```python
# Importing all the packages
import matplotlib.pyplot as plt
import numpy as np

# Given population
population = [65,82, 81, 67, 57,59 ,66 ,75 ,82, 70]
```

QUESTION A and B:

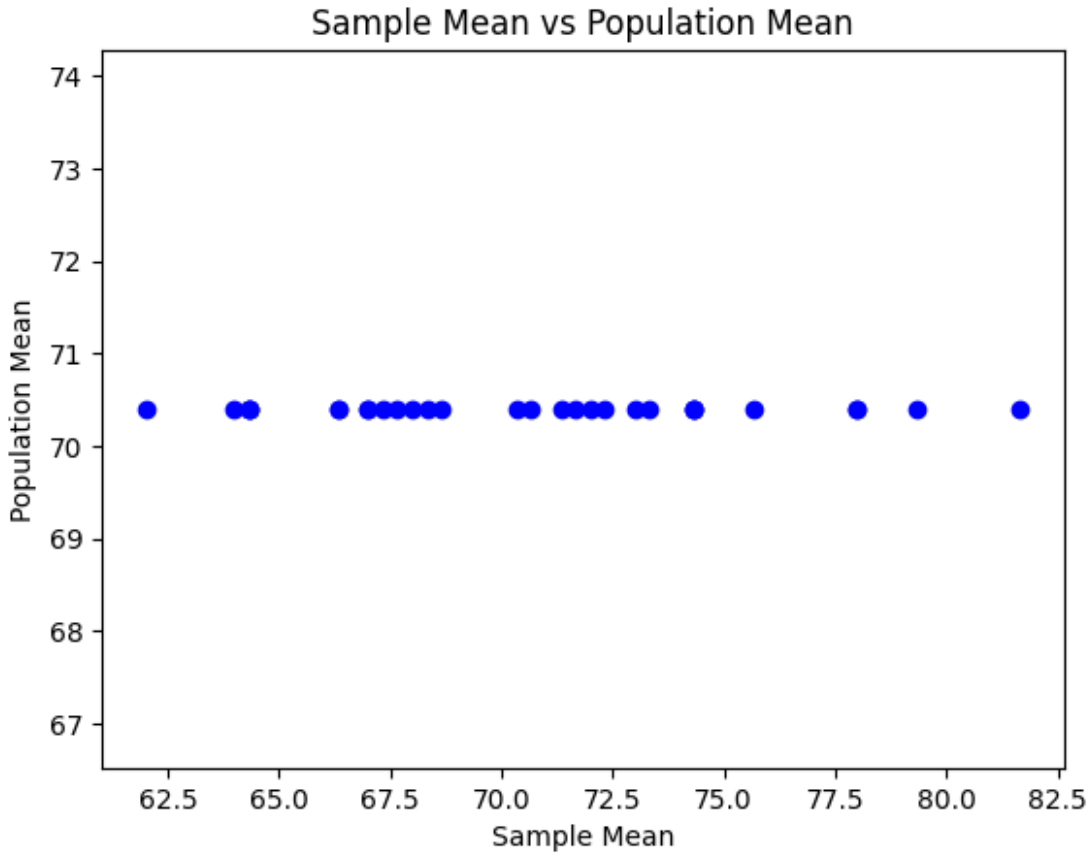
- Write a function/script in python, randomly selecting 3 data points from the population (N=10), and evaluate the sample mean and standard deviation. Repeat the code 'a' number of times, where a ≥ 15.

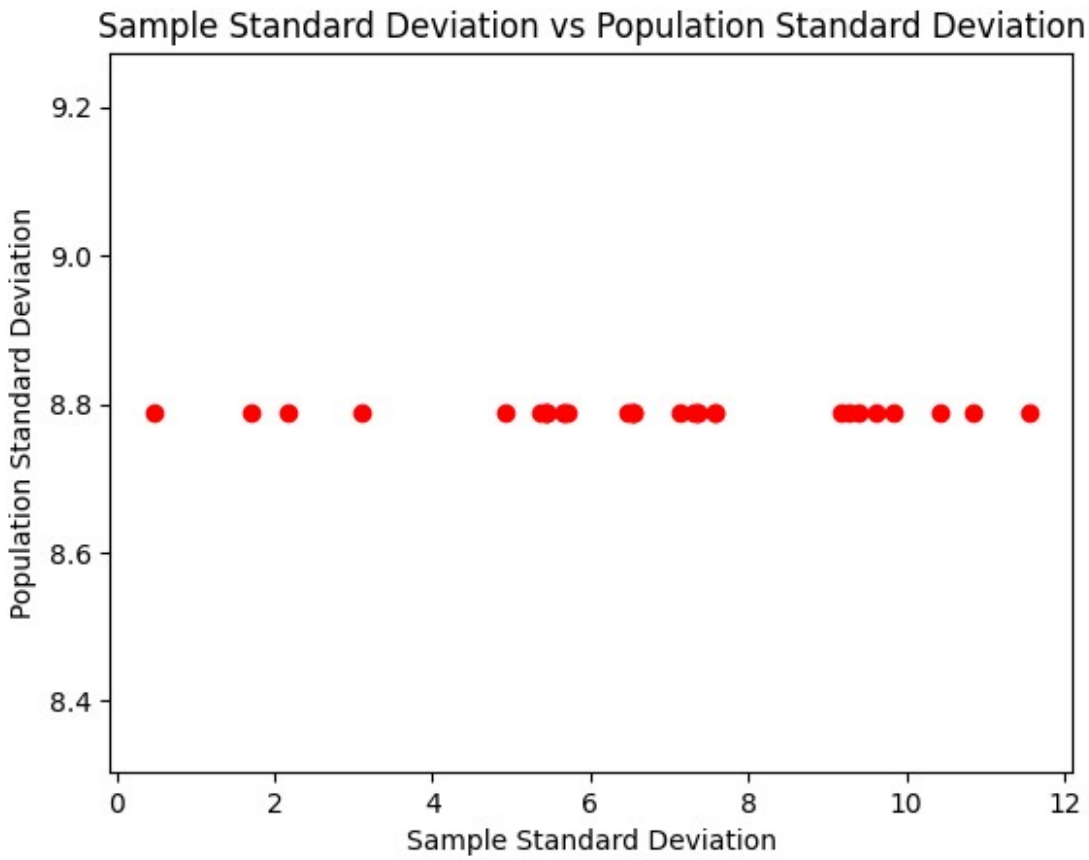
- Plot

1. Sample mean vs population mean for every repetition.
2. Sample standard deviation vs population standard deviation for every repetition.

```python
N = len(population)

# Defining the number of repetitions (a >= 15)
a = 30

# Initializing lists to store the sample mean and standard deviation
for every repetition
sample_means = []
sample_stds = []


for i in range(a):

    # Randomly selecting 3 data points from the population
    sample = np.random.choice(population, size=3, replace=False)

    # Evaluating the sample mean and standard deviation
    sample_mean = np.mean(sample)
    sample_std = np.std(sample)

    # Adding the sample mean and standard deviation to the respective
lists
    sample_means.append(sample_mean)
    sample_stds.append(sample_std)

# Plotting sample mean vs population mean
plt.scatter(sample_means, [np.mean(population)] * a, color='blue')
plt.xlabel('Sample Mean')
plt.ylabel('Population Mean')
plt.title('Sample Mean vs Population Mean')
plt.show()

# Plotting sample standard deviation vs population standard deviation
```

```
plt.scatter(sample_stds, [np.std(population)] * a, color='red')
plt.xlabel('Sample Standard Deviation')
plt.ylabel('Population Standard Deviation')
plt.title('Sample Standard Deviation vs Population Standard
Deviation')
plt.show()
```



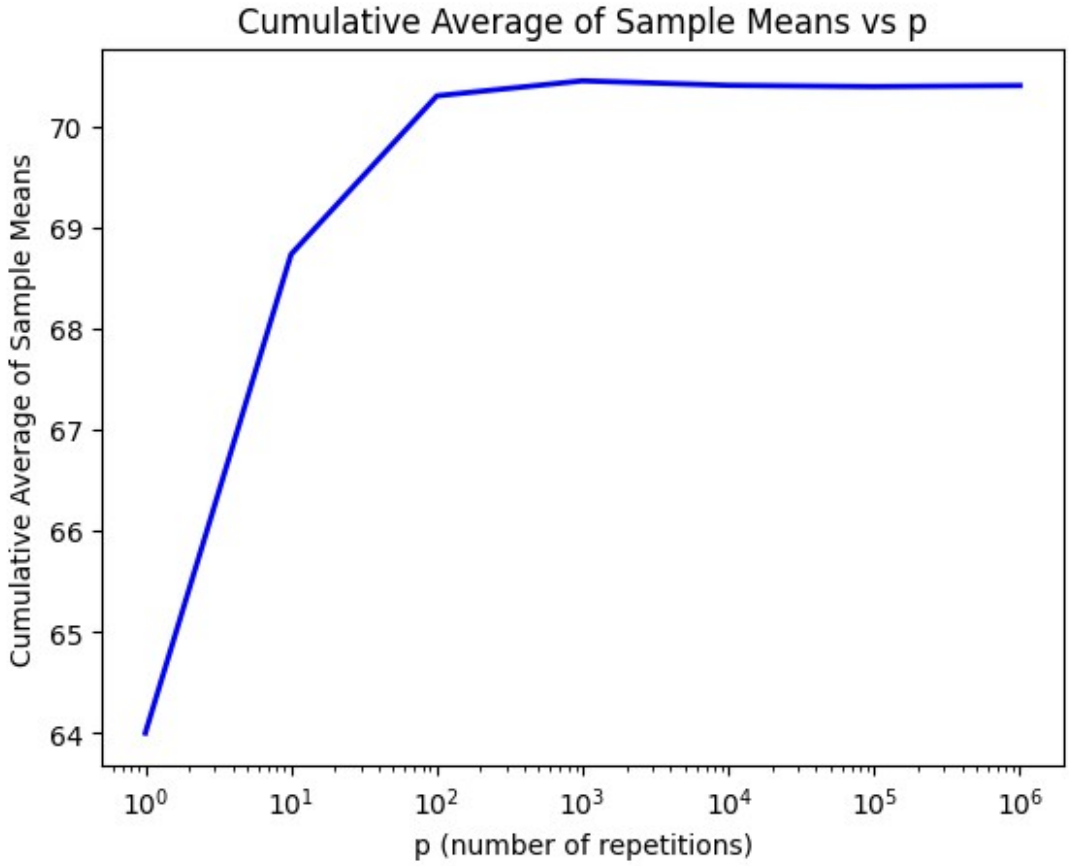


QUESTION C:

- Assign an array p = [1, 10, 100, 1000, 10000, 100000], and plot the cumulative average of the 'p' sample means (of the randomly selected 3 data points).

```
# Defining array p
p =
[pow(10,0),pow(10,1),pow(10,2),pow(10,3),pow(10,4),pow(10,5),pow(10,6)
]

# Initializing a list to store the cumulative average of the sample
means for every p
cumulative_avg = []


for i, n in enumerate(p):

    # Initializing a list to store the sample means for every
repetition
    sample_means = []

    # Loop for n number of repetitions
    for j in range(n):
        # Randomly selecting 3 data points from the population
```

```
            sample = np.random.choice(population, size=3, replace=False)

            # Evaluating the sample mean
            sample_mean = np.mean(sample)

            # Adding the sample mean to the sample means list
            sample_means.append(sample_mean)

        # Evaluating the cumulative average of the sample means
        cumulative_avg.append(np.mean(sample_means))

# Plotting the cumulative average of the sample means
plt.plot(p, cumulative_avg, color='blue',linewidth=2)
plt.xscale('log')
plt.xlabel('p (number of repetitions)')
plt.ylabel('Cumulative Average of Sample Means')
plt.title('Cumulative Average of Sample Means vs p')
plt.show()
```



Observations on the plots:

- From the plot of Sample Mean vs Sample deviation we can observe that the majority of the sample means are clustered around the population mean and this increases if choose a higher value of a.

- From the plot of Sample Standard Deviation vs Population standard deviation we can see clustering around population std value but we cannot comment of the specific pattern or insight from this plot.
- From the plot of cummulative average of sample means vs p we can find that as p increases and reaches very high values then the curve nearly becomes equal to a a straight line at a constant value equal to the population mean.