Payroll Assignment

Now it's assignment time! Let's start by testing the knowledge we have gained so far.

Introduction

This Jupyter notebook is part of your learning experience in the study of applied statistics.

You will work with data sets that contain payroll data of employees of a particular company.

In this exercise, you will perform the following tasks:

1 - Load and study the data.

2 - Clean the data and prepare it for further analysis.

3 - Conduct a hypothesis test for the data using Z-scores.

4 - Conduct a t-test for the data.

Task 1 - Load and study the data

Load the libraries.

In [2]:
```
# Load "numpy" and "pandas" for manipulating numbers and data frames
###################### CODE HERE ##########################
import numpy as np
import pandas as pd
```

Load the csv file as pandas dataframe.

In [3]:
```
# Read in the "Payroll_2015.csv" file as a Pandas Data Frame and store it as "df_2015"
# Note: Make sure the code and the data are in the same folder or specify the appropriate path
###################### CODE HERE ##########################
df_2015=pd.read_csv(r"C:\Users\vaish\Downloads\Copy of Payroll_2015.csv")
```

Reference:

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.head.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.head.html)

In [4]:
```
# Take a brief look at the data frame "df_2015" using ".head()"
###################### CODE HERE ##########################
df_2015.head()
```

Out[4]:

	Row ID	Projected Annual Salary
0	114364	$38857.68
1	114365	$38857.68
2	114366	$35078.40
3	114367	$35078.40
4	114368	$35078.40

In [5]:
```
# Study the description of the data
# Note: Make sure the code and the data description are in the same folder or specify the appropriate path
with open(r'C:\Users\vaish\Downloads\Copy of Payroll_2015_Feature_Description.txt', 'r') as f:
    print(f.read())
```

```
The data has 2 columns and 59678 rows.
The columns are as follows:
1. Row ID : the ids
2. Projected Annual Salary : the salaries in $s.
```

Reference:-https://www.w3schools.com/python/pandas/ref_df_info.asp (https://www.w3schools.com/python/pandas/ref_df_info.asp)

In [6]:
```
# Look at basic information about the data frame "df_2015" using ".info()"
###################### CODE HERE ##########################
df_2015.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 59767 entries, 0 to 59766
Data columns (total 2 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   Row ID                  59767 non-null  int64 
 1   Projected Annual Salary  59767 non-null  object
dtypes: int64(1), object(1)
memory usage: 934.0+ KB
```

In [7]:
```
# Read in the "Payroll_2016_Sample.csv" file as a Pandas Data Frame and store it as "df_2016_sample"
# Note: Make sure the code and the data are in the same folder or specify the appropriate path
###################### CODE HERE ##########################
df_2016_sample=pd.read_csv(r'C:\Users\vaish\Downloads\Copy of Payroll_2016_Sample.csv')
```

In [8]:
```
# Take a brief look at the data frame "df_2016_sample" using ".head()"
###################### CODE HERE ##########################
df_2016_sample.head()
```

Out[8]:

	Row ID	Projected Annual Salary
0	206226	$80659.44
1	236669	$37688.40
2	232394	$80137.44
3	190682	$99764.64
4	218049	$57795.84

In [9]:
```
# Study the description of the data
# Note: Make sure the code and the data description are in the same folder or specify the appropriate path
with open(r'C:\Users\vaish\Downloads\Copy of Payroll_2016_Sample_Feature_Description.txt','r') as f:
  print(f.read())
```

```
The data has 2 columns and 59678 rows.
The columns are as follows:
1. Row ID : the ids
2. Projected Annual Salary : the salaries in $s.
```

In [10]:
```
# Look at basic information about the data frame "df_2016_sample" using ".info()"
###################### CODE HERE ##########################
df_2016_sample.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11091 entries, 0 to 11090
Data columns (total 2 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   Row ID                  11091 non-null  int64 
 1   Projected Annual Salary  11091 non-null  object
dtypes: int64(1), object(1)
memory usage: 173.4+ KB
```

Reference:- https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html (https://docs.scipy.org/doc/scipy/reference/tutorial/stats.html)

https://www.statsmodels.org/stable/generated/statsmodels.stats.weightstats.ztest.html
(https://www.statsmodels.org/stable/generated/statsmodels.stats.weightstats.ztest.html)

In [11]:
```
# Load "scipy.stats" for scientific and statistical methods
# Load "statsmodels.stats.weightstats.ztest" for Z-tests
###################### CODE HERE ######################
from scipy import stats
from statsmodels.stats.weightstats import ztest
```

Observations:

We are interested in checking whether average annual salaries have increased from 2015 to 2016.

The data from 2015 is used only to arrive at a mean value for the null hypotheses in this exercise.

The data on which the actual hypothesis testing is done is the data from 2016.

Note: The 2016 data set is a sample and not the actual population data for 2016

In [12]:
```
##########################################################################
```

Task 2 - Clean the data and prepare it for further analysis

Reference:- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rename.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rename.html)

Change the column name to something much more interpretable.

In [13]:
```
# Rename the "Projected Annual Salary" feature to "CTC" for both data frames
# Note: Use the ".rename()" method and update the original data frames "df_2015" and "df_2016_sample"
# Note: The old and new column names must be passed to the "columns" parameter as a dictionary
###################### CODE HERE ##########################
df_2015=df_2015.rename(columns={'Projected Annual Salary':'CTC'})
df_2016_sample=df_2016_sample.rename(columns={'Projected Annual Salary':'CTC'})
```

In [14]:
```
# Remove the "$" symbol from the "CTC" feature in both data frames using ".str.replace()" method
# Note: Pass the values "$" and "" as two parameters to the ".str.replace()" function for the "CTC" feature
# Note: The "regex" parameter must be set to "True"
###################### CODE HERE ##########################
df_2015['CTC']=df_2015['CTC'].str.replace('$','',regex=True)
df_2016_sample['CTC']=df_2016_sample['CTC'].str.replace('$','',regex=True)
```

In [15]:
```
# Convert the "CTC" feature to the "float" data type in both data frames using the ".astype()" method
# Note: Pass the value "float" as a parameter to the ".astype()" function
###################### CODE HERE ##########################
df_2015['CTC']=df_2015['CTC'].astype(float)
df_2016_sample['CTC']=df_2016_sample['CTC'].astype(float)
```

In [16]:
```
# Remove any entries in the data frame "df_2015" that contain "CTC" values as 0
# Note: You can update the original data frame "df_2015" to contain "CTC" values greater than 0
###################### CODE HERE ##########################
df_2015=df_2015[df_2015['CTC']>0]
```

In [17]:
```
# Remove any entries in the data frame "df_2016_sample" that contain "CTC" values as 0
# Note: You can update the original data frame "df_2016_sample" to contain "CTC" values greater than 0
###################### CODE HERE ##########################
df_2016_sample=df_2016_sample[df_2016_sample['CTC']>0]
```

Observation:

A rigorous way to check whether the compensations have increased from 2015 to 2016 or not is by using hypothesis tests.

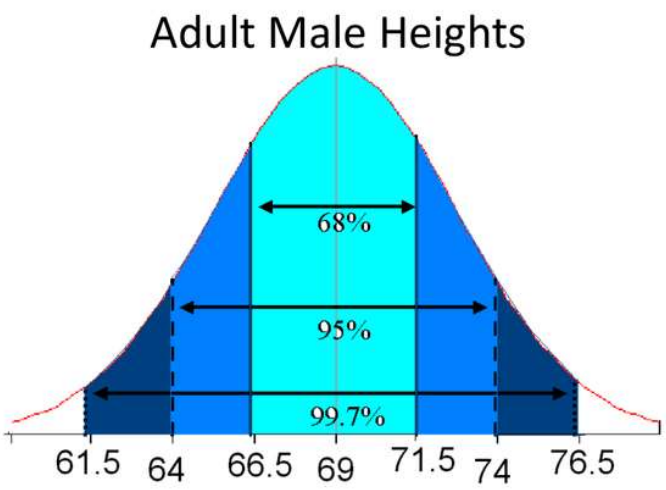
In [18]:
```
##########################################################################
```

Task 3 - Conduct a hypothesis test for the data using Z-scores

The null hypothesis is: The annual compensation of employees does not increase from the year 2015 to 2016.

The alternate hypothesis is: The annual compensation of employees increases from the year 2015 to 2016.

Z-Scores are measurements of how far from the center (mean) a data value falls.



Ex: A man who stands 71.5 inches tall is 1 standard deviation ABOVE the mean.(*z-score = 1*)

Ex: A man who stands 64 inches tall is 2 standard deviations BELOW the mean. (*z-score = -2*)

In [19]:
```
# Since we do not have the population standard deviation for the 2016 data, we will use the adjusted formula
# The adjusted formula for the Z-score is "z = (X - m) / s"
# "X" is the actual mean of the 2016 sample
# "m" is the mean of the 2016 sample under the null hypothesis, so we can use the mean of the 2015 data in pl
# "s" is either the population standard deviation for 2016 or the adjusted sample standard deviation for 2016
# We do not have the population standard deviation for 2016
# So, we will use the adjusted formula "s = standard deviation of sample / sqrt(number of observations in the
# Calculate a Z-score for the 2016 sample data using the appropriate formula and store it as "z_score"
##################### CODE HERE ##########################
num=df_2016_sample['CTC'].mean()-df_2015['CTC'].mean()
den=df_2016_sample['CTC'].std()/np.sqrt(len(df_2016_sample))
z_score=num/den
```

In [20]:
```
# Since this is a one-tailed test, the critical Z-score for a confidence level of 95% is about 1.65
# Print the value of "z_score"
##################### CODE HERE ##########################
z_score
```

Out[20]: 6.888610601050524

Reference:- https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html)

In [36]:
```
# Calculate the p-value associated with the Z-score using "stats.norm.sf(abs(z_score))"
# Note: Store the resulting p-value as "p_value_z"
##################### CODE HERE ##########################
p_value_z=stats.norm.sf(abs(z_score))
```

In [37]:
```
# Print the value of "p_value_z"
##################### CODE HERE ##########################
p_value_z
```

Out[37]: 2.8169961393097742e-12

Please refer to this article. It is useful for thorough study.

Reference:- https://www.statology.org/z-test-python/ (https://www.statology.org/z-test-python/)

A more brief video for the type of tests with some examples.

In [38]:
```
# Run a Z-test for the data using the "ztest()" method
# Store the resulting Z-score as "z_stat" and the resulting p-value as "p_val_z"
# Note: The parameter "x1" must be set to "df_2016_sample['CTC']"
# Note: The parameter "x2" must be set to "None" as this is a single sample test
# Note: The parameter "alternative" must be set to "larger" as this is a one-tailed (upper) test
# Note: The parameter "value" must be set to the null hypothesis (2015) mean, that is "df_2015['CTC'].mean()
###################### CODE HERE ##########################
z_stat,p_val_z=ztest(x1=df_2016_sample['CTC'],x2=None,alternative='larger',value=df_2015['CTC'].mean())
```

In [39]:
```
# Print the value of "z_stat"
###################### CODE HERE ##########################
z_stat
```

Out[39]: 6.888610601050499

In [40]:
```
# Print the value of "p_val_z"
###################### CODE HERE ##########################
p_val_z
```

Out[40]: 2.816996139310275e-12

Observations:

The calculated Z-statistic (about 6.89) is greater than the critical Z-statistic (about 1.65).

The calculated p-value (nearly 0) is less than 0.05.

So, the null hypothesis may be safely rejected.

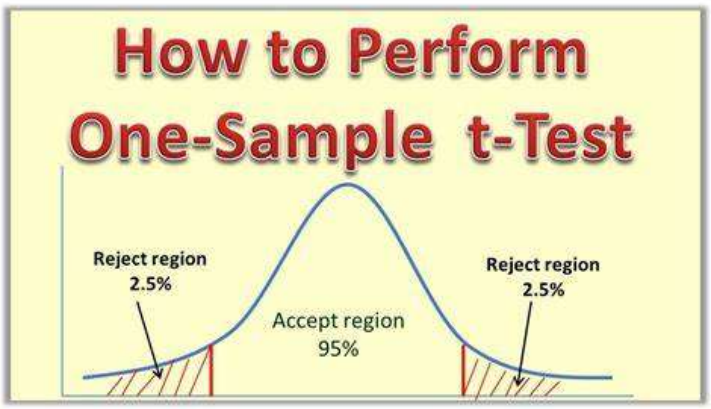
Thus, the alternate hypothesis that the annual salaries increase from the year 2015 to the year 2016 is true.

In [26]:
```
##########################################################################
```

Task 4 - Conduct a t-test for the data

The null hypothesis is: The annual compensation of employees does not increase from the year 2015 to 2016.

The alternate hypothesis is: The annual compensation of employees increases from the year 2015 to 2016.



In [41]:
```
# The formula for the t-statistic is "t = (X - m) / s"
# "X" is the actual mean of the 2016 sample
# "m" is the mean of the 2016 sample under the null hypothesis, so we can use the mean of the 2015 data in pl
# "s" is the adjusted sample standard deviation for 2016
# The formula for "s" is "s = standard deviation of sample / sqrt(number of observations in the sample)"
# Calculate a t-statistic for the 2016 sample data using the appropriate formula and store it as "t_score"
###################### CODE HERE ##########################
num=df_2016_sample['CTC'].mean()-df_2015['CTC'].mean()
den=df_2016_sample['CTC'].std()/np.sqrt(len(df_2016_sample))
t_score=num/den
```

In [42]:
```
# Since this is a one-tailed test with many degrees of freedom, the critical t-statistic for a confidence lev
# Print the value of "t_score"
###################### CODE HERE ##########################
t_score
```

Out[42]: 6.888610601050524

Reference:- https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.t.html (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.t.html)

In [43]:
```
# Calculate the p-value associated with the t-statistic using "stats.t.sf(abs(t_score), len(df_2016_sample) -
# Note: Store the resulting p-value as "p_value_t"
###################### CODE HERE ##########################
p_value_t=stats.t.sf(abs(t_score), len(df_2016_sample) - 1)
```

In [44]:
```
# Print the value of "p_value_t"
###################### CODE HERE ##########################
p_value_t
```

Out[44]: 2.9698553116313644e-12

Reference:- https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_1samp.html (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_1samp.html)

In [47]:
```
# Run a t-test for the data using the "stats.ttest_1samp()" method
# Store the resulting t-statistic as "t_stat" and the resulting p-value as "p_val_t"
# Note: The parameter "a" must be set to "df_2016_sample['CTC']"
# Note: The parameter "popmean" must be set to the null hypothesis (2015) mean, that is "df_2015['CTC'].mean
# Note: The parameter "alternative" must be set to "greater" as this is a one-tailed (upper) test
###################### CODE HERE ##########################
t_stat,p_val_t=stats.ttest_1samp(a=df_2016_sample['CTC'],popmean=df_2015['CTC'].mean(),alternative='greater'
```

In [48]:
```
# Print the value of "t_stat"
###################### CODE HERE ##########################
t_stat
```

Out[48]: 6.888610601050499

In [49]:
```
# Print the value of "p_val_t"
###################### CODE HERE ##########################
p_val_t
```

Out[49]: 2.9698553116318915e-12

Observations:

The t-distribution becomes nearly equivalent to the standard normal distribution for large sample sizes.

The calculated t-statistic (about 6.89) is greater than the critical t-statistic (about 1.65).

The calculated p-value (nearly 0) is less than 0.05.

So, the null hypothesis may be safely rejected.

Thus, the alternate hypothesis that the annual salaries increase from the year 2015 to the year 2016 is true.

In [34]:
```
##########################################################################
```

Conclusion

We can use hypothesis testing methods such the Z-score method and the Student's t-test to verify various hypotheses

In [35]:
```
##########################################################################
```

FEEDBACK

We hope you've enjoyed this course so far. We're committed to help you use "Stats and maths for data science" course to its full potential, so that you have a great learning experience. And that's why we need your help in form of a feedback here.

Please fill this feedback form
https://forms.zohopublic.in/cloudyml/form/CloudyMLStatisticsFeedbackForm/formperma/WV946wnf0sDM_tOlH87RxZR9yMceKWGrtuF
(https://forms.zohopublic.in/cloudyml/form/CloudyMLStatisticsFeedbackForm/formperma/WV946wnf0sDM_tOlH87RxZR9yMceKWGrtu