PSTAT160ASpring2020 Python HW 2

April 15, 2020

1 Python Homework 1

Release date: Friday, April 13rd **Due date:** Friday, April 17th, 11:59 p.m. via GauchoSpace

Instruction: Please upload your pdf or html file with your code and result on GauchoSpace with filename "PythonHW1_YOURPERMNUMBER".

The purpose of this Python Homework is to get a little bit familiar with sampling from a distribution with the **NumPy Package**.

Attention: Don't forget to import the necessary packages!

[1]:
```
import numpy as np
import numpy.random as npr
import matplotlib
import statistics
import math
import scipy.stats as stats
from matplotlib import pyplot
```

1.1 Problem 1 (6 Points)

1. Simulate 100,000 realizations from the binomial distribution with N=1000 trails and success probability p=0.3.

[2]:
```
# binomial with n=1000 and p=0.3
n=1000
p=0.3
size=100000
xbin=np.random.binomial(n,p,size)
print(xbin)
```

```
[307 276 286 … 280 277 324]
```

2. Compute the empirical mean and the empricial standard deviation of your sample and compare these values with the theoretical values.

[3]:
```
# calculate the empirical mean
#print("Standard Deviation of sample is % s " % (np.mean(xbin)))
```

```
mean=np.mean(xbin)
print(mean)
```

```
300.01067
```

[4]:
```
#theoretical value
mean1=n*p
print(mean1)
```

```
300.0
```

3. Plot a histogram of your sample with the absolute number of counts for each bin. Choose 25 bins.

[5]:
```
#emprical std
#print("Standard Deviation of sample is % s " % (statistics.stdev(xbin)))
std=np.std(xbin)
print(std)
```

```
14.530878024094072
```

3. Plot a histogram of your sample with the absolute number of counts for each bin. Choose 25 bins.

[6]:
```
#theoretical std
std1=math.sqrt(n*p*(1-p))
print(std1)
```

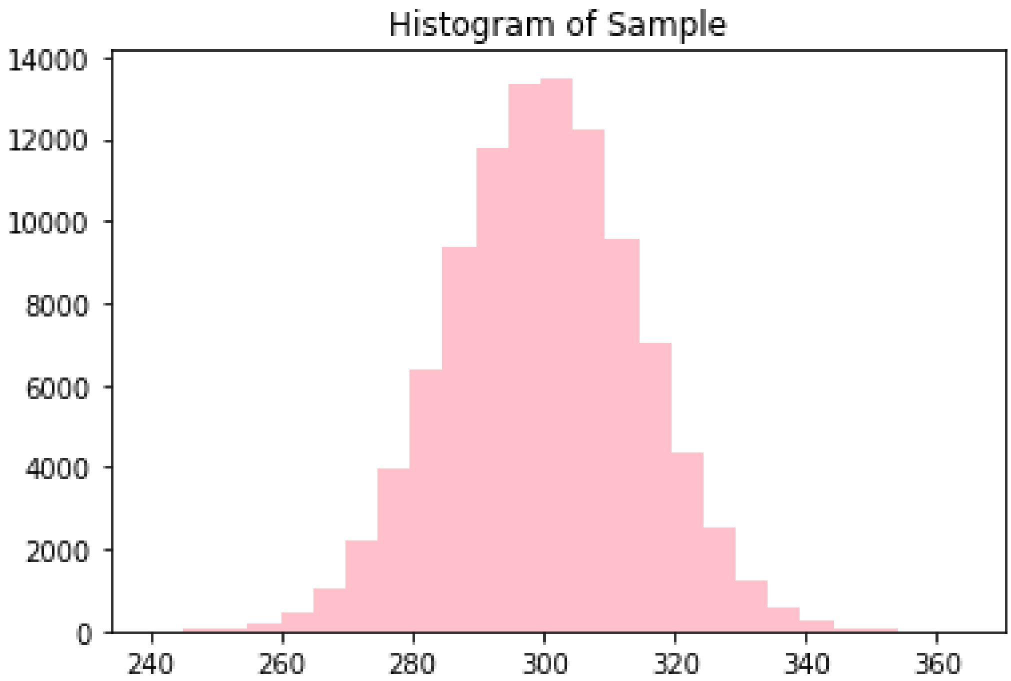
```
14.491376746189438
```

[7]:
```
print("According to my empirical mean and the empricial standard deviation and␣
↪compare these values with the theoretical values, I can see that they are␣
↪pretty close to each other.")
```

```
According to my empirical mean and the empricial standard deviation and compare
these values with the theoretical values, I can see that they are pretty close
to each other.
```

3. Plot a histogram of your sample with the absolute number of counts for each bin. Choose 25 bins.

[8]:
```
# histogram
pyplot.hist(xbin,
        bins = 25,
        facecolor = 'pink')
pyplot.title('Histogram of Sample')
pyplot.show()
```



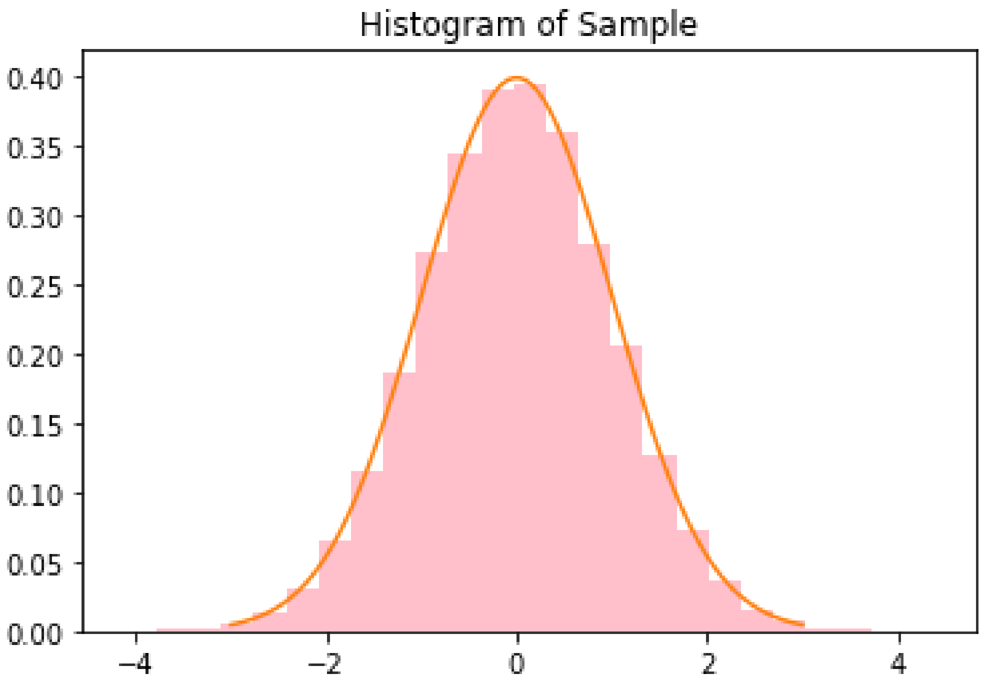
4. Standardize your sample, that is, subtract the emprical mean and divide by the empricial standard deviation.

[9]:
```
# standardize sample
standardize=(xbin-mean)/std
print(standardize)
```

```
[ 0.48099846 -1.65238948 -0.96419982 … -1.37711362 -1.58357052
  1.65092088]
```

5. Plot a histogram of your standardized sample with the counts normalized to form a probability density. Choose again 25 bins. Compare your histrogram with the density of the standard normal distribution by inserting its density into the histogram plot.

[10]:
```
# # histogram of standardized sample
pyplot.hist(standardize,
        bins = 25,
        facecolor = 'pink',density=True)
norm=np.linspace(-3,3,100)
pyplot.plot(norm,stats.norm.pdf(norm))
pyplot.title('Histogram of Sample')
pyplot.show()
```



[11]:
```
print("they look mostly the same")
```

```
they look mostly the same
```

1.2 Problem 2 (4 Points)

1. Implement the simulation of a biased 6-sided die which takes the values 1,2,3,4,5,6 with probabilities 1/8,1/12,1/8,1/12,1/12,1/2.

[12]:
```
# six side die
def simulation(n):
    return np.random.choice(range(1,7),n,p=[1/8,1/12,1/8,1/12,1/12,1/2])
```

2. Plot a histrogramm with 1,000,000 simulations to check if the relative counts of each number is approximately equal to the corresponding specified probabilities.

Remark: Specify the bins of your histogram correctly.

[13]:
```
# check equality
x=simulation(1000000)
pyplot.hist(x,bins=6,density=True,facecolor = 'pink')
pyplot.title("problem2 simulation")
pyplot.xlabel("value of dice")
pyplot.ylabel("probability")
pyplot.show()
```

[14]:
```
print("The probability showed in the histrogram for each die looks is␣
↪approximately same as the corresponding probabilities that are presupposed␣
↪in the problem.")
```

```
The probability showed in the histrogram for each die looks is approximately
same as the corresponding probabilities that are presupposed in the problem.
```