Lab No 3

Name - Hitesh Choudhary
Prn - 20200802146

Questions To performed on the Structured Dataset:

1. In this lab assignment you will use MLE to find the best fit parameters for a statistical distribution. You will work with a dataset and estimate the parameters that maximize the likelihood of observing that data? Dataset You will work with a dataset of exam scores from a hypothetical class. The scores are assumed to follow a normal distribution.

2. Estimate the probability of getting heads when flipping a coin based on a sequence of 10 coin flips: H,T,H,H,T,H,T,H,H,H . Using MLE Determine the most likely value for the probability “p” of getting heads when flipping the coin. Write a py. Program to calculate the MLE estimate for “p” and provide the result.

Generate exam scores

```
import numpy as np
from scipy.stats import norm

n = 1000

mean = 50

std = 10

scores = norm.rvs(loc = mean, scale = std, size = n)

print(scores)
```

```
 64.93567874 71.04557617 48.02599174 61.50506216 30.1870728  51.15651748
 52.22030645 55.41650842 43.5619179  48.27147165 60.8409693  51.29944358
 50.80223329 53.62500823 47.88248026 31.39874503 75.9085207  33.91576468
 48.73818771 38.70783583 52.10673928 46.90821138 55.05236667 52.30091181
 55.13226777 50.97489757 63.44434331 48.19043141 51.18537771 50.93576127
 55.00451909 55.74974032 36.00496667 61.17159397 69.08730745 35.79081154
 45.95817172 43.43300136 53.40467988 72.21668144 74.25721317 66.97498328
 47.94135233 40.08161065 44.2630645  62.77271445 50.58709991 55.67255702
 43.70903815 59.05602061 56.93719561 63.33166287 37.01457843 32.48903695
 52.49619356 54.07585589 59.15325235 53.85413005 48.28644475 52.87706725
 56.34182942 41.0677031  47.24446762 51.13332521 56.25785777 53.71042208
 42.31644638 46.46978397 44.97711081 58.00077081 34.31889548 55.62592678
 38.52740881 49.00206953 47.99334836 42.62314321 54.9856785  48.02510172
 51.13386089 42.77333892 57.66064371 52.1467159  65.88930466 42.06666991
 45.75642676 36.60963206 37.71775871 64.34077386 39.43455383 50.92694525
 52.86182656 54.63010781 50.26179235 44.4507864  58.39545919 53.5408004
 52.93626696 49.2699737  38.18057679 55.81071384 41.60915248 46.14408254
 51.31948634 39.33621135 61.30156467 54.03555275 49.79784529 47.62857769
 66.13028568 50.62699731 58.66832107 64.4905624  53.88761409 50.31198672
 40.64419819 55.7921574  52.15463039 56.96539809 48.36271918 56.6967473
 38.07133938 53.7119134  55.87176718 68.82743144 64.4743867  33.37699197
 39.65614189 48.82372018 64.37650056 42.59152781 33.6592584  51.276385
 37.75424642 57.67924473 46.47931145 39.52663222 41.73799573 50.0573138
 36.73688802 51.64716816 59.93328115 51.27131661 55.93699871 60.07610769
 40.70831264 55.6983712  60.55497967 48.46171447 46.57044825 53.41915261
 51.18178617 40.73868117 51.41160529 51.19866603 49.26378309 44.53255651
 66.28514361 39.32853394 52.32710074 61.66341259 43.44146053 59.19469803
 52.84848884 47.52658667 52.95314015 40.98902304 62.91802584 49.8010317
 46.91663422 33.37410876 59.63581229 57.24451707 36.63288137 57.54048289
 66.40517858 48.33317046 39.09822545 57.80507165 43.24533897 37.41521523
 54.30333946 54.3404552  52.22841773 52.50846578 63.13758266 37.63647025
 29.53701637 60.84540022 38.12601333 33.92311863 46.70675867 33.28132257
 40.71996857 53.31500733 54.16160561 57.86424634 47.32405845 59.45442112
 46.5577459  46.31986269 69.48573539 31.53098114 61.47363688 51.21818222
 41.02610275 48.11948179 51.60976866 61.04929469]
```

Finding MLE

```
mean_mle = scores.mean()

def likelihood(x,mu,sigma):
    return np.prod(norm.pdf(x,loc=mu,scale = sigma))

sigma_mle = np.sqrt(np.sum((scores-mean_mle)**2)/n)

print(mean_mle)
```

```
49.89480904281794
```

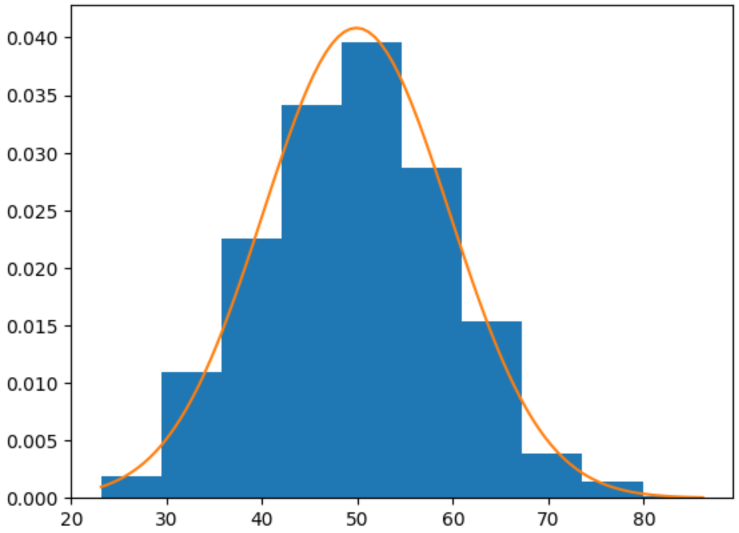
Plotting MLE

```
import matplotlib.pyplot as plt
plt.hist(scores,density = True, bins = 10)

x = np.linspace(min(scores), max(scores), 100)
```

```
plt.plot(x, norm.pdf(x, loc = mean_mle, scale = sigma_mle))
```

```
[<matplotlib.lines.Line2D at 0x79205834ce80>]
```



Finding MLE for Coin

```
def calculate_mle(sequence):
    num_heads = sequence.count('H')
    total_flips = len(sequence)
    probability = num_heads / total_flips
    return probability

def visualize_sequence(sequence):
    heads = sequence.count('H')
    tails = sequence.count('T')
    x = ['Heads', 'Tails']
    y = [heads, tails]
    plt.bar(x, y)
    plt.xlabel('Outcome')
    plt.ylabel('Count')
    plt.title('Coin Flip Sequence')
    plt.show()
```

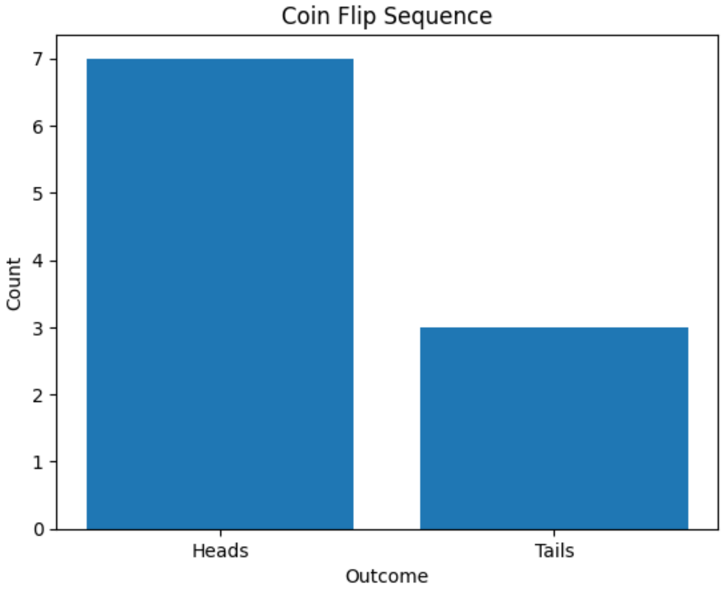
```
sequence = 'HTHHTHTHHH'

# Calculate MLE
probability = calculate_mle(sequence)
print(f"MLE probability of getting heads: {probability}")

# Visualize sequence
```

```
visualize_sequence(sequence)
```

```
MLE probability of getting heads: 0.7
```


Coin Flip Sequence
Count
Heads
Tails
Outcome