HR DATA ANALYSIS

This project focuses on analysing HR data to identify key factors influencing employee retention, satisfaction, and overall workforce dynamics within the organization. Using a comprehensive dataset encompassing various attributes such as employee demographics, job roles, compensation, and satisfaction metrices, we performed extensive data analysis to uncover patterns and trends.

Utkarsh Sharma

sharma.utkarsh2402@gmail.com

Project Overview

The HR analytics project aimed to uncover key factors influencing employee retention, satisfaction, and overall workforce dynamics within the organization. By leveraging a comprehensive dataset that included various employee attributes, the project sought to identify patterns and trends that could inform strategic HR decisions.

1. **Dataset:** The dataset comprised columns such as 'EmpID', 'Age', 'AgeGroup', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department', 'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount', 'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate', 'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction', 'MaritalStatus', 'MonthlyIncome', 'SalarySlab', 'MonthlyRate', 'NumCompaniesWorked', 'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion', 'YearsWithCurrManager', 'BusinessTravel1', and 'Overtime'.

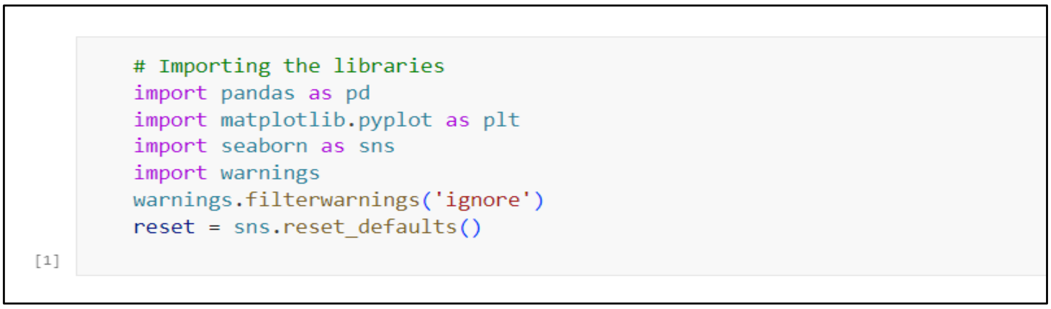
2. **Objectives:**
   - Identify key factors contributing to employee attrition.
   - Analyse the relationship between job satisfaction and employee retention.
   - Explore the impact of compensation, benefits, and work-life balance on employee satisfaction.
   - Develop targeted retention strategies based on demographic and job-related factors.

3. **Key Findings:**
   - **Attrition Patterns:** This metric shows that how many employees leave the company at a specific period. And in this key metrics we found that the attrition rate of this dataset is 16.08%.
   - **Impact of travelling -** In this metric we analyse that the job satisfaction of the employees according to their different business travel. And we found that those employees who are travel occasionally are highest satisfied with their jobs as the average rate is 2.89 compared to Frequent traveller, non-traveller and rare traveller. Rare travellers have the lowest satisfaction rate with the average rate is 2.70.
   - **Compensation and Benefits-** Monthly income and salary slab data indicate that competitive compensation is crucial for retaining top talent. Employees in higher salary slabs are less likely to leave the company. Regular salary reviews and performance-based incentives can motivate employees to stay.
   - **Demographic Factors -** Gender, marital status, and education field also play a role in employee satisfaction and retention. Tailored initiatives that address the unique needs of different demographic groups can foster a more inclusive and supportive work environment.

Cleaning of the data

In this phase we need to identify and remove inaccurate, duplicates, nulls, and irrelevant data, which affect our dataset when we analyse. It is a very crucial process for extracting the accurate insights from the data.
Here are some steps of how you clean this data: -

Step1- First you need import all the library.

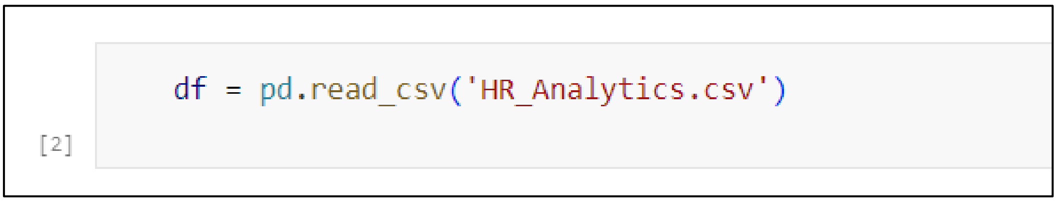


```
# Importing the libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
reset = sns.reset_defaults()
```

[1]

We use the variable called 'reset' to reset all the figure size used in this analysis.

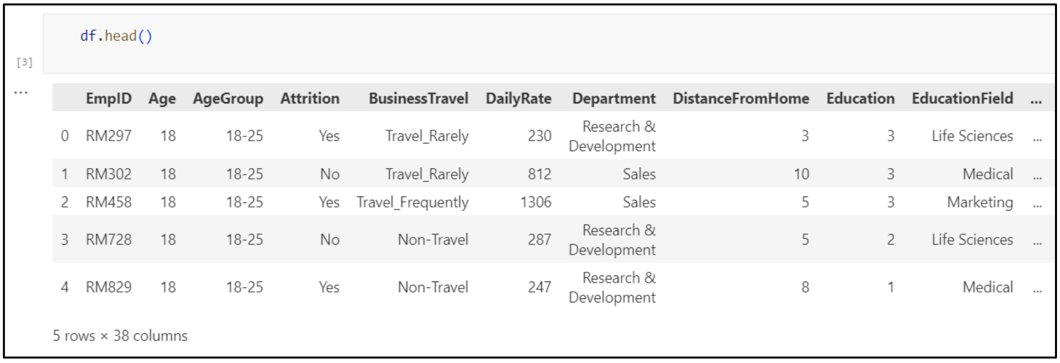
Step2- We need to read the database by using the panda's library.



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

[2]

Step3- Now we need to see the top 5 rows from the dataset.



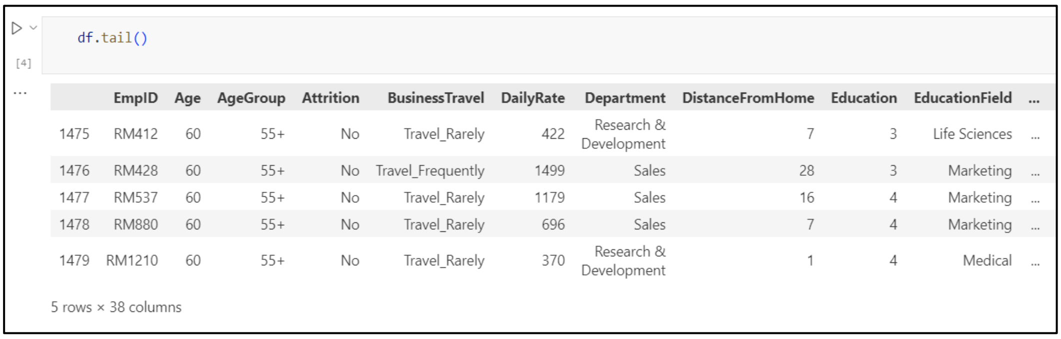
```
df.head()
```

[3]

	EmpID	Age	AgeGroup	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	...
0	RM297	18	18-25	Yes	Travel_Rarely	230	Research & Development	3	3	Life Sciences	...
1	RM302	18	18-25	No	Travel_Rarely	812	Sales	10	3	Medical	...
2	RM458	18	18-25	Yes	Travel_Frequently	1306	Sales	5	3	Marketing	...
3	RM728	18	18-25	No	Non-Travel	287	Research & Development	5	2	Life Sciences	...
4	RM829	18	18-25	Yes	Non-Travel	247	Research & Development	8	1	Medical	...

5 rows × 38 columns

Step4 – Then we need to see the last 5 rows.



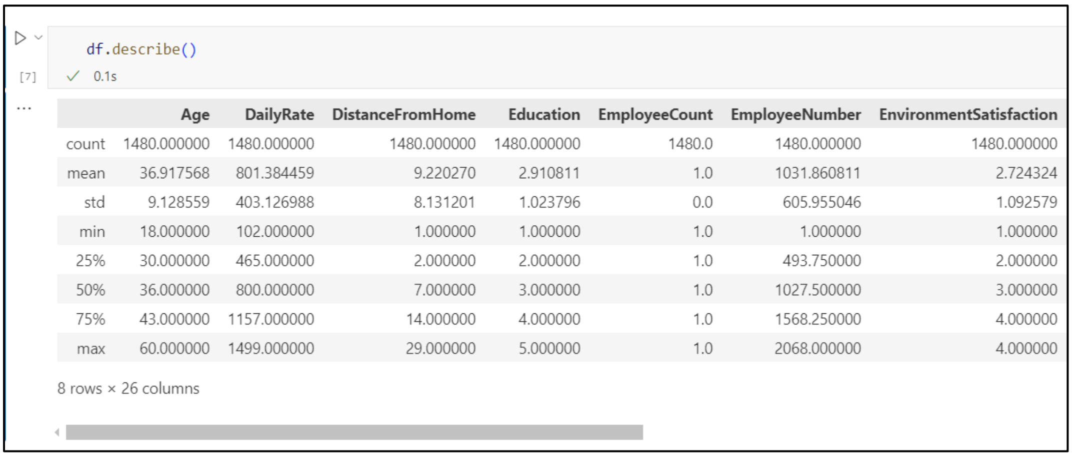
```
df.tail()
```

[4]

	EmpID	Age	AgeGroup	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	...
1475	RM412	60	55+	No	Travel_Rarely	422	Research & Development	7	3	Life Sciences	...
1476	RM428	60	55+	No	Travel_Frequently	1499	Sales	28	3	Marketing	...
1477	RM537	60	55+	No	Travel_Rarely	1179	Sales	16	4	Marketing	...
1478	RM880	60	55+	No	Travel_Rarely	696	Sales	7	4	Marketing	...
1479	RM1210	60	55+	No	Travel_Rarely	370	Research & Development	1	4	Medical	...

5 rows × 38 columns

Step5- We need to see all the statistics values or 5 pointer theory.



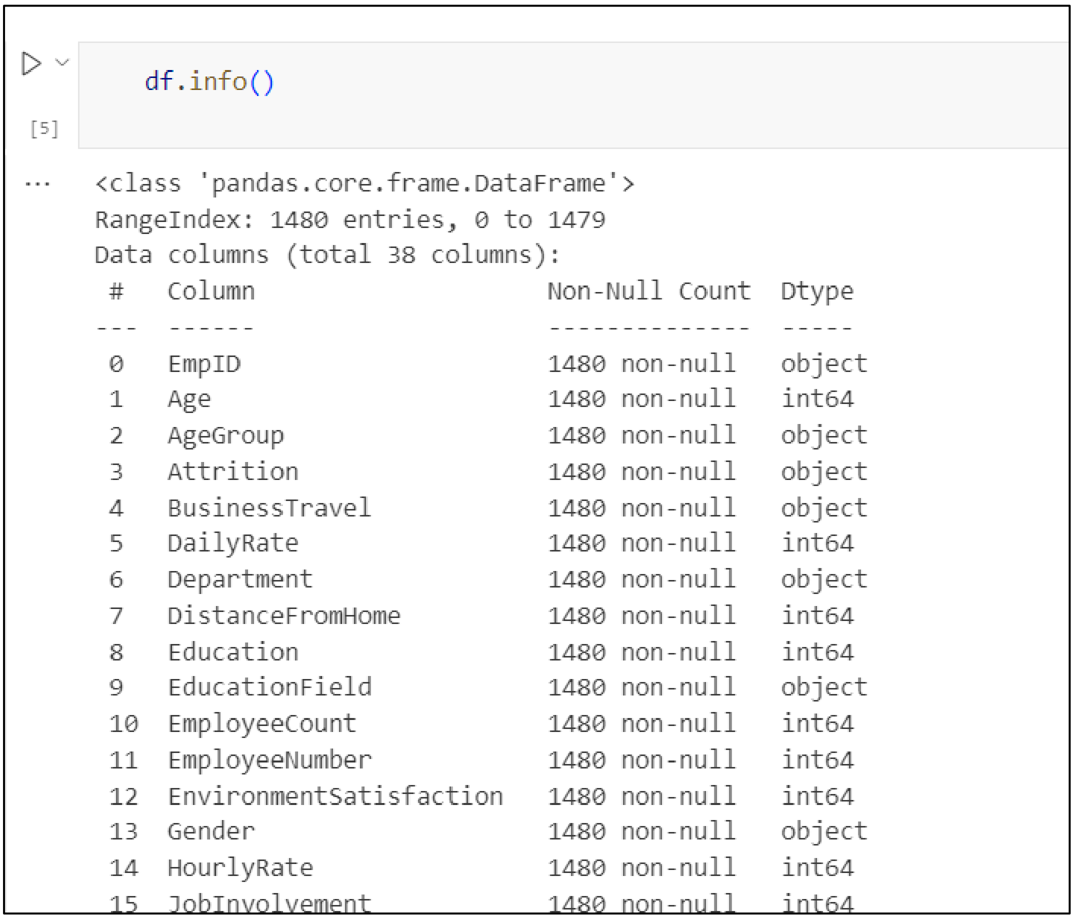
```
df.describe()
```

[7] ✓ 0.1s

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction
count	1480.000000	1480.000000	1480.000000	1480.000000	1480.0	1480.000000	1480.000000
mean	36.917568	801.384459	9.220270	2.910811	1.0	1031.860811	2.724324
std	9.128559	403.126988	8.131201	1.023796	0.0	605.955046	1.092579
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000
25%	30.000000	465.000000	2.000000	2.000000	1.0	493.750000	2.000000
50%	36.000000	800.000000	7.000000	3.000000	1.0	1027.500000	3.000000
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1568.250000	4.000000
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000

8 rows × 26 columns

Step6- We need to see the information related to the dataset.

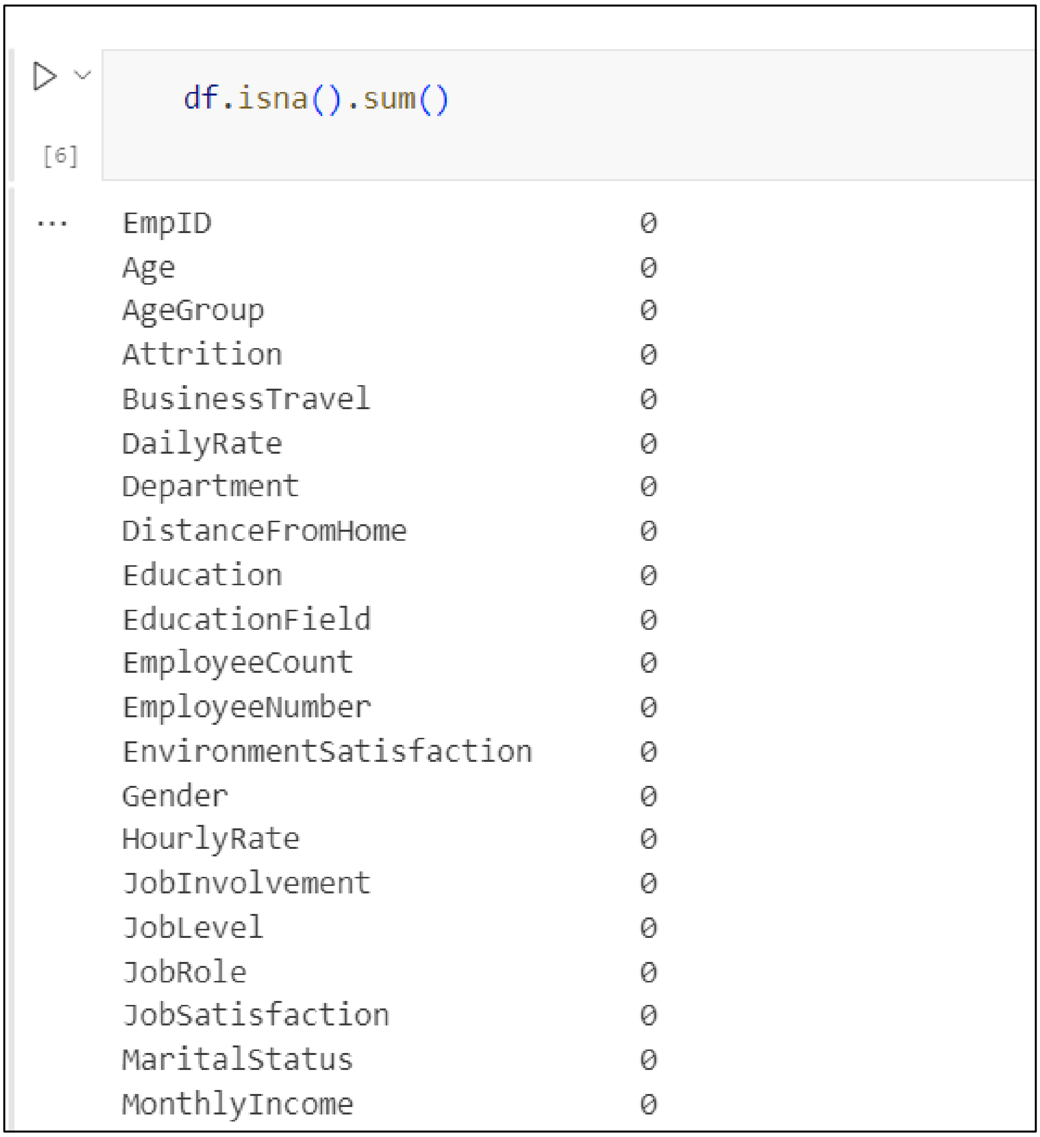


```
df.info()
```

[5]

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1480 entries, 0 to 1479
Data columns (total 38 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   EmpID                    1480 non-null   object
 1   Age                      1480 non-null   int64
 2   AgeGroup                 1480 non-null   object
 3   Attrition                1480 non-null   object
 4   BusinessTravel           1480 non-null   object
 5   DailyRate                1480 non-null   int64
 6   Department               1480 non-null   object
 7   DistanceFromHome         1480 non-null   int64
 8   Education                1480 non-null   int64
 9   EducationField           1480 non-null   object
 10  EmployeeCount            1480 non-null   int64
 11  EmployeeNumber           1480 non-null   int64
 12  EnvironmentSatisfaction  1480 non-null   int64
 13  Gender                   1480 non-null   object
 14  HourlyRate               1480 non-null   int64
 15  JobInvolvement           1480 non-null   int64
```

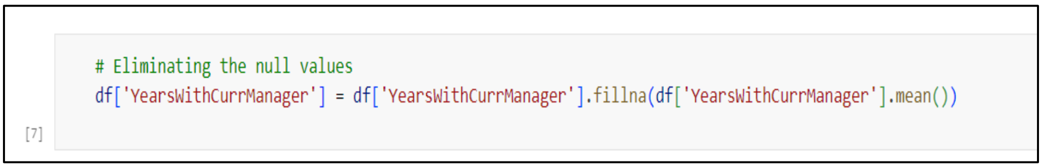
Step7- We need to check all the null values in the dataset.



```
[6]
df.isna().sum()
```

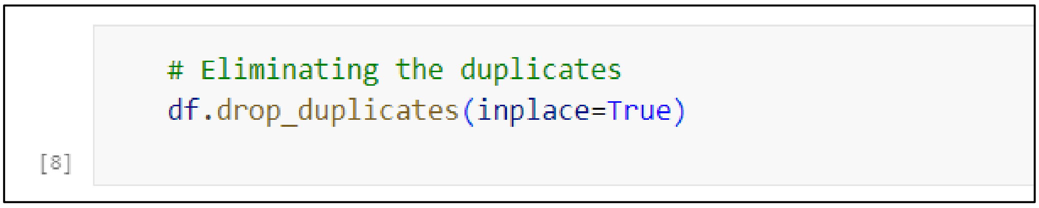
```
EmpID                       0
Age                         0
AgeGroup                    0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
```

Step8 – After identifying all the null values we need to fill those null values with mean/ median/ mode according to the dataset requirement.



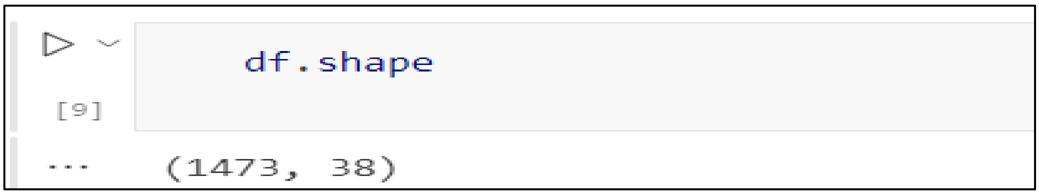
```
# Eliminating the null values
df['YearsWithCurrManager'] = df['YearsWithCurrManager'].fillna(df['YearsWithCurrManager'].mean())
[7]
```

Step9 – We need to remove all the null values (if present in the dataset).



```
# Eliminating the duplicates
df.drop_duplicates(inplace=True)
```

Step10 – Shape function will show the count of rows and columns. Here 1473 are the total rows and 38 columns.



```
df.shape
```

```
(1473, 38)
```

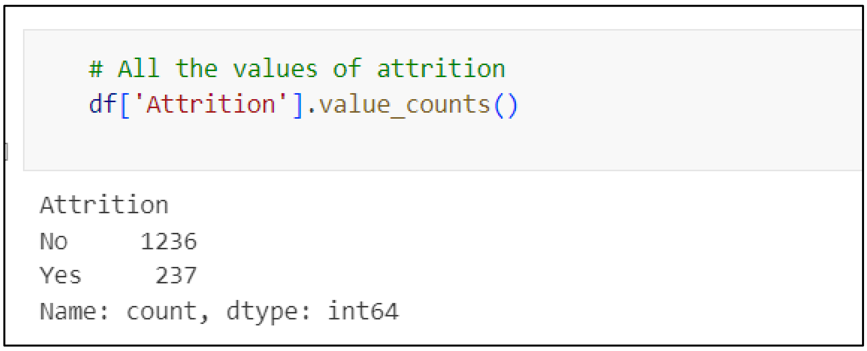
So, these are the steps of cleaning the dataset. We need to do these steps to follow the further data analysis. And the next step after the cleaning the dataset is data analysis/ interpretation.

Data Analysis

In this module we review and analyse data results to draw conclusions and make informed decisions. It involves understanding the implications of the data and identifying patterns, trends, and relationships. We analyse lots of attributes of this data like- Age Group, Business Travel, Job Role, Job Satisfaction etc.

Attrition Rate:

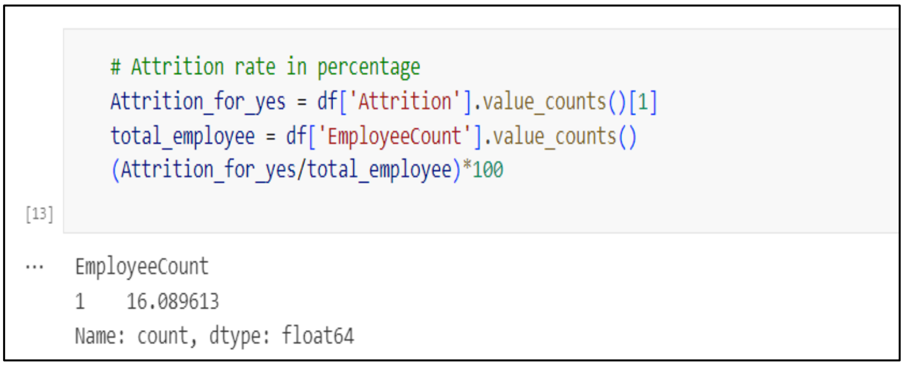
Attrition rate, often referred to as employee turnover rate, is a measure of the rate at which employees leave an organization over a specific period. It is typically expressed as a percentage.



```
# All the values of attrition
df['Attrition'].value_counts()
```

```
Attrition
No      1236
Yes      237
Name: count, dtype: int64
```

- In this we analyse count of unique values in this column and as per the results there are ‘Yes’=237 and ‘No’=1236



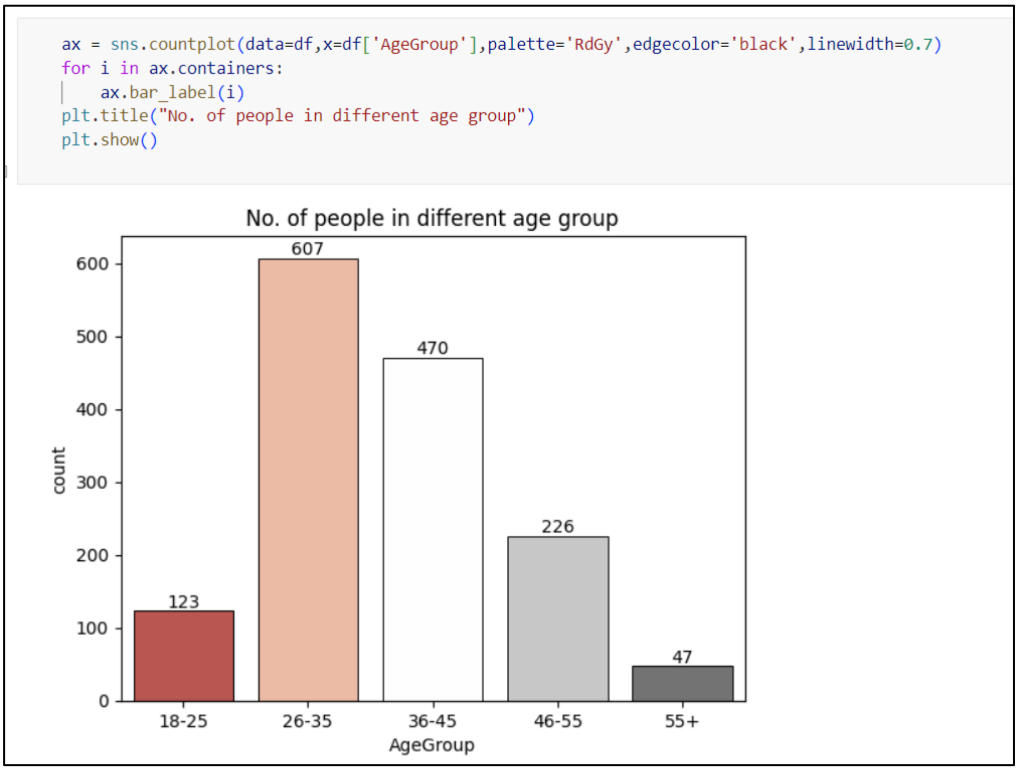
```
# Attrition rate in percentage
Attrition_for_yes = df['Attrition'].value_counts()[1]
total_employee = df['EmployeeCount'].value_counts()
(Attrition_for_yes/total_employee)*100
```

```
EmployeeCount
1    16.089613
Name: count, dtype: float64
```

- In this we analyse the attrition rate, means that what percent of the population are retained by the company which is 16.08%

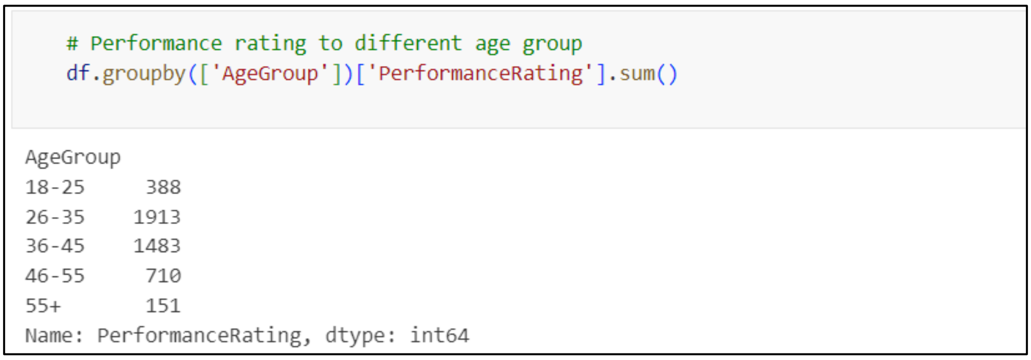
Age Group:

An age group is a category of people who fall within a specific age range. Age groups are often used in demographic studies, marketing, health research, and other fields to analyse and compare behaviours, preferences, and outcomes among different segments of the population.



```
ax = sns.countplot(data=df,x=df['AgeGroup'],palette='RdGy',edgecolor='black',linewidth=0.7)
for i in ax.containers:
    ax.bar_label(i)
plt.title("No. of people in different age group")
plt.show()
```

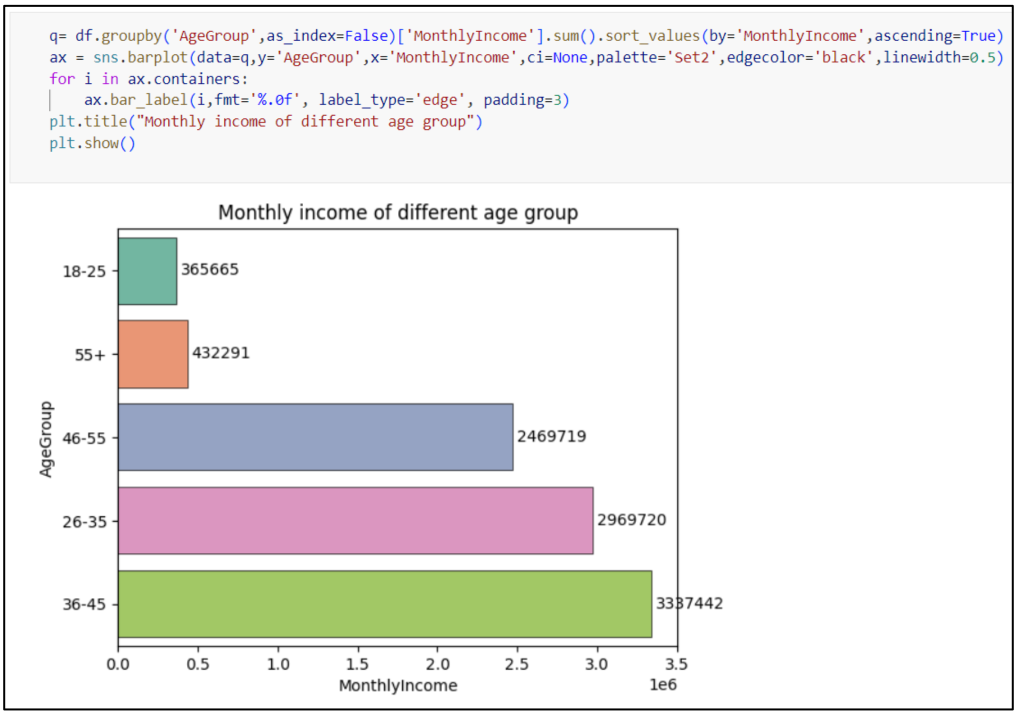
- This visual shows the number of employees in different age groups.
- This visual indicates the age demographic conditions of the company



```
# Performance rating to different age group
df.groupby(['AgeGroup'])['PerformanceRating'].sum()
```

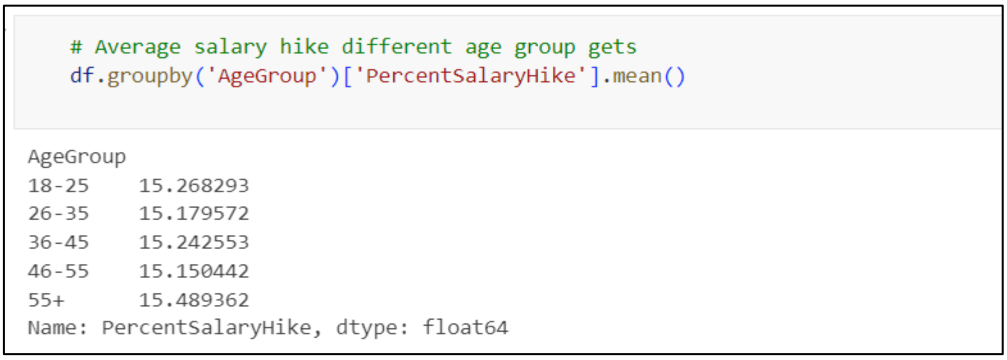
```
AgeGroup
18-25     388
26-35    1913
36-45    1483
46-55     710
55+       151
Name: PerformanceRating, dtype: int64
```

- This metric shows the performance status of employees of each age group.
- As the employees with the age group of 26-35 have the highest performance rating.



```
q= df.groupby('AgeGroup',as_index=False)['MonthlyIncome'].sum().sort_values(by='MonthlyIncome',ascending=True)
ax = sns.barplot(data=q,y='AgeGroup',x='MonthlyIncome',ci=None,palette='Set2',edgecolor='black',linewidth=0.5)
for i in ax.containers:
    ax.bar_label(i,fmt='%.0f', label_type='edge', padding=3)
plt.title("Monthly income of different age group")
plt.show()
```

- This visual indicates the monthly income of employees with different age groups.
- As in this horizontal bar graph employees with the age group of 36-45 earns highest.
- The employees whose age group is 18-25 are earns least.



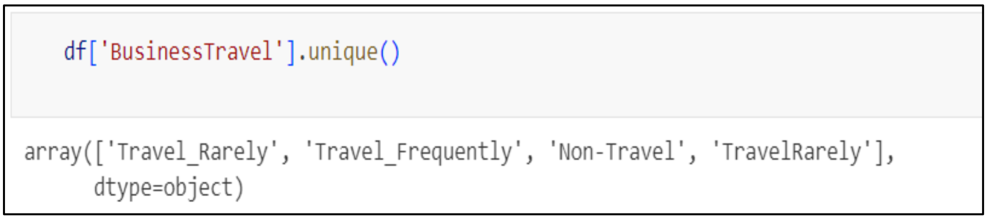
```
# Average salary hike different age group gets
df.groupby('AgeGroup')['PercentSalaryHike'].mean()
```

```
AgeGroup
18-25     15.268293
26-35     15.179572
36-45     15.242553
46-55     15.150442
55+       15.489362
Name: PercentSalaryHike, dtype: float64
```

- This metric indicates what is the average salary hike each age group received.
- Almost all the age group gets the same salary hike as their salary hike percentage lies between 15% - 15.5 %.

Business Travel:

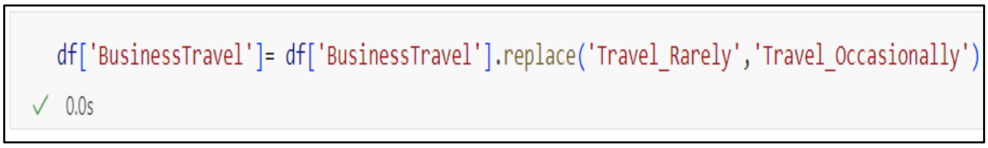
Business travel refers to the activities associated with traveling to different locations for professional purposes. This can include a variety of arrangements, from working from home, rarely traveller, frequently traveller and occasionally traveller.



```
df['BusinessTravel'].unique()
```

```
array(['Travel_Rarely', 'Travel_Frequently', 'Non-Travel', 'TravelRarely'],
      dtype=object)
```

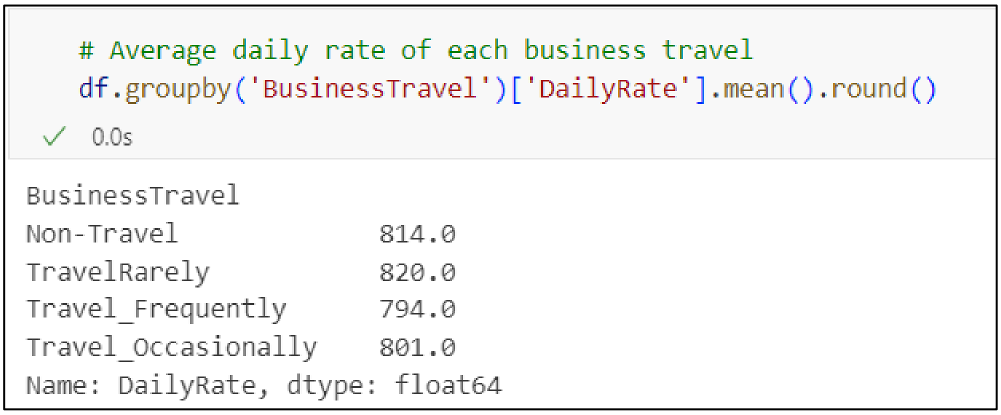
- This code shows what are the unique values present in the business travel columns.



```
df['BusinessTravel']= df['BusinessTravel'].replace('Travel_Rarely','Travel_Occasionally')
```

✓ 0.0s

- In the unique column there are two data with the same name ‘Travel Rarely’, so we need to replace one value with ‘Travel Occasionally’.

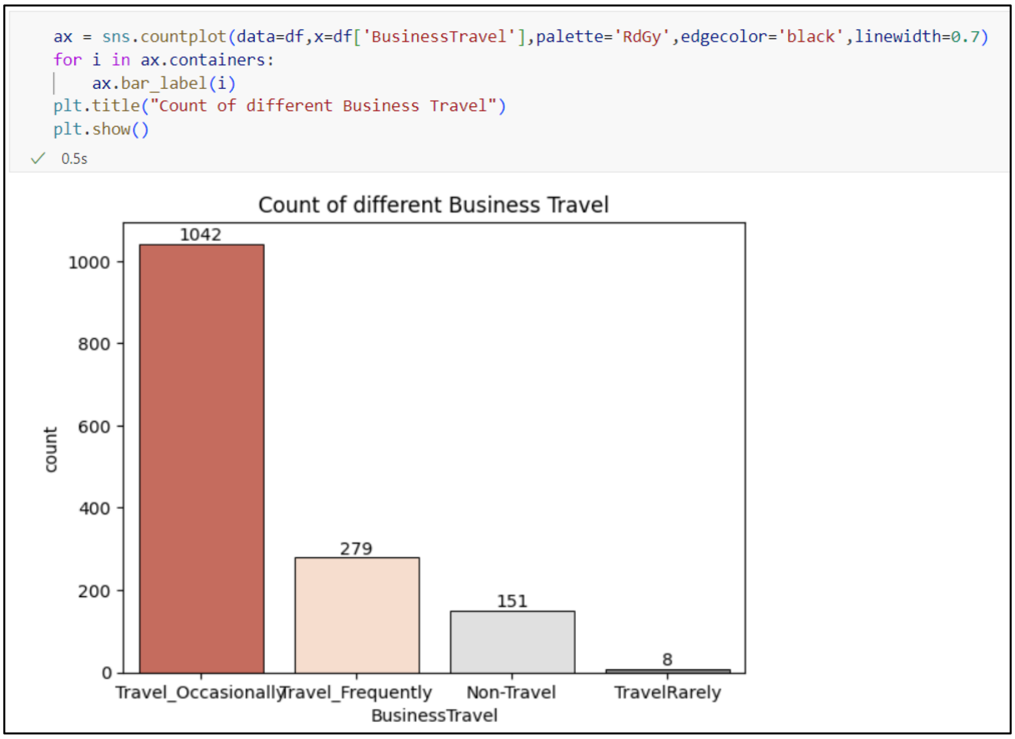


```
# Average daily rate of each business travel
df.groupby('BusinessTravel')['DailyRate'].mean().round()
```

✓ 0.0s

```
BusinessTravel
Non-Travel             814.0
TravelRarely           820.0
Travel_Frequently      794.0
Travel_Occasionally    801.0
Name: DailyRate, dtype: float64
```

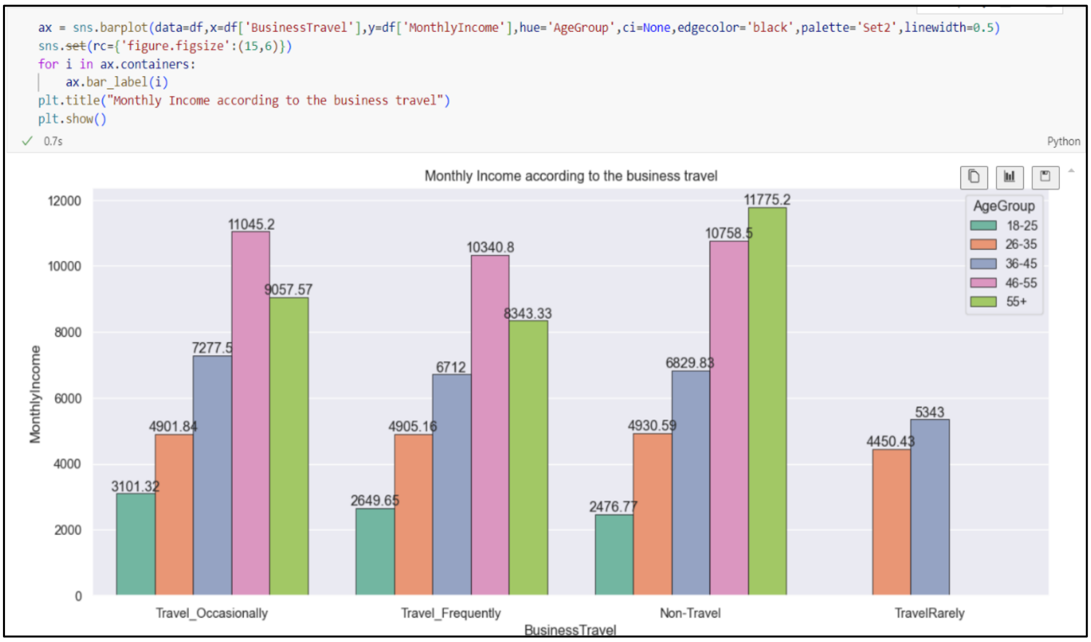
- This metric shows the average daily rate of each business travel.
- The employee who travels rarely have the highest daily rate on the average basis.



```
ax = sns.countplot(data=df,x=df['BusinessTravel'],palette='RdGy',edgecolor='black',linewidth=0.7)
for i in ax.containers:
    ax.bar_label(i)
plt.title("Count of different Business Travel")
plt.show()
```

✓ 0.5s

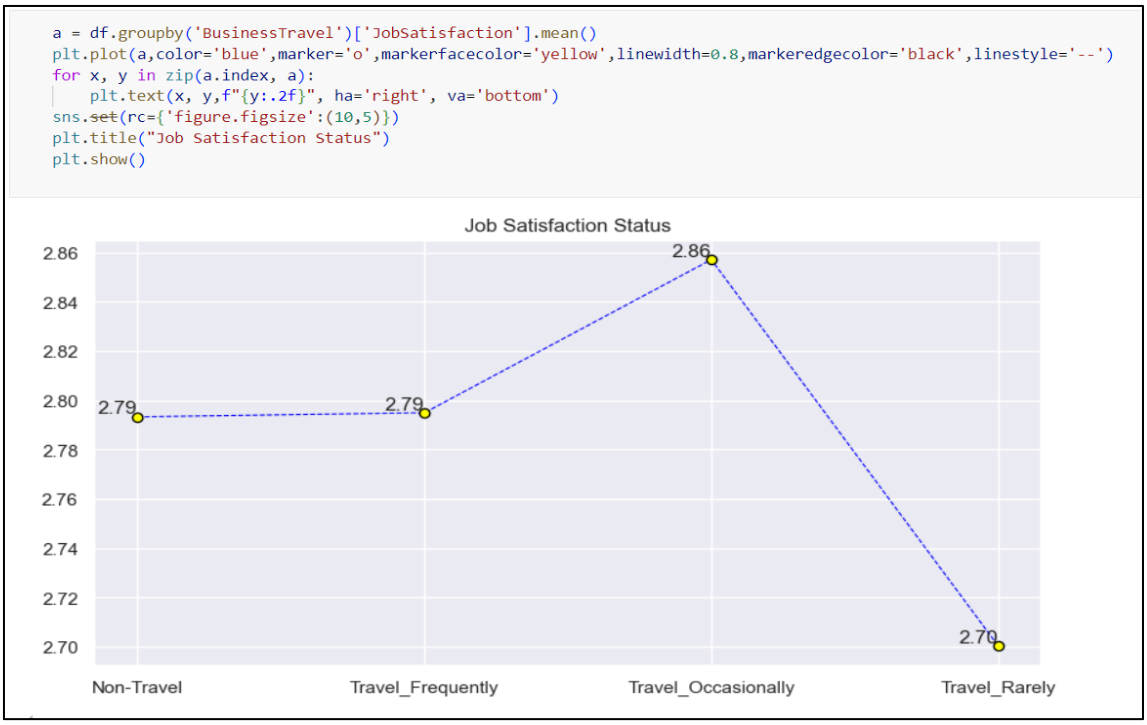
- This bar chats shows the total count of employees in each business travel
- The employees who are travelling occasionally are the highest.
- The employees who are travelling rarely are the least.



```
ax = sns.barplot(data=df,x=df['BusinessTravel'],y=df['MonthlyIncome'],hue='AgeGroup',ci=None,edgecolor='black',palette='Set2',linewidth=0.5)
sns.~~set~~(rc={'figure.figsize':(15,6)})
for i in ax.containers:
    ax.bar_label(i)
plt.title("Monthly Income according to the business travel")
plt.show()
```

✓ 0.7s Python

- This grouped bar chart shows monthly income earned by different age group and with different business travel.
- Those employees who are above age of 55 and are non-travel earns highest.
- These is no employees with the age group **18-25, 46-55,** and **55+** present in travel rarely.

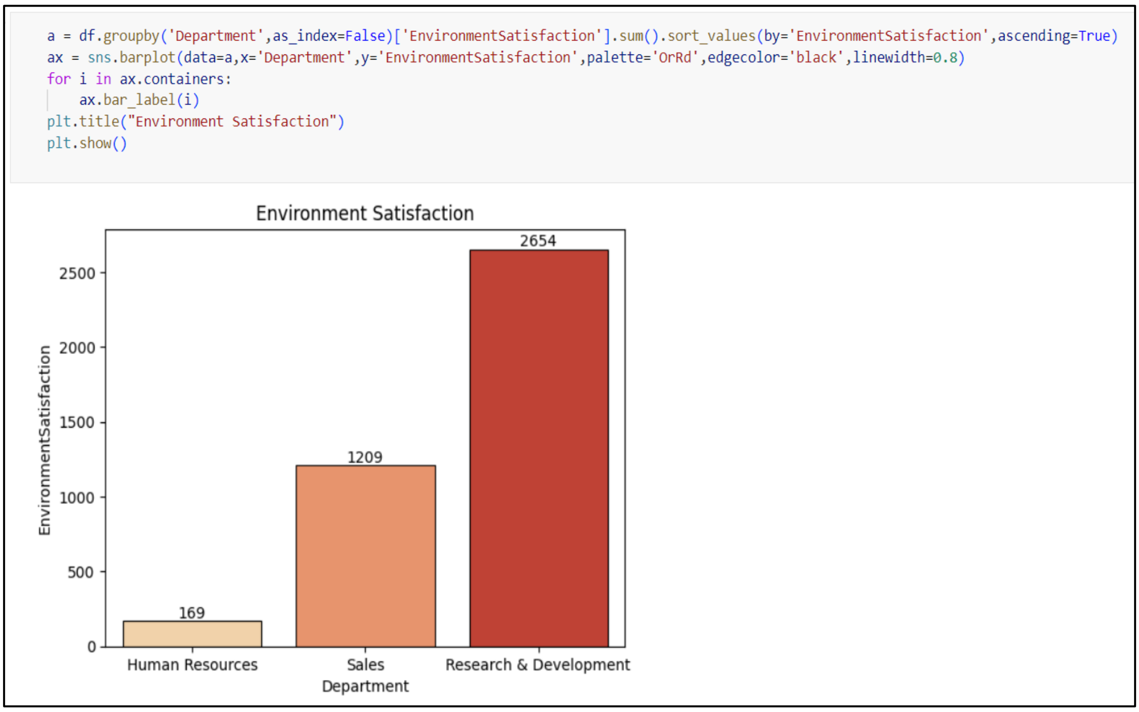


```
a = df.groupby('BusinessTravel')['JobSatisfaction'].mean()
plt.plot(a,color='blue',marker='o',markerfacecolor='yellow',linewidth=0.8,markeredgecolor='black',linestyle='--')
for x, y in zip(a.index, a):
    plt.text(x, y,f"{y:.2f}", ha='right', va='bottom')
sns.set(rc={'figure.figsize':(10,5)})
plt.title("Job Satisfaction Status")
plt.show()
```

- This visual shows that what are the average job satisfaction according to their business travel.
- The employees who are travelling occasionally have the highest job satisfaction.

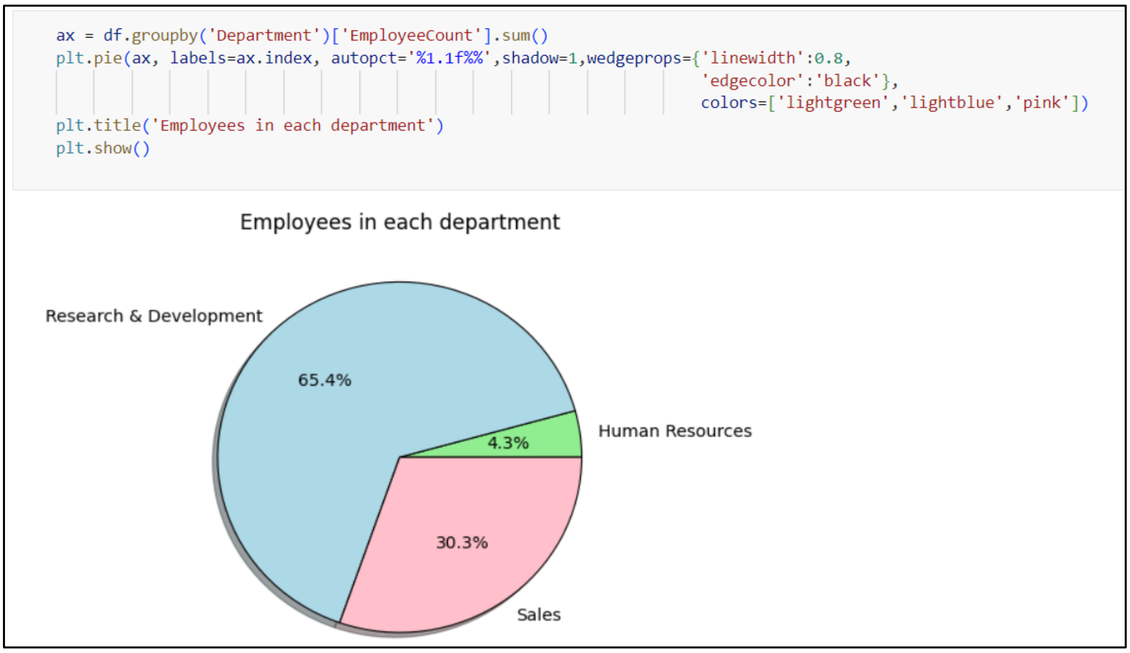
Department:

In a company, a department is a specialized division responsible for a specific area of the organization's operations or functions. Departments are typically organized based on tasks, roles, or expertise. In this data there three departments like Human Resources, Sales Department, Research and development.



```
a = df.groupby('Department',as_index=False)['EnvironmentSatisfaction'].sum().sort_values(by='EnvironmentSatisfaction',ascending=True)
ax = sns.barplot(data=a,x='Department',y='EnvironmentSatisfaction',palette='OrRd',edgecolor='black',linewidth=0.8)
for i in ax.containers:
    ax.bar_label(i)
plt.title("Environment Satisfaction")
plt.show()
```

- Environment satisfaction is basically the positive work environment which is crucial for the employees
- This visual shows the total environment satisfaction according to each department.
- As in this bar graph the Research and Development department has the highest environment satisfaction.

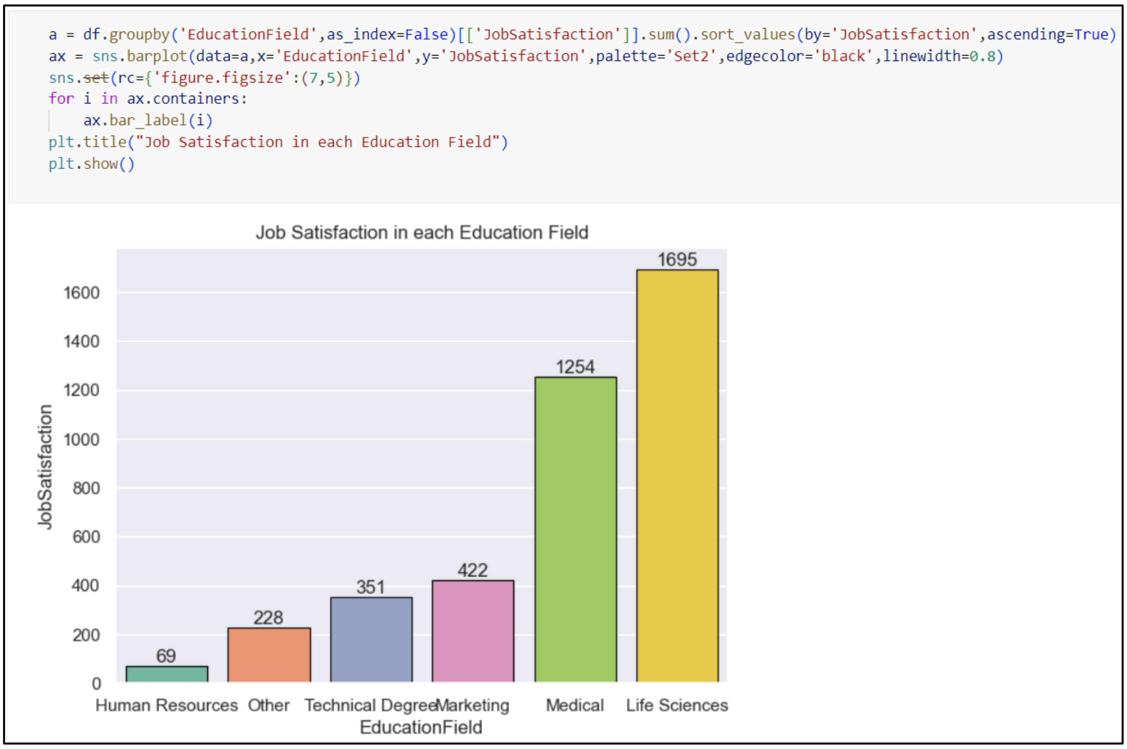


```
ax = df.groupby('Department')['EmployeeCount'].sum()
plt.pie(ax, labels=ax.index, autopct='%1.1f%%',shadow=1,wedgeprops={'linewidth':0.8,
                                                                    'edgecolor':'black'},
                                                                    colors=['lightgreen','lightblue','pink'])
plt.title('Employees in each department')
plt.show()
```

- This pie chart indicates the total number of employees in each department
- As this shows- **65.4%** of the people belongs to research and development department, **4.3%** of the people belongs to Human Resource department, and rest of the population belongs to Sales.

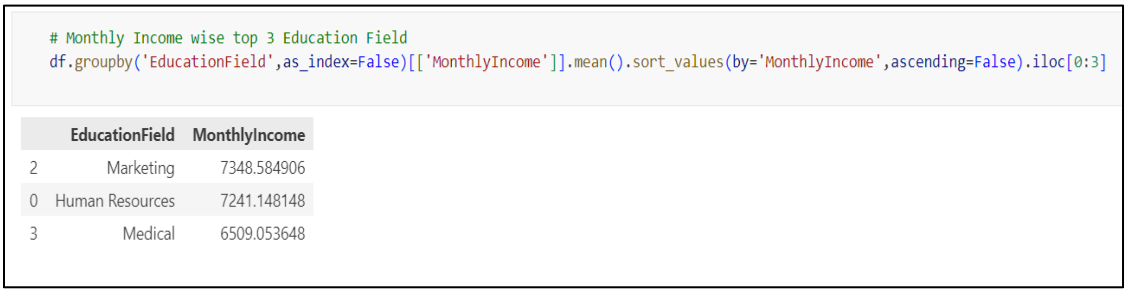
Education Field:

The education field is a broad and dynamic sector that encompasses various levels, types, and modes of teaching and learning.



```
a = df.groupby('EducationField',as_index=False)[['JobSatisfaction']].sum().sort_values(by='JobSatisfaction',ascending=True)
ax = sns.barplot(data=a,x='EducationField',y='JobSatisfaction',palette='Set2',edgecolor='black',linewidth=0.8)
sns.~~set~~(rc={'figure.figsize':(7,5)})
for i in ax.containers:
    ax.bar_label(i)
plt.title("Job Satisfaction in each Education Field")
plt.show()
```

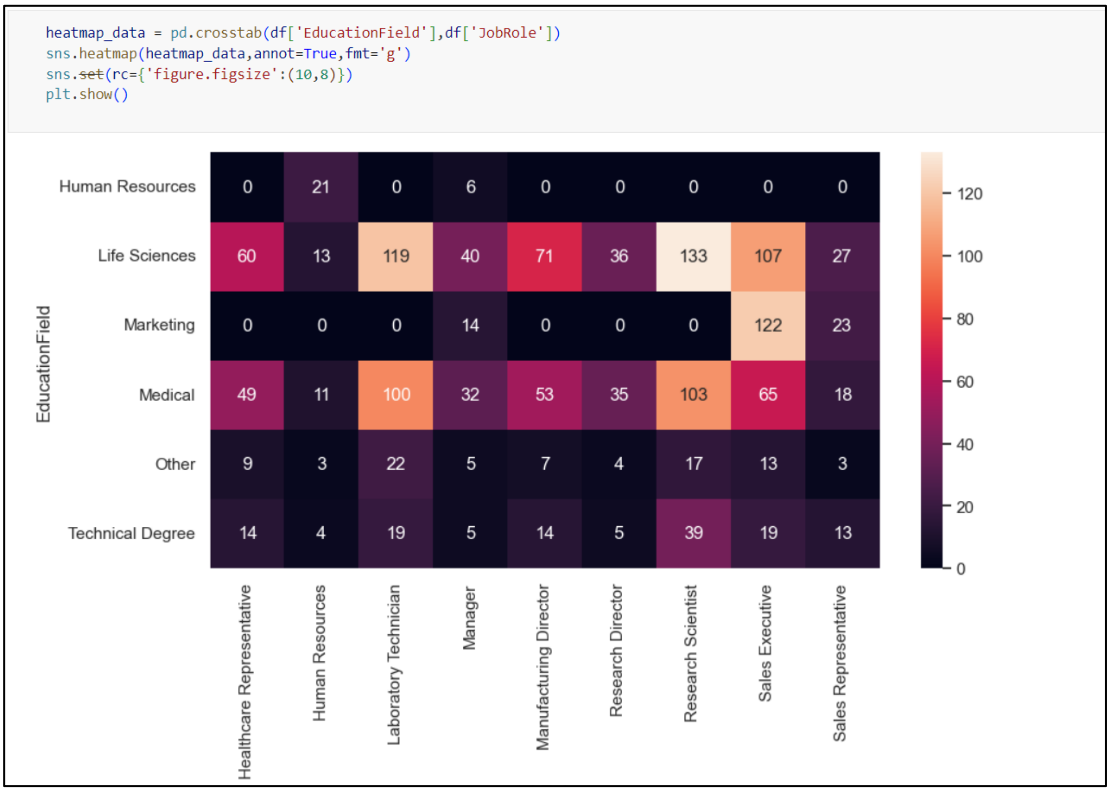
- This visual shows the job satisfaction in different areas of education field.
- As per this visual the employee who have the Life Science background have the highest job satisfaction.



```
# Monthly Income wise top 3 Education Field
df.groupby('EducationField',as_index=False)[['MonthlyIncome']].mean().sort_values(by='MonthlyIncome',ascending=False).iloc[0:3]
```

	EducationField	MonthlyIncome
2	Marketing	7348.584906
0	Human Resources	7241.148148
3	Medical	6509.053648

- This metrics shows the top 3 highest earning education field.
- Marketing has the highest average monthly salary of about **Rs. 7348.54**
- Human Resource is the second highest earning salary of about **Rs.7241.14.**
- Medical has the third highest earning salary of about **Rs.6509.05**



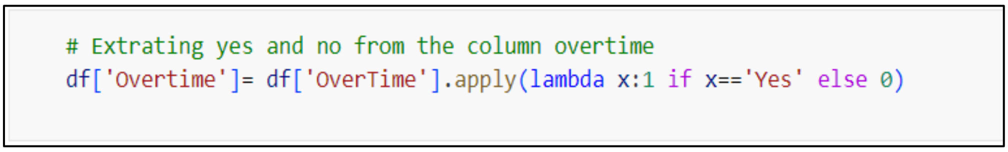
```
heatmap_data = pd.crosstab(df['EducationField'],df['JobRole'])
sns.heatmap(heatmap_data,annot=True,fmt='g')
sns.set(rc={'figure.figsize':(10,8)})
plt.show()
```

EducationField	Healthcare Representative	Human Resources	Laboratory Technician	Manager	Manufacturing Director	Research Director	Research Scientist	Sales Executive	Sales Representative
Human Resources	0	21	0	6	0	0	0	0	0
Life Sciences	60	13	119	40	71	36	133	107	27
Marketing	0	0	0	14	0	0	0	122	23
Medical	49	11	100	32	53	35	103	65	18
Other	9	3	22	5	7	4	17	13	3
Technical Degree	14	4	19	5	14	5	39	19	13

- This is called heatmap. This visual compare two categorical data at the same time.
- This shows how many employees come from different education field and pursue different Job role
- For example- There are 60 employees who have the education field of Life Science, and those 60 employees are working in Healthcare Representative.
- This visual shows analysis for all the respective Education field.

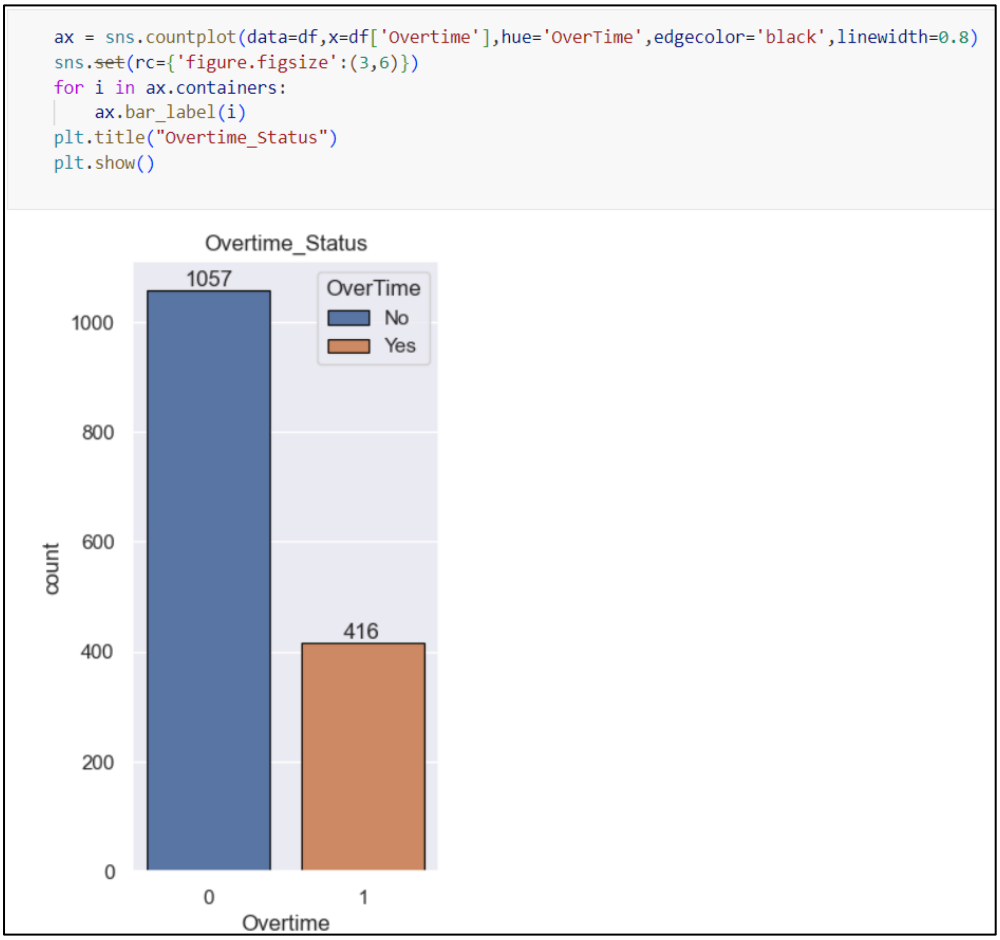
Overtime:

Overtime refers to the additional hours worked by an employee beyond their regular working hours. It often involves extra compensation for the extra time spent on the job.



```
# Extrating yes and no from the column overtime
df['Overtime']= df['OverTime'].apply(lambda x:1 if x=='Yes' else 0)
```

- We need to transform the data of overtime into numerical so that it will help us to create a visual for the analysis.

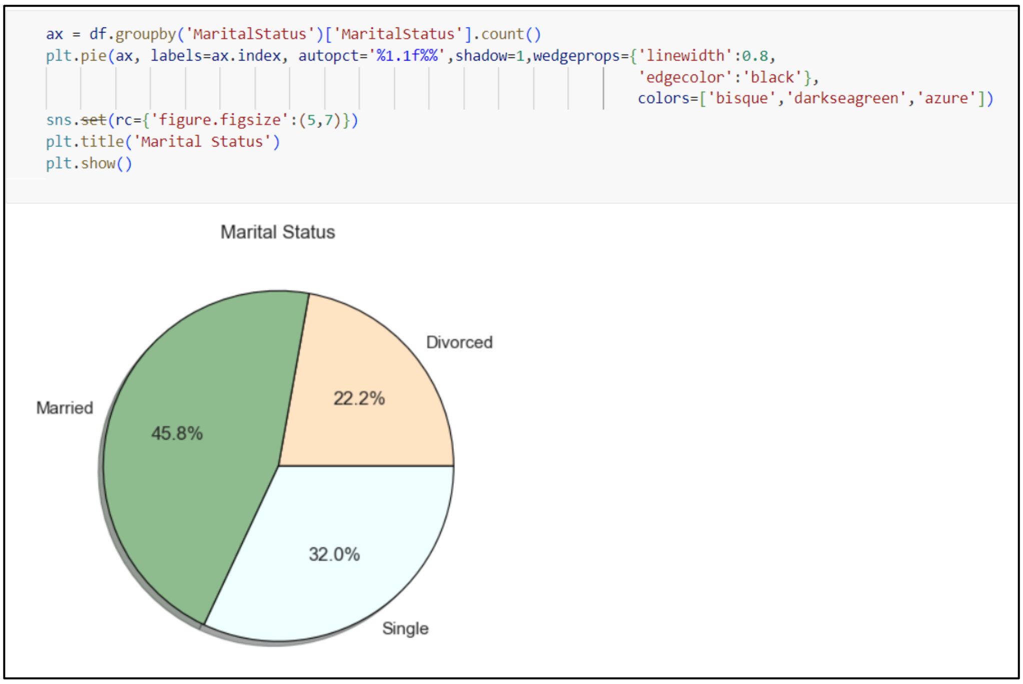


```
ax = sns.countplot(data=df,x=df['Overtime'],hue='OverTime',edgecolor='black',linewidth=0.8)
sns.set(rc={'figure.figsize':(3,6)})
for i in ax.containers:
    ax.bar_label(i)
plt.title("Overtime_Status")
plt.show()
```

- This bar graph shows the ratio of employees doing overtime.
- As per this visual, 416 employees are doing overtime and rest of them are doing their work on regular basis.

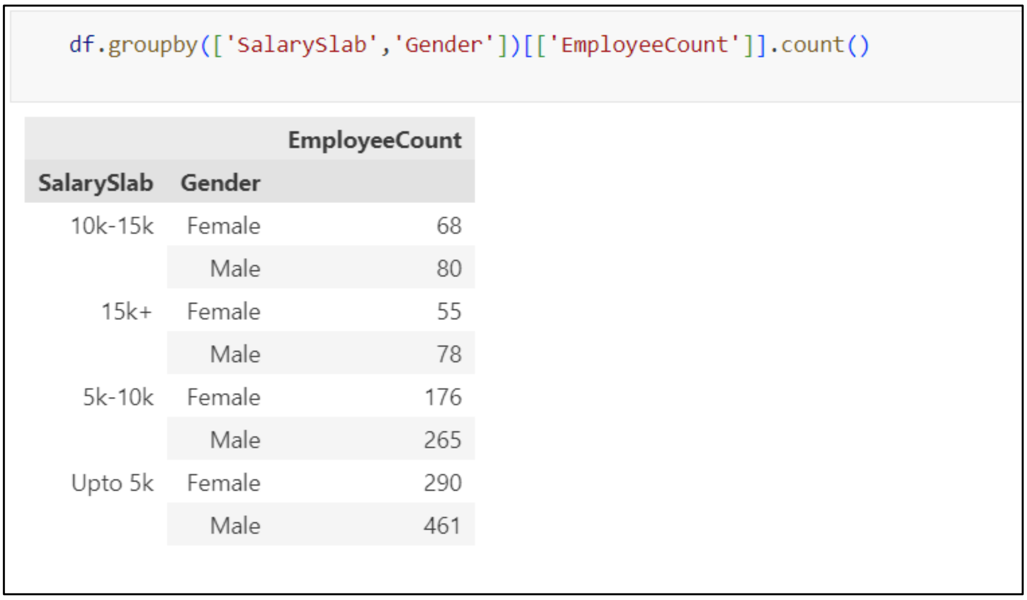
Miscellaneous Visuals:

Here are some visuals which are not part of any category



```
ax = df.groupby('MaritalStatus')['MaritalStatus'].count()
plt.pie(ax, labels=ax.index, autopct='%1.1f%%',shadow=1,wedgeprops={'linewidth':0.8,
                                                                      'edgecolor':'black'},
                                                                      colors=['bisque','darkseagreen','azure'])
sns.set(rc={'figure.figsize':(5,7)})
plt.title('Marital Status')
plt.show()
```

- This pie charts shows the marital status of the employees.
- There are 45.8% of the employees whose marital status is **Married.**
- There are 22.2% of the employees whose marital status is **Divorced.**
- There are 32.0% of the employees whose marital status is **Single.**



```
df.groupby(['SalarySlab','Gender'])[['EmployeeCount']].count()
```

		EmployeeCount
SalarySlab	**Gender**	
10k-15k	Female	68
	Male	80
15k+	Female	55
	Male	78
5k-10k	Female	176
	Male	265
Upto 5k	Female	290
	Male	461

- This metric shows the grouped data of the employees according to their salary slab, gender and their count.

Conclusion:

This report aims all the key insights extracted from the data that we analyse, all the attributes of this dataset like Age, Age Group, Attrition, Business Travel, Department, Education Field, etc are cleaned and interpreted with the help of visuals.
Here all the relevant insights are as follows: -

1. **Attrition Rate-** This visual shows that how many employees leave the company at a specific period. And in this key metrics we found that the attrition rate of this dataset is 16.08%.

2. **Impact of travelling -** In this metric we analyse that the job satisfaction of the employees according to their different business travel. And we found that those employees who are travel occasionally are highest satisfied with their jobs as the average rate is 2.89 compared to frequent traveller, non-traveller and rare traveller. Rare travellers have the lowest satisfaction rate with the average rate is 2.70.

3. **Compensation and Benefits-** Monthly income and salary slab data indicate that competitive compensation is crucial for retaining top talent. Employees in higher salary slabs are less likely to leave the company. Regular salary reviews and performance-based incentives can motivate employees to stay.

4. **Demographic Factors -** Gender, marital status, and education field also play a role in employee satisfaction and retention. Tailored initiatives that address the unique needs of different demographic groups can foster a more inclusive and supportive work environment.