

ATTRITION ASSIGNMENT

STEP 1 = LAUNCHING :

```
import pandas as p
```

```
import numpy as n
```

```
import matplotlib.pyplot as pl
```

```
data=pd.read_csv("general_data.csv")
```

->To find column names .

```
data.columns
```

```
In [6]: data.columns
Out[6]:
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',
       'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
       'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object')
```

->To find the data of first 5 rows.

```
data.head( )
```

```
In [7]: data.head()
Out[7]:
```

	Age	Attrition	...	YearsSinceLastPromotion	YearsWithCurrManager
0	51	No	...	0	0
1	31	Yes	...	1	4
2	32	No	...	0	3
3	38	No	...	7	5
4	32	No	...	0	4

```
[5 rows x 24 columns]
```

STEP 2 = DATA TREATMENT :

->To find out null values in the table.

`data.isnull()`

```
In [8]: data.isnull()
Out[8]:
```

	Age	Attrition	...	YearsSinceLastPromotion	YearsWithCurrManager
0	False	False	...	False	False
1	False	False	...	False	False
2	False	False	...	False	False
3	False	False	...	False	False
4	False	False	...	False	False
...
4405	False	False	...	False	False
4406	False	False	...	False	False
4407	False	False	...	False	False
4408	False	False	...	False	False
4409	False	False	...	False	False

[4410 rows x 24 columns]

->To find out duplicated values of table.

`data.duplicated()`

```
In [9]: data.duplicated()
Out[9]:
```

0	False
1	False
2	False
3	False
4	False
...	
4405	False
4406	False
4407	False
4408	False
4409	False

Length: 4410, dtype: bool

->To drop all duplicated values of the table.

```
data.drop_duplicates( )
```

```
In [11]: data.drop_duplicates()
Out[11]:
```

	Age	Attrition	...	YearsSinceLastPromotion	YearsWithCurrManager
0	51	No	...	0	0
1	31	Yes	...	1	4
2	32	No	...	0	3
3	38	No	...	7	5
4	32	No	...	0	4
...
4405	42	No	...	0	2
4406	29	No	...	0	2
4407	25	No	...	1	2
4408	42	No	...	7	8
4409	40	No	...	3	9

[4410 rows x 24 columns]

STEP 3 = UNIVARIATE ANALYSIS :

->To describe the whole table.

```
data1=data[['Age', 'Attrition', 'BusinessTravel', 'Department',
'DistanceFromHome','Education', 'EducationField', 'EmployeeCount',
'EmployeeID', 'Gender','JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
'NumCompaniesWorked', 'Over18', 'PercentSalaryHike',
'StandardHours','StockOptionLevel', 'TotalWorkingYears',
'TrainingTimesLastYear','YearsAtCompany', 'YearsSinceLastPromotion',
'YearsWithCurrManager']].describe( )
```

```

In [14]: data1
Out[14]:

```

	Age	...	YearsWithCurrManager
count	4410.000000	...	4410.000000
mean	36.923810	...	4.123129
std	9.133301	...	3.567327
min	18.000000	...	0.000000
25%	30.000000	...	2.000000
50%	36.000000	...	3.000000
75%	43.000000	...	7.000000
max	60.000000	...	17.000000

```

[8 rows x 16 columns]

```

->To find out median of each column .

```

data2=data[['Age', 'Attrition', 'BusinessTravel', 'Department',
'DistanceFromHome','Education', 'EducationField', 'EmployeeCount',
'EmployeeID', 'Gender','JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
'NumCompaniesWorked', 'Over18', 'PercentSalaryHike',
'StandardHours','StockOptionLevel', 'TotalWorkingYears',
'TrainingTimesLastYear','YearsAtCompany', 'YearsSinceLastPromotion',
'YearsWithCurrManager']].median( )

```

```

In [16]: data2
Out[16]:

```

Age	36.0
DistanceFromHome	7.0
Education	3.0
EmployeeCount	1.0
EmployeeID	2205.5
JobLevel	2.0
MonthlyIncome	49190.0
NumCompaniesWorked	2.0
PercentSalaryHike	14.0
StandardHours	8.0
StockOptionLevel	1.0
TotalWorkingYears	10.0
TrainingTimesLastYear	3.0
YearsAtCompany	5.0
YearsSinceLastPromotion	1.0
YearsWithCurrManager	3.0

```

dtype: float64

```

->To find out mean of each column.

```
data3=data[['Age', 'Attrition', 'BusinessTravel', 'Department',  
'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',  
'EmployeeID', 'Gender', 'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',  
'NumCompaniesWorked', 'Over18', 'PercentSalaryHike',  
'StandardHours', 'StockOptionLevel', 'TotalWorkingYears',  
'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',  
'YearsWithCurrManager']].mean( )
```

```
In [18]: data3  
Out[18]:  
Age                36.923810  
DistanceFromHome   9.192517  
Education          2.912925  
EmployeeCount      1.000000  
EmployeeID         2205.500000  
JobLevel           2.063946  
MonthlyIncome      65029.312925  
NumCompaniesWorked 2.694830  
PercentSalaryHike  15.209524  
StandardHours      8.000000  
StockOptionLevel   0.793878  
TotalWorkingYears  11.279936  
TrainingTimesLastYear 2.799320  
YearsAtCompany     7.008163  
YearsSinceLastPromotion 2.187755  
YearsWithCurrManager 4.123129  
dtype: float64
```

->To find out mode.

```
data4=data[['Age', 'Attrition', 'BusinessTravel', 'Department',  
'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',  
'EmployeeID', 'Gender', 'JobLevel', 'JobRole', 'MaritalStatus',  
'MonthlyIncome', 'NumCompaniesWorked', 'Over18',  
'PercentSalaryHike', 'StandardHours', 'StockOptionLevel',  
'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany',  
'YearsSinceLastPromotion', 'YearsWithCurrManager']].mode( )
```

```

In [20]: data4
Out[20]:
   Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager
0  35.0         No ...                   0.0                2.0
1   NaN         NaN ...                   NaN                NaN
2   NaN         NaN ...                   NaN                NaN
3   NaN         NaN ...                   NaN                NaN
4   NaN         NaN ...                   NaN                NaN
...  ...         ... ...                   ...                ...
4405 NaN         NaN ...                   NaN                NaN
4406 NaN         NaN ...                   NaN                NaN
4407 NaN         NaN ...                   NaN                NaN
4408 NaN         NaN ...                   NaN                NaN
4409 NaN         NaN ...                   NaN                NaN

[4410 rows x 24 columns]

```

->To find variance of each column.

```

data5=data[['Age', 'Attrition', 'BusinessTravel', 'Department',
'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
'EmployeeID', 'Gender', 'JobLevel', 'JobRole', 'MaritalStatus',
'MonthlyIncome', 'NumCompaniesWorked', 'Over18',
'PercentSalaryHike', 'StandardHours', 'StockOptionLevel',
'TotalWorkingYears', 'TrainingTimesLastYear', 'YearsAtCompany',
'YearsSinceLastPromotion', 'YearsWithCurrManager']].var( )

```

```

In [22]: data5
Out[22]:
Age                8.341719e+01
DistanceFromHome   6.569144e+01
Education          1.048438e+00
EmployeeCount      0.000000e+00
EmployeeID         1.621042e+06
JobLevel           1.224760e+00
MonthlyIncome      2.215480e+09
NumCompaniesWorked 6.244436e+00
PercentSalaryHike  1.338907e+01
StandardHours      0.000000e+00
StockOptionLevel   7.257053e-01
TotalWorkingYears  6.056298e+01
TrainingTimesLastYear 1.661465e+00
YearsAtCompany     3.751728e+01
YearsSinceLastPromotion 1.037935e+01
YearsWithCurrManager 1.272582e+01
dtype: float64

```

->To find skewness.

```
data6=data[['Age', 'Attrition', 'BusinessTravel', 'Department',  
'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',  
'EmployeeID', 'Gender', 'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',  
'NumCompaniesWorked', 'Over18', 'PercentSalaryHike',  
'StandardHours', 'StockOptionLevel', 'TotalWorkingYears',  
'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',  
'YearsWithCurrManager']].skew( )
```

```
In [24]: data6  
Out[24]:  
Age                0.413005  
DistanceFromHome   0.957466  
Education          -0.289484  
EmployeeCount       0.000000  
EmployeeID         0.000000  
JobLevel           1.024703  
MonthlyIncome      1.368884  
NumCompaniesWorked 1.026767  
PercentSalaryHike  0.820569  
StandardHours      0.000000  
StockOptionLevel   0.968321  
TotalWorkingYears  1.116832  
TrainingTimesLastYear 0.552748  
YearsAtCompany     1.763328  
YearsSinceLastPromotion 1.982939  
YearsWithCurrManager 0.832884  
dtype: float64
```

->To find out kurtosis.

```
data7=data[['Age', 'Attrition', 'BusinessTravel', 'Department',  
'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',  
'EmployeeID', 'Gender', 'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',  
'NumCompaniesWorked', 'Over18', 'PercentSalaryHike',  
'StandardHours', 'StockOptionLevel', 'TotalWorkingYears',  
'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',  
'YearsWithCurrManager']].kurt( )
```

```

In [26]: data7
Out[26]:
Age                -0.405951
DistanceFromHome   -0.227045
Education           -0.560569
EmployeeCount       0.000000
EmployeeID         -1.200000
JobLevel            0.395525
MonthlyIncome       1.000232
NumCompaniesWorked  0.007287
PercentSalaryHike   -0.302638
StandardHours       0.000000
StockOptionLevel    0.361086
TotalWorkingYears   0.912936
TrainingTimesLastYear 0.491149
YearsAtCompany      3.923864
YearsSinceLastPromotion 3.601761
YearsWithCurrManager 0.167949
dtype: float64

```

->To find standard deviation .

```

data8=data[['Age', 'Attrition', 'BusinessTravel', 'Department',
'DistanceFromHome','Education', 'EducationField', 'EmployeeCount',
'EmployeeID', 'Gender','JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
'NumCompaniesWorked', 'Over18', 'PercentSalaryHike',
'StandardHours','StockOptionLevel', 'TotalWorkingYears',
'TrainingTimesLastYear','YearsAtCompany', 'YearsSinceLastPromotion',
'YearsWithCurrManager']].std( )

```



```

In [29]: data8
Out[29]:
Age                9.133301
DistanceFromHome   8.105026
Education          1.023933
EmployeeCount      0.000000
EmployeeID         1273.201673
JobLevel           1.106689
MonthlyIncome      47068.888559
NumCompaniesWorked 2.498887
PercentSalaryHike   3.659108
StandardHours      0.000000
StockOptionLevel   0.851883
TotalWorkingYears  7.782222
TrainingTimesLastYear 1.288978
YearsAtCompany     6.125135
YearsSinceLastPromotion 3.221699
YearsWithCurrManager 3.567327
dtype: float64

```

INFERENCE :

	MEDIAN	MEAN	STANDARD DEAVIATION	SKEWNESS	KURTOSIS	VARIANCE
Age	36	36.92	9.13	0.41	-0.4	83.41
DistanceFromHome	7	9.19	8.1	0.95	-0.22	65.69
Education	3	2.91	1.02	-0.28	-0.56	1.04
Employee Count	1	1	0	0	0	0
Employee ID	2205	2205.5	1273.2	0	-1.2	1.62E+06
Job Level	2	2.06	1.1	1.02	0.39	1.22
Monthly Income	49190	65029.31	47068.88	1.36	1	2.21E+09
Num Companies Worked	2	2.69	2.49	1.02	0.0072	6.22
Percent Salary Hike	14	15.2	3.65	0.82	-0.3	13.38
Standard Hours	8	8	0	0	0	0
Stock Option Level	1	0.79	0.85	0.96	0.36	72.57
Toatl Working Hours	10	11.27	7.78	1.11	0.91	60.56
Training Times Lasr Year	3	2.79	1.2	0.55	0.49	1.66
Years At Company	5	7	6.125	1.76	3.92	37.51
Years Since Last Promotion	1	2.18	3.22	1.98	3.6	10.37
Years With Current Manager	3	4.12	3.56	0.83	0.16	12.72

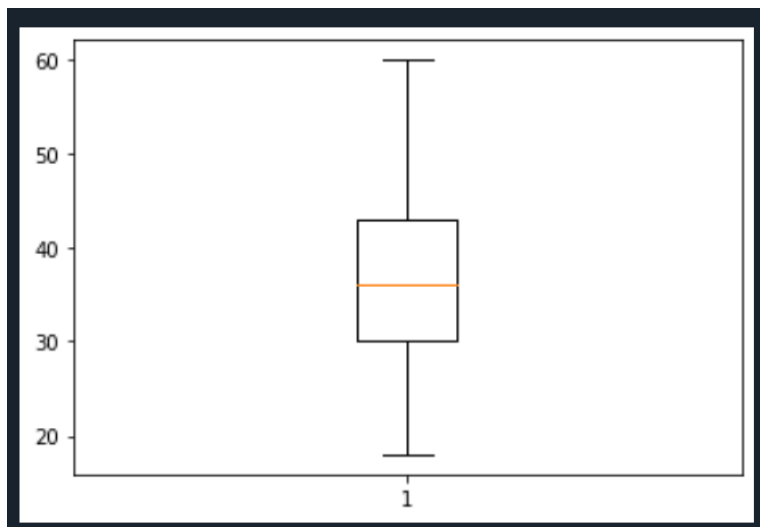
->All the above variables show positive skewness; while Age & Mean_distance_from_home are leptokurtic and all other variables are platykurtic.

OUTLIERS :

There's no regression found while plotting Age, MonthlyIncome, TotalWorkingYears , YearsAtCompany, etc., on a scatter plot.

```
box_plot=data.Age
```

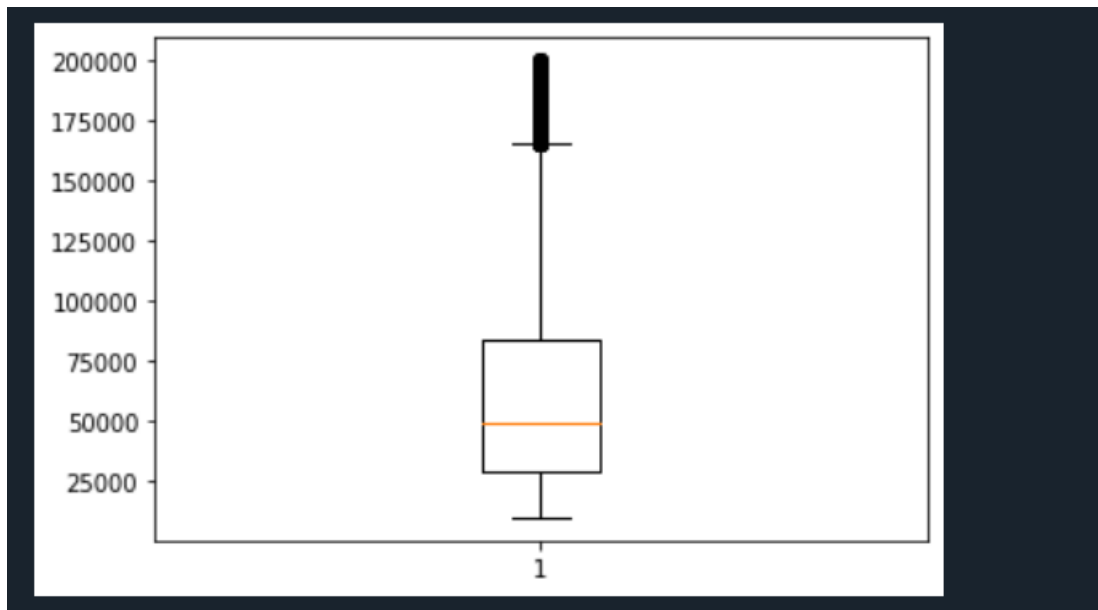
```
pl.boxplot(box_plot)
```



Age is normally distributed without any outliers

```
box_plot=data.MonthlyIncome
```

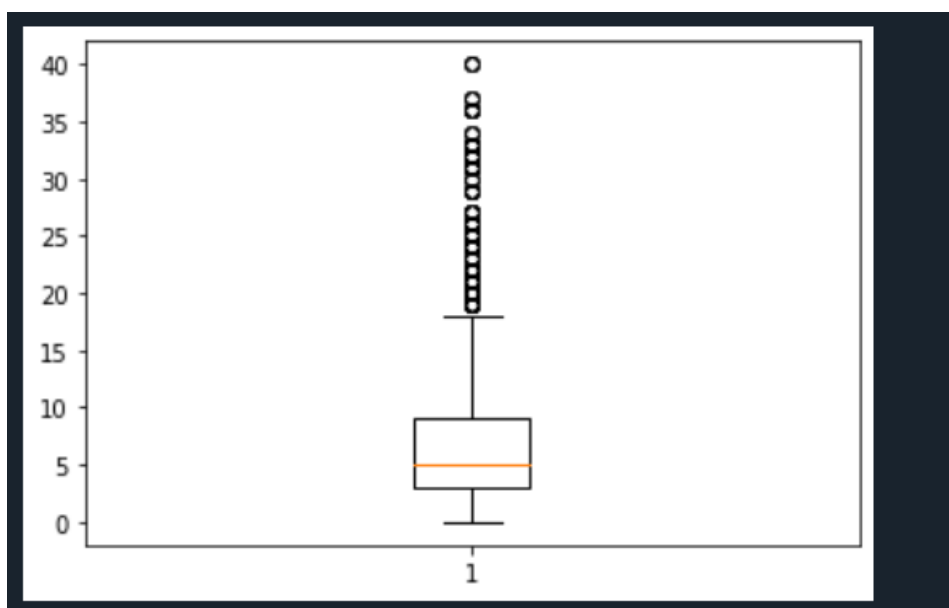
```
pl.boxplot(box_plot)
```



Monthly Income is Right skewed with several outliers

```
box_plot=data.YearsAtCompany
```

```
pl.boxplot(box_plot)
```



Years at company is also Right Skewed with several outliers observed.

4>STATISTICAL TEST (MANN-WHITNEY)

```
import pandas as pd
```

```
data=pd.read_csv("general_data.csv")
```

```
data.head()
```

```
   Age  Attrition  ...  YearsSinceLastPromotion  YearsWithCurrManager
0   51         No  ...                      0                      0
1   31         Yes  ...                      1                      4
2   32         No  ...                      0                      3
3   38         No  ...                      7                      5
4   32         No  ...                      0                      4

[5 rows x 24 columns]
```

```
data.columns
```

```
In [5]: data.columns
Out[5]:
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',
      'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',
      'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
      'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',
      'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
      'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object')
```

MANN WHTNEY TEST

```
import pandas as pd
```

```
df=pd.read_csv('general_data.csv')
```

```
dummy=pd.get_dummies(df['Attrition'])
```

```
df2=pd.concat((df,dummy),axis=1)
```

```
df2=df2.drop(['Attrition'],axis=1)
df2=df2.drop(['No'],axis=1)
df2=df2.rename(columns={"Yes":"Attrition"})
df2.head()
```

```
Out[9]:
```

	Age	BusinessTravel	...	YearsWithCurrManager	Attrition
0	51	Travel_Rarely	...	0	0
1	31	Travel_Frequently	...	4	1
2	32	Travel_Frequently	...	3	0
3	38	Non-Travel	...	5	0
4	32	Travel_Rarely	...	4	0

ATTRITION VS DISTANCE FROM HOME.

H_0 = There is no significant difference between attrition yes and no for distance from home

H_A = There is significant difference between attrition yes and no for distance from home

```
from scipy.stats import mannwhitneyu
stats,p=mannwhitneyu(df2.Attrition,df2.DistanceFromHome)
print(stats,p)
```

```
In [12]: stats,p=mannwhitneyu(df2.Attrition,df2.DistanceFromHome)
In [13]: print(stats,p)
221832.0 0.0
```

As the P value of 0.0 is < 0.05 , the H_0 is rejected and H_A is accepted.

So there is difference in attrition and distance from home.

ATTRITION VS TOTAL WORKING YEARS

H_0 = There is no significant difference between attrition yes and no for total working years.

H_A = There is significant difference between attrition yes and no for total working years.

```
stats,p=mannwhitneyu(df2.Attrition,df2.TotalWorkingYears)
print(stats,p)
```

```
In [18]: stats,p=mannwhitneyu(df2.Attrition,df2.TotalWorkingYears)
In [19]: print(stats,p)
170527.5 0.0
```

As the P value of 0.0 is < 0.05 , the H_0 is rejected and H_A is accepted.

So there is difference in attrition and total working years.

ATTRITION VS YEARS AT COMPANY

H_0 = There is no significant difference between attrition yes and no for years at company.

H_A = There is significant difference between attrition yes and no for years at company.

```
stats,p=mannwhitneyu(df2.Attrition,df2.YearsAtCompany)
print(stats,p)
```

```
In [20]: stats,p=mannwhitneyu(df2.Attrition,df2.YearsAtCompany)
In [21]: print(stats,p)
520357.5 0.0
```

As the P value of 0.0 is < 0.05 , the H_0 is rejected and H_A is accepted.

So there is difference in attrition and years at company.

ATTRITION VS YEARS WITH CURRENT MANAGER

H_0 = There is no significant difference between attrition yes and no for years with current manager.

H_A = There is significant difference between attrition yes and no for years with current manager.

```
stats,p=mannwhitneyu(df2.Attrition,df2.YearsWithCurrManager)
print(stats,p)
```

```
In [24]: stats,p=mannwhitneyu(df2.Attrition,df2.YearsWithCurrManager)
In [25]: print(stats,p)
2101288.5 0.0
```

As the P value of 0.0 is < 0.05 , the H_0 is rejected and H_A is accepted.

So there is difference in attrition and years with current manager.