Real World Applications of the Binomial Distribution

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.

We can think of each employee as a binomially distributed random variable with the following parameters:

n = 50

p = 4%

```
# Import libraries
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns# Call Center Simulation
```

```

# Number of employees to simulate
employees = 100
# Cost per employee
wage = 200
# Number of independent calls per employee
n = 50
# Probability of success for each call
p = 0.04
# Revenue per call
revenue = 100
# Binomial random variables of call center employees
conversions = np.random.binomial(n, p, size=employees)
# Print some key metrics of our call center
print('Average Conversions per Employee: ' + str(round(np.mean(conversions), 2)))
print('Standard Deviation of Conversions per Employee: ' + str(round(np.std(conversions
print('Total Conversions: ' + str(np.sum(conversions)))
print('Total Revenues: ' + str(np.sum(conversions)*revenue))
print('Total Expense: ' + str(employees*wage))
print('Total Profits: ' + str(np.sum(conversions)*revenue - employees*wage))




```

```
Average Conversions per Employee: 2.18
Standard Deviation of Conversions per Employee: 1.47
Total Conversions: 218
Total Revenues: 21800
Total Expense: 20000
Total Profits: 1800
```