```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
#loading the required libraries
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

About Dataset

The dataset consists of several predictor variables and one target variable,Attrition.

Data Loading

```
df=pd.read_csv('/content/drive/MyDrive/Data Analytics Python Projects/HR_Analytics/HR-Employee-Attrition.csv')
```

```
df.head(10)
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education
0	41	Yes	Travel_Rarely	1102	Sales	1	2
1	49	No	Travel_Frequently	279	Research & Development	8	1
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2
3	33	No	Travel_Frequently	1392	Research & Development	3	4
4	27	No	Travel_Rarely	591	Research & Development	2	1
5	32	No	Travel_Frequently	1005	Research & Development	2	2
6	59	No	Travel_Rarely	1324	Research & Development	3	3
7	30	No	Travel_Rarely	1358	Research & Development	24	1
8	38	No	Travel_Frequently	216	Research & Development	23	3
9	36	No	Travel_Rarely	1299	Research & Development	27	3

10 rows × 35 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
```

```
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

Statistical Measure of HR Analytics Data

```
print(df.describe())
```

```
       std       9.135373   403.509100          8.106864     1.024165            0.0
       min      18.000000   102.000000          1.000000     1.000000            1.0
       25%      30.000000   465.000000          2.000000     2.000000            1.0
       50%      36.000000   802.000000          7.000000     3.000000            1.0
       75%      43.000000  1157.000000         14.000000     4.000000            1.0
       max      60.000000  1499.000000         29.000000     5.000000            1.0

              EmployeeNumber  EnvironmentSatisfaction   HourlyRate  JobInvolvement  \
       count     1470.000000              1470.000000  1470.000000     1470.000000
       mean      1024.865306                 2.721769    65.891156        2.729932
       std        602.024335                 1.093082    20.329428        0.711561
       min          1.000000                 1.000000    30.000000        1.000000
       25%        491.250000                 2.000000    48.000000        2.000000
       50%       1020.500000                 3.000000    66.000000        3.000000
       75%       1555.750000                 4.000000    83.750000        3.000000
       max       2068.000000                 4.000000   100.000000        4.000000

                 JobLevel  ...  RelationshipSatisfaction  StandardHours  \
       count  1470.000000  ...               1470.000000         1470.0
       mean      2.063946  ...                  2.712245           80.0
       std       1.106940  ...                  1.081209            0.0
       min       1.000000  ...                  1.000000           80.0
       25%       1.000000  ...                  2.000000           80.0
       50%       2.000000  ...                  3.000000           80.0
       75%       3.000000  ...                  4.000000           80.0
       max       5.000000  ...                  4.000000           80.0

              StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
       count       1470.000000        1470.000000            1470.000000
       mean           0.793878          11.279592               2.799320
       std            0.852077           7.780782               1.289271
       min            0.000000           0.000000               0.000000
       25%            0.000000           6.000000               2.000000
       50%            1.000000          10.000000               3.000000
       75%            1.000000          15.000000               3.000000
       max            3.000000          40.000000               6.000000

              WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
       count      1470.000000     1470.000000         1470.000000
       mean          2.761224        7.008163            4.229252
       std           0.706476        6.126525            3.623137
       min           1.000000        0.000000            0.000000
       25%           2.000000        3.000000            2.000000
       50%           3.000000        5.000000            3.000000
       75%           3.000000        9.000000            7.000000
       max           4.000000       40.000000           18.000000

              YearsSinceLastPromotion  YearsWithCurrManager
       count              1470.000000           1470.000000
       mean                 2.187755              4.123129
       std                  3.222430              3.568136
       min                  0.000000              0.000000
```

```
max                 15.000000            17.000000

[8 rows x 26 columns]
```

Data Analysis

Checking is there any null value or not

```
df.isnull().sum()
```

```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

Insights: 1.No null value is depicted

Checking Duplicate

```
df.duplicated().sum()
```

```
0
```

Insights: No duplicate

```
df['StandardHours'].value_counts()
# there is only 1 class in StandardHours
```

```
80    1470
Name: StandardHours, dtype: int64
```

Drop Over 18 since in Age column, all employees are older than 18 so it's meaningless Employercount, StandardHours,EmployeeNumber also offer no meaning

```
df = df.drop(['EmployeeCount','Over18','StandardHours','EmployeeNumber'], axis =1)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 31 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EnvironmentSatisfaction   1470 non-null   int64
 9   Gender                    1470 non-null   object
 10  HourlyRate                1470 non-null   int64
 11  JobInvolvement            1470 non-null   int64
 12  JobLevel                  1470 non-null   int64
 13  JobRole                   1470 non-null   object
 14  JobSatisfaction           1470 non-null   int64
 15  MaritalStatus             1470 non-null   object
 16  MonthlyIncome             1470 non-null   int64
 17  MonthlyRate               1470 non-null   int64
 18  NumCompaniesWorked        1470 non-null   int64
 19  OverTime                  1470 non-null   object
 20  PercentSalaryHike         1470 non-null   int64
 21  PerformanceRating         1470 non-null   int64
 22  RelationshipSatisfaction  1470 non-null   int64
 23  StockOptionLevel          1470 non-null   int64
 24  TotalWorkingYears         1470 non-null   int64
 25  TrainingTimesLastYear     1470 non-null   int64
 26  WorkLifeBalance           1470 non-null   int64
 27  YearsAtCompany            1470 non-null   int64
 28  YearsInCurrentRole        1470 non-null   int64
 29  YearsSinceLastPromotion   1470 non-null   int64
 30  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(23), object(8)
memory usage: 356.1+ KB
```

```
df['Age'].value_counts().sort_index(ascending = True)
```

```
18     8
19     9
20    11
21    13
22    16
23    14
24    26
25    26
26    39
27    48
28    48
29    68
30    60
31    69
32    61
33    58
34    77
35    78
36    69
37    50
38    58
39    42
40    57
41    40
42    46
43    32
44    33
45    41
46    33
47    24
48    19
49    24
50    30
51    19
52    18
53    19
54    18
55    22
56    14
57     4
58    14
59    10
```

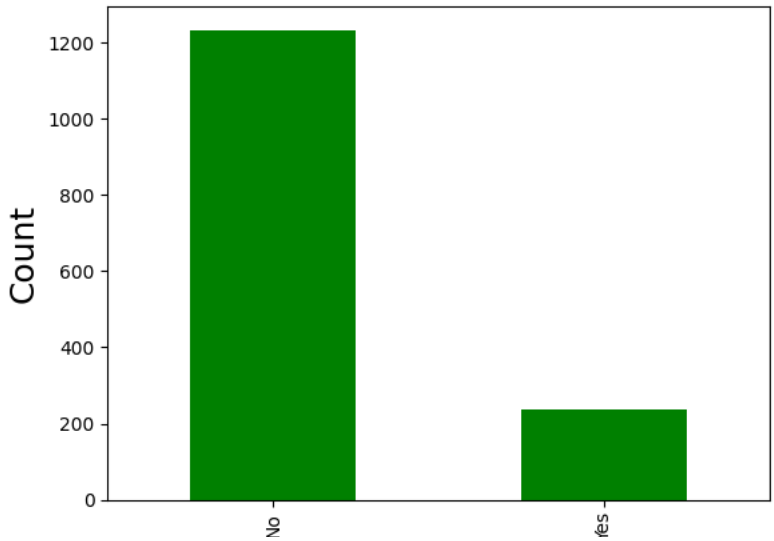
```
60     5
Name: Age, dtype: int64
```

Insights: All employees are over 18

Total number of Attrition and non-attrition

```
df['Attrition'].value_counts().plot(kind = 'bar', color = 'green')
print(df.Attrition.value_counts())
plt.xlabel('Attrition',fontsize=18)
plt.ylabel('Count',fontsize=18)
```

```
No     1233
Yes     237
Name: Attrition, dtype: int64
Text(0, 0.5, 'Count')
```



Checking all categorical Data

```
for col in df.describe(include= 'object').columns:
  print(col)
  print(df[col].unique())
  print('-'*50)
```

```
Attrition
['Yes' 'No']
--------------------------------------------------
BusinessTravel
['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']
--------------------------------------------------
Department
['Sales' 'Research & Development' 'Human Resources']
--------------------------------------------------
EducationField
['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
 'Human Resources']
--------------------------------------------------
Gender
['Female' 'Male']
--------------------------------------------------
JobRole
['Sales Executive' 'Research Scientist' 'Laboratory Technician'
 'Manufacturing Director' 'Healthcare Representative' 'Manager'
 'Sales Representative' 'Research Director' 'Human Resources']
--------------------------------------------------
MaritalStatus
['Single' 'Married' 'Divorced']
--------------------------------------------------
OverTime
['Yes' 'No']
--------------------------------------------------
```

Insights:

1. No-> Non-Attrition(Total 1233)
2. Yes-> Attritions(Total 237)

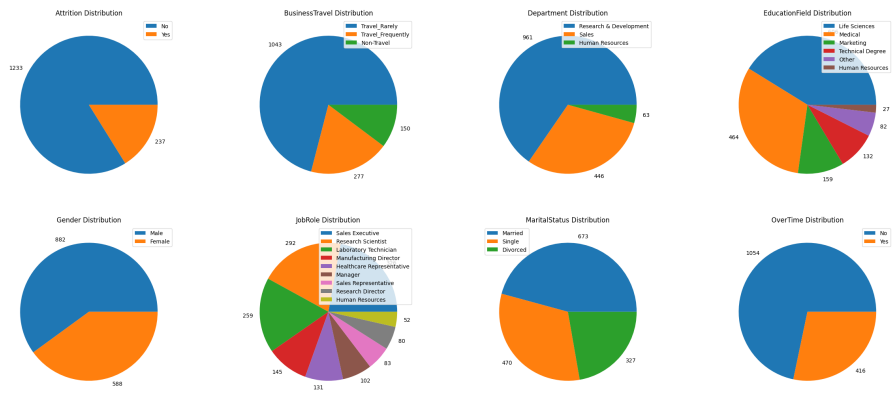
Select all categorical columns to graph piechart

```
cat_cols = df.select_dtypes(['object']).columns

cat_cols
```

```
Index(['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender',
       'JobRole', 'MaritalStatus', 'OverTime'],
      dtype='object')
```

```
plotnumber=1
plt.figure(figsize=(30,26),facecolor='white')
for col in cat_cols:
    if(plotnumber<=9):
        plt.subplot(4,4,plotnumber)
        plt.pie(df[col].value_counts(), labels=df[col].value_counts().values)
        plt.title(col+" Distribution")
        plt.legend(df[col].value_counts().index)
        plotnumber+=1
```

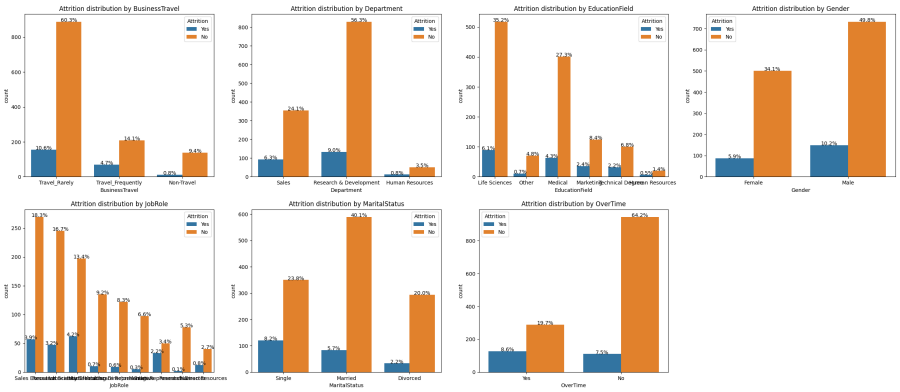



```
# select columns which are categorical except Attrition to graph against Attrition
cat_cols_2 = df.drop('Attrition',axis=1).select_dtypes(['object']).columns
cat_cols_2
```

```
Index(['BusinessTravel', 'Department', 'EducationField', 'Gender', 'JobRole',
       'MaritalStatus', 'OverTime'],
      dtype='object')
```

```
plotnumber=1
plt.figure(figsize=(30,26),facecolor='white')
for col in cat_cols_2:
    if(plotnumber<=9):
```

```
        plt.subplot(4,4,plotnumber)
        sns.countplot(x=col, hue='Attrition', data=df)
        plt.title("Attrition distribution by " + col)
        ax = plt.gca()
        total_height = len(df['Attrition'])
        for p in ax.patches:
          percentage = '{:.1f}%'.format(100 * p.get_height() / total_height)
          x = p.get_x() + p.get_width() / 2
          y = p.get_height()
          ax.annotate(percentage, (x, y), ha='center')
        plotnumber+=1
```



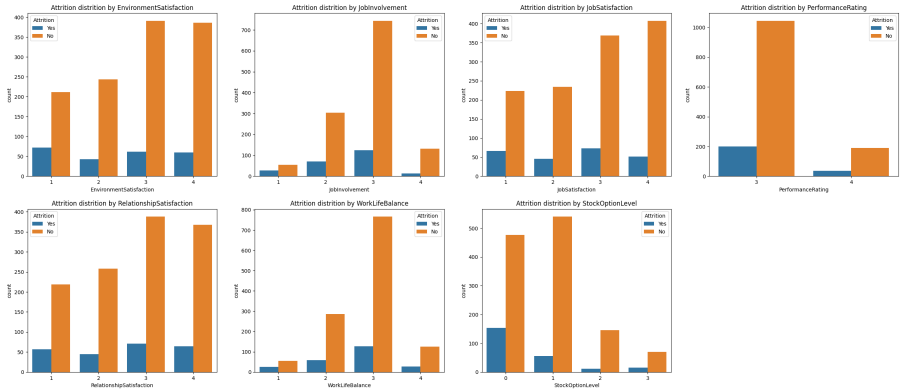
Insights:

1. People who do not travel has the least attrition rate of all.
2. Attrition among men are more than women
3. Attrition rate among sale representative is high among all the job roles.
4. Attrition rate among Married is the least when compared to all the other marital status.
5. Attrition rate is high among people who work overtime

Select variables which are rating from surveying

```
rating_cols = ['EnvironmentSatisfaction','JobInvolvement', 'JobSatisfaction',
               'PerformanceRating', 'RelationshipSatisfaction', 'WorkLifeBalance','StockOptionLevel']
```

```
plotnumber=1
plt.figure(figsize=(30,26),facecolor='white')
for col in rating_cols:
    if(plotnumber<=9):
        plt.subplot(4,4,plotnumber)
        sns.countplot(x=col, hue='Attrition', data=df)
        plt.title("Attrition distrition by " + col)
        plotnumber+=1
```

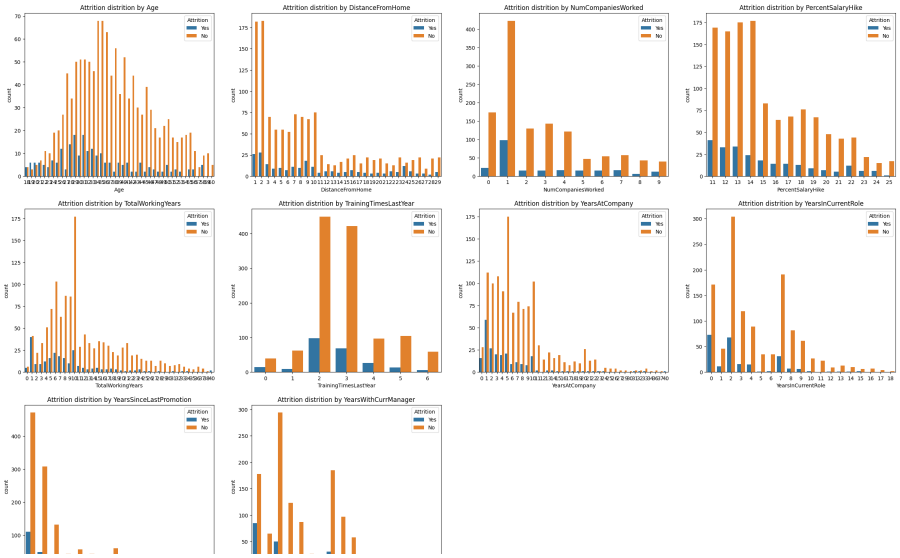


Insights:

1. At jobinvolvement = 1, both attrition rates are very low.
2. At Performancerating = 3, both attrition rates are at their highest.
3. At Relationshipsatifaction and WorkLifeBalance = 3, both attrition rates are at their highest.

```
num_cols = ['Age', 'DistanceFromHome','NumCompaniesWorked', 'PercentSalaryHike',
       'TotalWorkingYears', 'TrainingTimesLastYear',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager']
```

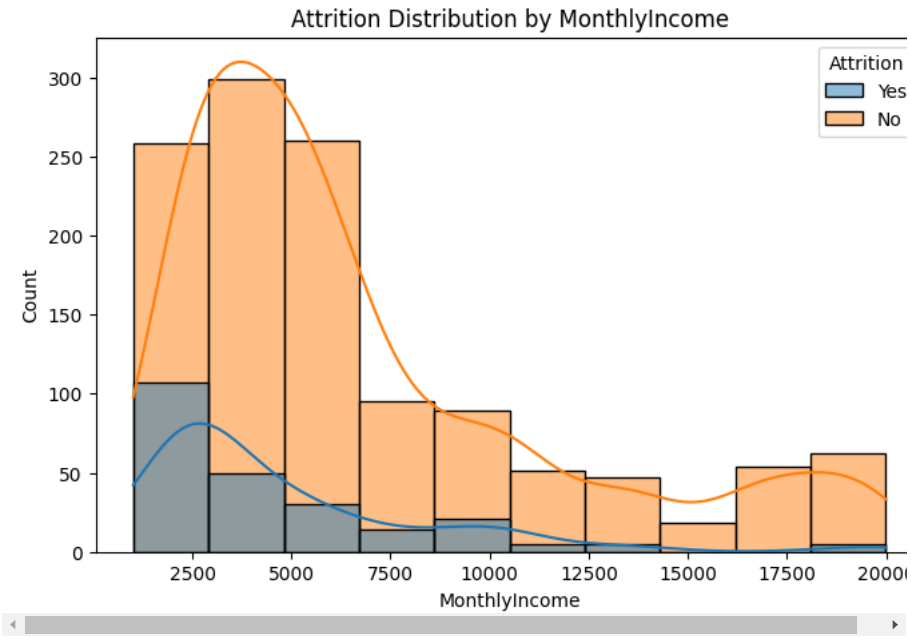
```
plotnumber=1
plt.figure(figsize=(30,26),facecolor='white')
for col in num_cols:
    if(plotnumber<=11):
        plt.subplot(4,4,plotnumber)
        sns.countplot(x=col, hue='Attrition', data=df)
        plt.title("Attrition distrition by " + col)
        plotnumber+=1
```



Insights:

1. Between the age groups 29-33 , Attritiion rate is high
2. Attrition rate is high among employees upto 13%

```
plt.figure(figsize=(8,5))
plt.title('Attrition Distribution by MonthlyIncome')
sns.histplot(data=df,x='MonthlyIncome',hue='Attrition',bins=10,kde=True)
plt.show()
```



Insights:

Attrition rate is high among employees with monthly income range upto 2700