

In [ ]:

In [ ]:

```
#Import packages
import numpy as np
import statistics
```

## EXERCISE 8

Let  $X_1, \dots, X_n$  be independent and identically distributed random variables having unknown mean  $\mu$ . For given constants  $a < b$ , we are interested in estimating  $p = P\{a < \sum_{i=1}^n X_i/n - \mu < b\}$ .

### (a) Explain how we can use the bootstrap approach to estimate $p$ .

It is possible to create a range of means from the one sample ( $X_i$ ), which can be used to calculate the proportion of bootstrapped means subtracted  $\mu$  that fall within  $a$  and  $b$ .

### (b) Estimate $p$ if $n = 10$ and the values of the $X_i$ are 56, 101, 78, 67, 93, 87, 64, 72, 80, and 69. Take $a = -5$ , $b = 5$ .

In [ ]:

```
a = -5
b = 5
Xi = [56, 101, 78, 67, 93, 87, 64, 72, 80, 69]
original_sample_mean = np.mean(Xi)

# Get bootstrap samples from data set Xi
bootstrap_means = []
n = 10000
for i in range(n):
    bootstrap_sample = np.random.choice(Xi, size = len(Xi), replace=True)
    bootstrap_means.append(np.mean(bootstrap_sample))

# Calc the differences from the original mean
differences = bootstrap_means - original_sample_mean
#print(differences)

# Calc differences that is within [a,b]
dif_prop = []
for dif in differences:
    if dif >= a and dif <= b:
        dif_prop.append(dif)

estimated_prop = len(dif_prop)/len(differences)
print(f"Estimated p: {estimated_prop}") # estimate p
```

Estimated p: 0.7683

In the following three exercises  $X_1, \dots, X_n$  is a sample from a distribution whose variance is (the unknown)  $\sigma^2$ . We are planning to estimate  $\sigma^2$  by the sample variance  $S^2 = \sum_{i=1}^n (X_i - \bar{X})^2 / (n-1)$ , and we want to use the bootstrap technique to estimate  $\text{Var}(S^2)$ .

If  $n = 15$  and the data are: 5, 4, 9, 6, 21, 17, 11, 20, 7, 10, 21, 15, 13, 16, 8

### what is the bootstrap estimate of $\text{Var}(S^2)$ ?

In [ ]:

```

n = 15
N = 10000
data = [5, 4, 9, 6, 21, 17, 11, 20, 7, 10, 21, 15, 13, 16, 8]
original_sample_mean = np.mean(data)

bootstrap_means = []
bootstrap_var = []

for i in range(N):
    bootstrap_sample = np.random.choice(data, size = len(data), replace=True)
    s = [(xi-original_sample_mean)**2 for xi in bootstrap_sample]
    s2 = sum(s)/(n-1)
    bootstrap_var.append(s2)

# Sanity check with imported functions

print(f"Variance from original sample: {statistics.variance(data)}") # var from orig
print(f"Estimated variance with bootstrapping: {sum(bootstrap_var)/N}") # estimated

```

Variance from original sample: 34.31428571428572  
 Estimated variance with bootstrapping: 34.23207571428577

## Write a subroutine that takes as input a “data” vector of

observed values, and which outputs the median as well as the bootstrap estimate of the variance of the median, based on  $r = 100$  bootstrap replicates. Simulate  $N = 200$  Pareto distributed random variates with  $\beta = 1$  and  $k = 1.05$ .

In [ ]:

In [ ]:

```

r = 100 # bootstrap replicates
N = 200 # number of Pareto distributed random variates with
beta = 1
k = 1.05

# Compute the mean and the median (of the sample)
#sample = np.random.pareto(1, size = N)
sample = (np.random.pareto(k, N) + 1) * beta
sample_mean = np.mean(sample)
sample_median = np.median(sample)

#print(sample_mean)
#print(sample_median)

# Make the bootstrap estimate of the variance of the sample mean
number_of_bootstraps = 10000
bootstrap_var = []
bootstrap_medians = []

for i in range(number_of_bootstraps):
    bootstrap_sample = np.random.choice(sample, size = len(sample), replace=True)

    # median
    bootstrap_medians.append(np.median(bootstrap_sample))

    # var
    s = [(xi-sample_mean)**2 for xi in bootstrap_sample]

```

```
s2 = sum(s)/(N-1)
bootstrap_var.append(s2)
```

```
estimate_var = sum(bootstrap_var)/number_of_bootstraps
estimate_med = sum(bootstrap_medians)/number_of_bootstraps
```

```
print(f"Variance of sample: {np.var(sample)}")
print(f"Estimated variance based on bootstrapping: {estimate_var}")
print("")
print(f"Median of sample:{np.median(sample)}")
print(f"Estimated median based on bootstrapping: {estimate_med}")
# Make the bootstrap estimate of the variance of the sample median
```

```
# Compare the precision of the estimated median with the precision of the estimated mean
print("Compare the precision of the estimated median with the precision of the estimated mean")
print("The estimated median is more precise (closer to the actual median) compared to the estimated mean")
```

Variance of sample: 880.8063174045823

Estimated variance based on bootstrapping: 886.4578685906159

Median of sample:1.9114882713041397

Estimated median based on bootstrapping: 1.9002796691756276

Compare the precision of the estimated median with the precision of the estimated mean

The estimated median is more precise (closer to the actual median) compared to the estimated mean. This is probably due to the nature of mean and median, whereas the middle value (the median) will fluctuate less compared to the mean. The mean can skew the data with only one large datapoint, which will not affect the median.

In [ ]:

In [ ]: