# Confidence Intervals

This lesson will focus on the importance and calculation of confidence intervals.

WE'LL COVER THE FOLLOWING



A point estimate can give us a rough idea of a population parameter. But there is a chance that it contains some errors. We need to take many samples to reduce these errors. But taking many samples is not feasible all the time. As in our example of Data Scientists, it might not be feasible to take many different samples of 100 data scientists. What do we do if we face such a situation? We make *confidence intervals*.

## Confidence intervals #

A **confidence interval** is a range of values above and below a point estimate that might contain the true value of a population parameter. The interval is associated with a *confidence level*. The bigger the confidence level, the wider the range of the interval. The confidence level is decided before calculating the confidence intervals. Confidence intervals are usually reported with point estimates to show how reliable the estimates are.

The intervals are calculated from samples which means for each sample, there will be a different interval. Commonly, a 95% confidence level is used, which means there is a 95% chance that the true population parameter lies in the range. Or in other words, if we take 100 samples and calculate confidence intervals for each of the 100 samples with 95% confidence, the true population parameter will lie in 95 of these intervals.

## Calculating confidence intervals #

We can calculate confidence intervals from point estimates. We add and subtract a margin of error to the point estimate to calculate both boundaries of the interval. The margin of error depends on the:

- Confidence level chosen
- Size of the data
- Standard deviation of the data which denotes the spread of the data

There are two ways of calculating confidence intervals that depend on whether we know the standard deviation of the population or not.

## Confidence intervals using z-values #

If we know the standard deviation of the population then we can calculate the margin of error as

$$margin\ of\ error = z^* \frac{\sigma}{\sqrt{n}}$$

$z^*$= the critical z-value

$\sigma$ = standard deviation of the population

$n$ = sample size

The **critical $z^*$ value** for a 95% confidence interval is the z-score that tells us the number of standard deviations we must go above or below the mean to obtain an area of 95% in a standard normal distribution.

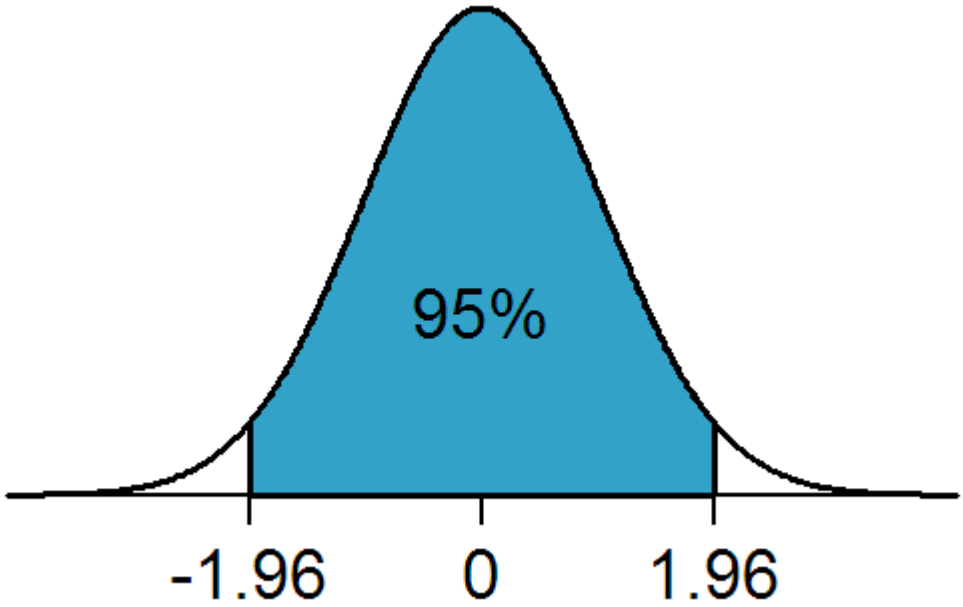
Let's see how we can calculate confidence intervals in Python. We will be using the Default of Credit Card Clients Dataset. We will sample from the `AGE` column as we did in the last lesson.

```
import pandas as pd
import scipy.stats as st
import math

df = pd.read_csv('credit_card_cleaned.csv')

# Calculate population parameters
pop_mean = df['AGE'].mean()
pop_std_dev = df['AGE'].std()
```

```
# Sample
sample_size = 100

sample = df['AGE'].sample(n = sample_size,random_state = 5)
sample_mean = sample.mean()

# Confidence interval caclulations
cl = 0.95
quantile = 1 - ( (1-cl)/2 )
z_value = st.norm.ppf(q = quantile)

margin_error = z_value * (pop_std_dev / math.sqrt(sample_size))
CI = (sample_mean - margin_error,sample_mean + margin_error)


print('Sample Mean : ',sample_mean)
print('margin of error : ', margin_error)
print('CI : ',CI)
print('Population Mean : ', pop_mean)
```

We calculate the population mean and standard deviation in **lines 8-9** using the functions `mean` and `std` respectively. We set `sample_size` to `100` in **line 12**. In the next line, we sample from the `AGE` column using the function `sample` by passing it `n = sample_size` and a random state for repeatability. Then we take the sample mean in **line 14**.

Now to calculate the confidence interval, we need to have the $z^*$ value. We know from the below graph that for a standard normal distribution, the z-score for obtaining an area of 95% is 1.96.




However we can calculate this value using the Python library `scipy` which

However, we can calculate this value using the Python library `scipy` which has a `stats` module that has functions related to statistics. We import the

library `scipy.stats` as `st`. Then we use the function `norm.ppf` to calculate the z-score. The `norm` is for functions related to the normal distribution. This function `norm.ppf` expects a quantile. We can calculate the quantile for a confidence interval using the formula

$$quantile = 1 - \frac{(1 - CL)}{2}$$

We set the confidence level as `cl` to `0.95` in **line 17**. Then, we calculate the quantile using the above formula in the next line and finally use the function `norm.ppf` in **line 19** to retrieve the z-score. We calculate the margin of error using its formula in **line 21**. To calculate the square root, we have imported the library `maths` which has the function `sqrt` that we have used in **line 21**. In the next line, we store the confidence interval values in a *tuple*. Then we print all of our findings in **lines 25-28**.

From the output, we see that the population mean lies in between the confidence interval.

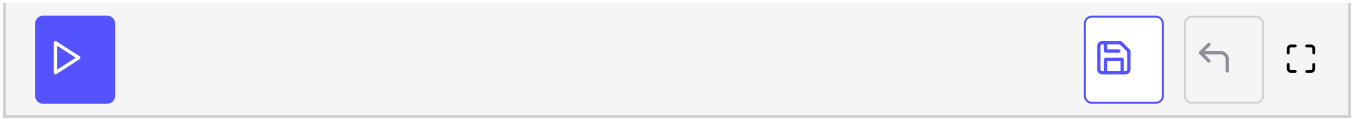
## Confidence intervals using t-values #

In most cases, we do not know the standard deviation of the population. In that case, we use the standard deviation of the sample instead of the population. But since using the standard deviation of the sample may induce some error, therefore, to account for the error, we calculate **t-values** instead of z-values. t-values are drawn from a **t-distribution**, which is similar to a standard normal distribution, but it goes wider as the sample size falls. But for greater sample sizes, t-distribution is very close to a normal distribution.

Let's see an example of calculating confidence intervals using the t-distribution. We will use the same data but this time we can assume that the credit card data that we have is a sample of the population.

```
import pandas as pd
import scipy.stats as st
import math

df = pd.read_csv('credit_card_cleaned.csv')

# Population parameter
```

```
pop_mean = df['AGE'].mean()

# Sample
sample_size = 100
sample = df['AGE'].sample(n = sample_size,random_state = 5)
sample_mean = sample.mean()
sample_std_dev = sample.std()

# Calculation of confidence interval
cl = 0.95
quantile = 1 - ( (1-cl)/2 )
t_value = st.t.ppf(q = quantile,df = sample_size-1)

margin_error = t_value * (sample_std_dev / math.sqrt(sample_size))
CI = (sample_mean - margin_error,sample_mean + margin_error)


print('Sample Mean : ',sample_mean)
print('margin of error : ', margin_error)
print('CI : ',CI)
print('Population Mean : ', pop_mean)
```



The code is almost the same as above. The first difference is that we have calculated the standard deviation of the sample in **line 14** and used that in the formula of the margin of error in **line 21**. Also, we have used the t-distribution instead of normal distribution. Therefore, we have used the function `t.ppf` instead of `norm.ppf` to calculate the t-values. From the output, we can see that the margin of error is greater, in this case, to account for the variability induced by using the standard deviation of the sample.

So, now whenever we have data that we know is not a population, we can report confidence intervals with whichever statistic we report in our findings. In the next lesson, we will use confidence intervals again to learn about *hypothesis testing*.