EE 381 Project 5

Confidence Intervals

by Alexandra McBride

November 20, 2019

California State University, Long Beach

Dr. Anastasios Chassiakos

Problem 1

Introduction

This problem involves the measurements of statistics from a large population of ball bearings that has size N, mean μ in grams and standard deviation σ in grams. From that population, we take a sample, or subset, of the ball bearing from the population that has size n. That sample has a distribution of the sample mean $\bar{X}$, where $\mu_X = \mu$ and $\sigma_X = \sigma / \sqrt{n}$. We want to find the relationship between $\bar{X}$ and μ. To do this, we want to take samples of the population starting from size 1 up to size n and find the sample mean $\bar{X}$ for each of them. We will then plot each $\bar{X}$ on the graph and plot the confidence intervals for the 95% confidence interval, which says that approximately 95% of the $\bar{X}$ values will fall between those intervals. We will do another plot for the 99% confidence interval. Lastly, we will observe these results and draw our conclusions from them.

Methodology

For this problem, I will be using Python in the PyCharm IDE. The tools used will include methods from the `numpy` and `mathplotlib.pyplot` libraries.

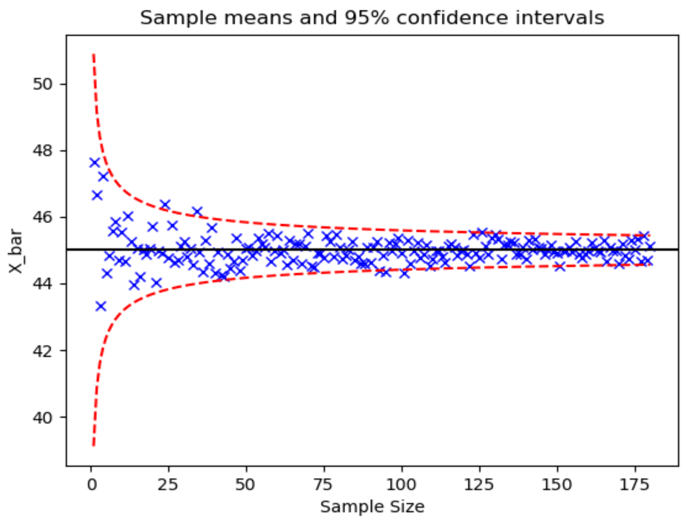
The program will create N random variables that are normally distributed. From this, it takes a sample of those random variables of size n and finds the mean $\bar{X}$ and standard deviation using the sample. The program then begins plotting two graphs where both have each $\bar{X}$, then at the end plot the lower and upper interval in red for the 95% confidence interval for one of the graphs, the plots the lower and upper interval in green for the 99% confidence interval.

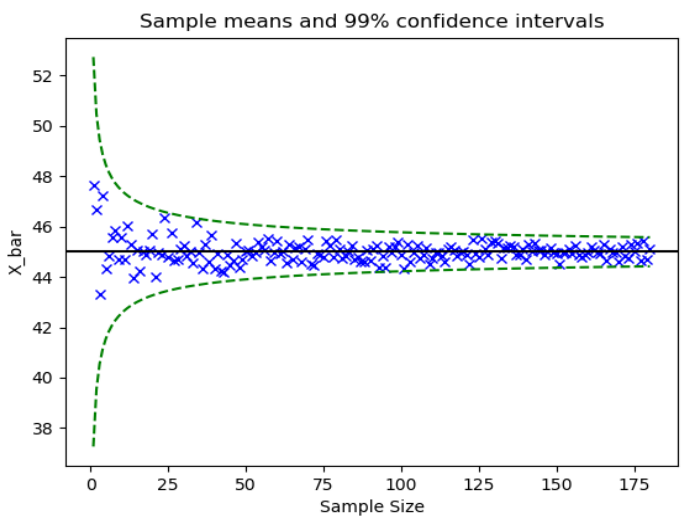
Results and Conclusion

The values that were used to get these results were:

$$N = 1200000 \quad \mu = 45 \quad \sigma = 3$$

Here are the graphical results, which include both the 95% confidence interval and 99% confidence interval:





From the results in the 95% confidence interval, we see that the lower and upper intervals include almost all the $\bar{X}$ values. From the results in the 99% confidence interval, we see that the lower and upper intervals include all the $\bar{X}$ values. These are results expected, and $\bar{X}$ appears to normally stay very close to the value of μ.

Appendix

```
import matplotlib.pyplot as plt
import numpy as np
import math
# #Setting the values
N=1200000; mu=45; sigma=3; n=180 #ranges from 1 to 180
```

```
# #Creating N X variables
B=np.random.normal(mu,sigma,N)
#
#Creating graph for confidence interval
pos_interval_95=np.zeros((n,1))
neg_interval_95=np.zeros((n,1))
pos_interval_99=np.zeros((n,1))
neg_interval_99=np.zeros((n,1))



sample_sizes = list(range(1,n+1))
for i in sample_sizes:
    X=np.random.choice(B,i)
    X_bar=np.mean(X)
    S_hat=sigma/np.sqrt(i)
    print(S_hat)
    plt.figure(0)
    plt.plot(i, X_bar, 'b', marker="x")
    plt.figure(1)
    plt.plot(i, X_bar, 'b', marker="x")
    neg_interval_95[i-1] = mu - 1.96*S_hat
    pos_interval_95[i-1]= mu + 1.96*S_hat
    neg_interval_99[i - 1] = mu - 2.58* S_hat
    pos_interval_99[i - 1] = mu + 2.58 * S_hat

plt.figure(0)
plt.title("Sample means and 95% confidence intervals")
plt.xlabel("Sample Size")
plt.ylabel("X_bar")
plt.plot(sample_sizes, neg_interval_95,'r',linestyle='--')
plt.plot(sample_sizes, pos_interval_95,'r',linestyle='--')
plt.axhline(mu,color='k')

plt.savefig("95%.png")

plt.figure(1)
plt.title("Sample means and 99% confidence intervals")
plt.xlabel("Sample Size")
plt.ylabel("X_bar")
plt.plot(sample_sizes, neg_interval_99,'g',linestyle='--')
plt.plot(sample_sizes, pos_interval_99,'g',linestyle='--')
plt.axhline(mu,color='k')

plt.savefig("99%.png")
plt.close("all")
```

Problem 2

Introduction

This problem involves the measurements of statistics from a large population of ball bearings that has size N, mean μ in grams and standard deviation σ in grams. However, we realize that it is unrealistic that the population mean μ and standard deviation σ could be those exact values. Realistically, we need to take sample of the population of different sizes in order to come to a conclusion on those values. We start by taking a sample of n bearings, then find the sample mean $\bar{X}$, sample standard deviation $\hat{S}$, and then the 95% confidence intervals [$\mu_{lower,}\mu_{upper}$]for a normal distribution, then we will see if μ is within the interval. If it is, then the experiment was successful. We then find the 95% confidence intervals[$\mu_{lower,}\mu_{upper}$] for Student's t distribution, then see if μ is within the interval. We find the 99% confidence intervals for both a normal distribution and for Student's t distribution and check if μ is within both the intervals for them. We will then repeat this 10,000 times, then get the percentage of success for both 95% and 99% Confidence for a Normal Distribution and both 95% and 99% Confidence for a Student's t distribution.

Methodology

For this problem, I will be using Python in the PyCharm IDE. The tools used will include methods from the `numpy` and `mathplotlib.pyplot` libraries.

The program will create N random variables that are normally distributed. From this, it will take a sample of those random variables of size n and finds the mean $\bar{X}$ and standard deviation $\hat{S}$ using the sample. The program will then calculate the 95% confidence intervals [$\mu_{lower,}\mu_{upper}$] for a normal distribution, and if μ is within the interval we will add to a counter that keep track of the number of successes. We will also test μ against the 95% confidence intervals for a Student's t distribution, 99% confidence intervals for a normal distribution, and 99% confidence intervals for a Student's t distribution. We will then repeat this 10,000 times, then get the percentage of success for both 95% and 99% Confidence for a Normal Distribution and both 95% and 99% Confidence for a Student's t distribution. Then, we test different values on n by changing the value of n and repeat this process for them.

Results and Conclusion

Here are the table results, which include percentages for the success rate of each along with the sample sizes:

Sample Size (n)	**95% Confidence** (Using Normal Distribution)	**99% Confidence** (Using Normal Distribution)	**95% Confidence** (Using Student's t Distribution)	**99% Confidence** (Using Student's t Distribution)
5	0.8782	0.9398	0.9388	0.9892
40	0.9409	0.9842	0.9475	0.9884
120	0.9457	0.99	0.9472	0.9911

From these results, we can see that the Student's t distribution appears to have more success and therefore is a more accurate confidence interval. We also see as the sample size gets larger, the percentage of success between 95% rates for both the normal distribution and the Student's t distribution gets closer as the n increases, as well as between the 99% rates for both the normal distribution and the Student's t distribution. The success rate also increases as n gets larger.

Appendix

```
#Problem 2
# size 5
n=120; M=10000; z=1.96 ;t=1.98  #change t when n changes
numOfSuccess__normal_95=0
numOfSuccess__t_95=0
for i in range(M):
 X = np.random.choice(B, n)
 X_bar=sum(X)/n
 total=0
 for x in X:
   total+=((x-X_bar)**2)
 S_hat=np.sqrt(total/(n-1))

 pos_interval_normal_95=X_bar+z*(S_hat/np.sqrt(n))
 neg_interval_normal_95=X_bar-z*(S_hat/np.sqrt(n))

 if mu <= pos_interval_normal_95 and mu >= neg_interval_normal_95:
    numOfSuccess__normal_95+=1

 pos_interval_t_95=X_bar+t*(S_hat/np.sqrt(n))
 neg_interval_t_95=X_bar-t*(S_hat/np.sqrt(n))

 if mu <= pos_interval_t_95 and mu >= neg_interval_t_95:
    numOfSuccess__t_95+=1

normal_percent_95 = numOfSuccess__normal_95/M
print(normal_percent_95)
t_percent_95 = numOfSuccess__t_95/M
print(t_percent_95)
```

```
z = 2.58;
t = 2.62  # change t when n changes
numOfSuccess__normal_99=0
numOfSuccess__t_99=0
for i in range(M):
 X = np.random.choice(B, n)
 X_bar=sum(X)/n
 total=0
 for x in X:
    total+=((x-X_bar)**2)
 S_hat=np.sqrt(total/(n-1))

 pos_interval_normal_99=X_bar+z*(S_hat/np.sqrt(n))
 neg_interval_normal_99=X_bar-z*(S_hat/np.sqrt(n))

 if mu <= pos_interval_normal_99 and mu >= neg_interval_normal_99:
     numOfSuccess__normal_99+=1

 pos_interval_t_99=X_bar+t*(S_hat/np.sqrt(n))
 neg_interval_t_99=X_bar-t*(S_hat/np.sqrt(n))

 if mu <= pos_interval_t_99 and mu >= neg_interval_t_99:
     numOfSuccess__t_99+=1

normal_percent_99 = numOfSuccess__normal_99/M
print(normal_percent_99)
t_percent_99 = numOfSuccess__t_99/M
print(t_percent_99)
```