Documentation

1. (1.ipynb)

1)

$E[X_n] = E[X_1 + X_2 + \cdots + X_n]$ where $X_1, X_2, \ldots X_n$ are indepent trials

$E[X_n] = \sum E[X_i]$

$\boxed{E[X_n] = 0}$

$E[X] = 1 \times \frac{1}{2} - 1 \times \frac{1}{2} = 0$

$E[X^2] = 1 \times \frac{1}{2} + 1 \times \frac{1}{2} = 1$

$E[X_1 X_2] = E[X_1] \cdot E[X_2] = 0$

$$Var(X_n) = E[X_n^2] - (E[X_n])^2$$

$$= E[X_n^2]$$

$$E[X_n^2] = E[(X_1 + X_2 + \cdots + X_n)^2]$$

$$= E\left[\sum_{i=1}^{n} X_i^2 + \sum_{i \neq j} X_i X_j\right]$$

$$= \sum E[X_i^2] + \sum E[X_i X_j]$$

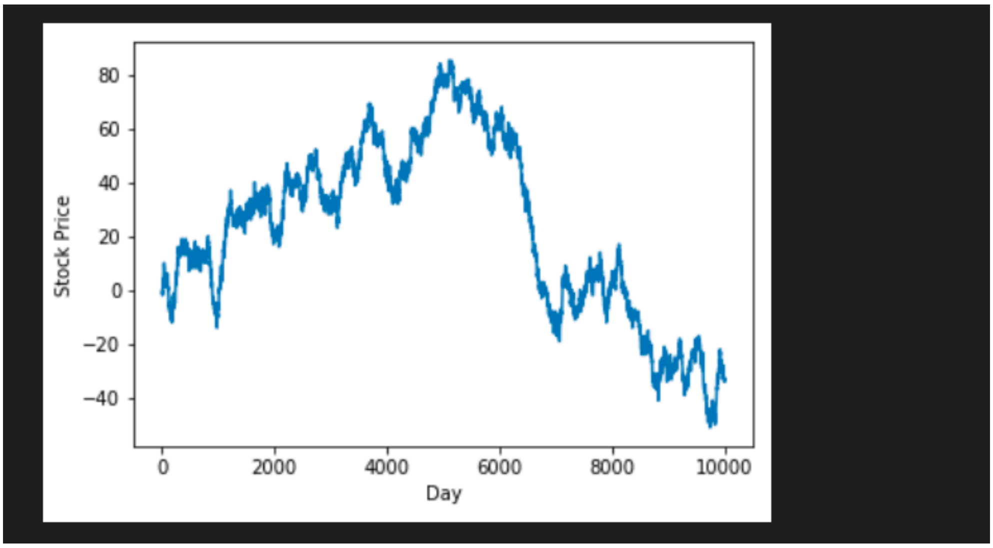
$$= n$$

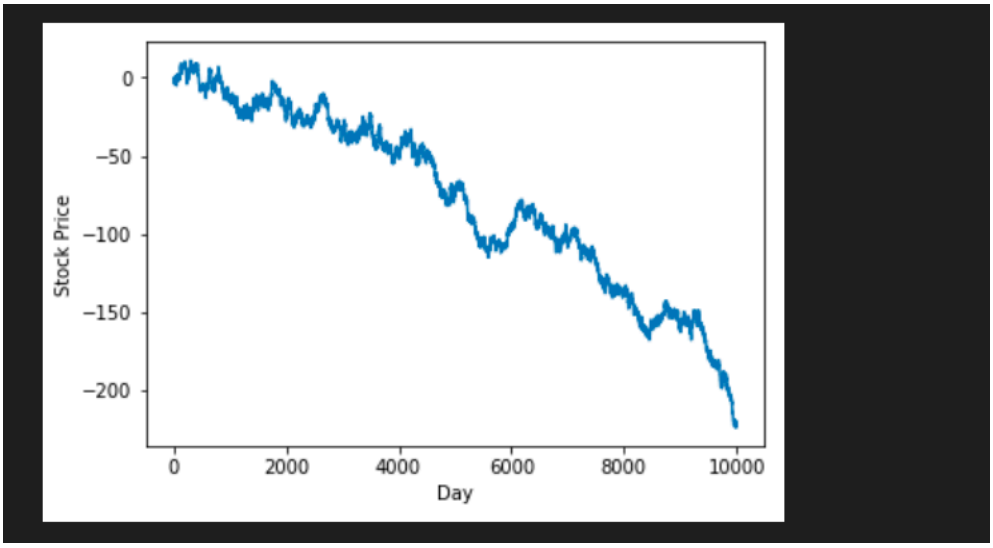
$\boxed{Var(X_n) = n}$

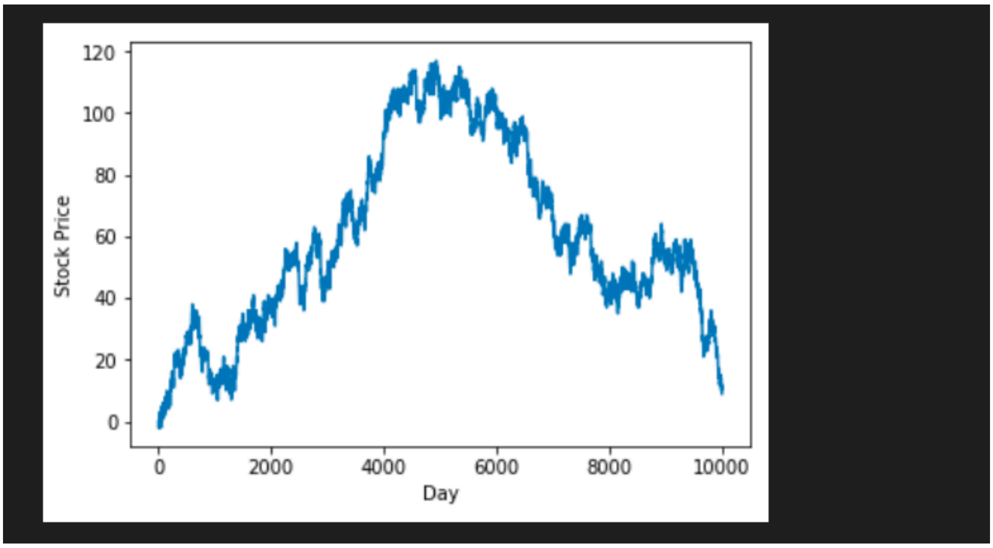
Implementation

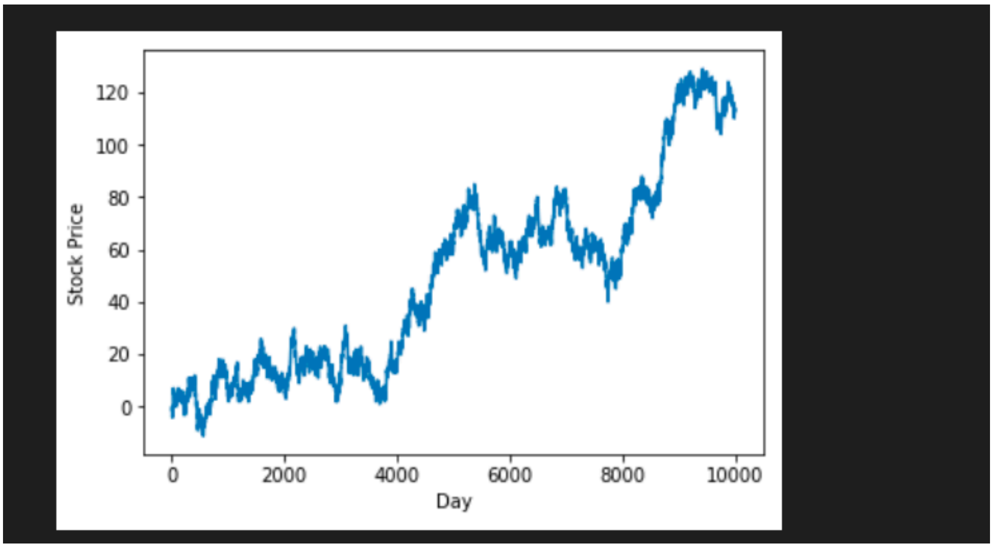
- I randomly choose the stock price at any day for every trial and add it to the stock price at the end of the previous day.
- My initial stock price is 0 hence the -ve stock prices.

Results









Observations

The variance of Xn is n and the mean is 0. As n increases the variance increases. It can be seen that if the graph keeps going high (or low) for too long, its starts to go low (or high) to compensate for it.

As the variance is a function of n, the graph expands vertically with increasing n. Therefore, Xn's at higher n's have very high values at some runs of the code and much lower values in another run.

2. (2.ipynb)

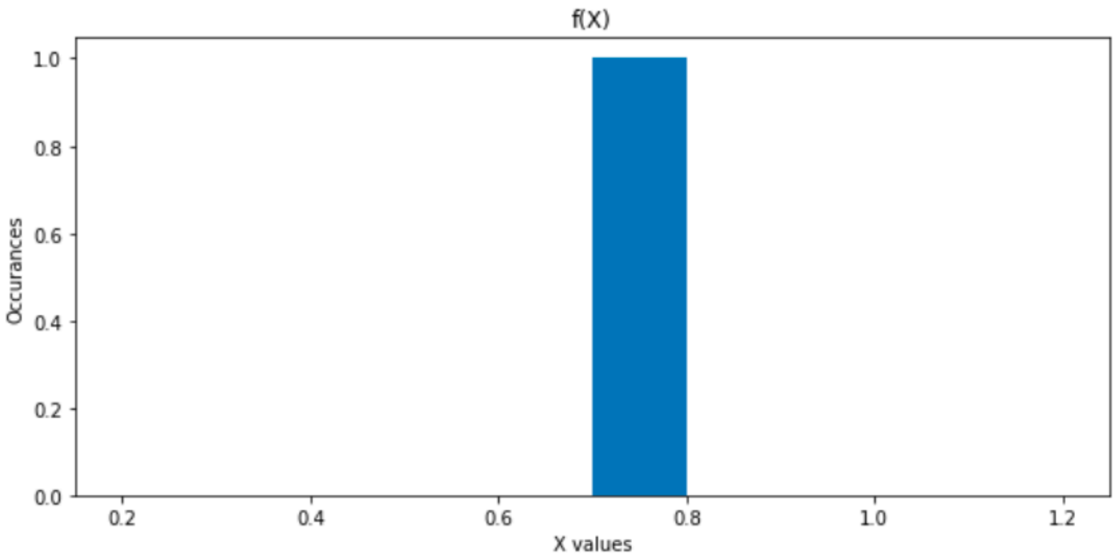
Implementation

- The first graph is the distribution of the statistic X. As X is supposed to be uniformly distributed between 0 and 1, based on the value of n , I compute the frequencies of each X.
- We had to find the distribution of the statistic Xn (mean of n random variables). For each value of n, for 10000 trials I computed different values of Xn and stored their frequencies. If the same value of Xn occurred for another trial, the frequency corresponding that Xn would be increased.
- if n=5, in each trial I find X1,X2,X3,X4,X5 from uniform distribution and find their mean. If this mean is one of the means already there then i increase occurance of this mean.

Results

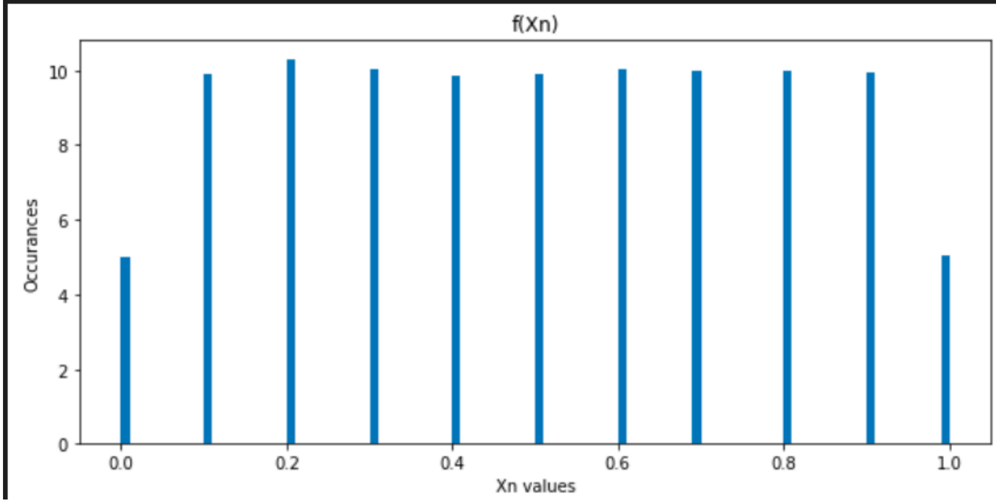
n=1

f(X) -

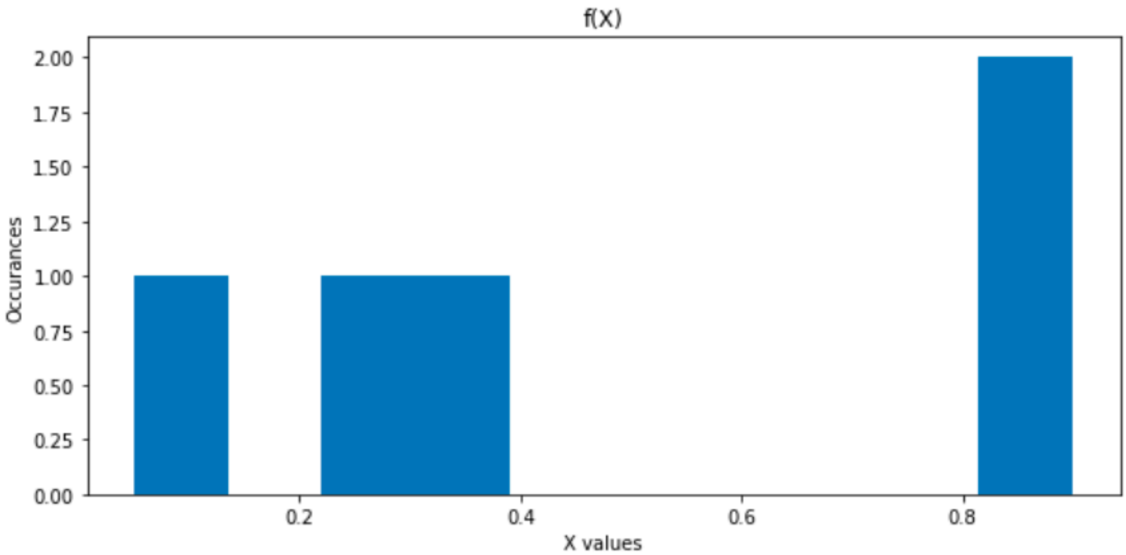


f(Xn) -

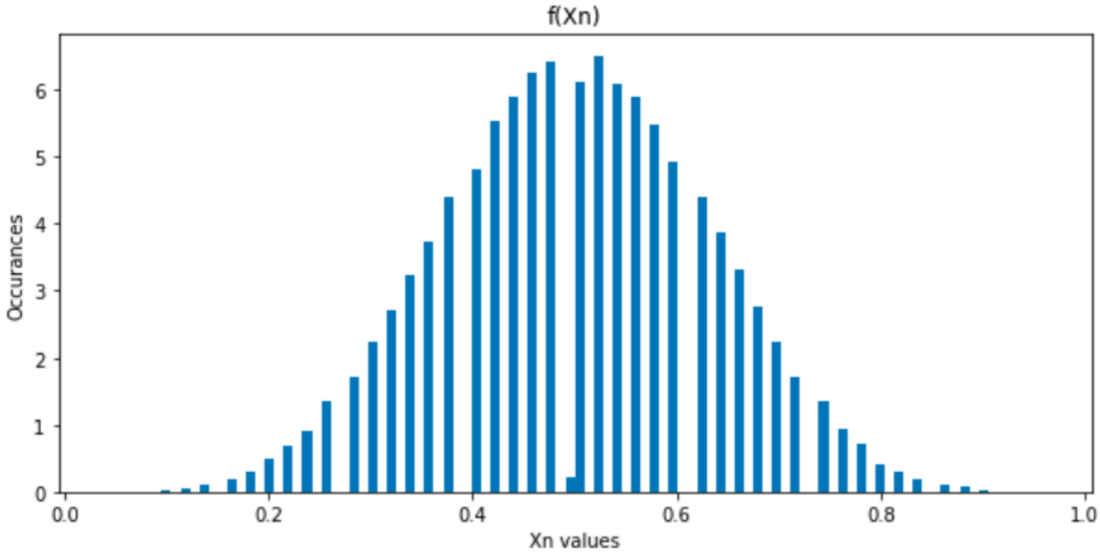
```
('Obsered Expectation = ', 0.499568, 'Variance = ', 0.085192)
('Theoretical Expectation = ', 0.5, ' Variance = ', 0.08333333333333333)
```



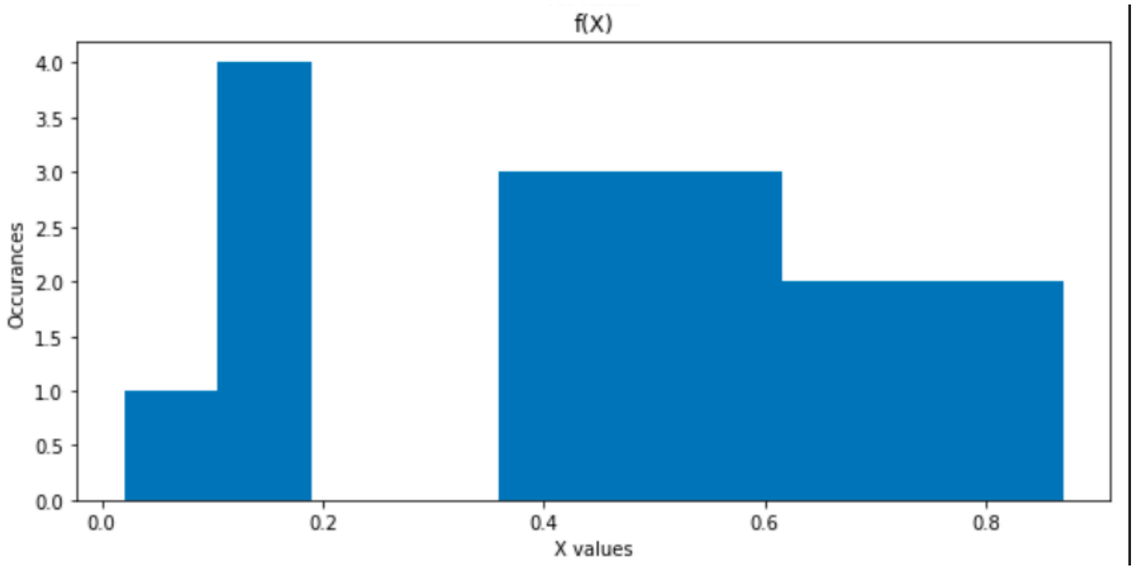
n=5



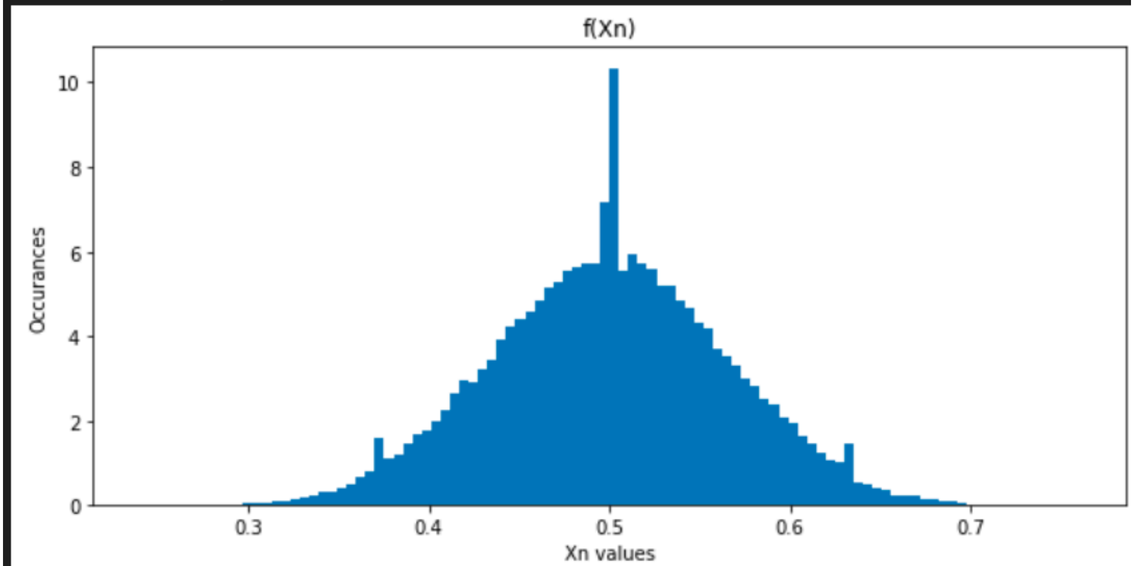
```
'Obsered Expectation = ', 0.500403, 'Variance = ', 0.017088)
'Theoretical Expectation = ', 0.5, ' Variance = ', 0.016666666666666666)
```



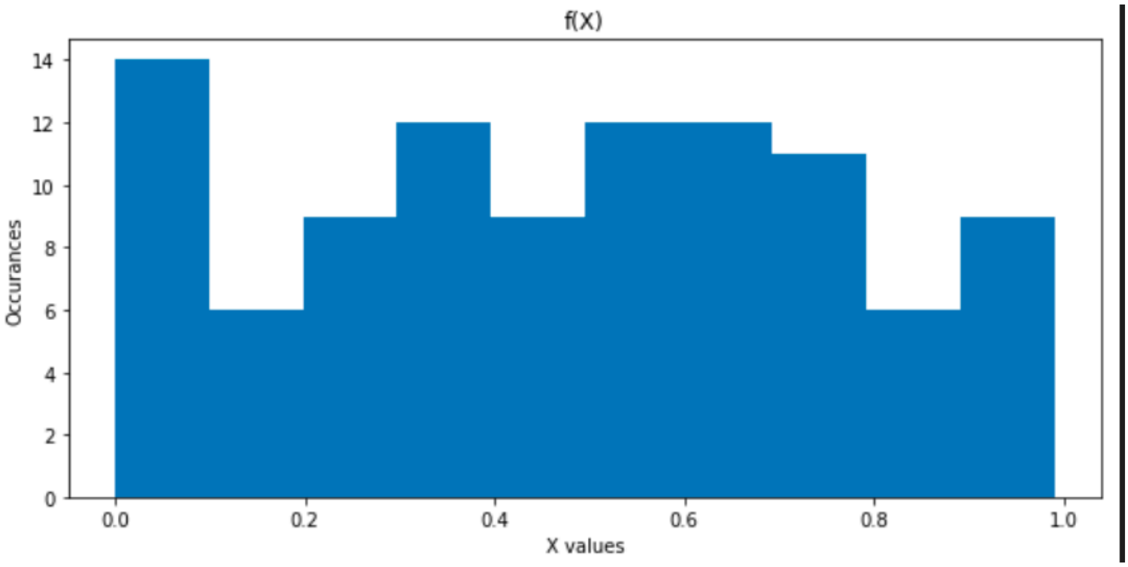
n=20



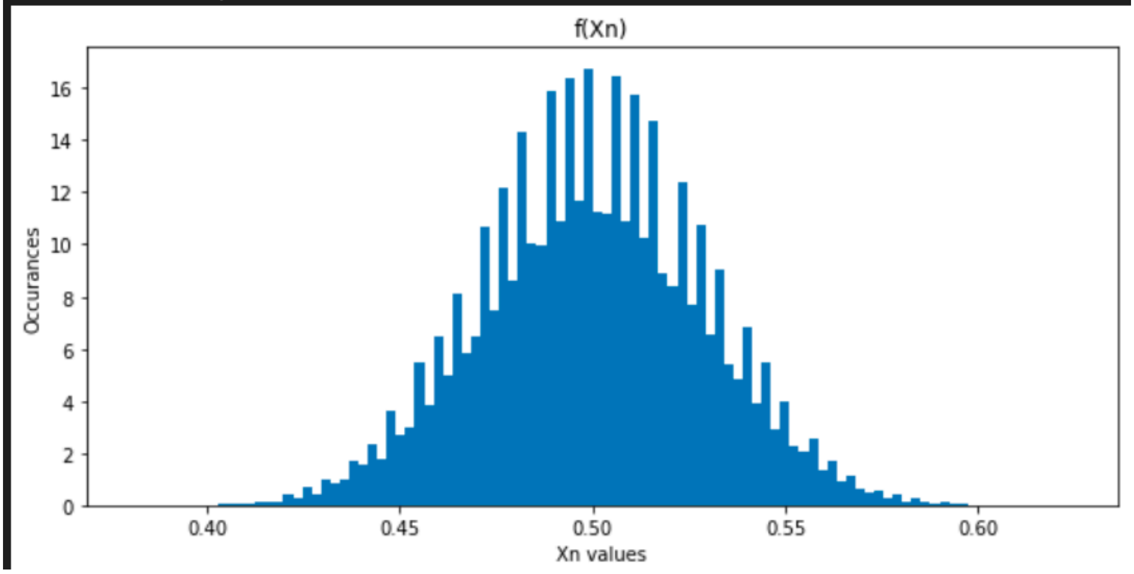

```
('Obsered Expectation = ', 0.50006, 'Variance = ', 0.004226)
('Theoretical Expectation = ', 0.5, ' Variance = ', 0.004166666666666667)
```



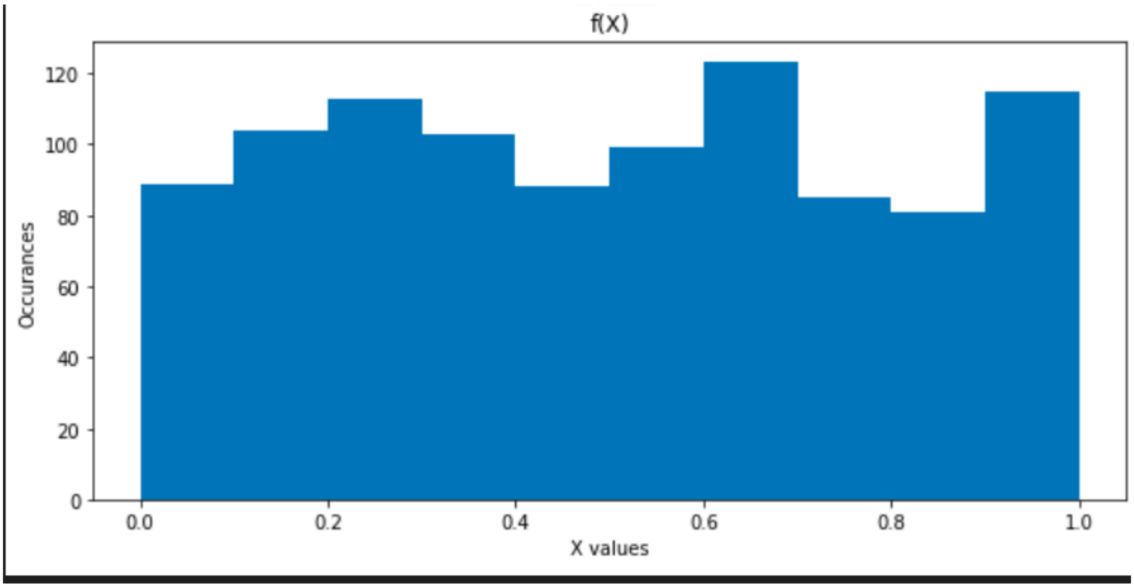
n=100



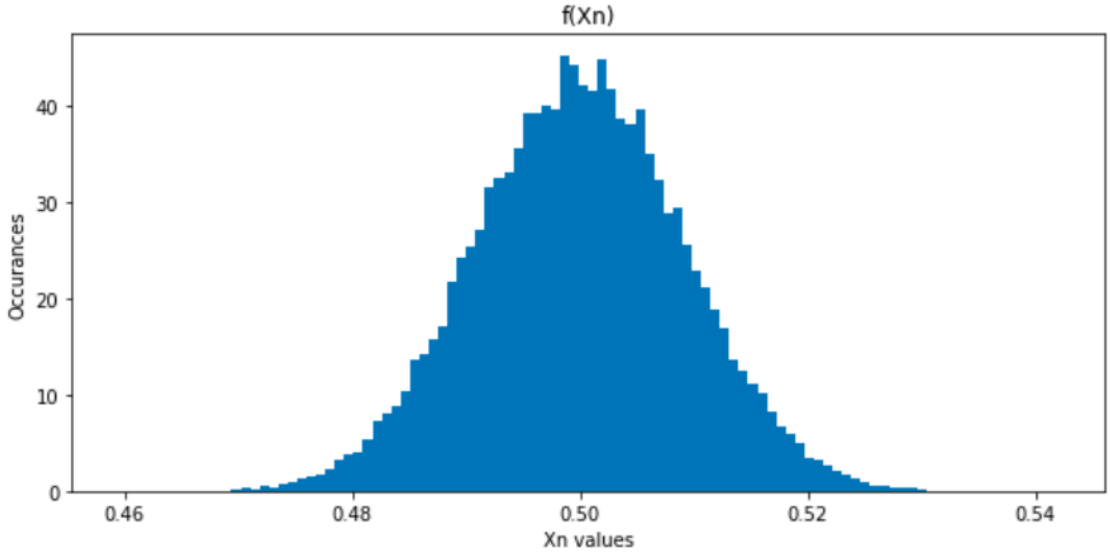

```
('Obsered Expectation = ', 0.500103, 'Variance = ', 0.000846)
('Theoretical Expectation = ', 0.5, ' Variance = ', 0.0008333333333333334)
```



n=1000




```
'Obsered Expectation = ', 0.500017, 'Variance = ', 8.5e-05)
'Theoretical Expectation = ', 0.5, ' Variance = ', 8.333333333333333e-05)
```



Observations

- Graphs of f(X) at different value of n show that the graph becomes more and more uniformly distributed with increasing value of n
- For f(x) as i increase the value of n i..e increase the number of random variables the graph becomes flatter and flatter proving that the distribution is uniform
- Graphs of f(Xn) at different values of n become increasing concentrated around the mean and variance decreases with each n. f(Xn) takes the highest values/occurrences at Xn close to the mean
- The theoretical expectation of Xn is mean of X and the variance of Xn is the variance(X)/ n^2. We can agree with this at every n based on the above outputs
- As the n we chose for f(Xn) is very high, the theoretical mean and variance match at every n with the observed.

3. Random number checker (randomNo.ipynb)

Implementation

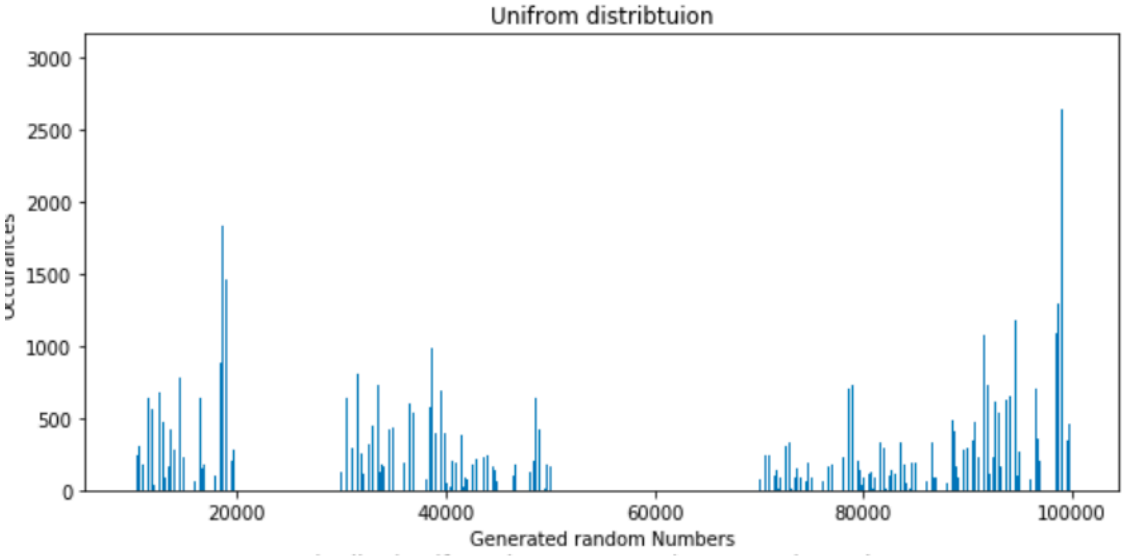
- We have a variable here called numberSize = 5. Based on numberSize the random number is generated. As our numberSize is 5 here, 5 ten digit number are generated and then a digit is picked from each number using the distribution.
- In the first cell of the code, the distribution used is a uniform distribution. From this distribution we made 5 ten digit numbers and for 10000 trials we kept generating a number.
- To generate the number, we randomly pick a value from the distribution array and corresponding to that value we pick a digit from each of the 5 numbers and form a number
- The first graph is the plot of the number against its occurrences
- The Second graph is the plot of the number against its occurrences if truly random numbers were generated in every trial

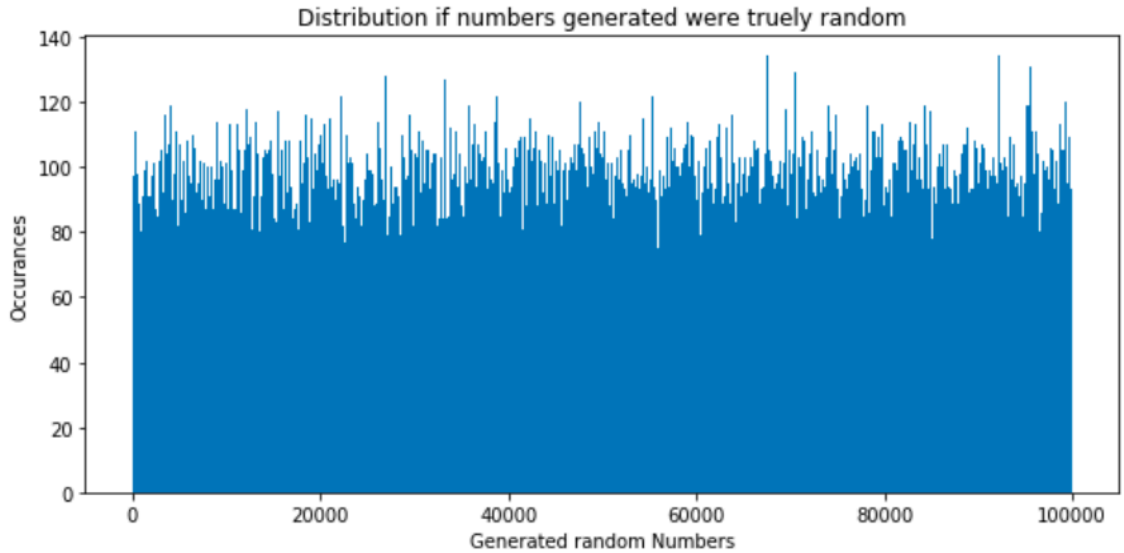
- In the second cell, the distribution used is exponential with **lambda =5.** We form a distribution array with values 0-9 based on the exponential distribution.

Results

Cell 1 - uniform distribution

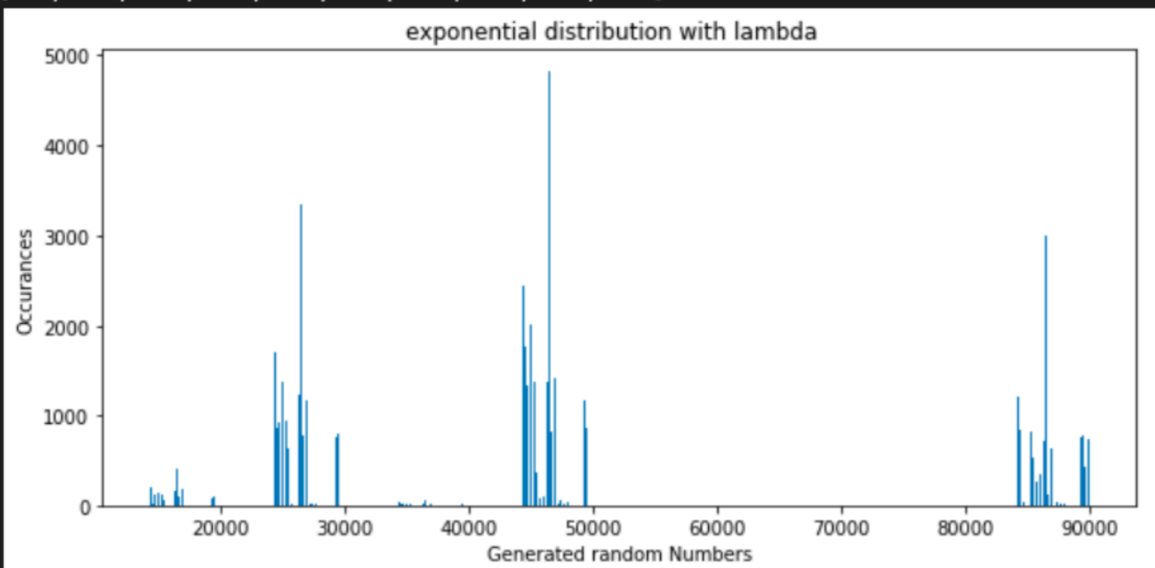
```
Below are the generated 10 digit numbers from which a random number sequence is generated
['9', '7', '3', '4', '1', '3', '9', '1', '8', '9']
['0', '9', '6', '8', '2', '4', '1', '8', '3', '8']
['9', '0', '5', '6', '9', '5', '6', '5', '5', '9']
['6', '7', '7', '5', '6', '8', '3', '1', '1', '8']
['2', '8', '7', '8', '7', '0', '6', '6', '5', '1']
```

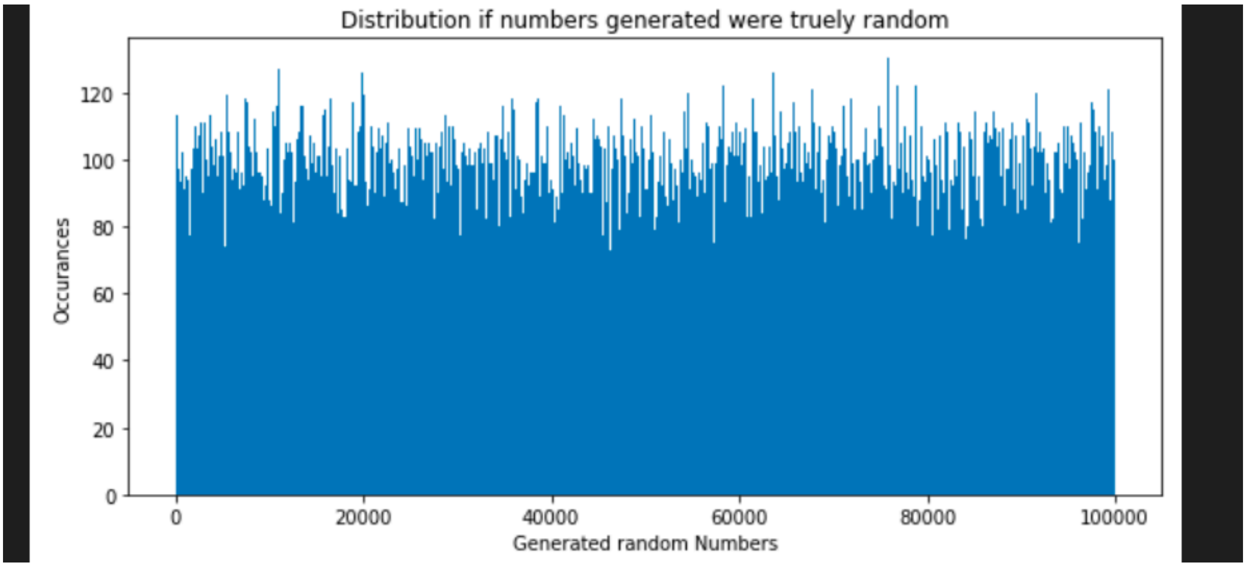




Cell 2 exponential distribution

```
Below are the generated 10 digit numbers from which a random number sequence is generated
['3', '1', '4', '4', '4', '2', '8', '2', '8', '8']
['7', '4', '4', '4', '6', '5', '9', '6', '6', '6']
['3', '6', '6', '4', '9', '2', '4', '3', '3', '3']
['4', '6', '4', '5', '6', '6', '4', '7', '4', '4']
['4', '6', '8', '2', '7', '8', '9', '8', '5', '6']
```





Observations

- Considering the first case, there are spikes at different positions whenever the I run the code. Here the spikes are at 10000. If the numbers were random then the graph would have been something like the one right below the graph. Hence, the numbers are not really random using the mentioned strategy
- Considering the second case, again there are spikes and the graph is clearly not uniformly distributed. The fact that **lambda = 5,** the 10 digit numbers generated will tend to have more 5's and there's some reason to believe that spikes are not really unexpected.
- Therefore, using the mentioned strategy the generated numbers can be highly biased and not random.

4. Central Limit Theorem (CLT.ipynb)

Theory

- To prove that Sn = X1 + X2 + … tends to normal as n increases. Skewed distributions tend to converge more slowly than symmetric distributions.
- The mean and variance must approximate to - (2n/3,n/18)

Implementation

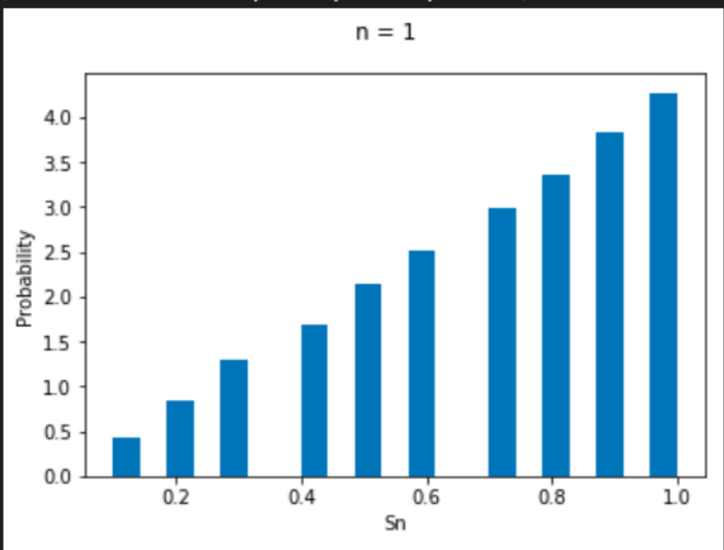
- The distribution I have chosen is f(x) = 2x where 0<x<1.
- I generate the distribution in the first part of the code. n is adjusted here
- For 100000 trials I generate n samples and find their sum (**Sn**) in each trial. If the generated sum is already there then it's occurrence is increased.
- The graph of the Sn is plotted against the probability of each Sn

Results

n=1

Here, we get the distribution that we have chosen it self.

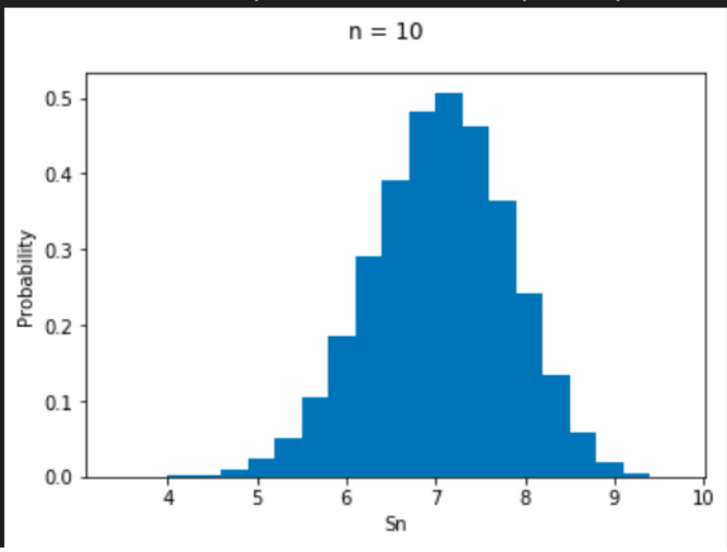
```
The mean and variance of f(x) = 2x is
Mean = 0.66 and variance = 0.055
('Mean result', 0.7002990000001057, 'Var result', 0.06016301222911956)
('Theoretical mean', 0.66, 'Var', 0.055)
```



n=10

The distribution is slightly normal

```
The mean and variance of f(x) = 2x is
('Mean result', 6.999988000000055, 'Var result', 0.5986685865420092)
('Theoretical mean', 6.6000000000000005, 'Var', 0.55)
```



n=100

The distribution is very similar to normal and the mean and variance are close to the theoretical mean and variance

```
The mean and variance of f(x) = 2x is
('Mean result', 69.99990799999964, 'Var result', 6.009705488591162)
('Theoretical mean', 66.0, 'Var', 5.5)
```