Model of the Central limit theorem

May 25, 2020

1 The Central Limit Theorem

1.0.1 One of the most, if not the most important theorem in statistics, is the Central Limit Theorem.

The Central Limit Theorem states the following: If a sample is taken from a given a set of independent random variables, regardless of its original distribution, the sample distribution will be normally distributed and the sample mean will be equal to the population mean as the sample size increases. In more mathematical terms, let $(X_1, X_2, X_3, ..., X_n)$ be a set of indepedent random variables. The sample distributon $\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$ as n $\rightarrow \infty$. Here μ represents the population mean and σ^2 is the population variance, and n is the sample size. **Note:** There are certain calculations that can be made to ensure the sample distribution will follow a normal distribution. However it is accepted that a sample size of thirty will be sufficient.

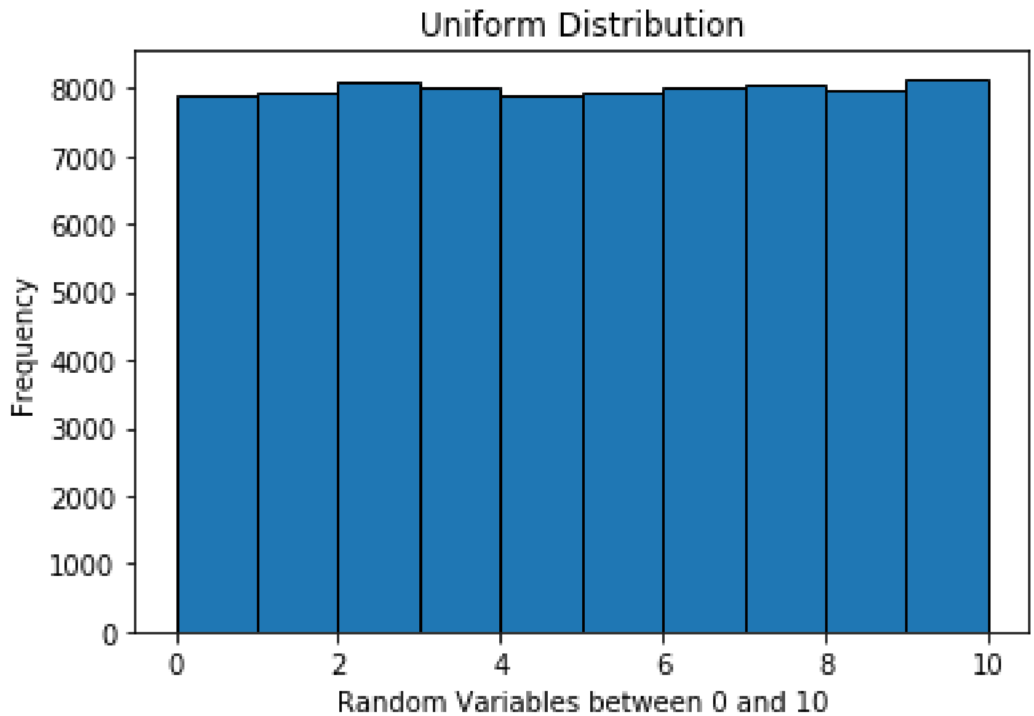
1.0.2 The purpose of this notebook is to model this theorem using different values of n.

[1]:
```
import numpy as np
import math
import random
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

2 Here we will be drawing samples from a uniform distribution. However, as stated previously, the result will be the same with any other distribution. Visually, using a uniform distribution has more of an impact.

[2]:
```
np.random.seed(3)
distribution = np.random.uniform(0,10,80000)
plt.title("Uniform Distribution")
plt.xlabel('Random Variables between 0 and 10')
plt.ylabel("Frequency")
plt.hist(distribution,edgecolor = 'black')
```

```
plt.show()
print('The population mean is: ',round(np.mean(distribution),6))
```




```
The population mean is:  5.014729
```

3 Now random samples of size n will be taken from the uniform distribution.

[3]:
```
xbar = []
num_samples = 20000
n = 5
k = n
for i in range(0,num_samples):
    sample = random.choices(distribution,k=n)
    avg = np.mean(sample)
    xbar.append(avg)
mu  = round(np.mean(xbar),6)
std = round(np.std(xbar),6)
print("The sample mean is equal to", mu,"with a standard deviation of",std ,"␣
 ↪when n =",n)
```

```
The sample mean is equal to 5.011843 with a standard deviation of 1.303086  when
n = 5
```

These are the results we get from drawing the following samples:

[5]:
```
import pandas as pd
data = {'Sample size n':  [2,5,10,20,30],
       "Mean": [ 5.034652,5.011843,5.013351,5.016147,5.014476 ],
       "Standard deviation": [2.054247,1.303086,0.915367,0.641544,0.525793]}
df =pd.DataFrame(data, columns = ["Sample size n","Mean", "Standard deviation"],
                index = ['A', "B", 'C','D', "E"])
df
```

[5]:

	Sample size n	Mean	Standard deviation
A	2	5.034652	2.054247
B	5	5.011843	1.303086
C	10	5.013351	0.915367
D	20	5.016147	0.641544
E	30	5.014476	0.525793

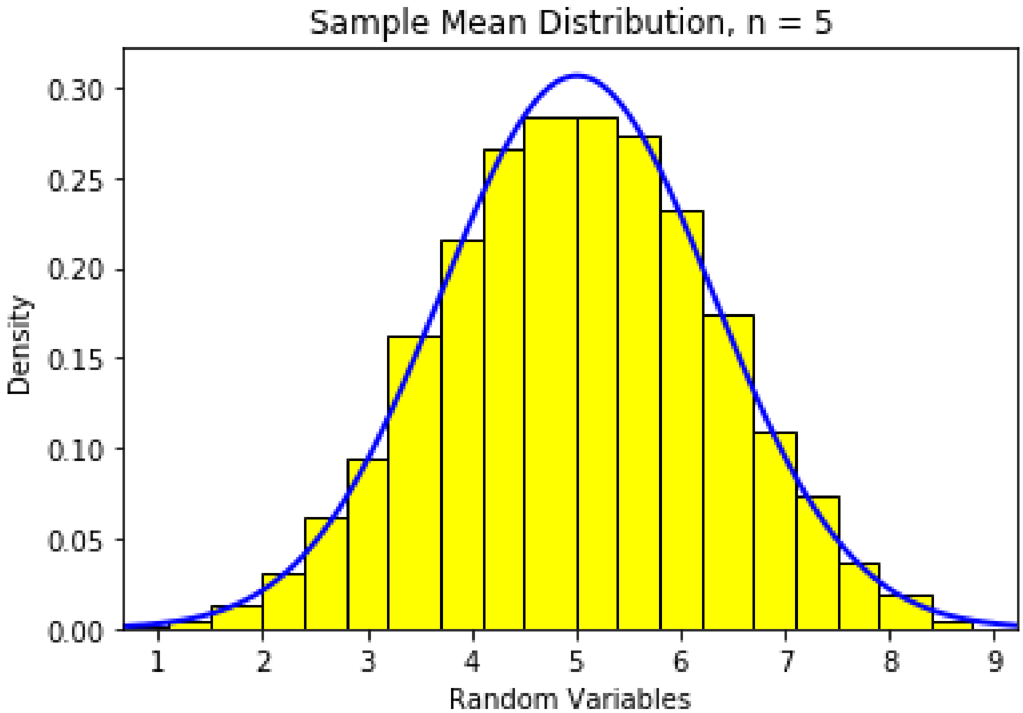
4 Now we will use the formula, Class width = $\frac{Max-Min}{Numberofclasses}$ and use that as our increment from one bin to another. This will create 20 different classes

[6]:
```
num_bins = 20
step = (max(xbar) - min (xbar))/ num_bins
num_classes = []
for q in range(0,num_bins):
    incr = min(xbar) + q*step
    num_classes.append(round(incr,1))
```

5 Now to display our results with a histogram

[7]:
```
norm_xbar = np.linspace(min(xbar), max(xbar), num_samples)
plt.title("Sample Mean Distribution, n = 5")
plt.xlim(min(xbar),max(xbar))
plt.xlabel('Random Variables')
plt.ylabel("Density")
plt.hist(xbar, bins= num_classes, edgecolor = "black", align = "mid", density =␣
 ↪True,color = "yellow")
plt.plot(norm_xbar,norm.pdf(norm_xbar,mu,std),linewidth=2, color = "b")
```

[7]: [<matplotlib.lines.Line2D at 0x160fcb0bc88>]



6 Summary

It is important to notice not only how closely the histogram follows the normal distribution, but how much more dense or "skinnier" the graph becomes as n increases. According to this theorem, thats exactly what should happen. Increasing the sample size will make the standard deviation become smaller, making the graph spike up at the mean. It is incredible to see this theorem work. Its precision and simplicity is what makes it one of the most significant theorems in statistics and all of mathematics.

[157]:

```

```

[ ]:

```

```