```
### IBM HR Analytics Employee Attrition & Performance ###
```

```
# import data manipulation package
import pandas as pd
import numpy as np
# import data visualisation package
import matplotlib.pyplot as plt
import seaborn as sns
# importing the warnings library
import warnings
warnings.filterwarnings('ignore')
```

```
# set pandas options
pd.set_option('display.max_columns', 35)
```

```
# load dataset
df=pd.read_csv('ibmhr.csv')
df
```

Out[110]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel	JobRole
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	94	3	2	Sales Executive
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	Male	61	2	2	Research Scientist
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male	92	2	1	Laboratory Technician
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	Female	56	3	1	Research Scientist
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	Male	40	3	1	Laboratory Technician
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1465	36	No	Travel_Frequently	884	Research & Development	23	2	Medical	1	2061	3	Male	41	4	2	Laboratory Technician
1466	39	No	Travel_Rarely	613	Research & Development	6	1	Medical	1	2062	4	Male	42	2	3	Healthcare Representative
1467	27	No	Travel_Rarely	155	Research & Development	4	3	Life Sciences	1	2064	2	Male	87	4	2	Manufacturing Director
1468	49	No	Travel_Frequently	1023	Sales	2	3	Medical	1	2065	4	Male	63	2	2	Sales Executive
1469	34	No	Travel_Rarely	628	Research & Development	8	3	Medical	1	2068	2	Male	82	4	2	Laboratory Technician

1470 rows × 35 columns

```
df.head()
```

Out[111]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel	JobRole	JobSati
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	94	3	2	Sales Executive	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	3	Male	61	2	2	Research Scientist	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male	92	2	1	Laboratory Technician	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	4	Female	56	3	1	Research Scientist	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	1	Male	40	3	1	Laboratory Technician	

```
# check data shape
df.shape
```

Out[116]: (1470, 35)

```
# check number of duplicated data
print(f'Number of duplicated data: {df.duplicated().sum()}')
```

```
Number of duplicated data: 0
```

```
# check missing value
df.isnull().sum() / len(df) * 100
```

Out[120]:

```
Age                         0.0
Attrition                   0.0
BusinessTravel              0.0
DailyRate                   0.0
Department                  0.0
DistanceFromHome            0.0
Education                   0.0
EducationField              0.0
EmployeeCount               0.0
EmployeeNumber              0.0
EnvironmentSatisfaction     0.0
Gender                      0.0
HourlyRate                  0.0
JobInvolvement              0.0
JobLevel                    0.0
JobRole                     0.0
JobSatisfaction             0.0
MaritalStatus               0.0
MonthlyIncome               0.0
MonthlyRate                 0.0
NumCompaniesWorked          0.0
Over18                      0.0
OverTime                    0.0
PercentSalaryHike           0.0
PerformanceRating           0.0
RelationshipSatisfaction    0.0
StandardHours               0.0
StockOptionLevel            0.0
TotalWorkingYears           0.0
TrainingTimesLastYear       0.0
WorkLifeBalance             0.0
YearsAtCompany              0.0
YearsInCurrentRole          0.0
YearsSinceLastPromotion     0.0
YearsWithCurrManager        0.0
dtype: float64
```

```
# check data types
df.dtypes
```

Out[122]:

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

```
# check data decribe
df.describe()
```

Out[124]:

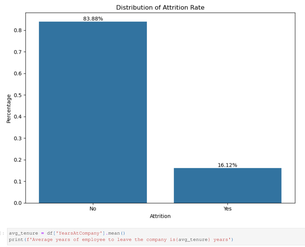
	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	JobSatisfaction	MonthlyIncome	MonthlyRate	NumCompanie
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	147
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	2.729932	2.063946	2.728571	6502.931293	14313.103401	2
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	0.711561	1.106940	1.102846	4707.956783	7117.786044	2
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	1.000000	1.000000	1.000000	1009.000000	2094.000000	0
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	2.000000	1.000000	2.000000	2911.000000	8047.000000	1
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	3.000000	2.000000	3.000000	4919.000000	14235.500000	2
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	3.000000	3.000000	4.000000	8379.000000	20461.500000	4
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	4.000000	5.000000	4.000000	19999.000000	26999.000000	9

```
df['Attrition'].value_counts(normalize=True)
```

Out[126]:

```
Attrition
No     0.838776
Yes    0.161224
Name: proportion, dtype: float64
```

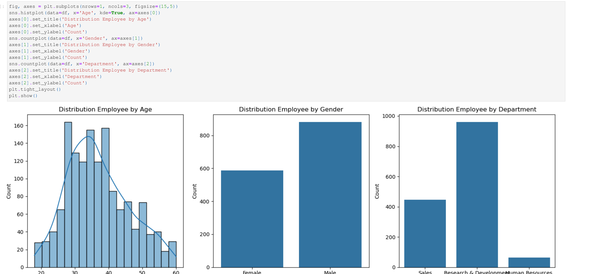
```
attrition = df['Attrition'].value_counts(normalize=True)
plt.figure(figsize=(6,6))
ax = sns.barplot(x=attrition.index, y=attrition)
for p in ax.patches:
    ax.annotate(f'{p.get_height() * 100:.2f}%',
                (p.get_x() + p.get_width() / 2.,
                 p.get_height()),
                ha='center', va='bottom')
plt.title('Distribution of Attrition Rate')
plt.xlabel('Attrition')
plt.ylabel('Percentage')
plt.tight_layout()
plt.show()
```




```
avg_tenure = df['YearsAtCompany'].mean()
print(f'Average years of employee to leave the company is{avg_tenure} years')
```

```
Average years of employee to leave the company is7.008163265306122 years
```

```
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(15,5))
sns.histplot(data=df, x='Age', kde=True, ax=axes[0])
axes[0].set_title('Distribution Employee by Age')
axes[0].set_xlabel('Age')
axes[0].set_ylabel('Count')
sns.countplot(data=df, x='Gender', ax=axes[1])
axes[1].set_title('Distribution Employee by Gender')
axes[1].set_xlabel('Gender')
axes[1].set_ylabel('Count')
sns.countplot(data=df, x='Department', ax=axes[2])
axes[2].set_title('Distribution Employee by Department')
axes[2].set_xlabel('Department')
axes[2].set_ylabel('Count')
plt.tight_layout()
plt.show()
```



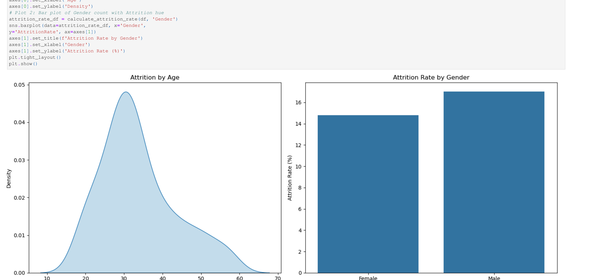

```
df_attrition = df[df['Attrition'] == 'Yes']
df_attrition.head()
```

Out[131]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	Gender	HourlyRate	JobInvolvement	JobLevel	JobRole	Job
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	2	Female	94	3	2	Sales Executive	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	4	Male	92	2	1	Laboratory Technician	
14	28	Yes	Travel_Rarely	103	Research & Development	24	3	Life Sciences	1	19	3	Male	50	2	1	Laboratory Technician	
21	36	Yes	Travel_Rarely	1218	Sales	9	4	Life Sciences	1	27	3	Male	82	2	1	Sales Representative	
24	34	Yes	Travel_Rarely	699	Research & Development	6	1	Medical	1	31	2	Male	83	3	1	Research Scientist	

```
# Pengat untuk Menghitung Attrition Rate
def calculate_attrition_rate(df, column):
    attrition_counts = df.groupby([column, 'Attrition']).size().unstack(fill_value=0)
    attrition_rate = attrition_counts['Yes'] /attrition_counts.sum(axis=1) * 100
    attrition_rate_df = attrition_rate.reset_index()
    attrition_rate_df.columns = [column, 'AttritionRate']
    return attrition_rate_df

fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15,8))
# Plot 1: KDE plot of Age with Attrition hue
sns.kdeplot(data=df,x='Age', hue='Attrition', ax=axes[0])
axes[0].set_title('Distribution by Age')
axes[0].set_xlabel('Age')
axes[0].set_ylabel('Density')
# Plot 2: Bar plot of Gender count with Attrition hue
attrition_rate_df = calculate_attrition_rate(df, 'Gender')
sns.barplot(data=attrition_rate_df, x='Gender',
y='AttritionRate', ax=axes[1])
axes[1].set_title(f'Attrition Rate by Gender')
axes[1].set_xlabel('Gender')
axes[1].set_ylabel('Attrition Rate (%)')
plt.tight_layout()
plt.show()
```




```
###  Objective of Project

# 1. Understand Current Turnover Rates: Gain a comprehensive understanding of
# the current employee turnover rate and analyze the demographic distribution of
# attrition by age, gender, education, department, and job role.
# 2. Identify Key Factors Influencing Turnover: Examine the main factors
# contributing to employee attrition, including job satisfaction (indicators (job
# involvement and work-life balance), salary factors (monthly income and salary
# hikes), and benefit factors (stock option levels), to uncover patterns and
# correlations that drive higher attrition rates.
```

```
#####      END Of Project      #####
```