```
# Salary Prediction Data Science Project

# Performing Data Cleaning, Data analysis, Data visualization, Train
ML Models, Feature Engineering, creating web app using streamlet

import pandas as pd

data = pd.read_excel("H:\\Data Aalystics\\Projects\\DS & ML\\Salary
Prediction\\Employees.xlsx")

data.head()
```

```
   No First Name Last Name  Gender Start Date  Years
Department  \
0   1     Ghadir     Hmshw    Male 2018-04-04      2     Quality
Control
1   2       Omar    Hishan    Male 2020-05-21      0     Quality
Control
2   3      Ailya    Sharaf  Female 2017-09-28      3  Major Mfg
Projects
3   4       Lwiy     Qbany    Male 2018-08-14      2
Manufacturing
4   5      Ahmad     Bikri    Male 2020-03-11      0
Manufacturing

                Country Center  Monthly Salary  Annual Salary  Job
Rate  \
0                 Egypt   West            1560          18720
3.0
1          Saudi Arabia   West            3247          38964
1.0
2          Saudi Arabia   West            2506          30072
2.0
3  United Arab Emirates   Main            1828          21936
3.0
4                 Egypt   Main             970          11640
5.0

   Sick Leaves  Unpaid Leaves  Overtime Hours
0            1              0             183
1            0              5             198
2            0              3             192
3            0              0               7
4            0              5             121
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 689 entries, 0 to 688
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
```

```
---  ------           --------------  -----
 0   No               689 non-null    int64
 1   First Name       689 non-null    object
 2   Last Name        689 non-null    object
 3   Gender           689 non-null    object
 4   Start Date       689 non-null    datetime64[ns]
 5   Years            689 non-null    int64
 6   Department       689 non-null    object
 7   Country          689 non-null    object
 8   Center           689 non-null    object
 9   Monthly Salary   689 non-null    int64
 10  Annual Salary    689 non-null    int64
 11  Job Rate         689 non-null    float64
 12  Sick Leaves      689 non-null    int64
 13  Unpaid Leaves    689 non-null    int64
 14  Overtime Hours   689 non-null    int64
dtypes: datetime64[ns](1), float64(1), int64(7), object(6)
memory usage: 80.9+ KB
```

```
data.shape
```

```
(689, 15)
```

```
data.isna().sum()
```

```
No                 0
First Name         0
Last Name          0
Gender             0
Start Date         0
Years              0
Department         0
Country            0
Center             0
Monthly Salary     0
Annual Salary      0
Job Rate           0
Sick Leaves        0
Unpaid Leaves      0
Overtime Hours     0
dtype: int64
```

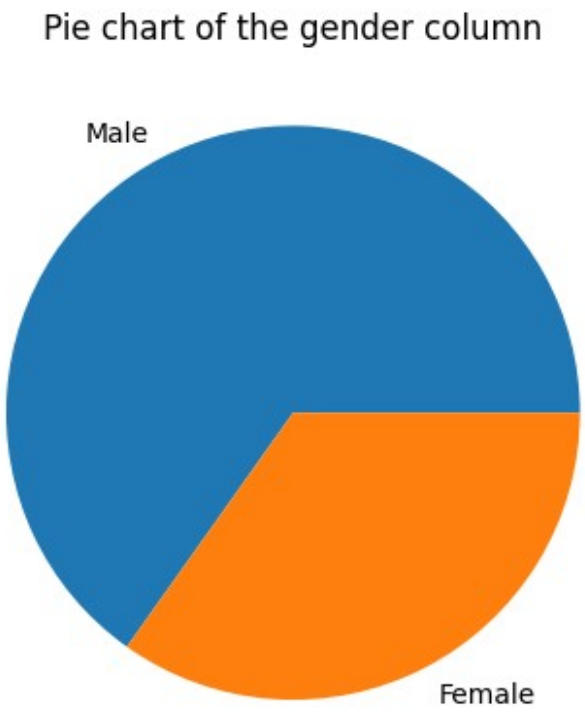
```
#data.dropna(inplace = true) droping null values in the dataset
```

```
data.duplicated().sum()
```

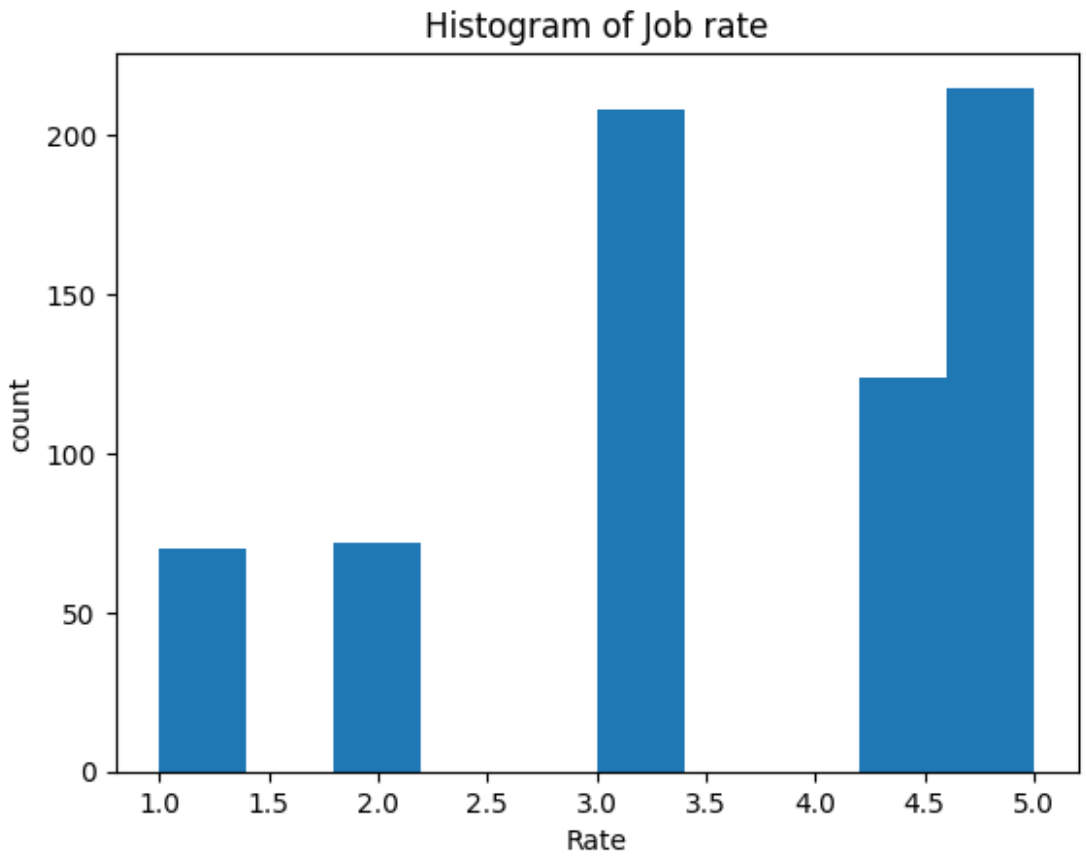
```
0
```

```
#data.drop_duplicates(inplace = true) droping duplicates values in the
dataset
```

```
import matplotlib.pyplot as plt

data["Gender"].value_counts().sort_values (ascending =
False).plot(kind = "pie")
plt.title("Pie chart of the gender column")
plt.ylabel("")
plt.show()
```




```
plt.hist(data["Job Rate"])
plt.title("Histogram of Job rate")
plt.xlabel('Rate')
plt.ylabel('count')
plt.show()
```



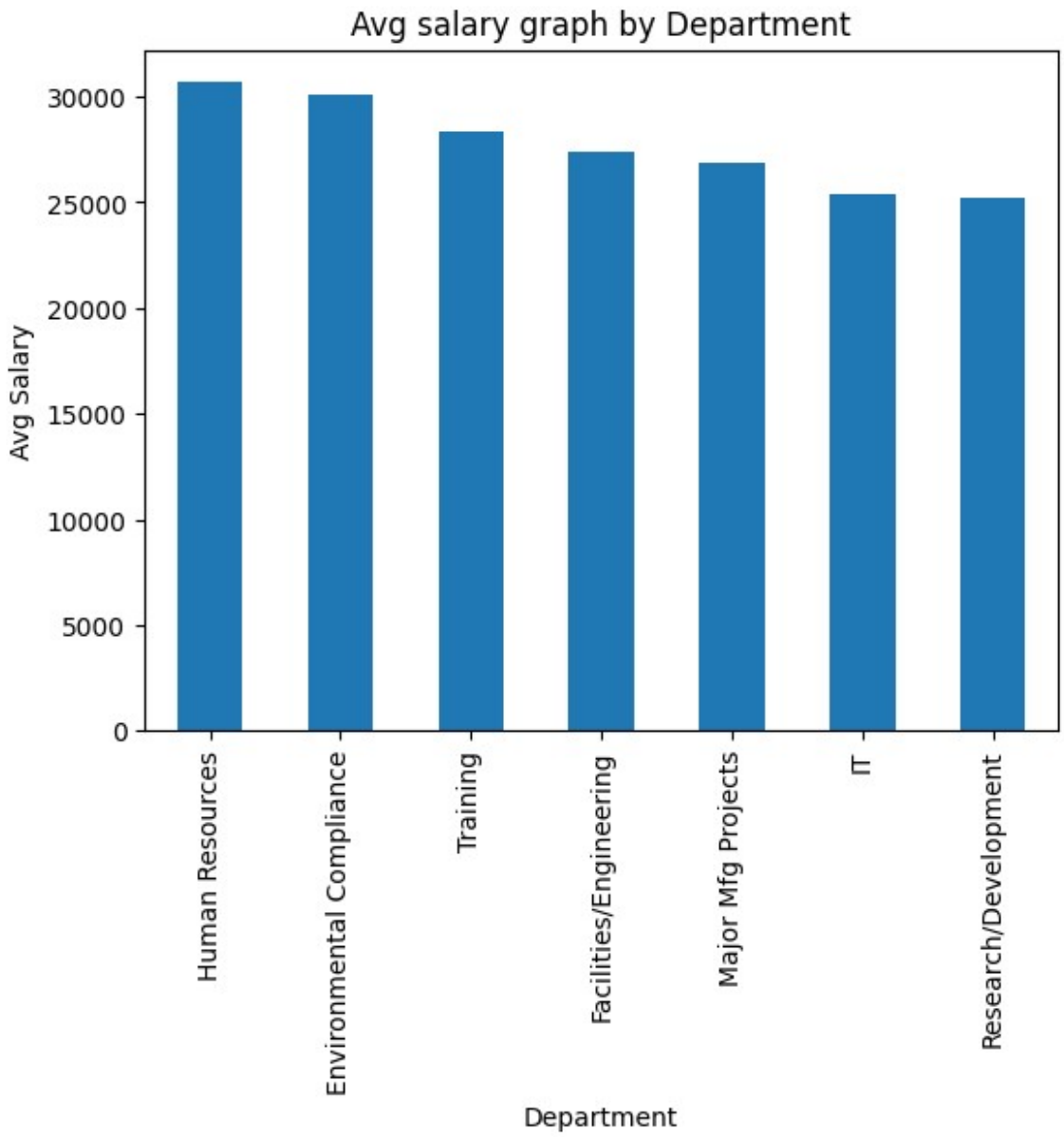

```
data["Job Rate"]. describe()
```

```
count    689.000000
mean       3.586357
std        1.350125
min        1.000000
25%        3.000000
50%        3.000000
75%        5.000000
max        5.000000
Name: Job Rate, dtype: float64
```

```
data.head()
```

```
   No First Name Last Name  Gender Start Date  Years
Department  \
0   1     Ghadir     Hmshw    Male 2018-04-04      2     Quality
Control
1   2       Omar    Hishan    Male 2020-05-21      0     Quality
Control
2   3      Ailya    Sharaf  Female 2017-09-28      3  Major Mfg
Projects
3   4       Lwiy     Qbany    Male 2018-08-14      2
```

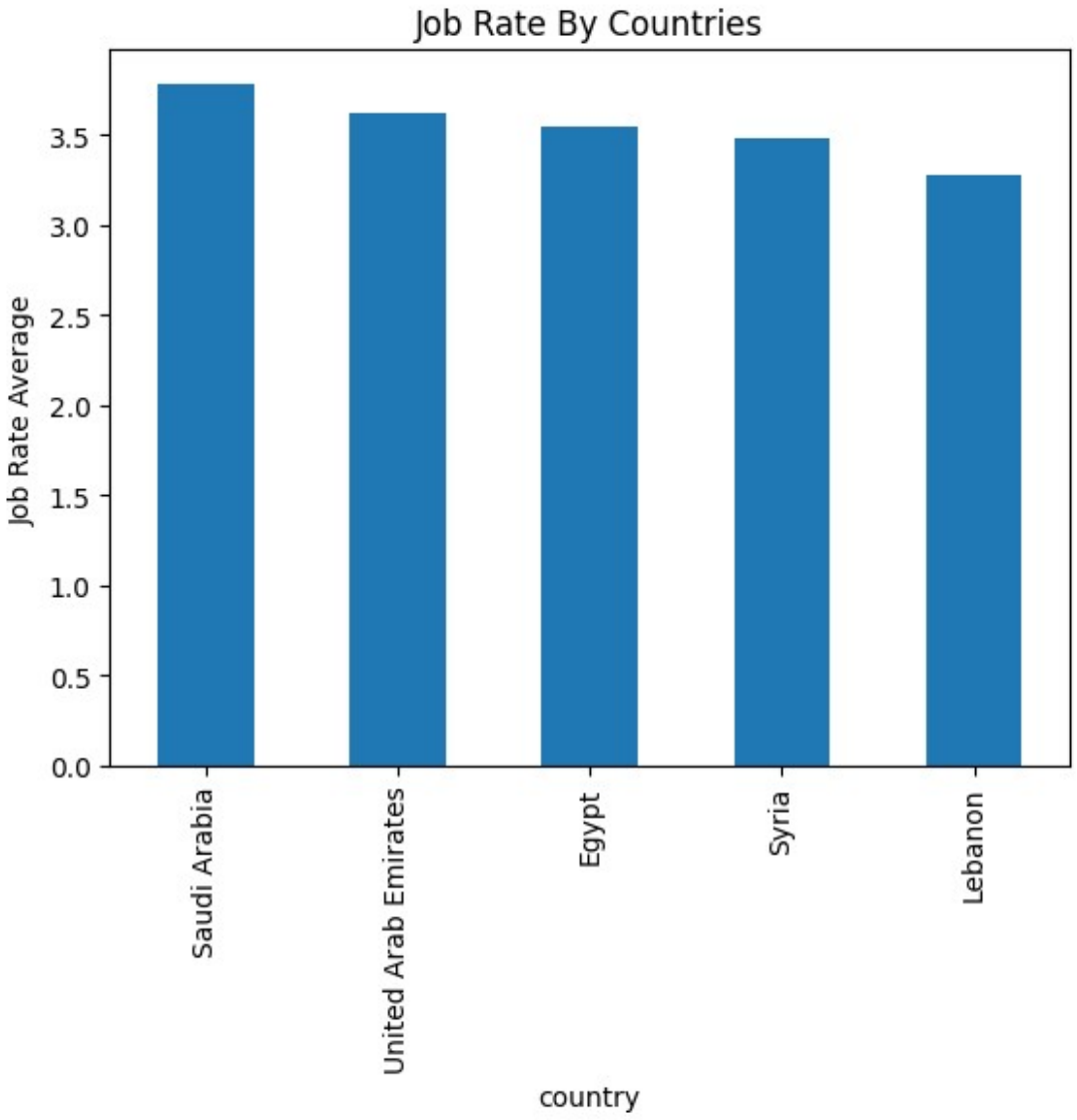
```
Manufacturing
4   5      Ahmad     Bikri    Male 2020-03-11      0
Manufacturing

                 Country Center  Monthly Salary  Annual Salary  Job
Rate  \
0                   Egypt   West            1560          18720
3.0
1           Saudi Arabia   West            3247          38964
1.0
2           Saudi Arabia   West            2506          30072
2.0
3  United Arab Emirates   Main            1828          21936
3.0
4                   Egypt   Main             970          11640
5.0

   Sick Leaves  Unpaid Leaves  Overtime Hours
0            1              0             183
1            0              5             198
2            0              3             192
3            0              0               7
4            0              5             121

data.groupby("Department")["Annual
Salary"].mean().sort_values(ascending = False).head(7).plot(kind =
"bar")
plt.title("Avg salary graph by Department")
plt.xlabel("Department")
plt.ylabel("Avg Salary")
plt.show()
```



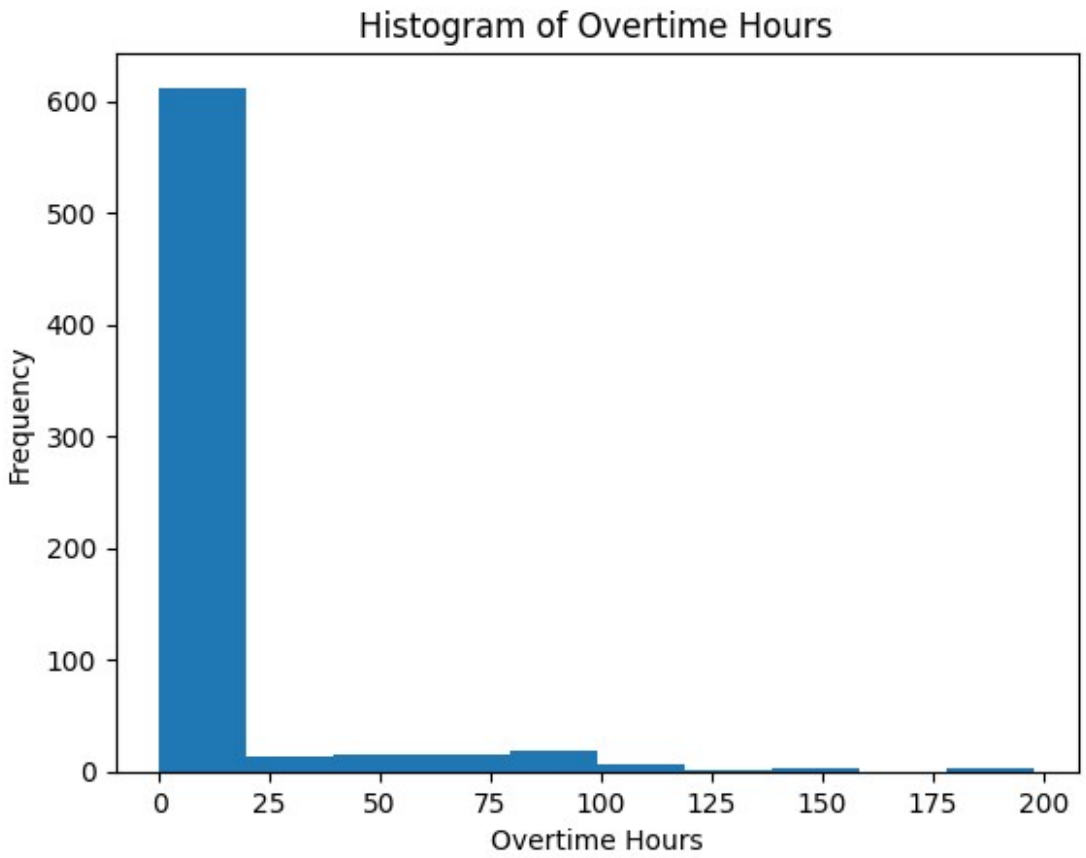

```
data.groupby("Center")["Monthly Salary"].mean().sort_values(ascending
= False)

Center
East     2274.021277
West     2068.672269
North    2064.811594
Main     2054.776892
South    1981.153846
Name: Monthly Salary, dtype: float64

data["Country"].unique()
```

```
array(['Egypt', 'Saudi Arabia', 'United Arab Emirates', 'Syria',
       'Lebanon'], dtype=object)
```

```
data.groupby("Country")["Job Rate"].mean().sort_values(ascending=
False).plot (kind ='bar')
plt.title("Job Rate By Countries")
plt.xlabel("country")
plt.ylabel("Job Rate Average")
plt.show()
```




```
plt.hist(data["Overtime Hours"])
plt.title("Histogram of Overtime Hours")
plt.xlabel("Overtime Hours")
plt.ylabel("Frequency")
plt.show()
```




```
data["Overtime Hours"].describe()
```

```
count    689.000000
mean      13.702467
std       25.692049
min        0.000000
25%        3.000000
50%        7.000000
75%       10.000000
max      198.000000
Name: Overtime Hours, dtype: float64
```

```
data["Annual Salary"].describe()
```

```
count      689.000000
mean     24818.420900
std       9159.470878
min       8436.000000
25%      17232.000000
50%      24924.000000
75%      32184.000000
max      41400.000000
Name: Annual Salary, dtype: float64
```

```
data.columns
```

```
Index(['No', 'First Name', 'Last Name', 'Gender', 'Start Date',
'Years',
       'Department', 'Country', 'Center', 'Monthly Salary', 'Annual
Salary',
       'Job Rate', 'Sick Leaves', 'Unpaid Leaves', 'Overtime Hours'],
      dtype='object')
```

```
X = data[["Years", "Job Rate"]]
y = data["Annual Salary"]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)
```

```
len(y_train)
```

```
551
```

```
len(X_test)
```

```
138
```

```
from sklearn.linear_model import LinearRegression
```

```
lr = LinearRegression()
```

```
lr.fit(X_train,y_train)
```

```
LinearRegression()
```

```
predslr = lr.predict(X_test)
```

```
from sklearn.metrics import mean_absolute_error
```

```
mean_absolute_error(predslr,y_test)
```

```
7470.017953159506
```

```
X
```

```
     Years  Job Rate
0        2       3.0
1        0       1.0
2        3       2.0
3        2       3.0
4        0       5.0
..     ...       ...
684      0       2.0
685      0       3.0
686      3       5.0
687      2       3.0
```

```
688      0       5.0

[689 rows x 2 columns]
```

```
import joblib
joblib.dump(lr,"linearmodel.pkl")
```

```
['linearmodel.pkl']
```

```
!streamlit run app.py
```