1. File .CSV is loaded and first five row are get displayed using ".head()" function .

```
In [2]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        dataset = pd.read_csv("general_data.csv")
```

```
In [3]: dataset.head()
```

Out[3]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	...	NumCompaniesWorke
0	51	No	Travel_Rarely	Sales	6	2	Life Sciences	1	1	Female	...	1.
1	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female	...	0.
2	32	No	Travel_Frequently	Research & Development	17	4	Other	1	3	Male	...	1.
3	38	No	Non-Travel	Research & Development	2	5	Life Sciences	1	4	Male	...	3.
4	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	5	Male	...	4.

5 rows × 24 columns

2. All table Columns are get show.

```
In [4]: dataset.columns
```

```
Out[4]: 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')
```

3. We have check is any null value is present in data.

```
In [5]: dataset.isnull()
```

Out[5]:

Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	...	NumCompaniesWorked	O
False	False	False	False	False	False	False	False	False	False	...	False	
False	False	False	False	False	False	False	False	False	False	...	False	
False	False	False	False	False	False	False	False	False	False	...	False	
False	False	False	False	False	False	False	False	False	False	...	False	
False	False	False	False	False	False	False	False	False	False	...	False	
...	...	...	...	...	...	...	...	...	...	...	...	
False	False	False	False	False	False	False	False	False	False	...	False	
False	False	False	False	False	False	False	False	False	False	...	False	
False	False	False	False	False	False	False	False	False	False	...	False	
False	False	False	False	False	False	False	False	False	False	...	False	
False	False	False	False	False	False	False	False	False	False	...	False	

ows × 24 columns

4. we have check is any duplicated data .

```
In [7]: dataset.duplicated()
```

```
Out[7]: 0       False
        1       False
        2       False
        3       False
        4       False
                ...
        4405    False
        4406    False
        4407    False
        4408    False
        4409    False
        Length: 4410, dtype: bool
```

5. If any duplicated data is present then using function ‘drop_duplicates()’ is get removed.

```
In [8]: dataset.drop_duplicates()
```

Out[8]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	...	NumCompaniesWo
0	51	No	Travel_Rarely	Sales	6	2	Life Sciences	1	1	Female	...	
1	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female	...	
2	32	No	Travel_Frequently	Research & Development	17	4	Other	1	3	Male	...	
3	38	No	Non-Travel	Research & Development	2	5	Life Sciences	1	4	Male	...	
4	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	5	Male	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4405	42	No	Travel_Rarely	Research & Development	5	4	Medical	1	4406	Female	...	
4406	29	No	Travel_Rarely	Research & Development	2	4	Medical	1	4407	Male	...	
4407	25	No	Travel_Rarely	Research & Development	25	2	Life Sciences	1	4408	Male	...	
4408	42	No	Travel_Rarely	Sales	18	2	Medical	1	4409	Male	...	
4409	40	No	Travel_Rarely	Research & Development	28	3	Medical	1	4410	Male	...	

4410 rows × 24 columns

6. By using ‘.describe()’ we are getting the info about mean,std,25%...ect.

```
In [11]: dataset1=dataset[['Age', 'Attrition','DistanceFromHome','Education','MonthlyIncome','NumCompaniesWorked','PercentSalaryH
             'YearsAtCompany','TrainingTimesLastYear','YearsSinceLastPromotion', 'YearsWithCurrManager']].describe()
         dataset1
```

Out[11]:

	Age	DistanceFromHome	Education	MonthlyIncome	NumCompaniesWorked	PercentSalaryHike	TotalWorkingYears	TrainingTimesLastYear	Yea
count	4410.000000	4410.000000	4410.000000	4410.000000	4391.000000	4410.000000	4401.000000	4410.000000	
mean	36.923810	9.192517	2.912925	65029.312925	2.694830	15.209524	11.279936	2.799320	
std	9.133301	8.105026	1.023933	47068.888559	2.498887	3.659108	7.782222	1.288978	
min	18.000000	1.000000	1.000000	10090.000000	0.000000	11.000000	0.000000	0.000000	
25%	30.000000	2.000000	2.000000	29110.000000	1.000000	12.000000	6.000000	2.000000	
50%	36.000000	7.000000	3.000000	49190.000000	2.000000	14.000000	10.000000	3.000000	
75%	43.000000	14.000000	4.000000	83800.000000	4.000000	18.000000	15.000000	3.000000	
max	60.000000	29.000000	5.000000	199990.000000	9.000000	25.000000	40.000000	6.000000	

7. '.mean()' function we get all columns mean.

```
In [12]: dataset1=dataset[['Age', 'Attrition','DistanceFromHome','Education','MonthlyIncome','NumCompaniesWorked','PercentSalaryH
             'YearsAtCompany','TrainingTimesLastYear','YearsSinceLastPromotion', 'YearsWithCurrManager']].mean()
         dataset1
```

```
Out[12]: Age                          36.923810
         DistanceFromHome              9.192517
         Education                     2.912925
         MonthlyIncome             65029.312925
         NumCompaniesWorked            2.694830
         PercentSalaryHike            15.209524
         TotalWorkingYears            11.279936
         TrainingTimesLastYear         2.799320
         YearsAtCompany                7.008163
         TrainingTimesLastYear         2.799320
         YearsSinceLastPromotion       2.187755
         YearsWithCurrManager          4.123129
         dtype: float64
```

8. '.mode()' function is used for getting mode values of all columns.

```
In [13]: dataset1=dataset[['Age', 'Attrition','DistanceFromHome','Education','MonthlyIncome','NumCompaniesWorked','PercentSalaryH
             'YearsAtCompany','TrainingTimesLastYear','YearsSinceLastPromotion', 'YearsWithCurrManager']].mode()
         dataset1
```

Out[13]:

	Age	Attrition	DistanceFromHome	Education	MonthlyIncome	NumCompaniesWorked	PercentSalaryHike	TotalWorkingYears	TrainingTimesLastYear	YearsAt
0	35	No	2	3	23420	1.0	11	10.0	2	

9. By using '.median()' function we get all median values of each columns.

```
In [14]: dataset1=dataset[['Age', 'Attrition','DistanceFromHome','Education','MonthlyIncome','NumCompaniesWorked','PercentSalaryH
             'YearsAtCompany','TrainingTimesLastYear','YearsSinceLastPromotion', 'YearsWithCurrManager']].median()
         dataset1
```

```
Out[14]: Age                          36.0
         DistanceFromHome              7.0
         Education                     3.0
         MonthlyIncome             49190.0
         NumCompaniesWorked            2.0
         PercentSalaryHike            14.0
         TotalWorkingYears            10.0
         TrainingTimesLastYear         3.0
         YearsAtCompany                5.0
         TrainingTimesLastYear         3.0
         YearsSinceLastPromotion       1.0
         YearsWithCurrManager          3.0
         dtype: float64
```

10. By using '.var() ' we are getting variance value .

```
In [15]: dataset1=dataset[['Age', 'Attrition','DistanceFromHome','Education','MonthlyIncome','NumCompaniesWorked','PercentSalaryH
             'YearsAtCompany','TrainingTimesLastYear','YearsSinceLastPromotion', 'YearsWithCurrManager']].var()
         dataset1
```

```
Out[15]: Age                      8.341719e+01
         DistanceFromHome         6.569144e+01
         Education                1.048438e+00
         MonthlyIncome            2.215480e+09
         NumCompaniesWorked       6.244436e+00
         PercentSalaryHike        1.338907e+01
         TotalWorkingYears        6.056298e+01
         TrainingTimesLastYear    1.661465e+00
         YearsAtCompany           3.751728e+01
         TrainingTimesLastYear    1.661465e+00
         YearsSinceLastPromotion  1.037935e+01
         YearsWithCurrManager     1.272582e+01
         dtype: float64
```

11. In '.skew()' function is used for asymmetry of the probability distribution of a real -valued random variable about its mean.

```
In [16]: dataset1=dataset[['Age', 'Attrition','DistanceFromHome','Education','MonthlyIncome','NumCompaniesWorked','PercentSalaryH
             'YearsAtCompany','TrainingTimesLastYear','YearsSinceLastPromotion', 'YearsWithCurrManager']].skew()
         dataset1

Out[16]: Age                        0.413005
         DistanceFromHome           0.957466
         Education                 -0.289484
         MonthlyIncome              1.368884
         NumCompaniesWorked         1.026767
         PercentSalaryHike          0.820569
         TotalWorkingYears          1.116832
         TrainingTimesLastYear      0.552748
         YearsAtCompany             1.763328
         TrainingTimesLastYear      0.552748
         YearsSinceLastPromotion    1.982939
         YearsWithCurrManager       0.832884
         dtype: float64
```

12. In '.kurt()' function give the info about measure of the "tailedness" of the probability distribution of a real -valued random variable.

```
In [17]: dataset1=dataset[['Age', 'Attrition','DistanceFromHome','Education','MonthlyIncome','NumCompaniesWorked','PercentSalaryH
             'YearsAtCompany','TrainingTimesLastYear','YearsSinceLastPromotion', 'YearsWithCurrManager']].kurt()
         dataset1

Out[17]: Age                       -0.405951
         DistanceFromHome          -0.227045
         Education                 -0.560569
         MonthlyIncome              1.000232
         NumCompaniesWorked         0.007287
         PercentSalaryHike         -0.302638
         TotalWorkingYears          0.912936
         TrainingTimesLastYear      0.491149
         YearsAtCompany             3.923864
         TrainingTimesLastYear      0.491149
         YearsSinceLastPromotion    3.601761
         YearsWithCurrManager       0.167949
         dtype: float64
```

13. Using '.std()' function we can get the amount of variation or dispersion of a set value.

```
In [18]: dataset1=dataset[['Age', 'Attrition','DistanceFromHome','Education','MonthlyIncome','NumCompaniesWorked','PercentSalaryH
             'YearsAtCompany','TrainingTimesLastYear','YearsSinceLastPromotion', 'YearsWithCurrManager']].std()
         dataset1

Out[18]: Age                        9.133301
         DistanceFromHome           8.105026
         Education                  1.023933
         MonthlyIncome          47068.888559
         NumCompaniesWorked         2.498887
         PercentSalaryHike          3.659108
         TotalWorkingYears          7.782222
         TrainingTimesLastYear      1.288978
         YearsAtCompany             6.125135
         TrainingTimesLastYear      1.288978
         YearsSinceLastPromotion    3.221699
         YearsWithCurrManager       3.567327
         dtype: float64
```

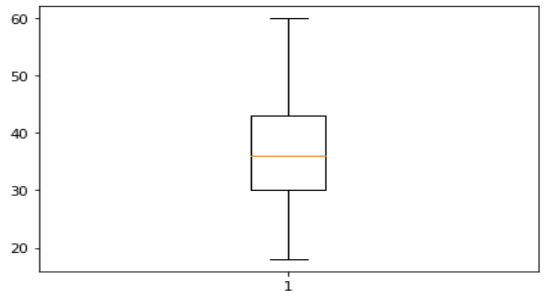
14. Inference of the analyst :

- All the above variable are showing positive skew except Education is showing negatively skew .
- In Kurtosis Age , Education , DistanceFromHome and percentSalaryHike are **Platykurtic.** And TotalWorkingYears, TrainingTimeLastYear, NumCompaniesWorked , YearWithCurrent Manager are **Mesokurtic .** YearsAtCompany, YearsSinceLastPromotion and MonthlyIncome are **Leptokurtic.**
- standard deviation of MonthlyIncome is more as compared to other Columns

15. Outliers :-

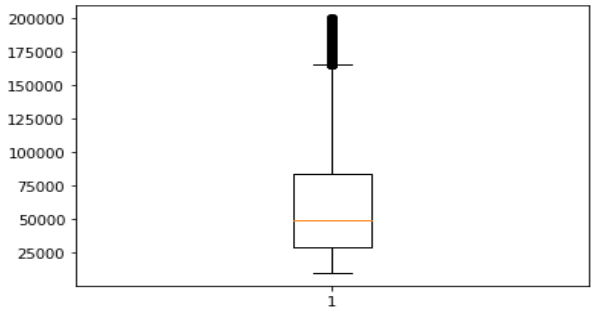
1. Age is normally skewed without any outliers.

```
In [7]: plt.boxplot(dataset.Age)
Out[7]: {'whiskers': [<matplotlib.lines.Line2D at 0x1913940c948>,
          <matplotlib.lines.Line2D at 0x1913942af88>],
         'caps': [<matplotlib.lines.Line2D at 0x19139437b08>,
          <matplotlib.lines.Line2D at 0x19139437fc8>],
         'boxes': [<matplotlib.lines.Line2D at 0x19139200bc8>],
         'medians': [<matplotlib.lines.Line2D at 0x19139437cc8>],
         'fliers': [<matplotlib.lines.Line2D at 0x1913942a648>],
         'means': []}
```



2. MonthlyIncome is negatively skewed with many outliers.

```
In [8]: plt.boxplot(dataset.MonthlyIncome)
Out[8]: {'whiskers': [<matplotlib.lines.Line2D at 0x191394d07c8>,
          <matplotlib.lines.Line2D at 0x191394d0d48>],
         'caps': [<matplotlib.lines.Line2D at 0x191394d0e48>,
          <matplotlib.lines.Line2D at 0x191394d0ec8>],
         'boxes': [<matplotlib.lines.Line2D at 0x191394c8f88>],
         'medians': [<matplotlib.lines.Line2D at 0x191394d6dc8>],
         'fliers': [<matplotlib.lines.Line2D at 0x191394d6ec8>],
         'means': []}
```



3. PercentSalaryHike is positively skewed with no any outliers.

```
In [9]: plt.boxplot(dataset.PercentSalaryHike)
```

```
Out[9]: {'whiskers': [<matplotlib.lines.Line2D at 0x1913953fcc8>,
          <matplotlib.lines.Line2D at 0x1913953fdc8>],
         'caps': [<matplotlib.lines.Line2D at 0x1913953fe48>,
          <matplotlib.lines.Line2D at 0x19139544d48>],
         'boxes': [<matplotlib.lines.Line2D at 0x1913953f548>],
         'medians': [<matplotlib.lines.Line2D at 0x19139544e48>],
         'fliers': [<matplotlib.lines.Line2D at 0x19139544ec8>],
         'means': []}
```