Student's T-Test

In 1908 **William Sealy Gosset**, an Englishman publishing under the pseudonym Student, developed the t-test and t distribution.

A t-test is a statistical test that is used to compare the means of two groups. It is often used in hypothesis testing to determine whether a process or treatment actually has an effect on the population of interest, or whether two groups are different from one another.

There are three types of t-tests we can perform based on the data at hand: One sample t-test. Independent two-sample t-test. Paired sample t-test.

In [ ]:
```
#import libararies
import pandas as pd
import seaborn as sns
import scipy as sc
import matplotlib.pyplot as plt
import numpy as np
```

In [ ]:
```
salarykadata = pd.read_csv('ml_data_salary.csv')
salarykadata.head()
```

Out[ ]:

	age	distance	YearsExperience	Salary
0	31.1	77.75	1.1	39343
1	31.3	78.25	1.3	46205
2	31.5	78.75	1.5	37731
3	32.0	80.00	2.0	43525
4	32.2	80.50	2.2	39891

In [ ]:
```
salarykadata.isna().sum()
```

Out[ ]:
```
age                0
distance           0
YearsExperience    0
Salary             0
dtype: int64
```

In [ ]:
```
#binning of age column into 3 catagories
bins = np.linspace(min(salarykadata['age']),max(salarykadata['age']),4)
age_groups= ['bachy','jawan','borhy']
salarykadata['age']= pd.cut(salarykadata['age'],bins,labels=age_groups, include_lowe
salarykadata['age']
```

Out[ ]:
```
0     bachy
1     bachy
2     bachy
3     bachy
4     bachy
5     bachy
6     bachy
```

```
7     bachy
8     bachy
9     bachy
10    bachy
11    bachy
12    bachy
13    bachy
14    jawan
15    jawan
16    jawan
17    jawan
18    jawan
19    jawan
20    jawan
21    jawan
22    borhy
23    borhy
24    borhy
25    borhy
26    borhy
27    borhy
28    borhy
29    borhy
Name: age, dtype: category
Categories (3, object): ['bachy' < 'jawan' < 'borhy']
```

In [ ]:
```
df = salarykadata[['age','YearsExperience','Salary']]
df.head()
```

Out[ ]:

	age	YearsExperience	Salary
0	bachy	1.1	39343
1	bachy	1.3	46205
2	bachy	1.5	37731
3	bachy	2.0	43525
4	bachy	2.2	39891

One-sample student's t-test

Test a sample with a known standard value. **Assumptions**

- Observations in each sample are independent and identically distributed.
- Observations in each sample are normally distributed.
- **Interpretation**

H0: the means of the samples are equal to the known value.

H1: the means of the samples are unequal to the known value.

In [ ]:
```
# 1 sample t test to compare the salary of young workers with 40000
#1. import libarary
from scipy.stats import ttest_1samp

#2. sub set of age by bachy jawan borhy
df_bachy=  df[df['age']=='bachy']
df_jawan =  df[df['age']=='jawan']
```

```
df_borhy=  df[df['age']=='borhy']

#3. t test
stat,p = ttest_1samp(df_jawan['Salary'],40000)
print('stat=%.3f,p=%.3f'% (stat,p))

#4. make a conditional argument for further case
if p > 0.05:
  print('There is no significance difference')
else:
    print('There is a significance difference')
```

```
stat=8.165,p=0.000
There is a significance difference
```

Independent student's t-test

Assumptions

- Observations in each sample are independent and identically distributed.
- Observations in each sample are normally distributed.
- Observations in each sample have the same variance.

Interpretation

H0: the means of the samples are equal.

H1: the means of the samples are unequal

In [ ]:

```
# 2 sample t test to compare the salary of jawan and borhy

#1. import libarary
from scipy.stats import ttest_ind

#2. sub set of age by bachy borhy jawan
df_jawan =  df[df['age']=='jawan']
df_borhy=  df[df['age']=='borhy']

#3. t test(unpaired/two sample/independent)
stat,p = ttest_ind(df_jawan['Salary'],df_borhy['Salary'])
print('stat=%.3f,p=%.3f'% (stat,p))

#4. make a conditional argument for further case
if p > 0.05:
  print('There is no significance difference')
else:
    print('There is a significance difference')
```

```
stat=-5.806,p=0.000
There is a significance difference
```

Paired student's t-test

Tests whether the means of two paired samples are significantly different. **Assumptions**

- Observations in each sample are independent and identically distributed.
- Observations in each sample are normally distributed.
- Observations in each sample have the same variance.

- Observations across each sample are paired.
- **Interpretation**

H0: the means of the samples are equal.

H1: the means of the samples are unequal.

In [ ]:
```
#binning the YearsExperience column
bins = np.linspace(min(df['YearsExperience']),max(df['YearsExperience']),3)
ex_grp= ['expert','newbee']
df['YearsExperience']= pd.cut(df['YearsExperience'],bins,labels=ex_grp, include_lowe
df['YearsExperience']
```

```
C:\Users\Azka\AppData\Local\Temp/ipykernel_11944/1679920551.py:3: SettingWithCopyWar
ning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  df['YearsExperience']= pd.cut(df['YearsExperience'],bins,labels=ex_grp, include_lo
west=True)
```

Out[ ]:
```
0     expert
1     expert
2     expert
3     expert
4     expert
5     expert
6     expert
7     expert
8     expert
9     expert
10    expert
11    expert
12    expert
13    expert
14    expert
15    expert
16    expert
17    expert
18    newbee
19    newbee
20    newbee
21    newbee
22    newbee
23    newbee
24    newbee
25    newbee
26    newbee
27    newbee
28    newbee
29    newbee
Name: YearsExperience, dtype: category
Categories (2, object): ['expert' < 'newbee']
```

In [ ]:
```
# 2 sample t test to compare the salary of young and experienced worker with young a

#1. import libarary
from scipy.stats import ttest_rel

#2. sub set of age by bachy borhy jawan
```

```
df_bachy=  df[df['age']=='bachy']
df_jawan =  df[df['age']=='jawan']

df_jawan_expert =  df_jawan[df_jawan['YearsExperience']=='expert']
df_jawan_expert.head()
df_jawan_newbee =  df_jawan[df_jawan['YearsExperience']=='newbee']
df_jawan_newbee.head()

# equaling the rows of the df_jawan_expert and df_jawan_newbee
df_male_1st= df_jawan_expert.sample(n=10,replace=True)
df_male_2nd= df_jawan_newbee.sample(n=10,replace=True)

#3. t test(paired/two sample/dependent)
stat,p = ttest_rel(df_jawan_expert['Salary'],df_jawan_newbee['Salary'])
print('stat=%.3f,p=%.3f'% (stat,p))

#4. make a conditional argument for further case
if p > 0.05:
  print('There is no significance difference')
else:
    print('There is a significance difference')
```

```
stat=-8.486,p=0.003
There is a significance difference
```