CSE17040-ProbDist

August 12, 2020

1 Call Center problem simulation for various Probability Distributions

CB.EN.U4CSE17040

Manojkumar V K

Problem

Imagine that we are data scientists tasked with improving the ROI (Return on Investment) of our company's call center, where employees attempt to cold call potential customers and get them to purchase our product. You look at some historical data and find the following:

The typical call center employee completes on average 50 calls per day.

The probability of a conversion (purchase) for each call is 4%.

The average revenue to your company for each conversion is 20.

The call center you are analyzing has 100 employees.

Each employee is paid 200 per day of work.

[1]:
```
import numpy as np
import matplotlib.pyplot as p
import seaborn as sns
from scipy.stats import uniform
from scipy.stats import norm
from scipy.stats import bernoulli
from scipy.stats import binom
from scipy.stats import poisson
```

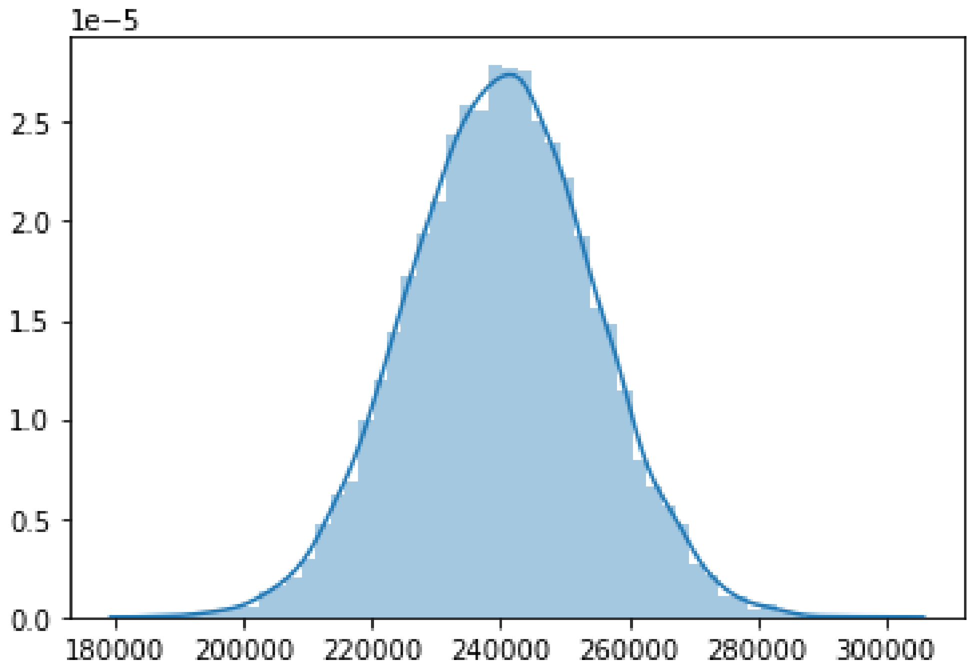
[2]:
```
a = 1
b = 50
employees = 100
wage = 200
n = 50
p = 0.04
revenue = 100
sims = 10000
mu = 2
```

1.1 Original Simulation

1.1.1 Uniform Distribution

```
[3]: uniform_conversions = [np.sum(uniform.rvs(size=employees, loc = a, scale=b))␣
      ↪for _ in range(sims)]
     uniform_profits = np.array(uniform_conversions)*revenue - employees*wage
     sns.distplot(uniform_profits)
```

```
[3]: <matplotlib.axes._subplots.AxesSubplot at 0x7f4ef57c3dd0>
```



```
[4]: print('Average Conversions per Employee                  : ' + str(round(np.
      ↪mean(uniform_conversions), 2)))
     print('Standard Deviation of Conversions per Employee : ' + str(round(np.
      ↪std(uniform_conversions))))
     print('Total Conversions                                 : ' + str(np.
      ↪sum(uniform_conversions)))
     print('Total Revenues                                    : ' + str(np.
      ↪sum(uniform_conversions)*revenue))
     print('Total Expense                                     : ' + str(employees*wage))
     print('Total Profits                                     : ' + str(np.
      ↪sum(uniform_conversions)*revenue - employees*wage))
```

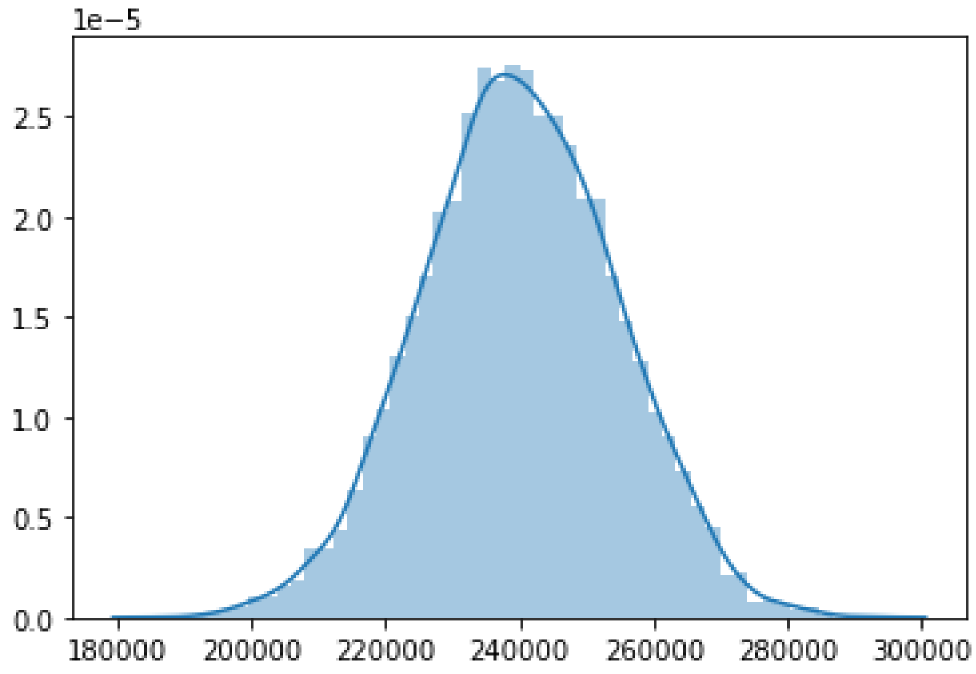
```
Average Conversions per Employee                  : 2600.65
Standard Deviation of Conversions per Employee : 144.0
```

```
Total Conversions                                : 26006513.87012773
Total Revenues                                   : 2600651387.012773
Total Expense                                    : 20000
Total Profits                                    : 2600631387.012773
```

1.1.2 Normal Distribution

[5]:
```
normal_conversions = [np.sum(uniform.rvs(size=employees, loc = a, scale=b)) for␣
 ↪_ in range(sims)]
normal_profits = np.array(normal_conversions)*revenue - employees*wage
sns.distplot(normal_profits)
```

[5]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef4f205d0>
```



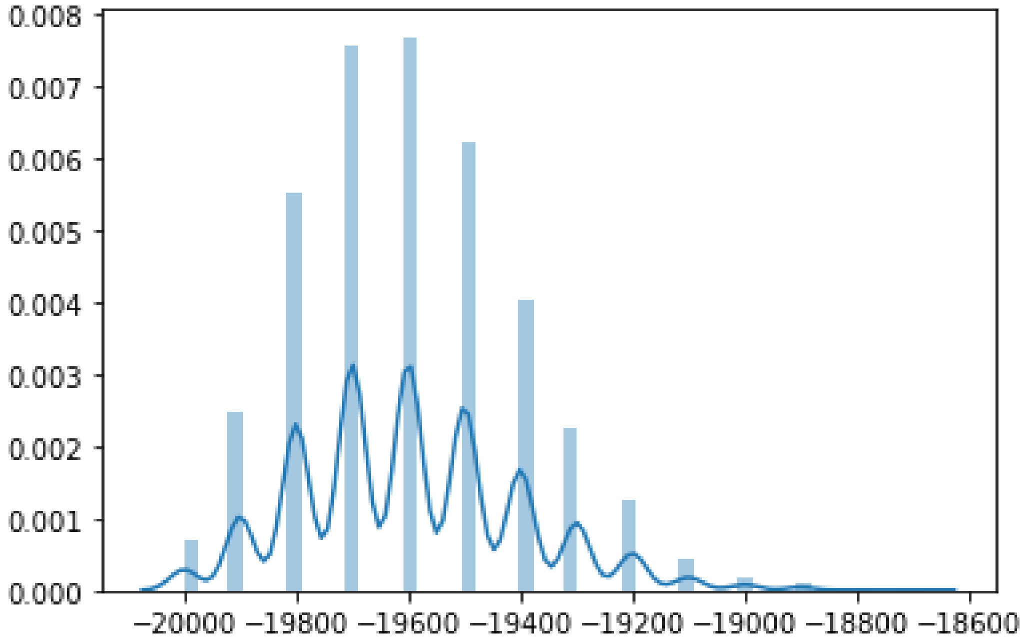
[6]:
```
print('Average Conversions per Employee                 : ' + str(round(np.
 ↪mean(normal_conversions), 2)))
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(normal_conversions))))
print('Total Conversions                                : ' + str(np.
 ↪sum(normal_conversions)))
print('Total Revenues                                   : ' + str(np.
 ↪sum(normal_conversions)*revenue))
print('Total Expense                                    : ' + str(employees*wage))
print('Total Profits                                 : ' + str(np.
 ↪sum(normal_conversions)*revenue - employees*wage))
```

```
Average Conversions per Employee                  : 2599.41
Standard Deviation of Conversions per Employee : 145.0
Total Conversions                                 : 25994103.340628948
Total Revenues                                    : 2599410334.062895
Total Expense                                     : 20000
Total Profits                                     : 2599390334.062895
```

1.1.3 Bernoulli Distribution

[7]:
```
bern_conversions = [np.sum(bernoulli.rvs(size=employees, p=p)) for _ in␣
 ↪range(sims)]
bern_profits = np.array(bern_conversions)*revenue - employees*wage
sns.distplot(bern_profits)
```

[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef4d96050>
```



[8]:
```
print('Average Conversions per Employee               : ' + str(round(np.
 ↪mean(bern_conversions), 2)))
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(bern_conversions))))
print('Total Conversions                              : ' + str(np.
 ↪sum(bern_conversions)))
```

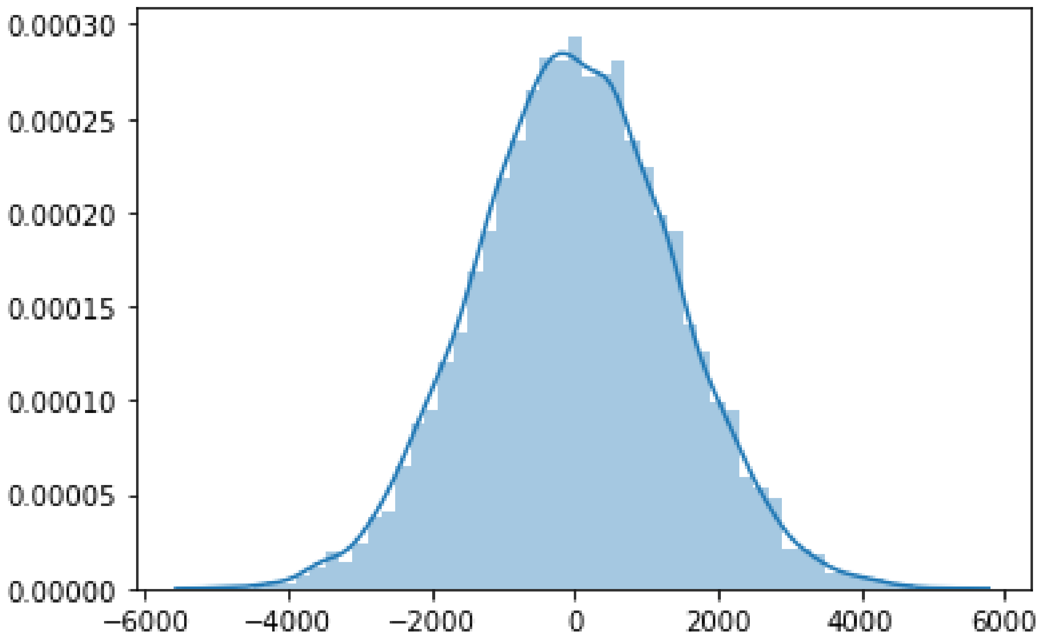
```
print('Total Revenues                                 : ' + str(np.
 ↪sum(bern_conversions)*revenue))
print('Total Expense                                  : ' + str(employees*wage))
print('Total Profits                                  : ' + str(np.
 ↪sum(bern_conversions)*revenue - employees*wage))
```

```
Average Conversions per Employee                : 4.03
Standard Deviation of Conversions per Employee : 2.0
Total Conversions                               : 40332
Total Revenues                                  : 4033200
Total Expense                                   : 20000
Total Profits                                   : 4013200
```

1.1.4 Binomial Distribution

[9]:
```
binom_conversions = [np.sum(binom.rvs(size=employees, p=p, n=n)) for _ in␣
 ↪range(sims)]
binom_profits = np.array(binom_conversions)*revenue - employees*wage
sns.distplot(binom_profits)
```

[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef5f8ebd0>
```



[10]:
```
print('Average Conversions per Employee               : ' + str(round(np.
 ↪mean(binom_conversions), 2)))
```

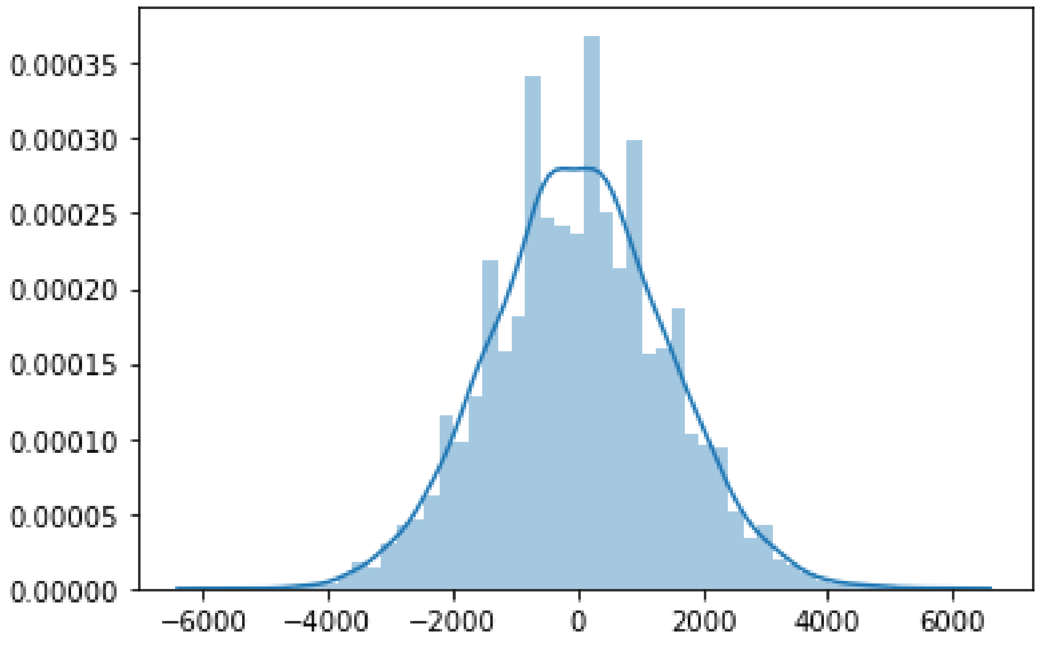
```
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(binom_conversions))))
print('Total Conversions                              : ' + str(np.
 ↪sum(binom_conversions)))
print('Total Revenues                                 : ' + str(np.
 ↪sum(binom_conversions)*revenue))
print('Total Expense                                  : ' + str(employees*wage))
print('Total Profits                                  : ' + str(np.
 ↪sum(binom_conversions)*revenue - employees*wage))
```

```
Average Conversions per Employee               : 199.88
Standard Deviation of Conversions per Employee : 14.0
Total Conversions                              : 1998771
Total Revenues                                 : 199877100
Total Expense                                  : 20000
Total Profits                                  : 199857100
```

1.1.5 Poisson Distribution

[11]:
```
poisson_conversions = [np.sum(poisson.rvs(mu=mu, size=employees)) for _ in␣
 ↪range(sims)]
poisson_profits = np.array(poisson_conversions)*revenue - employees*wage
sns.distplot(poisson_profits)
```

[11]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef4ad4ed0>
```



[12]:
```
print('Average Conversions per Employee                : ' + str(round(np.
 ↪mean(poisson_conversions), 2)))
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(poisson_conversions))))
print('Total Conversions                               : ' + str(np.
 ↪sum(poisson_conversions)))
print('Total Revenues                                  : ' + str(np.
 ↪sum(poisson_conversions)*revenue))
print('Total Expense                                   : ' + str(employees*wage))
print('Total Profits                                   : ' + str(np.
 ↪sum(poisson_conversions)*revenue - employees*wage))
```

```
Average Conversions per Employee               : 200.26
Standard Deviation of Conversions per Employee : 14.0
Total Conversions                              : 2002612
Total Revenues                                 : 200261200
Total Expense                                  : 20000
Total Profits                                  : 200241200
```

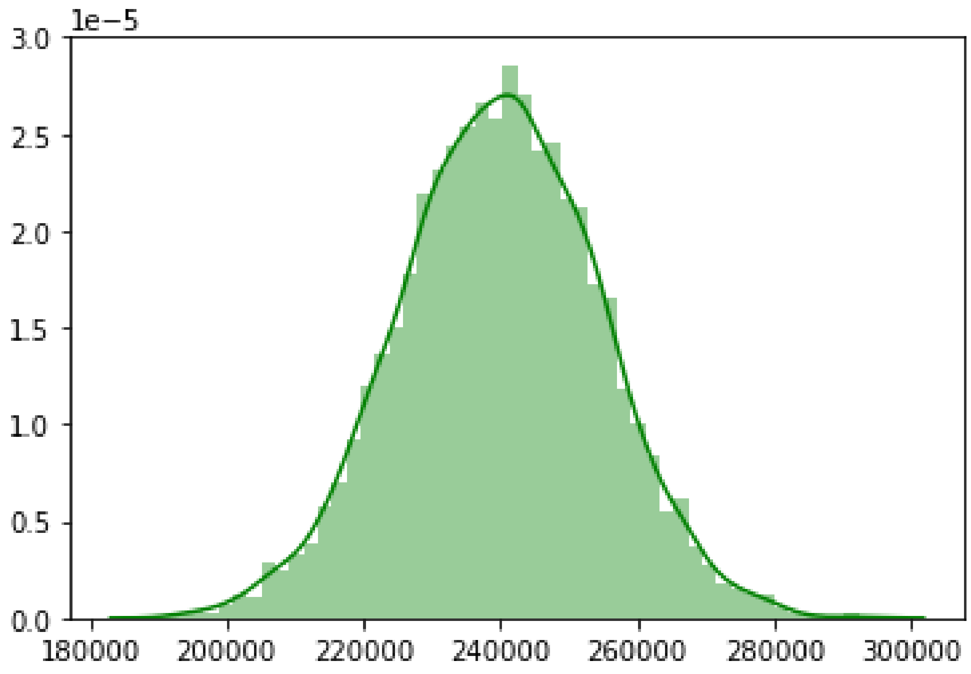
1.2 Improvised Simulation

[13]:
```
n = 55
p = 0.05
mu = 2.5
```

1.2.1 Uniform Distribution

[14]:
```
uniform_conversions = [np.sum(uniform.rvs(size=employees, loc = a, scale=b))␣
 ↪for _ in range(sims)]
uniform_profits = np.array(uniform_conversions)*revenue - employees*wage
sns.distplot(uniform_profits, color='green')
```

[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef498c2d0>
```



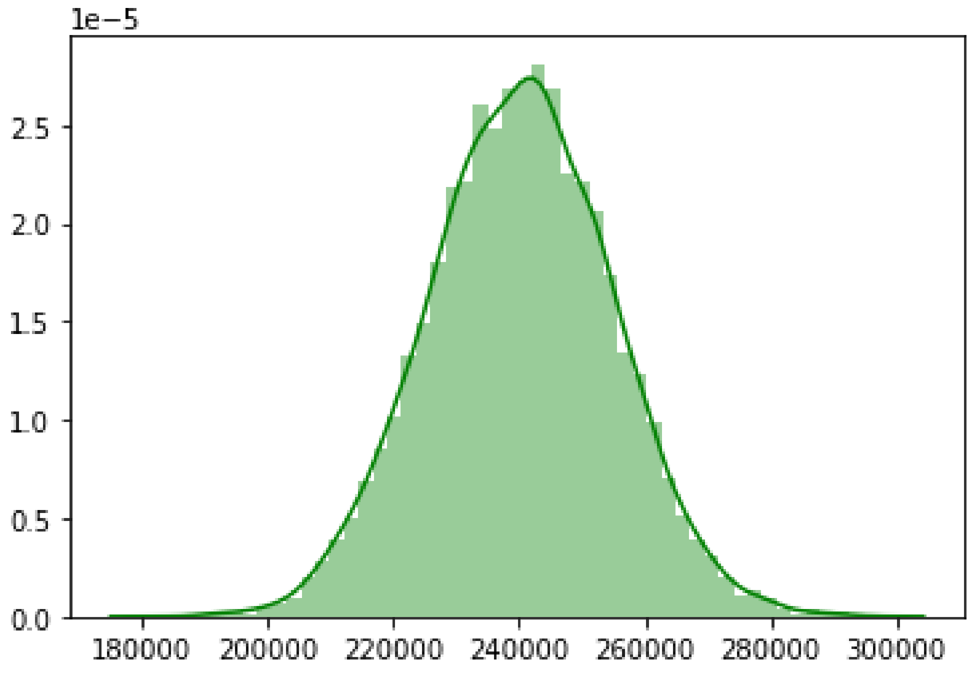
[15]:
```
print('Average Conversions per Employee                : ' + str(round(np.
 ↪mean(uniform_conversions), 2)))
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(uniform_conversions))))
print('Total Conversions                               : ' + str(np.
 ↪sum(uniform_conversions)))
print('Total Revenues                                  : ' + str(np.
 ↪sum(uniform_conversions)*revenue))
print('Total Expense                                   : ' + str(employees*wage))
print('Total Profits                                   : ' + str(np.
 ↪sum(uniform_conversions)*revenue - employees*wage))
```

```
Average Conversions per Employee               : 2598.45
Standard Deviation of Conversions per Employee : 145.0
Total Conversions                              : 25984525.68970747
Total Revenues                                 : 2598452568.970747
Total Expense                                  : 20000
Total Profits                                  : 2598432568.970747
```

1.2.2 Normal Distribution

[16]:
```
normal_conversions = [np.sum(uniform.rvs(size=employees, loc = a, scale=b)) for␣
 ↪_ in range(sims)]
normal_profits = np.array(normal_conversions)*revenue - employees*wage
sns.distplot(normal_profits, color='green')
```

[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef4898e90>
```



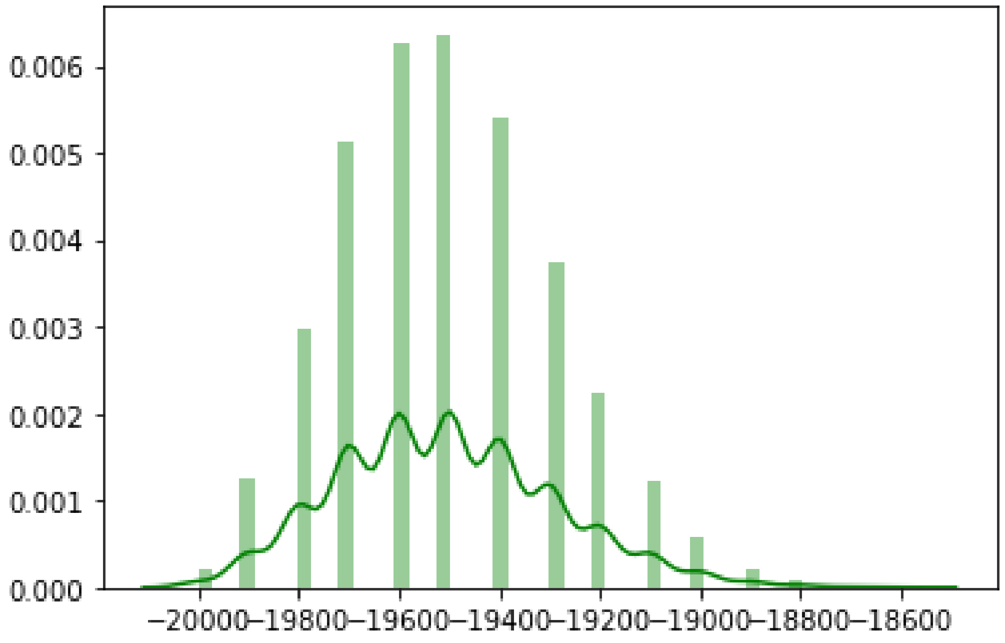
[17]:
```
print('Average Conversions per Employee               : ' + str(round(np.
 ↪mean(normal_conversions), 2)))
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(normal_conversions))))
print('Total Conversions                              : ' + str(np.
 ↪sum(normal_conversions)))
print('Total Revenues                                 : ' + str(np.
 ↪sum(normal_conversions)*revenue))
print('Total Expense                                  : ' + str(employees*wage))
print('Total Profits                                  : ' + str(np.
 ↪sum(normal_conversions)*revenue - employees*wage))
```

```
Average Conversions per Employee               : 2602.01
Standard Deviation of Conversions per Employee : 145.0
Total Conversions                              : 26020106.290446617
Total Revenues                                 : 2602010629.0446615
Total Expense                                  : 20000
Total Profits                                  : 2601990629.0446615
```

1.2.3 Bernoulli Distribution

[18]:
```
bern_conversions = [np.sum(bernoulli.rvs(size=employees, p=p)) for _ in␣
 ↪range(sims)]
bern_profits = np.array(bern_conversions)*revenue - employees*wage
sns.distplot(bern_profits, color='green')
```

[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef4720c50>
```



[19]:
```
print('Average Conversions per Employee               : ' + str(round(np.
 ↪mean(bern_conversions), 2)))
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(bern_conversions))))
print('Total Conversions                              : ' + str(np.
 ↪sum(bern_conversions)))
print('Total Revenues                                 : ' + str(np.
 ↪sum(bern_conversions)*revenue))
print('Total Expense                                  : ' + str(employees*wage))
print('Total Profits                                  : ' + str(np.
 ↪sum(bern_conversions)*revenue - employees*wage))
```

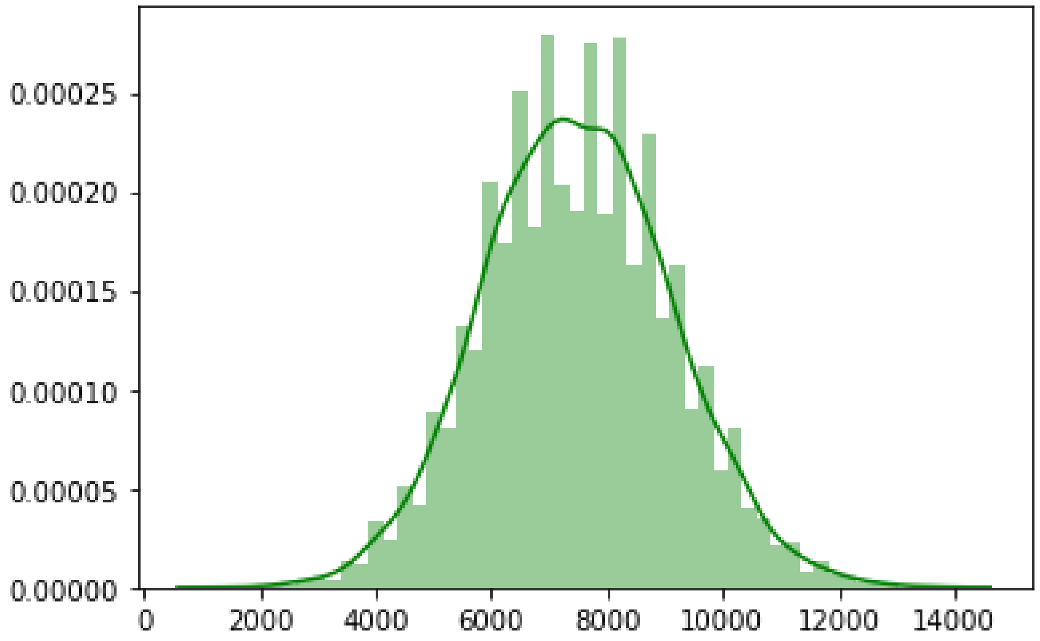
```
Average Conversions per Employee               : 4.94
```

```
Standard Deviation of Conversions per Employee : 2.0
Total Conversions                              : 49424
Total Revenues                                 : 4942400
Total Expense                                  : 20000
Total Profits                                  : 4922400
```

1.2.4 Binomial Distribution

[20]:
```
binom_conversions = [np.sum(binom.rvs(size=employees, p=p, n=n)) for _ in␣
 ↪range(sims)]
binom_profits = np.array(binom_conversions)*revenue - employees*wage
sns.distplot(binom_profits, color='green')
```

[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef472da10>
```



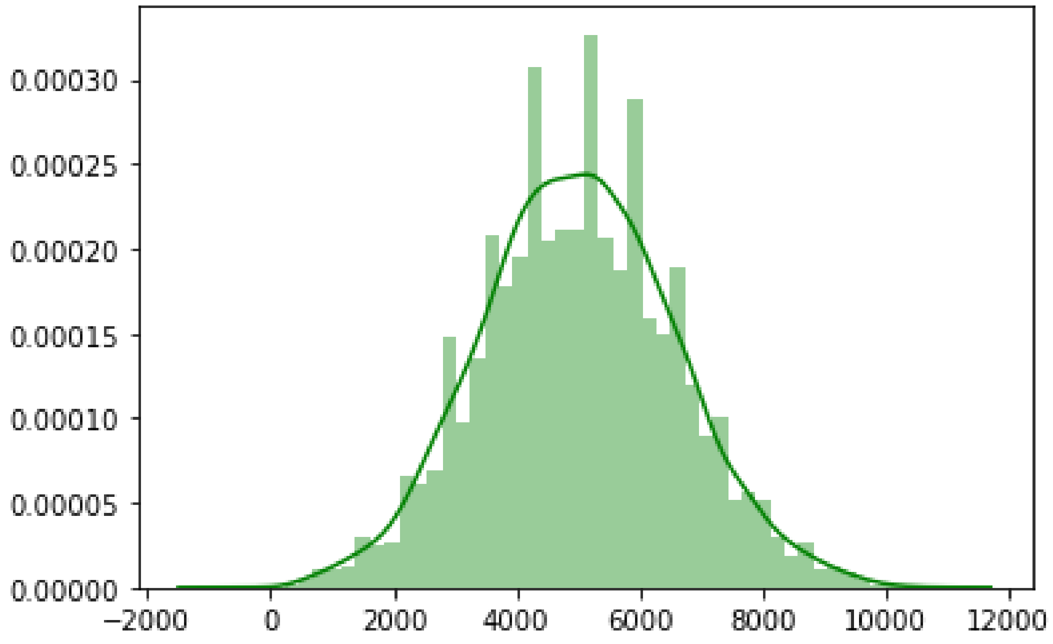
[21]:
```
print('Average Conversions per Employee               : ' + str(round(np.
 ↪mean(binom_conversions), 2)))
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(binom_conversions))))
print('Total Conversions                              : ' + str(np.
 ↪sum(binom_conversions)))
print('Total Revenues                                 : ' + str(np.
 ↪sum(binom_conversions)*revenue))
print('Total Expense                                  : ' + str(employees*wage))
print('Total Profits                                  : ' + str(np.
 ↪sum(binom_conversions)*revenue - employees*wage))
```

```
Average Conversions per Employee               : 274.93
Standard Deviation of Conversions per Employee : 16.0
Total Conversions                              : 2749300
Total Revenues                                 : 274930000
Total Expense                                  : 20000
Total Profits                                  : 274910000
```

1.2.5 Poisson Distribution

[22]:
```
poisson_conversions = [np.sum(poisson.rvs(mu=mu, size=employees)) for _ in␣
 ↪range(sims)]
poisson_profits = np.array(poisson_conversions)*revenue - employees*wage
sns.distplot(poisson_profits, color='green')
```

[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f4ef4565b10>
```



[23]:
```
print('Average Conversions per Employee               : ' + str(round(np.
 ↪mean(poisson_conversions), 2)))
print('Standard Deviation of Conversions per Employee : ' + str(round(np.
 ↪std(poisson_conversions))))
print('Total Conversions                              : ' + str(np.
 ↪sum(poisson_conversions)))
```

```
print('Total Revenues                                        : ' + str(np.
 ↪sum(poisson_conversions)*revenue))
print('Total Expense                                         : ' + str(employees*wage))
print('Total Profits                                         : ' + str(np.
 ↪sum(poisson_conversions)*revenue - employees*wage))
```

```
Average Conversions per Employee                : 250.07
Standard Deviation of Conversions per Employee : 16.0
Total Conversions                               : 2500691
Total Revenues                                  : 250069100
Total Expense                                   : 20000
Total Profits                                   : 250049100
```