IIITD

Tutorial Submission

Name: Jatin Kumar Sharma
Roll No: 2020563
MTH-102

Solution 1

##

…………………………………………………………………………………………………

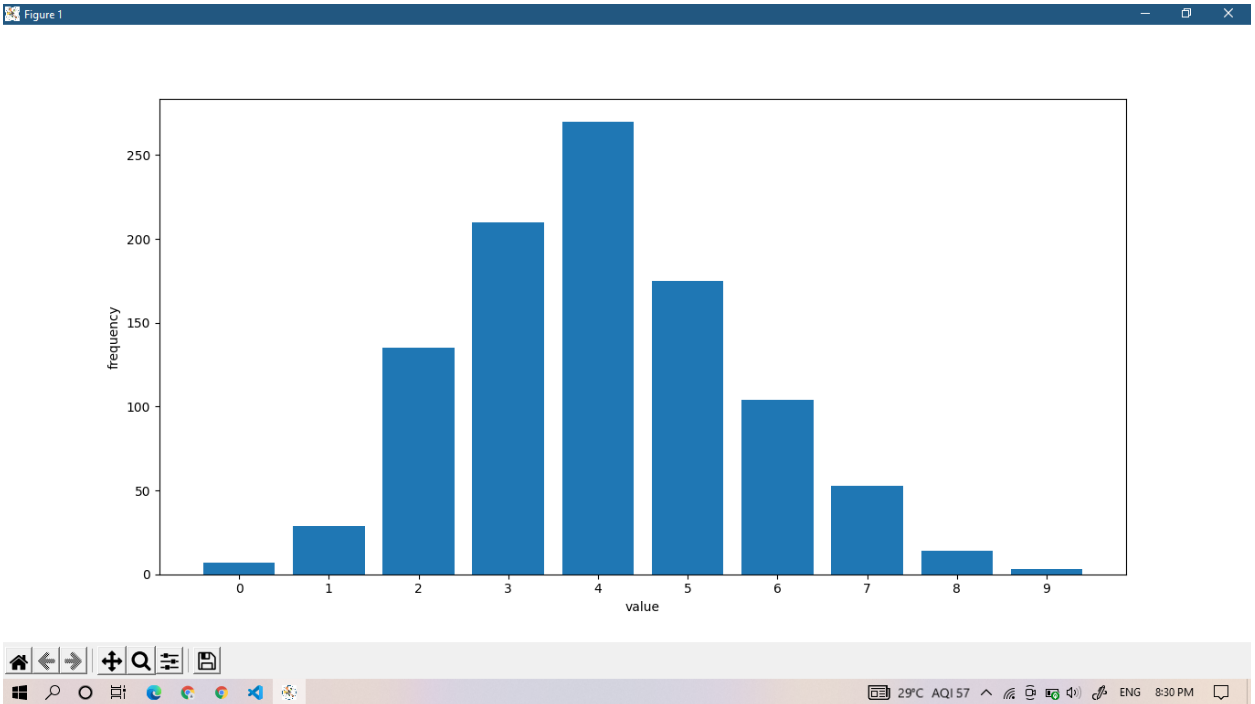
Solution 1.1. **Change n for Binomial RV:**

Parameters for BInomial RV:

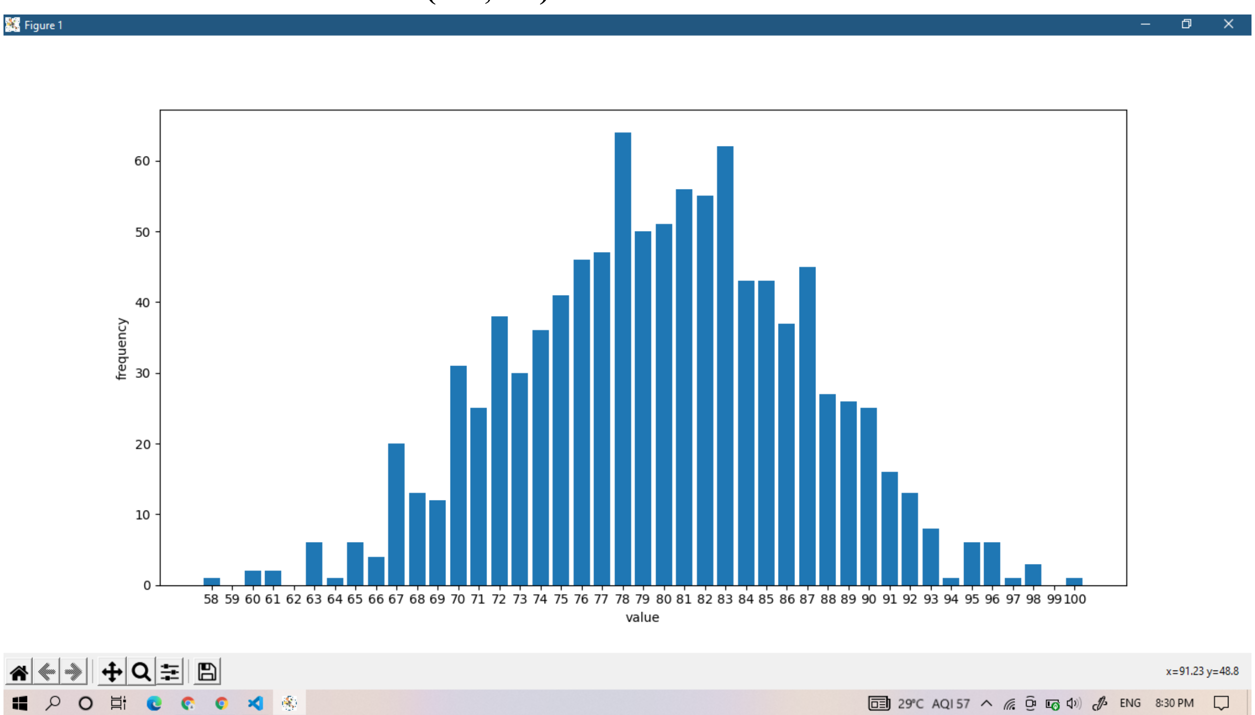
1. Sample size = 1000
2. Probability of success(p) = 0.5   //(a median value is taken)
3. Number of trials n : varying.

```
#@title  #Binomial RV
import matplotlib.pyplot as plt
from scipy.stats import uniform
import numpy as np

size = 1000

def Binomial(n, p = 0.5, m = size):
  binomial = np.empty(m, dtype=int)

  for i in range(m):
    data_uniform = uniform.rvs(size=n, loc=0, scale=1)
    bernoulli = np.where(data_uniform < p, 1, 0)
    binomial[i] = np.sum(bernoulli)
  rv = binomial
  bin_edges = np.arange(rv.min()-0.5, rv.max()+1.5, 1)
  plt.hist(rv, bins = bin_edges, rwidth = 0.8)
  plt.xticks(np.arange(rv.min(), rv.max()+1, 1))
  plt.xlabel('value')
  plt.ylabel('frequency')
  plt.show()
```

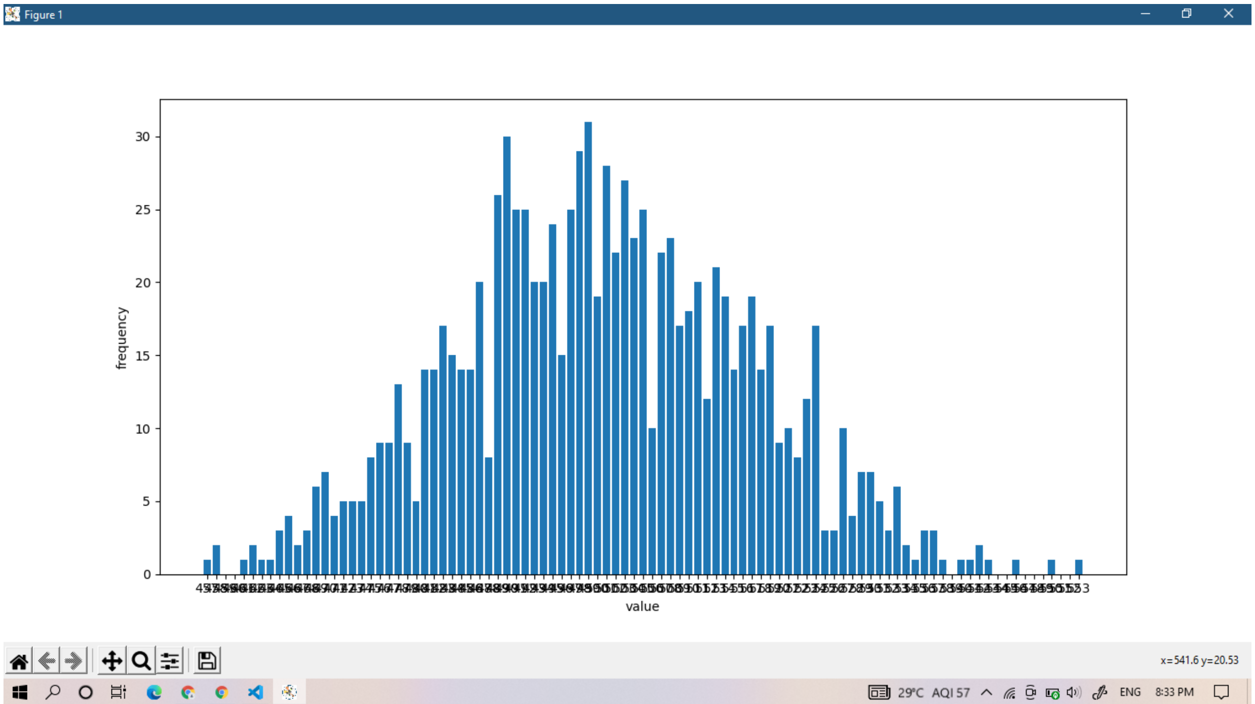
1.1..1: For n = 10 ⇒ Binomial(10, 0.5):



1.1.2. For n = 200 ⇒ Binomial(200, 0.5):



1.1.3. For n = 1000 ⇒ Binomial(1000, 0.5):



Explanation:

As we increase the value of n in binomial distribution the frequency of bigger values of Binomial RV increases whereas the frequency of lower values of Binomial RV decreases, The nature of the graph is the same here.

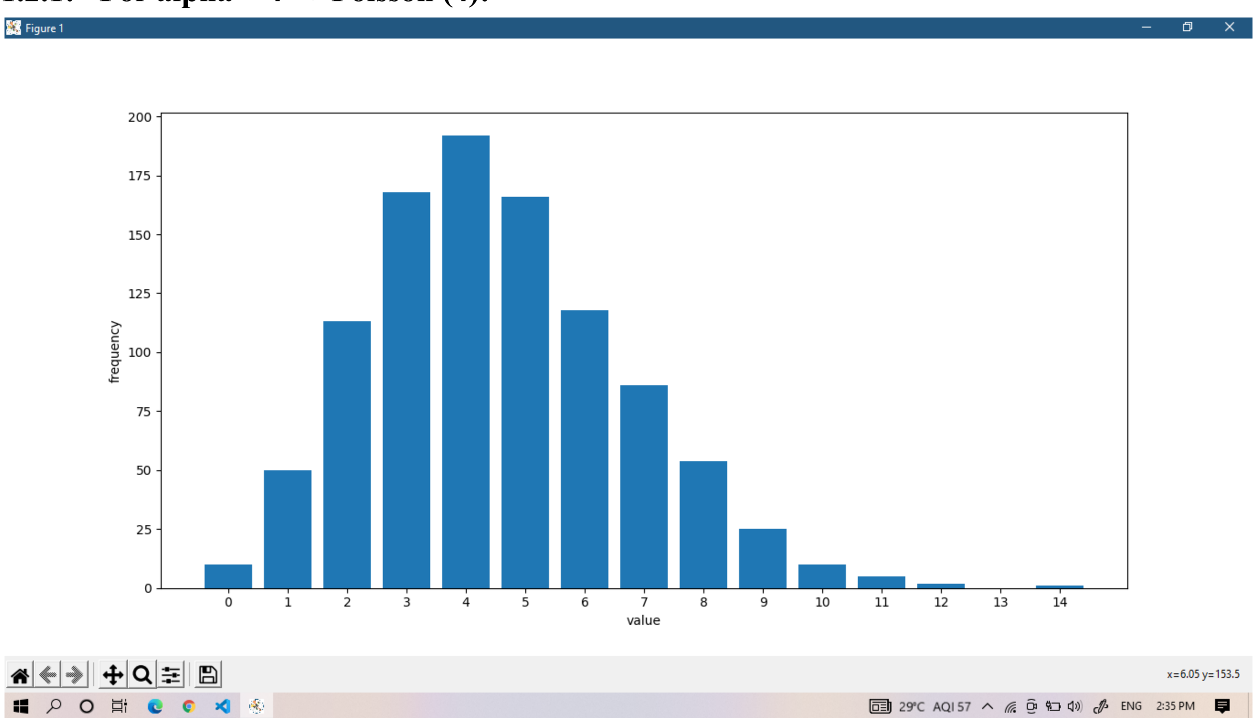
Solution 1.2. Change alpha for Poisson RV:

Parameters for **Poisson** RV:

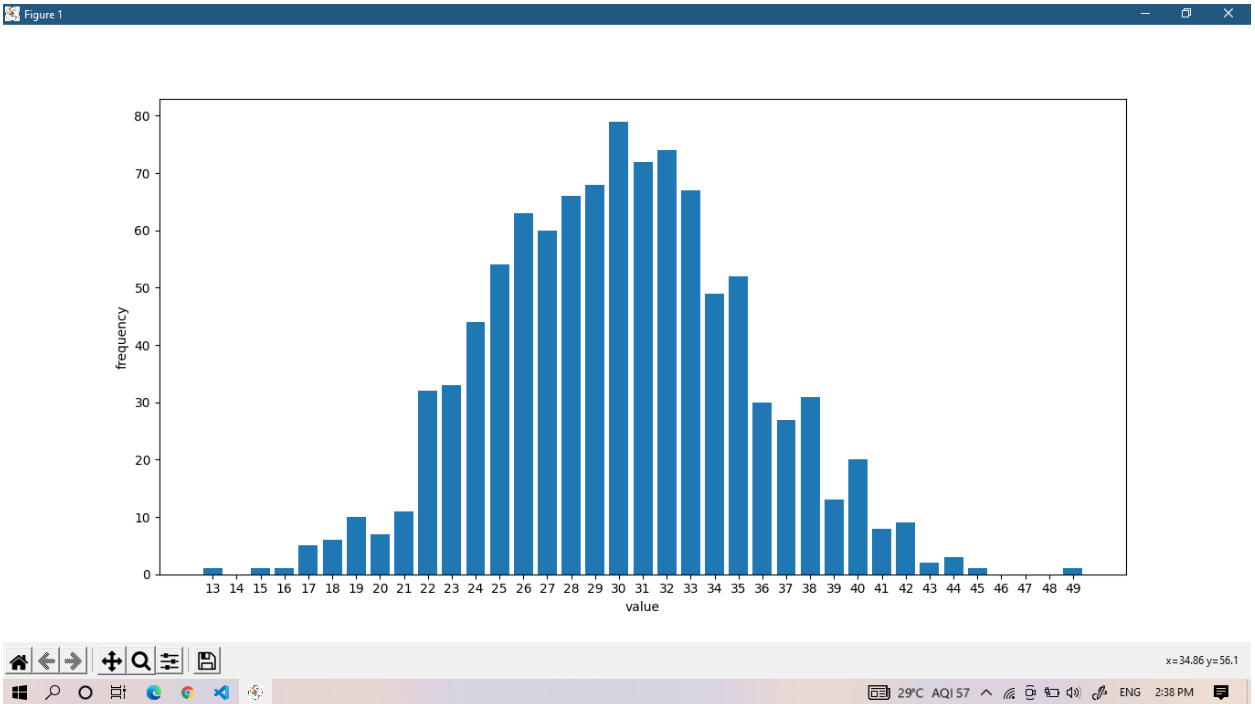
1. Sample size = 1000
2. Value of alpha : varying.

```
#@title  Poisson RV
from scipy.stats import poisson
import matplotlib.pyplot as ok
size = 1000
def Poisson(alpha):
  rv = poisson.rvs(alpha, size =  size)
  bin_edges = np.arange(rv.min()-0.5, rv.max()+1.5, 1)
  ok.hist(rv, bins = bin_edges, rwidth = 0.8)
  ok.xticks(np.arange(rv.min(), rv.max()+1, 1))
  ok.xlabel('value')
  ok.ylabel('frequency')
  ok.show()
```

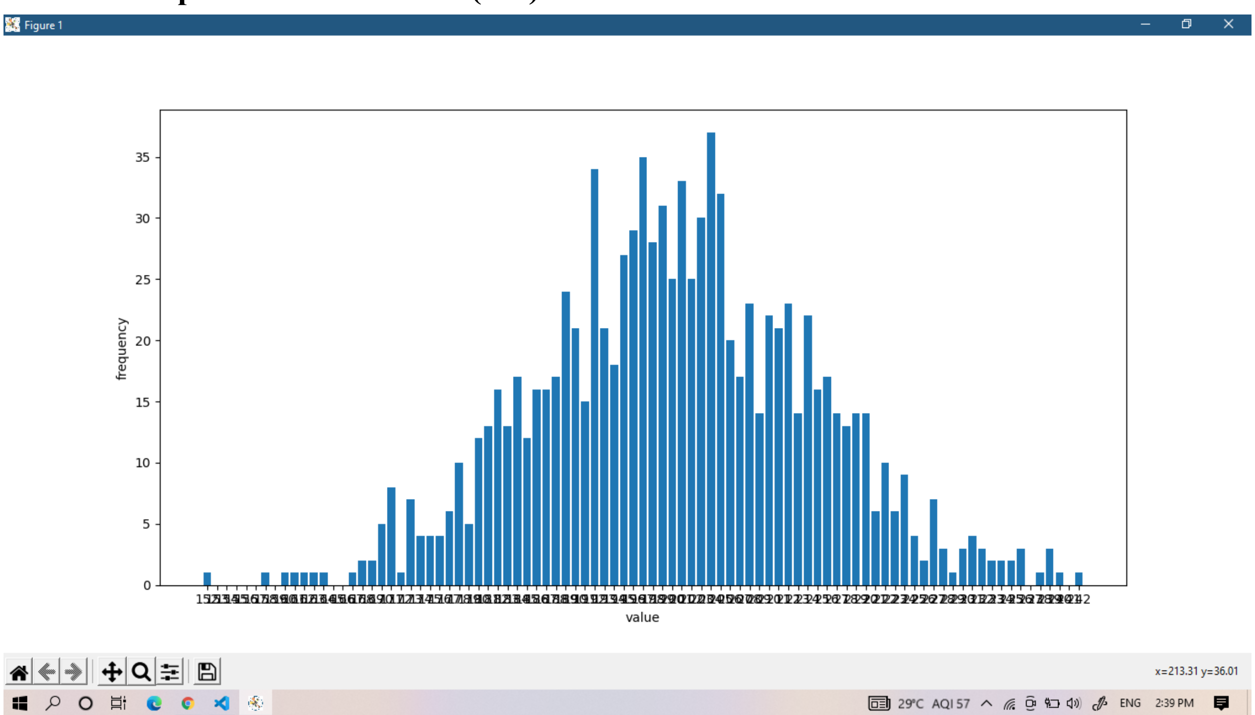
1.2.1: For alpha = 4 ⇒ Poisson (4):



1.2.2. For alpha = 30 ⇒ Poisson (30):



1.2.3. For alpha = 200 ⇒ Poisson (200):



…………………………………………………………………………………………………....
Explanation:

For other fixed parameters, when we increase alpha the RV gets more centralized. That means the frequency of values of RV decreases on both sides about a certain value of RV, so is the nature of plots .

…………………………………………………………………………………………………....

……………………………………………………………………………………………………

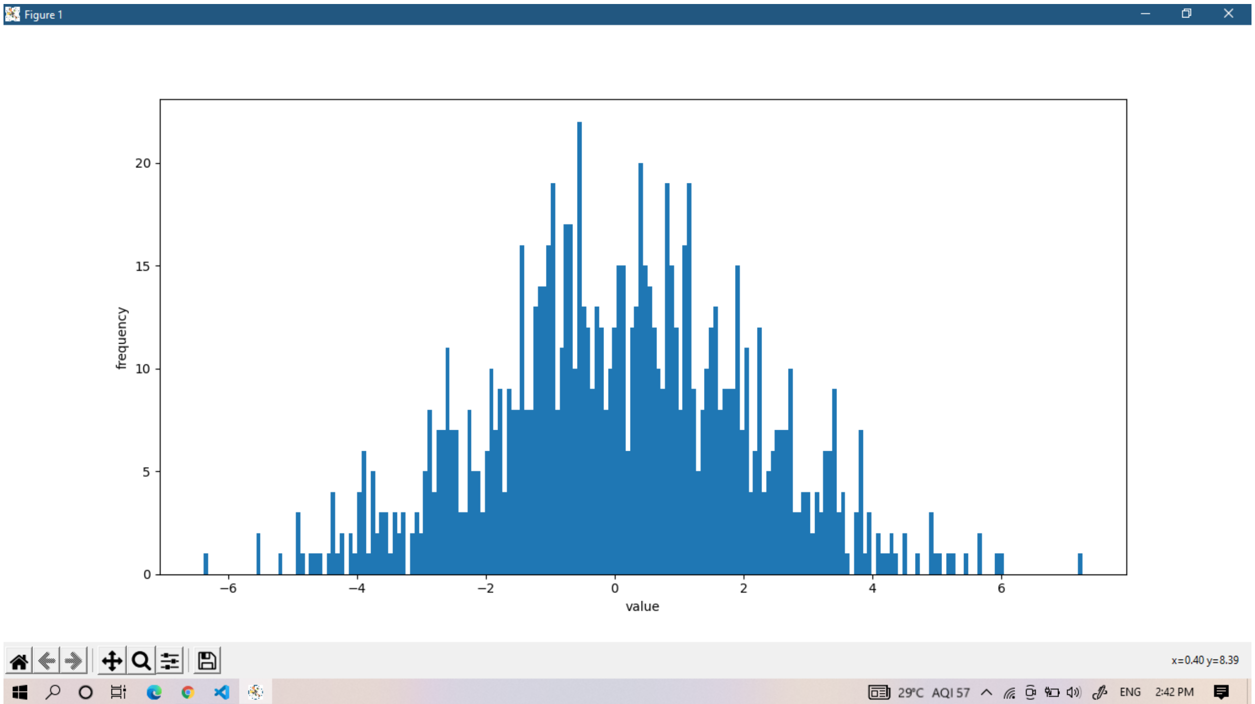
Solution 1.3. **Change mean and standard deviation for Gaussian RV:**

Parameters for **Gaussian** RV:

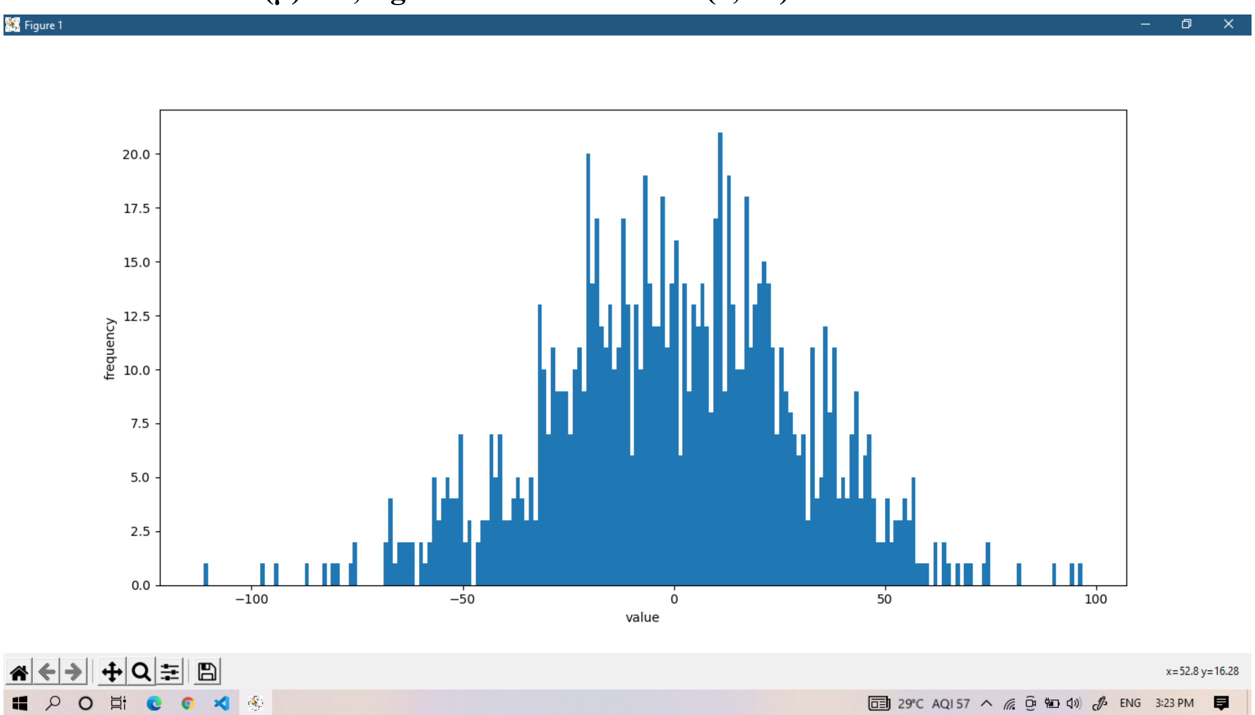
1. Sample size = 1000
2. Value of mean : varying
3. Value of standard deviation : varying

```
#@title Gaussian
import matplotlib.pyplot as ok
import numpy as np
import random

size = 1000

def Gaussian(mu, sig):
  nums=[]
  for i in range(size):
      nums.append(random.gauss(mu, sig))

  ok.hist(nums, bins= 200)
  ok.xlabel('value')
  ok.ylabel('frequency')
  ok.show()
```

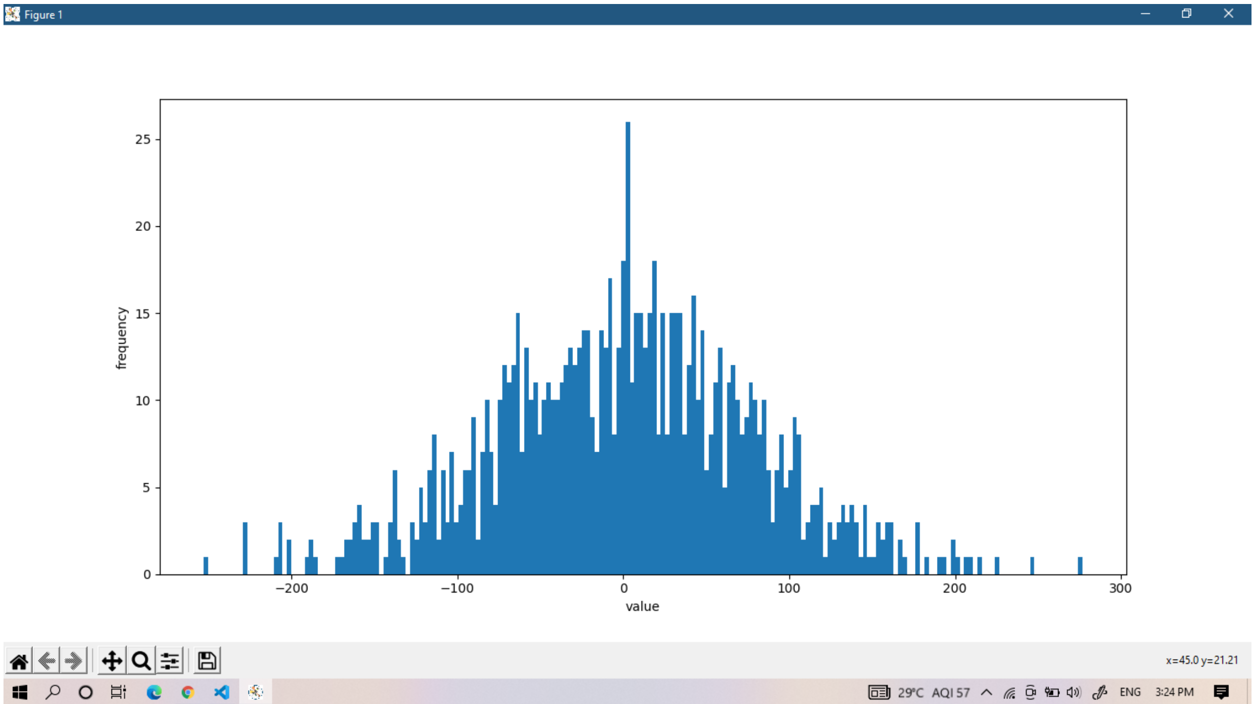
1.3.1: For fix mean(μ) = 0, sigma = 2 ⇒ Gaussian (0, 2):



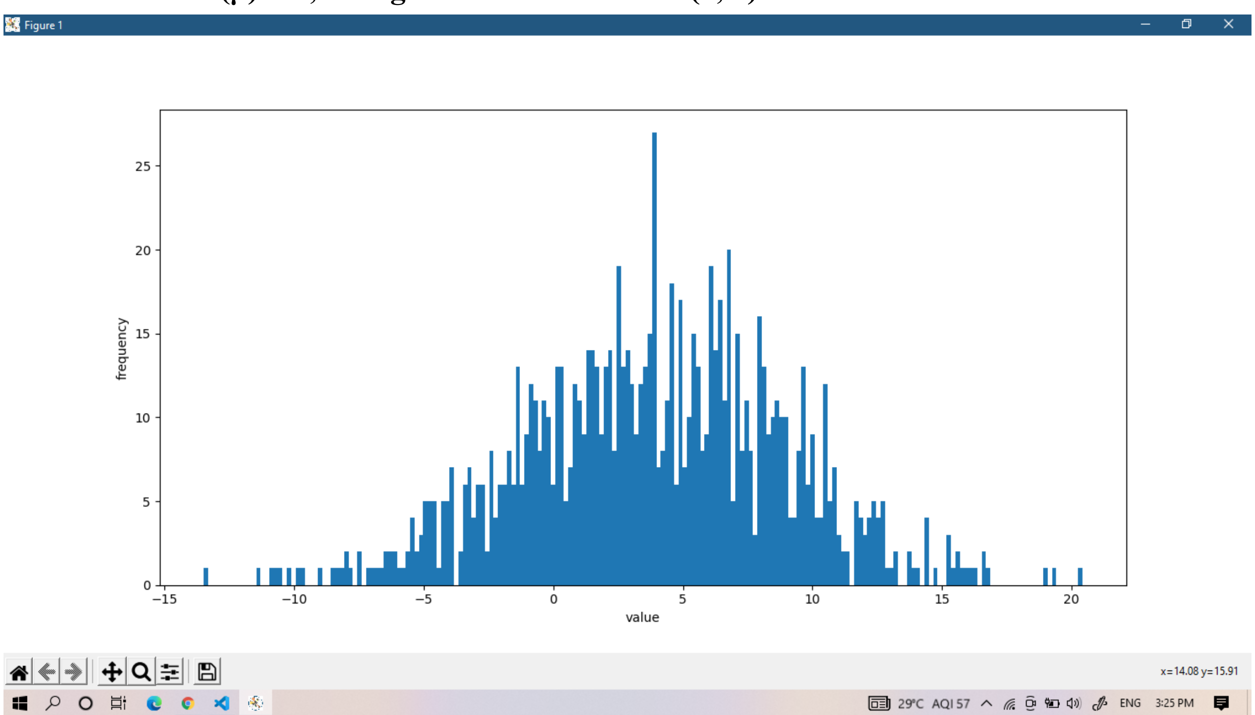
1.3.2. For fix mean(μ) = 0, sigma = 30 ⇒ Gaussian (0, 30):



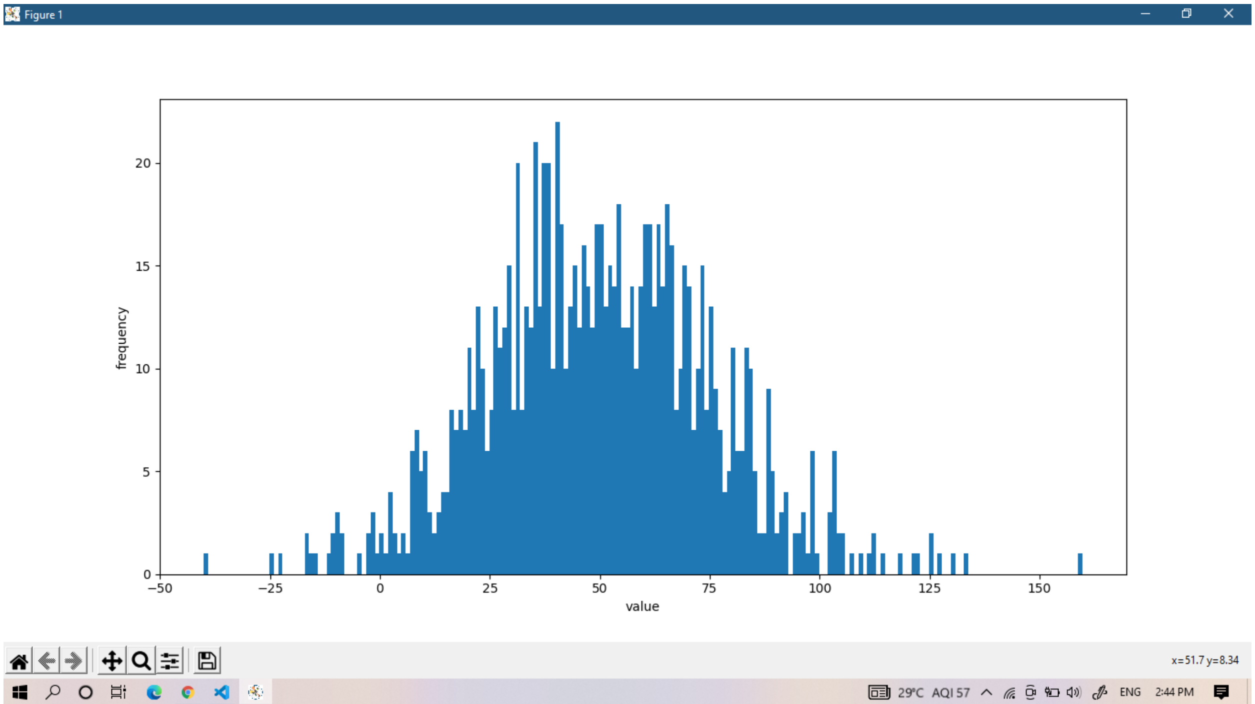
1.3.3. For mean(μ) = 0, sigma = 80 ⇒ Gaussian (0, 80):



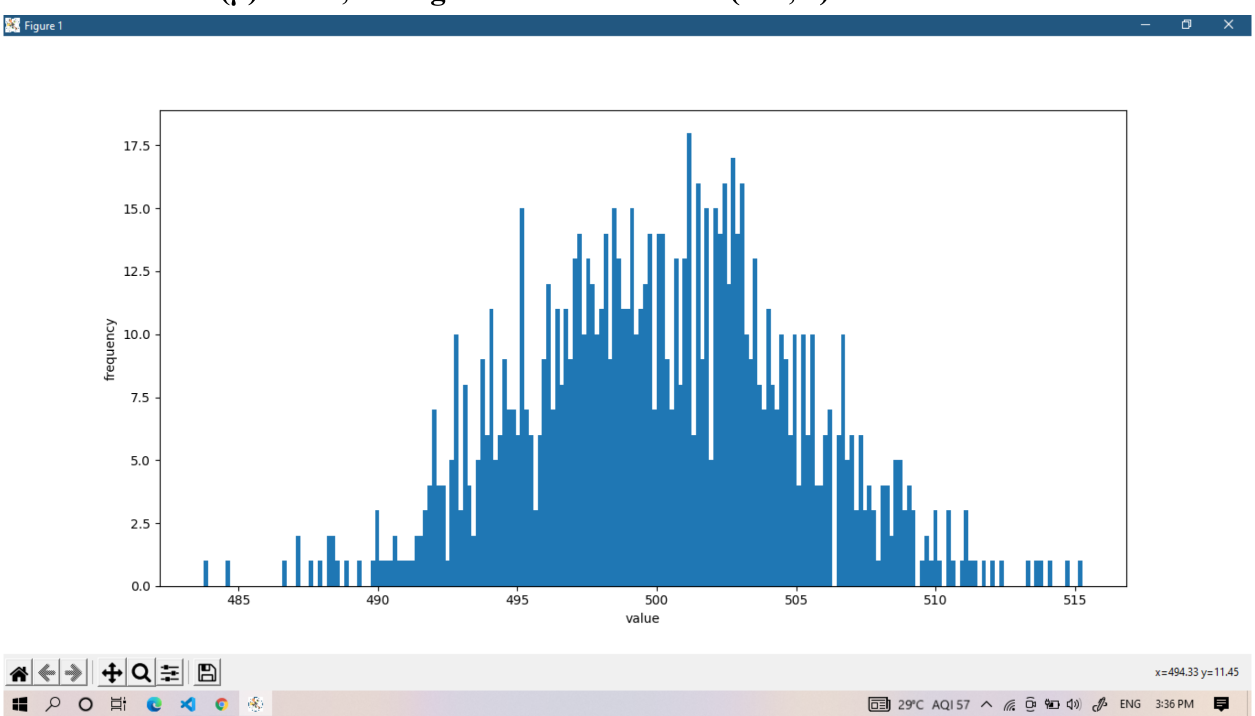
1.3.4. For mean(μ) = 4, fix sigma = 5 ⇒ Gaussian (4, 5):



1.3.5. For mean(μ) = 40, fix sigma = 5 ⇒ Gaussian (40, 5):



1.3.6. For mean(μ) = 500, fix sigma = 5 ⇒ Gaussian (500, 5):



…………………………………………………………………………………………………....

Explanation:

Fix mean(mu):

For a fixed value of mean, when we increase the standard deviation the frequency for the values those are more than mean starts to increase parallely the frequency for the values those are less than mean starts to increase as well, that means the diameter of the bell of the plot increases across centre approximately in equal proportion. The above graphs show approximately the same behaviour.

Fix Standard Deviation(sig):

For a fixed value of Standard Deviation, as we increase the centre of plot shifts to the value of mean, the plots here are showing this property as well.

…………………………………………………………………………………………………....

…………………………………………………………………………………………………

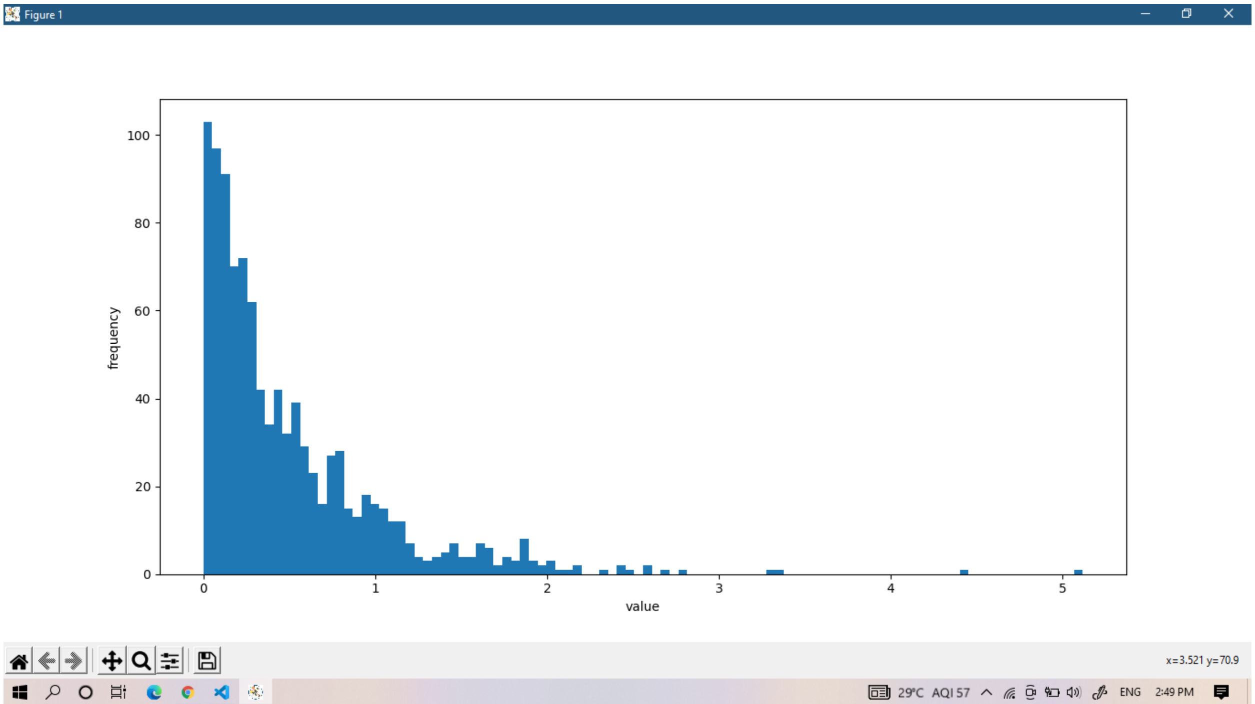
Solution 1.4. **Change lambda(λ) for Exponential RV:**

Parameters for **Exponential** RV:

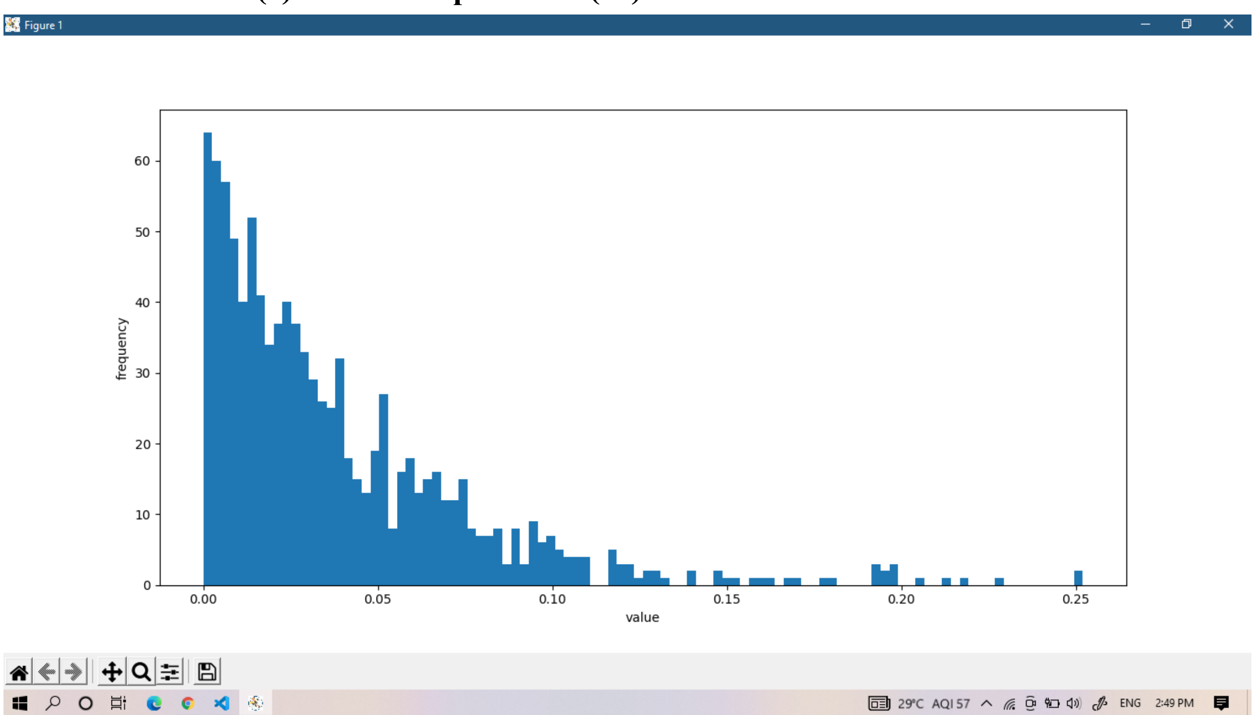
1. Sample size = 1000
2. lambda(λ) : varying

```
#@title Exponential
import matplotlib.pyplot as ok
from numpy import random

size=1000

def Exponential(lda):
  data = random.exponential(1.0/lda, size=size)
  ok.hist(data, bins = 100)
  ok.xlabel('value')
  ok.ylabel('frequency')
  ok.show()
```

1.4.1: For lambda(λ) = 2 $\Rightarrow$ Exponential (2):

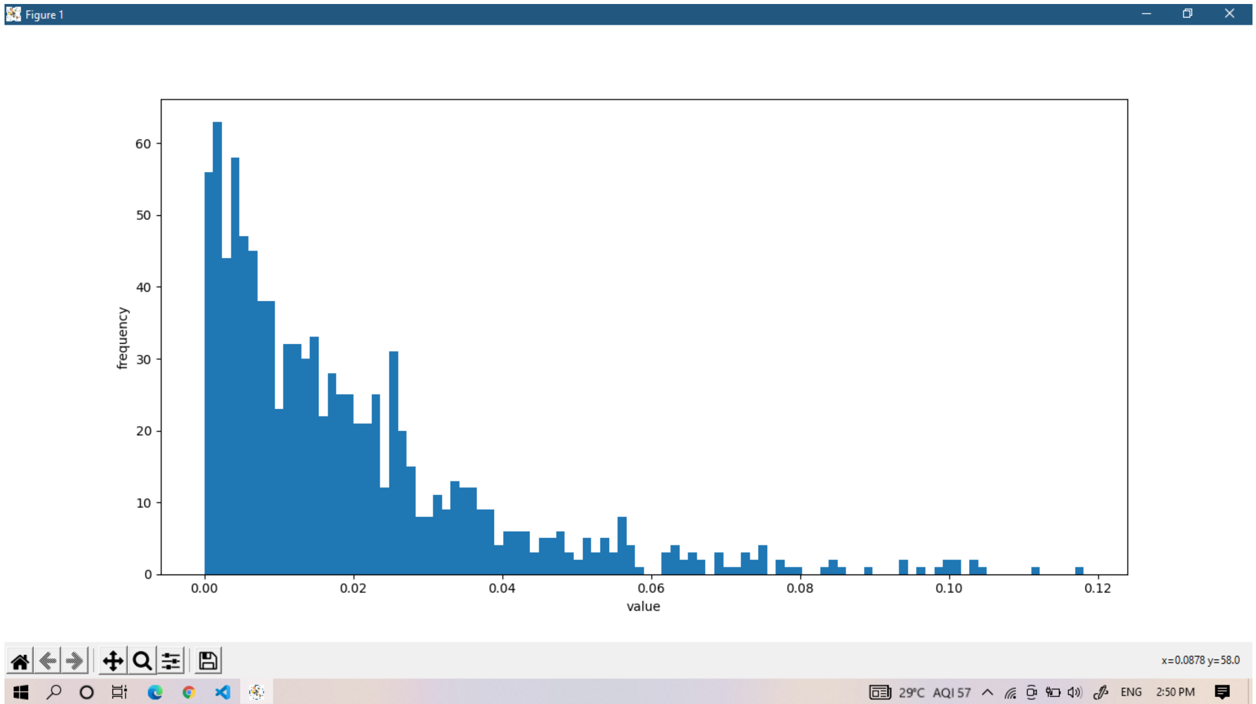


1.4.2. For lambda(λ) = 25 $\Rightarrow$ Exponential (25):



……………………………………………………………………………………………….

1.4.3. For lambda(λ) = 100 ⇒ Exponential (100):



……………………………………………………………………………………………….

Explanation:

Here we can see that frequency for lower value of RV is increasing and for higher values of RV the frequency is decreasing as the value **λ** increase.

……………………………………………………………………………………………….

……………………………………………………………………………………………….

###

Solution 2:

###

....

A. Demonstrate that the plot of 3 independent exponential RVs is erlang RV.

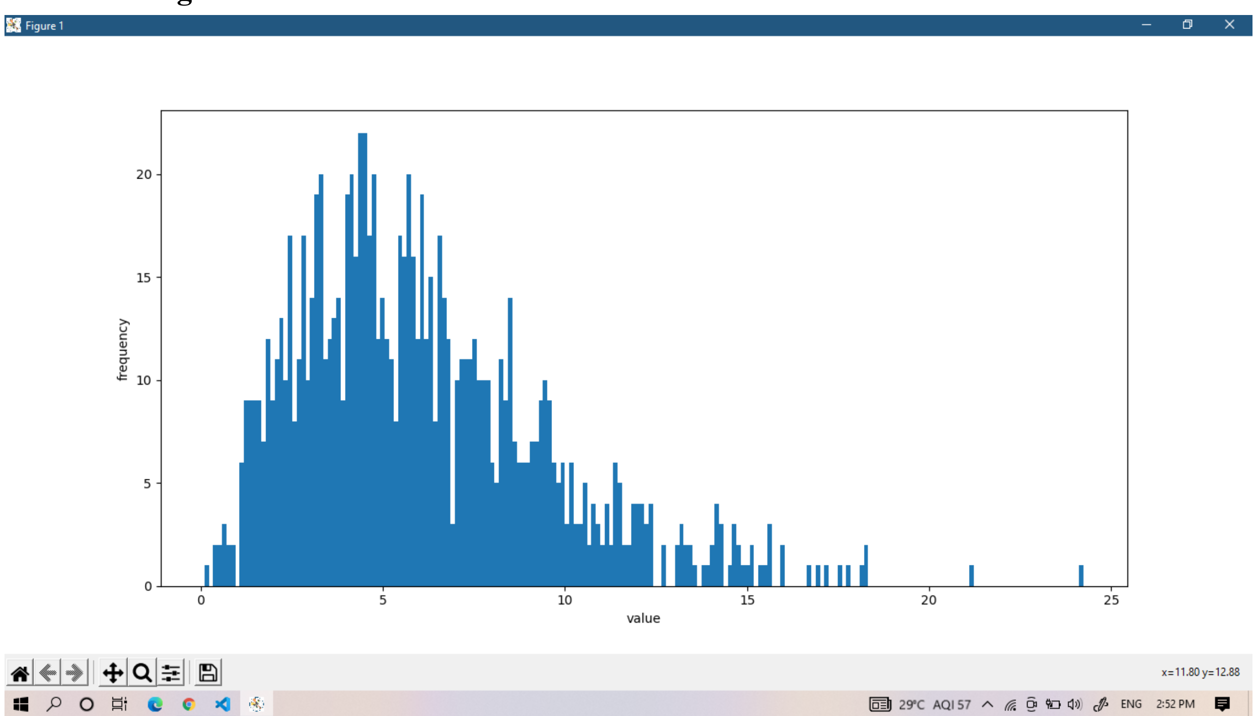
....

1. Plot an erlang RV (say n=3 and 1/λ = 2) :

```
#@title Erlang RV
from scipy.stats import erlang
import matplotlib.pyplot as plt

n = 3
lda = 1/2

y = erlang.rvs(n, loc = 0, scale = 1/lda, size = 1000)

plt.hist(y, bins = 200)
plt.xlabel('value')
plt.ylabel('frequency')
plt.show()
```

....

Plot of Erlang:

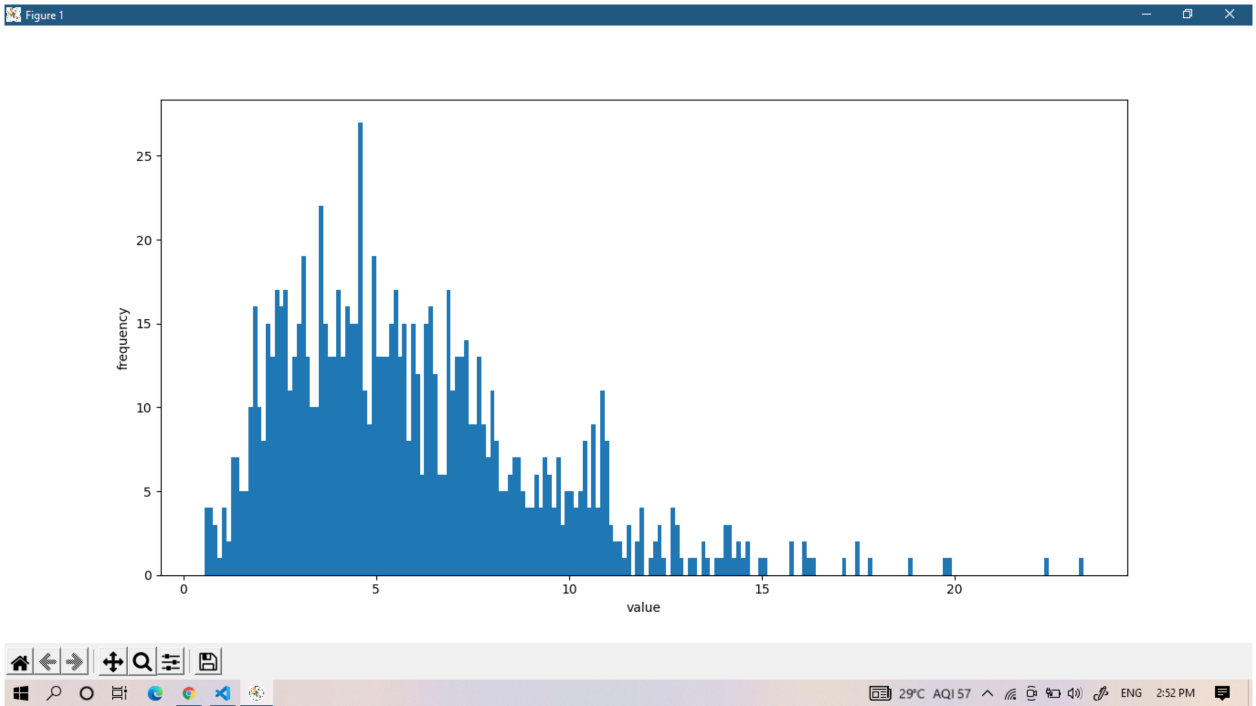


……....

2. Now generate n (=3) independent exponential RV ($1/\lambda = 2$).
3. Sum all n (=3) to generate a sample for a new RV Y.
4. Repeat 2. and 3. to generate m (say 1000) samples of Y.

```
#@title plot of 3 independent exponential RVs
import matplotlib.pyplot as plt
from numpy import random

# print(data)
Y = []

n = 3
lda = 1/2

#4 generate 1000 samples.
for i in range(1000):
  #2 generate 3 samples of exponential rv
  data = random.exponential(1/lda, size=n)
  #3 generate a RV by by summing up all 3 samples and store in Y
  Y.append(sum(data))

# plot Y
plt.hist(Y, bins = 200)
plt.xlabel('value')
plt.ylabel('frequency')
plt.show()
```

Plot of 3 independent exponential RVs:



Comparison of the 2 plots: The plots we got, not exactly but are nearly same that we can easily catch the behaviour of both the plots like the frequency of the values are increasing sharply for a while moving in +ve x-axis and after a certain value frequency start to decreases and when moving forward further on +ve x-axis frequency for a values decrease slowly in comparison to the frequency of the neighbour values. Hence the nature of both the plots is the same.

Observation: As the nature of both of the plots is quite similar, we can say that the plot of 3 independent exponential RVs is erlang RV.

……………………………………………………………………………………………...

B. Demonstrate the Central Limit Theorem using the sum of Binomial RVs.

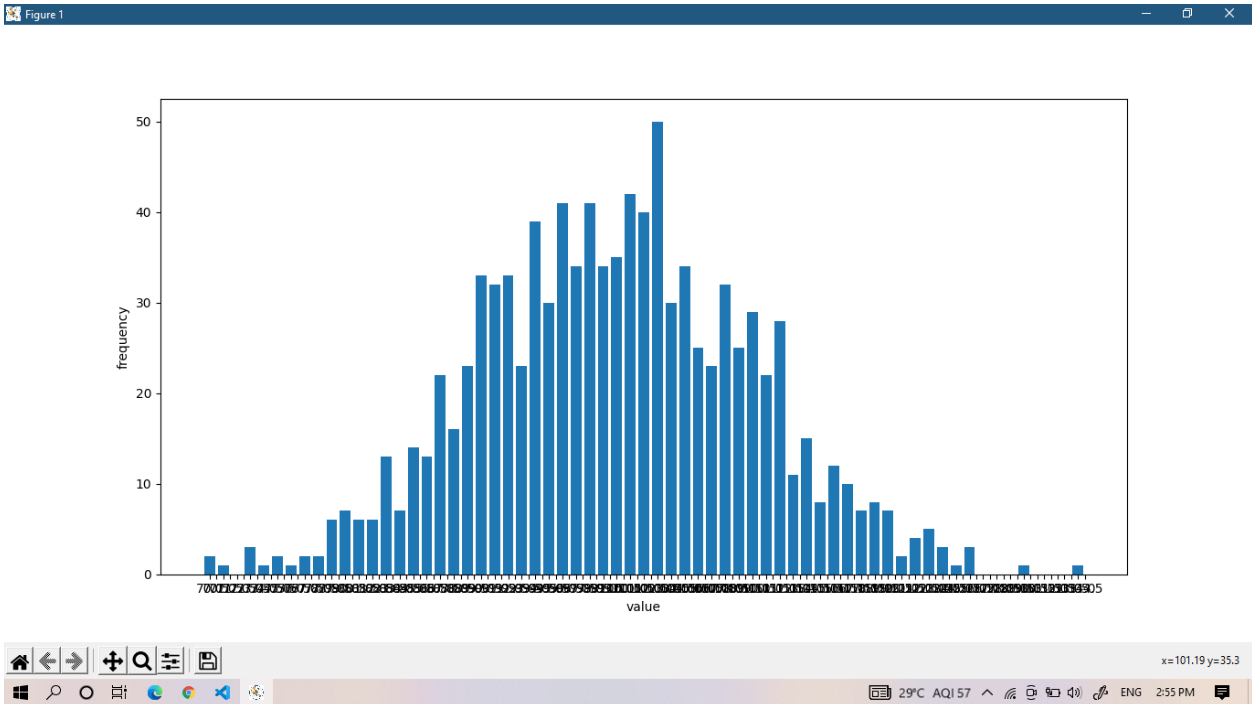
……………………………………………………………………………………………

1. Generate m (say 100) independent samples of Binomial RV with any parameters (n, p).
2. Sum all m (=100) to generate a sample for a new RV Y.
3. Repeat 1. and 2. to generate say 1000 samples of Y.

```
#generate m (say 100) independent samples of Binomial RV with any
parameters

from scipy.stats import uniform
import matplotlib.pyplot as plt
import numpy as np

#1/2 generate sum of 100 binomial variable generator
def binrvs(n, p):
  binomial = np.empty(100, dtype=int)

  for i in range(100):
    data_uniform = uniform.rvs(size=n, loc=0, scale=1)
    bernoulli = np.where(data_uniform < p, 1, 0)
    binomial[i] = np.sum(bernoulli)

  return sum(binomial)

#3 generate say 1000 samples of Y.
folks = np.empty(1000, dtype=int)
for i in range(1000):
  folks[i]  = binrvs(69, 0.0145)

#4 plot
bin_edges = np.arange(folks.min()-0.5, folks.max()+1.5, 1)
plt.hist(folks, bins = bin_edges, rwidth = 0.8)
plt.xticks(np.arange(folks.min(), folks.max()+1, 0.5))
plt.xlabel('value')
plt.ylabel('frequency')
plt.show()
```

....

4. **Plot:** Plot these values of Y.



....

5. **What does the shape of your plot look like?**

⇒ The shape of the plot looks like a bell. The shape of the plot of the Gaussian RV also looks like a bell, hence the plot can be compared with the Gaussian RV.

⇒ As the shape of the curve is similar to Gaussian hence it demonstrates the Central Limit Theorem.

....

##