ASSIGNMENT 4 – Confidence Interval

In this assignment we will first calculate confidence interval using population standard deviation which is unrealistic. The dataset we will be working with will be **countyComplete.csv** and will also be making use of the mosaic package.

We read the file by using the read.csv command.

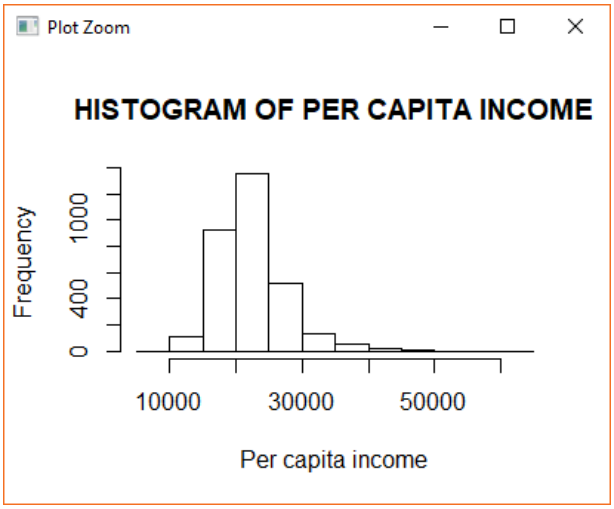
```
require( mosaic )
county = read.csv( "countyComplete.csv" )
```

We will then use the **mean** command to get the mean of the per capita income of the population whose confidence interval we are trying to compute.

```
mean_pop = mean(county$per_capita_income)
mean_pop
```

```
[1] 22504.7
```

We now check if the values of per capita income have a normal distribution.



Clearly the distribution is not normal. The population distribution is rightly skewed. Next we take a sample from this population and take its mean.

```
one_sample <- sample(county$per_capita_income, size=310, replace=TRUE)

mean_sample = mean(one_sample)

mean_sample
```

```
[1] 22029.89
```

We now compute the standard error of the mean using the standard deviation of the entire population. Standard error is calculated by dividing the standard deviation of the population by the square root of the number of observations in the sample. The process is shown below.

pop_var <- function(x){sum((x-mean(x))^2)/(length(x))}

pop_sd = pop_var(county$per_capita_income)^0.5

pop_se = pop_sd/(310^0.5)

pop_se

```
[1] 307.1428
```

We now use this standard error to calculate the confidence interval for the per capita income variable of the population.

low_CI_pop = mean_sample - pop_se * 1.96

up_CI_pop = mean_sample + pop_se * 1.96

CI_pop = c(lower = low_CI_pop, upper = up_CI_pop)

CI_pop

```
   lower    upper
21589.39 22793.39
```

Note that the above method calculating the confidence interval is unrealistic since in most practical situations, we only have information of the sample of a population and have no access to parameters of population such as population standard deviation in this case.

We will now implement two techniques to calculate the confidence interval for the mean of the per capita income of the population. We will start with the technique suggested in the OpenIntro Statistics book. For that we will calculate the standard error of the mean of the population in a different way. We will use the standard deviation of the sample **one_sample** in place of the standard deviation of the population. All the other variables to calculate the standard error will remain the same.

samp_sd = var(one_sample)^0.5

samp_sd

```
[1] 4403.369
```

samp_se = samp_sd/(310^0.5)

samp_se

```
[1] 250.0945
```

As we can see that the difference between the standard error calculated with sample standard deviation and population standard deviation varies greatly. This will surely have an effect on the confidence interval. Let us calculate that now.

low_CI_sam = mean_sample - samp_se * 1.96

up_CI_sam = mean_sample + samp_se * 1.96

CI_sam = c(lower = low_CI_sam, upper = up_CI_sam)

CI_sam

```
   lower    upper
21539.70 22520.07
```

We will now calculate the confidence interval of the population using the technique suggested in Kaplan's Statistical Modeling book. Here he suggests calculating the CI using a technique called **bootstrapping.** Here we create resamples of the sample in order to simulate the sampling variability of the population. The reason why the population parameters differ from the values calculated using a sample is due to sampling variability. In order to sufficiently explain the parameters of the population we need to account this sampling variability. Resampling is one way to account that. We will demonstrate the process below.

resamples <- mean(sample(one_sample, size=310, replace=TRUE))

for (i in 1:999){resamples <- c(resamples, mean(sample(one_sample, size=310, replace=TRUE)))}

The resamples variable above contains the simulated mean distribution of the population. Of course, this only accounts the variability of values present in the original **one_sample**. Nevertheless, this is a good approximation of the sampling variability of the population. The code below shows the first six values in the resamples variable.

head(resamples)

```
[1] 22573.78 21285.88 22065.31 22221.32 22824.50 22067.19
```

We will now compute the standard error using the resampling distribution. Here standard error is same as the standard deviation of the resampling distribution. We also calculate the mean value of the resampling distribution.

resamp_sd = var(resamples)^0.5

mean_resample = mean(resamples)

We are now ready to calculate the confidence interval using the standard error and mean of the resampling distribution. The code below shows the process.

low_CI_resam = mean_resample - resamp_sd * 1.96

up_CI_resam = mean_resample + resamp_sd * 1.96

CI_resam = c(lower = low_CI_resam, upper = up_CI_resam)

CI_resam

```
lower    upper
21666.81 22709.32
```

Now we are ready to compare the confidence interval calculated by all the three techniques. For the purpose of practicality, we will be ignoring the confidence interval generated by the population standard deviation as it is not feasible in most situations. Let us now recall the mean value of the population and the CI of both the techniques.

Population mean of per capita income : **22504.7**

CI using standard deviation of one sample : **21539.70** to **22520.07** with **95%** confidence.

CI using bootstrapping technique : **21666.81** to **22709.32** with **95%** confidence.

As we can see that the first technique barely captures the population mean and the technique suggested by Kaplan does capture the parameter more comfortably. But the size of the resampling distribution i.e. **1000** was decided rather arbitrarily. Let us now increase its value so as to see if it affects our estimate. We increase its value by more than twice i.e. **2200**. Let us implement the entire process one again.

```
resamples2 <- mean(sample(one_sample, size=310, replace=TRUE))

for (i in 1:2199){resamples2 <- c(resamples2, mean(sample(one_sample, size=310, replace=TRUE)))}

resamp2_sd = var(resamples2)^0.5

mean_resample2 = mean(resamples2)

low_CI_resam2 = mean_resample2 - resamp2_sd * 1.96

up_CI_resam2 = mean_resample2 + resamp2_sd * 1.96

CI_resam2 = c(lower = low_CI_resam2, upper = up_CI_resam2)

CI_resam2
```

```
lower    upper
21523.29 22509.88
```

Running the above code gives us confidence interval to be **21523.29** to **22509.88** with **95%** confidence which is in the similar range as the one we got from just one sample as it barely captures the population parameter. We can here observe that the method suggested by Kaplan's book can have no improvement over the one suggested by OpenIntro if we increase the resampling size.

One more thing to note here is that we have the population parameter i.e. **mean_pop = 22504.7** to show us which interval more comfortably captures the population parameter. In real world we won't be having this privilege as we will hardly ever have any population parameters to guide us. We might also be unsure of the true population size. In such uncertain scenario it might be difficult to accurately gauge the correct size of the resampling distribution in order to calculate the confidence interval. We can repeat this process multiple times in order to capture the most overlapping range from them to settle for one confidence interval range. This technique even though it can capture the population parameter more accurately than the single sample technique, due to its uncertain resampling size can be a hindrance where the population statistics are needed to be determined quickly.