for i in range(0,10000):
    a = random.random(1)
    var = array([a,stats.norm.ppf(a,loc=0,scale=1.73)],dtype=object)
    list_norm = np.append(list_norm,var)

for i in range(0,10000):
    a = random.random(1)
    var = array([a,stats.rayleigh.ppf(a,scale=1)],dtype=object)
    list_ray = np.append(list_ray,var)

for i in range(0,10000):
    a = random.random(1)
    var = array([a,stats.expon.ppf(a,scale=0.66)],dtype=object)
    list_expon = np.append(list_expon,var)


list_hist_norm = [list_norm[i] for i in range(1,20000,2)]
list_hist_ray = [list_ray[i] for i in range(1,20000,2)]
list_hist_expon = [list_expon[i] for i in range(1,20000,2)]
list_x = [i for i in range(-10,10)]
list_norm_pdf = [stats.norm.pdf(i,loc=0,scale=1.73) for i in range(-10,10)]
list_ray_pdf = [stats.rayleigh.pdf(i,loc=0,scale=1) for i in range(-10,10)]
list_expon_pdf = [stats.expon.pdf(i,loc=0,scale=0.66) for i in range(-10,10)]

plt.hist(list_hist_norm,bins= 100,density=True)
plt.plot(list_x,list_norm_pdf,color = 'r')
#plt.hist(list_hist_ray,bins= 100,density=True)
#plt.plot(list_x,list_ray_pdf,color = 'r')
#plt.hist(list_hist_expon,bins= 100,density=True)
#plt.plot(list_x,list_expon_pdf,color = 'r')
plt.show()
```

Normal distribution (mean = 0 var = 3)

red color denotes pdf

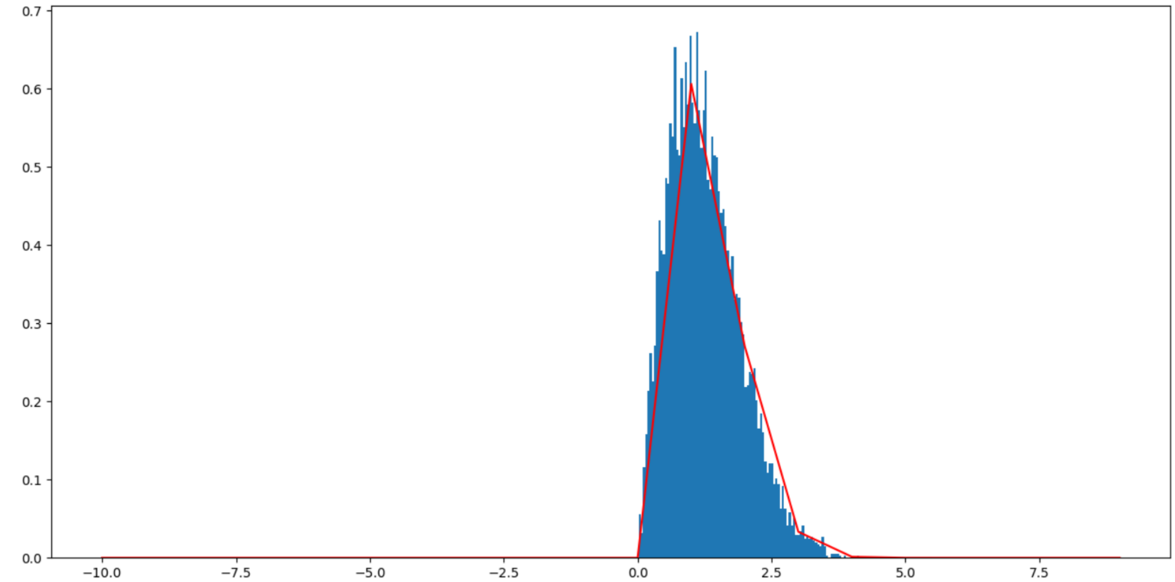
blue denotes histogram



Rayleigh distribution (scale = 1)

red denotes pdf

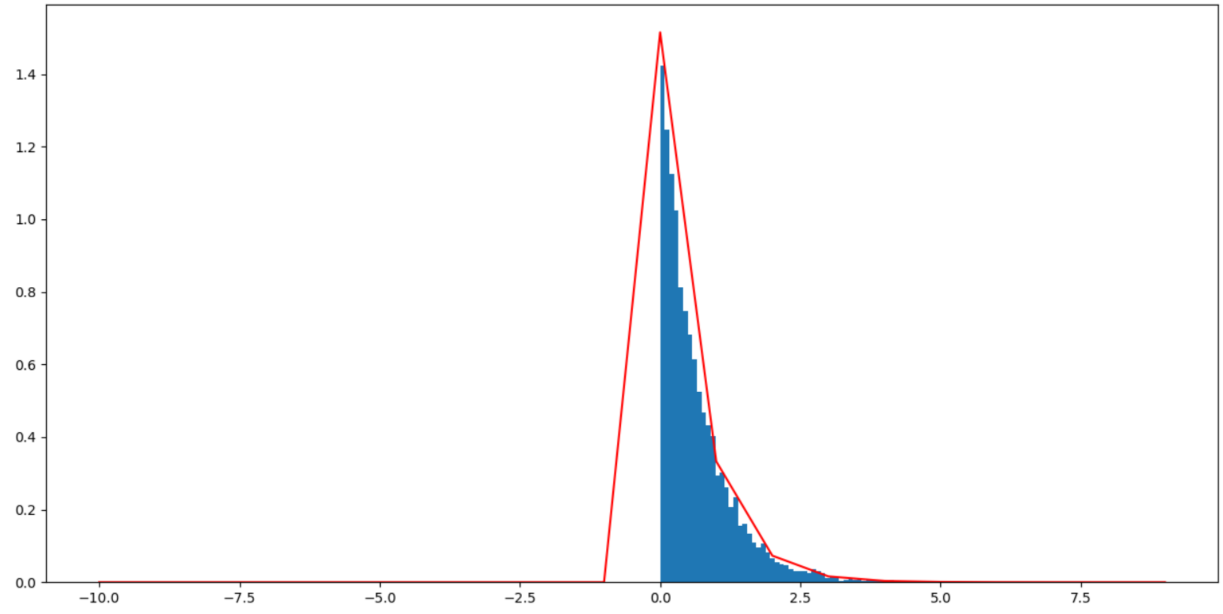
blue denotes histogram



Exponential distribution(lamda = 1.5)

red denotes color

blue denotes histogram



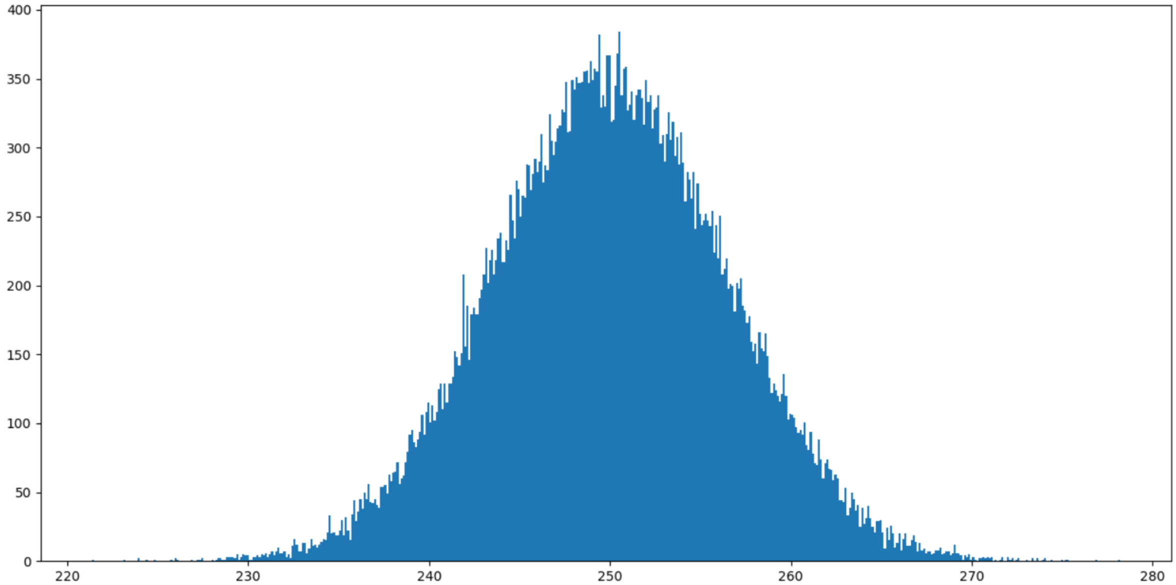
q7

```
from numpy import random
from matplotlib import pyplot as plt


def genrand():
    var = random.random(500)
    sum = 0
    for i in range(0,500):
        sum = sum+var[i]

    return sum

list_y = []
for i in range(0,50000):
    list_y.append(genrand())

plt.hist(list_y,bins=500)
plt.show()
```

genrand() function generates 500 random numbers from (0,1) and computes their sum

Histogram

The plot follows gaussian distribution



400
350
300
250
200
150
100
50
0
220
230
240
250
260
270
280