Loading the necessary libraries, reading the data set and viewinng the data

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
df=pd.read_csv('D:/DS_Files/LetsUpgrade-AI-ML/Day-7/Assignment/general_data.csv')
```

In [4]:
```python
df.head()
```

Out[4]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	...	NumCompaniesWorke
0	51	No	Travel_Rarely	Sales	6	2	Life Sciences	1	1	Female	...	1.0
1	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female	...	0.0
2	32	No	Travel_Frequently	Research & Development	17	4	Other	1	3	Male	...	1.0
3	38	No	Non-Travel	Research & Development	2	5	Life Sciences	1	4	Male	...	3.0
4	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	5	Male	...	4.0

5 rows × 24 columns

Exploring the data for different statistical parameters

In [5]:
```python
df.describe()
```

Out[5]:

	Age	DistanceFromHome	Education	EmployeeCount	EmployeeID	JobLevel	MonthlyIncome	NumCompaniesWorked	PercentSalaryHike	Sta
count	4410.000000	4410.000000	4410.000000	4410.0	4410.000000	4410.000000	4410.000000	4391.000000	4410.000000	
mean	36.923810	9.192517	2.912925	1.0	2205.500000	2.063946	65029.312925	2.694830	15.209524	
std	9.133301	8.105026	1.023933	0.0	1273.201673	1.106689	47068.888559	2.498887	3.659108	
min	18.000000	1.000000	1.000000	1.0	1.000000	1.000000	10090.000000	0.000000	11.000000	
25%	30.000000	2.000000	2.000000	1.0	1103.250000	1.000000	29110.000000	1.000000	12.000000	
50%	36.000000	7.000000	3.000000	1.0	2205.500000	2.000000	49190.000000	2.000000	14.000000	
75%	43.000000	14.000000	4.000000	1.0	3307.750000	3.000000	83800.000000	4.000000	18.000000	
max	60.000000	29.000000	5.000000	1.0	4410.000000	5.000000	199990.000000	9.000000	25.000000	

In [6]:
```python
df.columns #listing the colums
```

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')
```

In [7]:
```python
df.info()#listing the datatypes avaialble
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4410 entries, 0 to 4409
Data columns (total 24 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Age                      4410 non-null   int64
 1   Attrition                4410 non-null   object
 2   BusinessTravel           4410 non-null   object
 3   Department               4410 non-null   object
 4   DistanceFromHome         4410 non-null   int64
 5   Education                4410 non-null   int64
 6   EducationField           4410 non-null   object
 7   EmployeeCount            4410 non-null   int64
 8   EmployeeID               4410 non-null   int64
 9   Gender                   4410 non-null   object
 10  JobLevel                 4410 non-null   int64
 11  JobRole                  4410 non-null   object
 12  MaritalStatus            4410 non-null   object
 13  MonthlyIncome            4410 non-null   int64
 14  NumCompaniesWorked       4391 non-null   float64
```

Extracting the data with attrition level "yes" for analysis

In [19]:
```python
df_att=df[df['Attrition']=='Yes']
df_att
```

Out[19]:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	...	NumCompaniesWo
1	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female	...	
6	28	Yes	Travel_Rarely	Research & Development	11	2	Medical	1	7	Male	...	
13	47	Yes	Non-Travel	Research & Development	1	1	Medical	1	14	Male	...	
28	44	Yes	Travel_Frequently	Research & Development	1	2	Medical	1	29	Male	...	
30	26	Yes	Travel_Rarely	Research & Development	4	3	Medical	1	31	Male	...	
...	...	...	...	...	...	...	...	...	...	...	...	
4381	29	Yes	Travel_Rarely	Research & Development	7	1	Life Sciences	1	4382	Female	...	
4386	33	Yes	Travel_Rarely	Sales	11	4	Marketing	1	4387	Male	...	
4388	33	Yes	Travel_Rarely	Sales	1	3	Life Sciences	1	4389	Male	...	
4391	32	Yes	Travel_Rarely	Sales	23	1	Life Sciences	1	4392	Male	...	
4402	37	Yes	Travel_Frequently	Sales	2	3	Marketing	1	4403	Male	...	

711 rows × 24 columns

Checking null values and identifying the data types

In [22]:
```python
df_att.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 711 entries, 1 to 4402
Data columns (total 24 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      711 non-null    int64  
 1   Attrition                711 non-null    object 
 2   BusinessTravel           711 non-null    object 
 3   Department               711 non-null    object 
 4   DistanceFromHome         711 non-null    int64  
 5   Education                711 non-null    int64  
 6   EducationField           711 non-null    object 
 7   EmployeeCount            711 non-null    int64  
 8   EmployeeID               711 non-null    int64  
 9   Gender                   711 non-null    object 
 10  JobLevel                 711 non-null    int64  
 11  JobRole                  711 non-null    object 
 12  MaritalStatus            711 non-null    object 
 13  MonthlyIncome            711 non-null    int64  
 14  NumCompaniesWorked       707 non-null    float64
 15  Over18                   711 non-null    object 
 16  PercentSalaryHike        711 non-null    int64  
 17  StandardHours            711 non-null    int64  
 18  StockOptionLevel         711 non-null    int64  
 19  TotalWorkingYears        709 non-null    float64
 20  TrainingTimesLastYear    711 non-null    int64  
 21  YearsAtCompany           711 non-null    int64  
 22  YearsSinceLastPromotion  711 non-null    int64  
 23  YearsWithCurrManager     711 non-null    int64  
dtypes: float64(2), int64(14), object(8)
memory usage: 138.9+ KB
```

Here we have null values in 'NumCompaniesWorked' & 'TotalWorkingYears', which are negligible, hence will leave as it is

Analysis of attrition data

We will analyse the attrition percentage against each parameters

In [24]:
```python
df_att['Department'].value_counts()*100/df['Department'].value_counts()
```

Out[24]:
```
Research & Development    15.712799
Sales                     15.022422
Human Resources           30.158730
Name: Department, dtype: float64
```

From the above the Human Resourse department have higher attrition rate, ie is about 30%

In [27]:
```
df_att['BusinessTravel'].value_counts()*100/df['BusinessTravel'].value_counts()
```

Out[27]:
```
Travel_Rarely        14.956855
Travel_Frequently    24.909747
Non-Travel            8.000000
Name: BusinessTravel, dtype: float64
```

Employess who travels freequently also have a higher attrtion rate of 25%

In [38]:
```
df_att['Education'].value_counts()*100/df['Education'].value_counts()
```

Out[38]:
```
3    15.559441
4    15.577889
2    18.794326
1    15.294118
5    14.583333
Name: Education, dtype: float64
```

Attrition rate is almost similar in all education levels bu category 2(College) tops there with 19%

In [39]:
```
df_att['EducationField'].value_counts()*100/df['EducationField'].value_counts()
```

Out[39]:
```
Human Resources     40.740741
Life Sciences       16.666667
Marketing           15.723270
Medical             16.163793
Other               12.195122
Technical Degree    11.363636
Name: EducationField, dtype: float64
```

Here the people from Human Resourse Education field is more prone to attrition, ie 41%

In [40]:
```
df_att['Gender'].value_counts()*100/df['Gender'].value_counts()
```

Out[40]:
```
Male      16.666667
Female    15.306122
Name: Gender, dtype: float64
```

Gender has almost equal sharing in attrition but male dominate slightly

In [41]:
```
df_att['JobLevel'].value_counts()*100/df['JobLevel'].value_counts()
```

Out[41]:
```
1    15.469613
2    17.790262
3    14.678899
4    16.037736
5    13.043478
Name: JobLevel, dtype: float64
```

Attrition level is higher in JL2

In [34]:
```
df_att['JobRole'].value_counts()*100/df['JobRole'].value_counts()
```

Out[34]:
```
Healthcare Representative    14.503817
Human Resources              13.461538
Laboratory Technician        16.216216
Manager                      13.725490
Manufacturing Director       11.034483
Research Director            23.750000
Research Scientist           18.150685
Sales Executive              16.871166
Sales Representative         14.457831
Name: JobRole, dtype: float64
```

Here the post of research Direcor is the volatile position and attrition is about 23% followed by Research Scientist 18%

In [43]:
```
df_att['MaritalStatus'].value_counts()*100/df['MaritalStatus'].value_counts()
```

Out[43]:
```
Divorced    10.091743
Married     12.481426
Single      25.531915
Name: MaritalStatus, dtype: float64
```

Single personal are more in attrition ie 24%

Analysing the quantitiative parameters for more insight

Analysis of attrition only data

In [41]:
```
df_att[['Age', 'DistanceFromHome', 'MonthlyIncome', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear',
        'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].describe()
```

Out[41]:

	Age	DistanceFromHome	MonthlyIncome	PercentSalaryHike	TotalWorkingYears	TrainingTimesLastYear	YearsAtCompany	YearsSinceLastPromotion
count	711.000000	711.000000	711.000000	711.000000	709.000000	711.000000	711.000000	711.000000
mean	33.607595	9.012658	61682.616034	15.481013	8.255289	2.654008	5.130802	1.945148
std	9.675693	7.772368	44792.067695	3.775289	7.164018	1.154834	5.941598	3.148633
min	18.000000	1.000000	10090.000000	11.000000	0.000000	0.000000	0.000000	0.000000
25%	28.000000	2.000000	28440.000000	12.000000	3.000000	2.000000	1.000000	0.000000
50%	32.000000	7.000000	49080.000000	14.000000	7.000000	3.000000	3.000000	1.000000
75%	39.000000	15.000000	71040.000000	18.000000	10.000000	3.000000	7.000000	2.000000
max	58.000000	29.000000	198590.000000	25.000000	40.000000	6.000000	40.000000	15.000000

Analysis of whole data

In [43]:
```
df[['Age', 'DistanceFromHome', 'MonthlyIncome', 'PercentSalaryHike','TotalWorkingYears', 'TrainingTimesLastYear',
        'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].describe()
```

Out[43]:

	Age	DistanceFromHome	MonthlyIncome	PercentSalaryHike	TotalWorkingYears	TrainingTimesLastYear	YearsAtCompany	YearsSinceLastPromotion
count	4410.000000	4410.000000	4410.000000	4410.000000	4401.000000	4410.000000	4410.000000	4410.000000
mean	36.923810	9.192517	65029.312925	15.209524	11.279936	2.799320	7.008163	2.187755
std	9.133301	8.105026	47068.888559	3.659108	7.782222	1.288978	6.125135	3.221699
min	18.000000	1.000000	10090.000000	11.000000	0.000000	0.000000	0.000000	0.000000
25%	30.000000	2.000000	29110.000000	12.000000	6.000000	2.000000	3.000000	0.000000
50%	36.000000	7.000000	49190.000000	14.000000	10.000000	3.000000	5.000000	1.000000
75%	43.000000	14.000000	83800.000000	18.000000	15.000000	3.000000	9.000000	3.000000
max	60.000000	29.000000	199990.000000	25.000000	40.000000	6.000000	40.000000	15.000000

From the above, most of the parameters are rightskewed even for the attrition part also.

Outliers are there in Monthly Income,Total working years,Years at Company, Years Since Last Promotion and Years with current manager.