# Shri Ramdeobaba College of Engineering & Management, Nagpur

RCOEM
Shri Ramdeobaba College of
Engineering and Management, Nagpur

# TOOLS FOR DATA SCIENCE

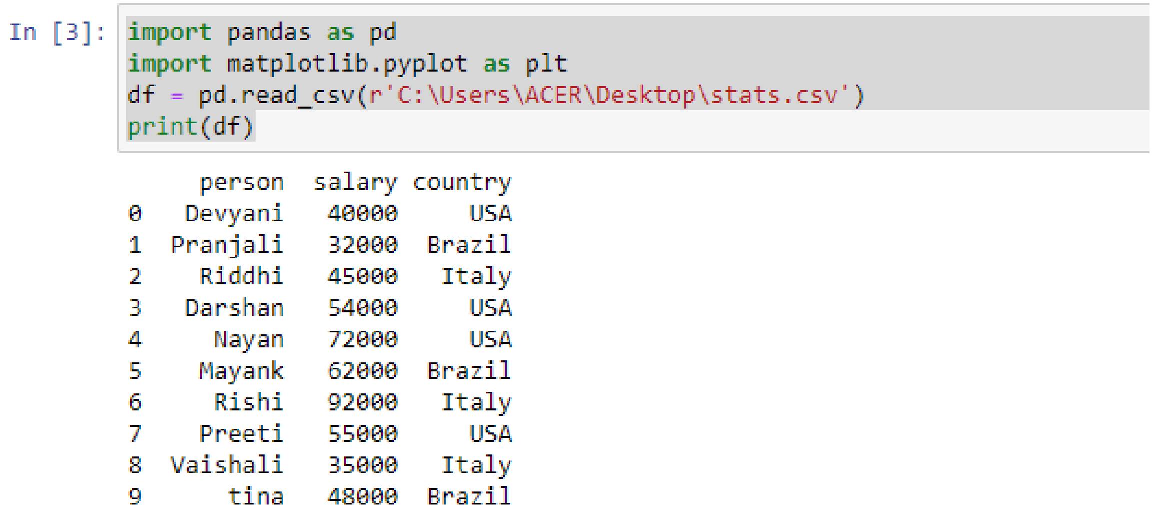
## TEACHER'S ASSESSMENT-01

| | |
|---|---|
| **Name** | DEVYANI THAKRE |
| **Branch/Roll.No.** | ECS/B-04 |

### 1. Data Analysis with Pandas and Matplotlib:

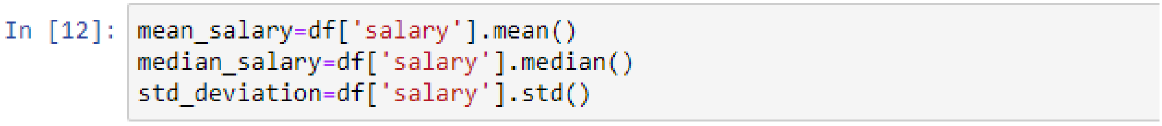
• Objective: Perform data analysis on a given dataset using Pandas and visualize the results using Matplotlib.

• Requirements: Choose a dataset (e.g., CSV, Excel, or any other format) related to a topic of interest (e.g., finance, sports, health). Use Pandas to load and clean the data. Perform basic statistical analysis (mean, median, standard deviation). Create meaningful visualizations using Matplotlib (e.g., bar chart, line plot, scatter plot). Provide insights or conclusions based on the analysis.

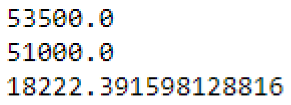


```
In [3]: import pandas as pd
        import matplotlib.pyplot as plt
        df = pd.read_csv(r'C:\Users\ACER\Desktop\stats.csv')
        print(df)
```

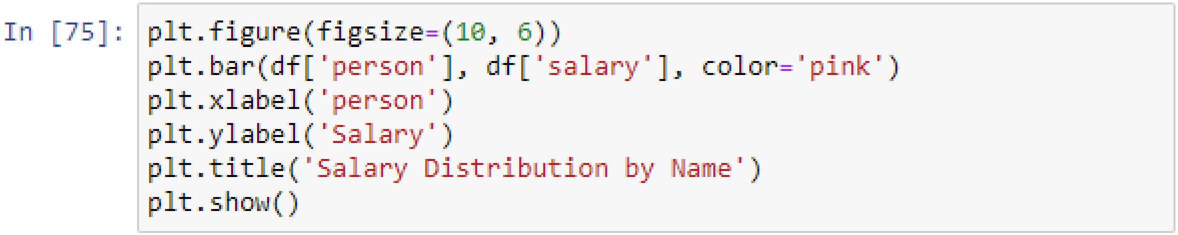
```
      person  salary country
0    Devyani   40000     USA
1   Pranjali   32000  Brazil
2     Riddhi   45000   Italy
3    Darshan   54000     USA
4      Nayan   72000     USA
5     Mayank   62000  Brazil
6      Rishi   92000   Italy
7     Preeti   55000     USA
8   Vaishali   35000   Italy
9       tina   48000  Brazil
```



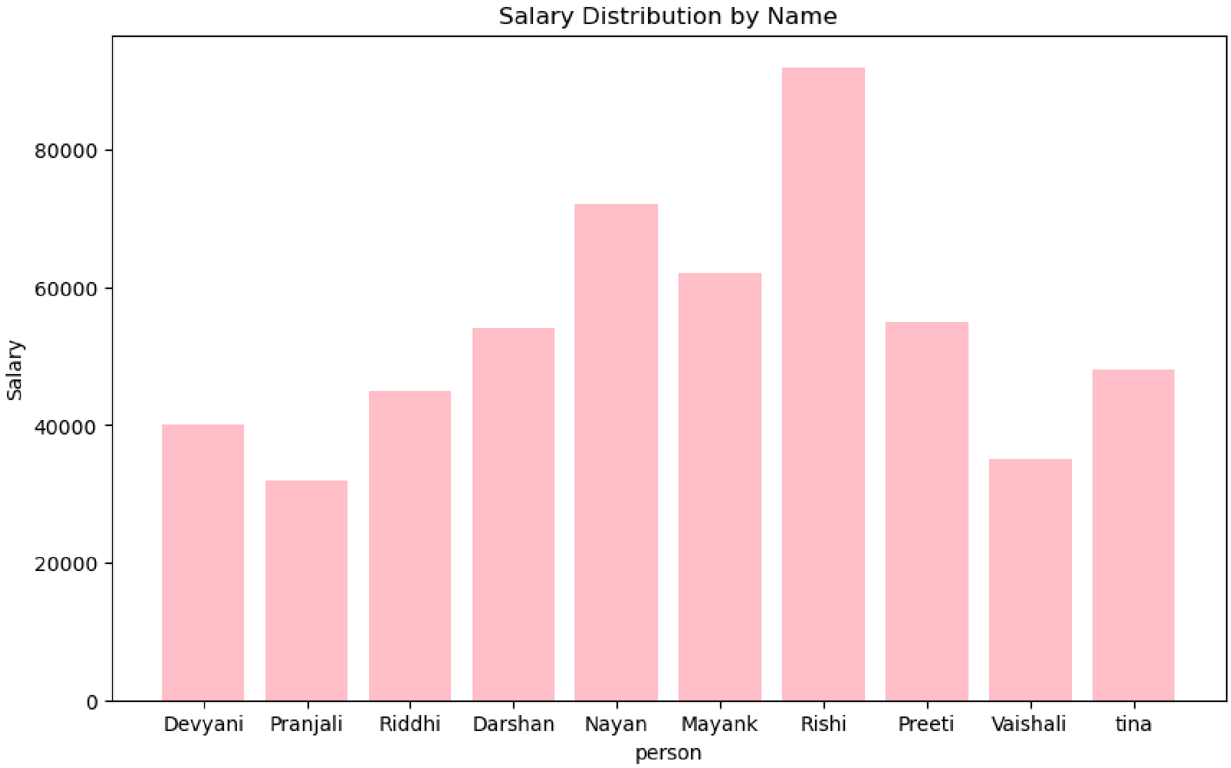
```
In [12]: mean_salary=df['salary'].mean()
         median_salary=df['salary'].median()
         std_deviation=df['salary'].std()
```



```
53500.0
51000.0
18222.391598128816
```



```
In [75]: plt.figure(figsize=(10, 6))
         plt.bar(df['person'], df['salary'], color='pink')
         plt.xlabel('person')
         plt.ylabel('Salary')
         plt.title('Salary Distribution by Name')
         plt.show()
```



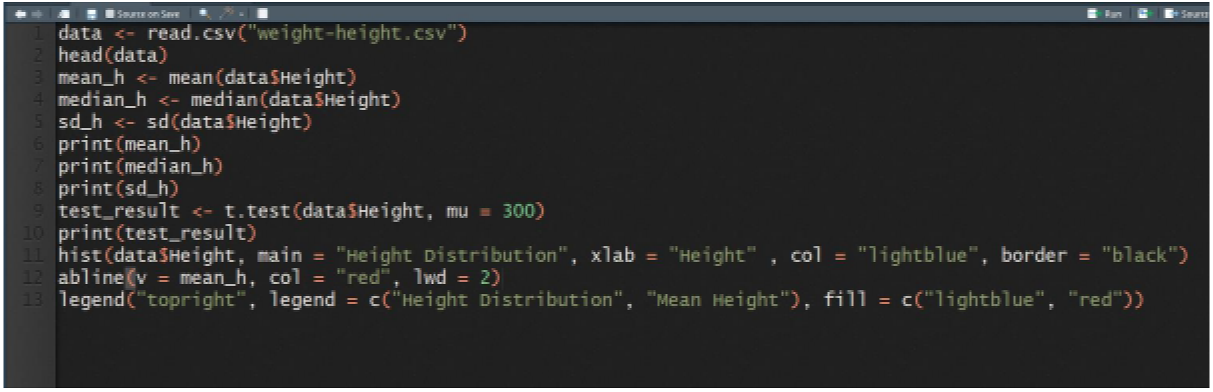
Conclusions:

Based on the bar charts, it seems that Rishi has the highest salary among all.

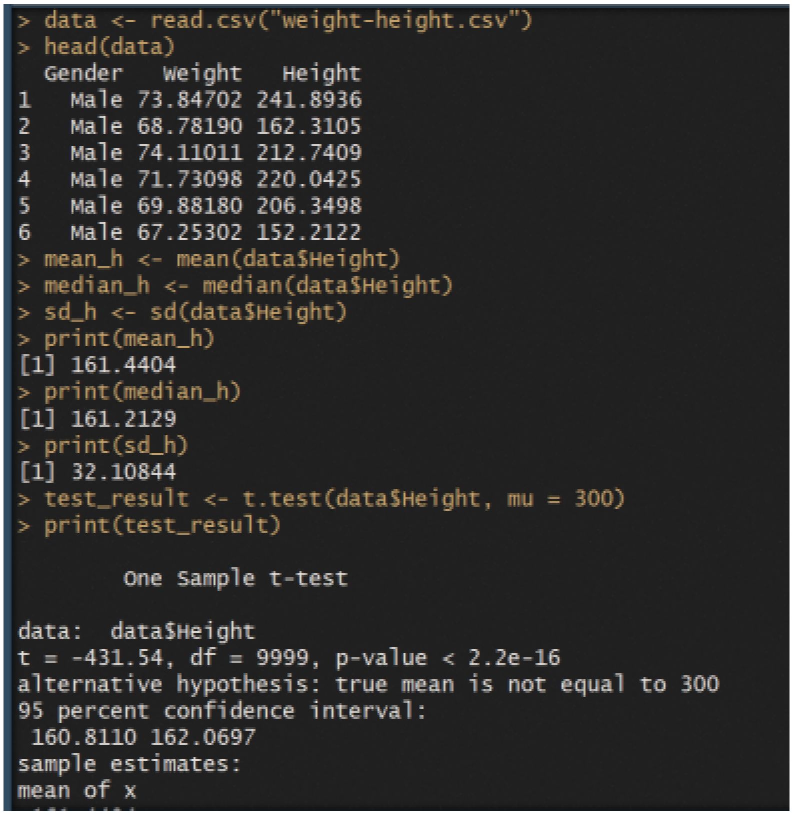
## 2. Statistical Analysis with R:

• Objective: Perform statistical analysis on a dataset using R's built-in statistical functions.

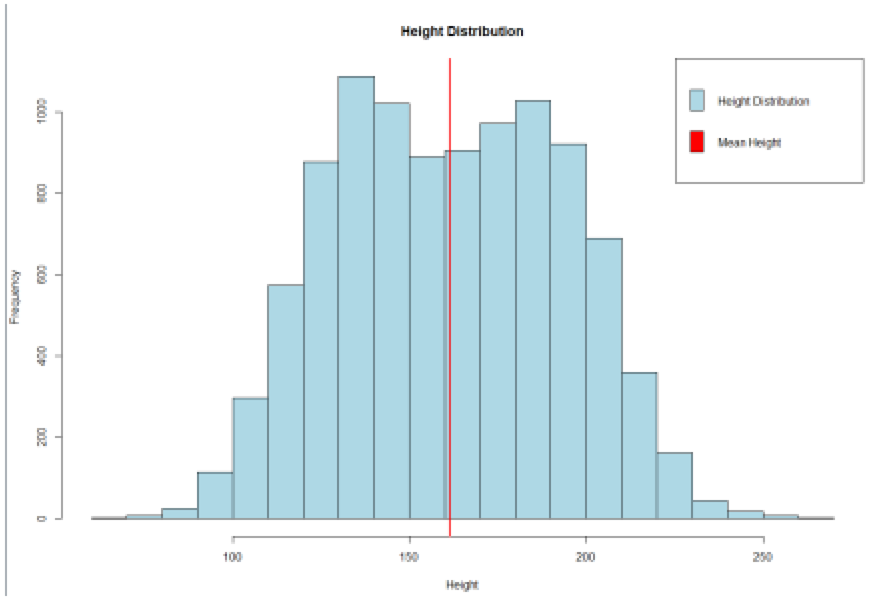
• Requirements: Choose a dataset suitable for statistical analysis (e.g., survey data, experiment results). Calculate descriptive statistics (mean, median, standard deviation) for relevant variables. Conduct hypothesis testing or create confidence intervals for specific hypotheses. Visualize the results using appropriate plots (e.g., histograms, violin plots). Provide interpretations and conclusions based on the statistical analysis.



```r
data <- read.csv("weight-height.csv")
head(data)
mean_h <- mean(data$Height)
median_h <- median(data$Height)
sd_h <- sd(data$Height)
print(mean_h)
print(median_h)
print(sd_h)
test_result <- t.test(data$Height, mu = 300)
print(test_result)
hist(data$Height, main = "Height Distribution", xlab = "Height" , col = "lightblue", border = "black")
abline(v = mean_h, col = "red", lwd = 2)
legend("topright", legend = c("Height Distribution", "Mean Height"), fill = c("lightblue", "red"))
```



```
> data <- read.csv("weight-height.csv")
> head(data)
  Gender   Weight   Height
1   Male 73.84702 241.8936
2   Male 68.78190 162.3105
3   Male 74.11011 212.7409
4   Male 71.73098 220.0425
5   Male 69.88180 206.3498
6   Male 67.25302 152.2122
> mean_h <- mean(data$Height)
> median_h <- median(data$Height)
> sd_h <- sd(data$Height)
> print(mean_h)
[1] 161.4404
> print(median_h)
[1] 161.2129
> print(sd_h)
[1] 32.10844
> test_result <- t.test(data$Height, mu = 300)
> print(test_result)

	One Sample t-test

data:  data$Height
t = -431.54, df = 9999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 300
95 percent confidence interval:
 160.8110 162.0697
sample estimates:
mean of x
```



• Conclusion: o The mean height of the individuals in the dataset is 161.4404 cm. o The median height is 161.2126 cm, indicating the central tendency. o The standard deviation of height is 32.108 cm, reflecting the spread of the data.

**3. Title: Data Analysis with Pandas and NumPy.**

**Problem Statement:**

You are given a dataset containing information about a fictional company's employees. The dataset (employee_data.csv) has the following columns: Employee_ID: Unique identifier for each employee. First_Name: First name of the employee. Last_Name: Last name of the employee. Department: Department in which the employee works. Salary: Salary of the employee. Joining_Date: Date when the employee joined the company.

**• Tasks:**

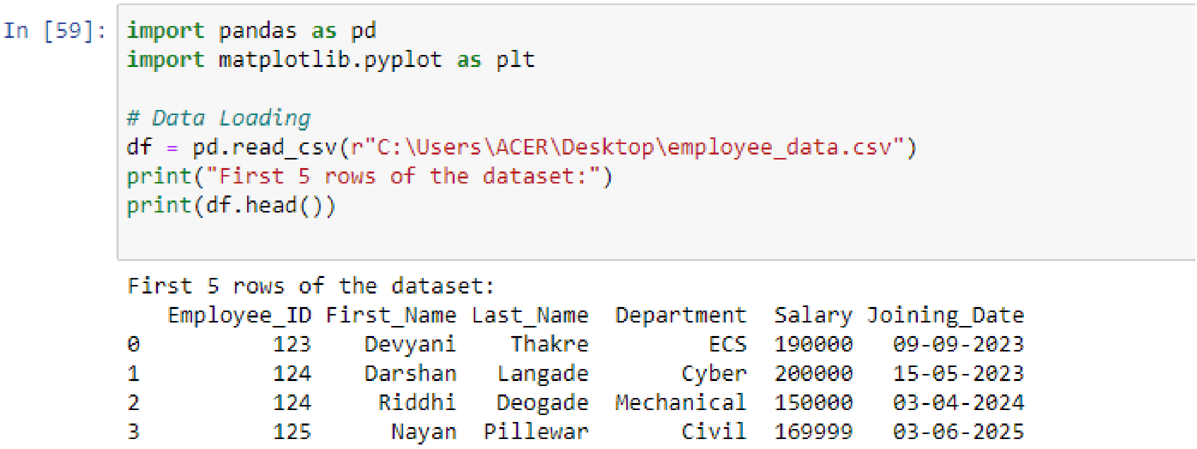
Data Loading: Load the dataset (employee_data.csv) into a Pandas DataFrame. Display the first 5 rows to get an overview of the data.

Data Cleaning: Check for and handle any missing values in the dataset. Convert the Joining_Date column to a datetime format.

Data Exploration: Calculate and display the average salary of employees in each department. Identify the employee with the highest salary and display their information.

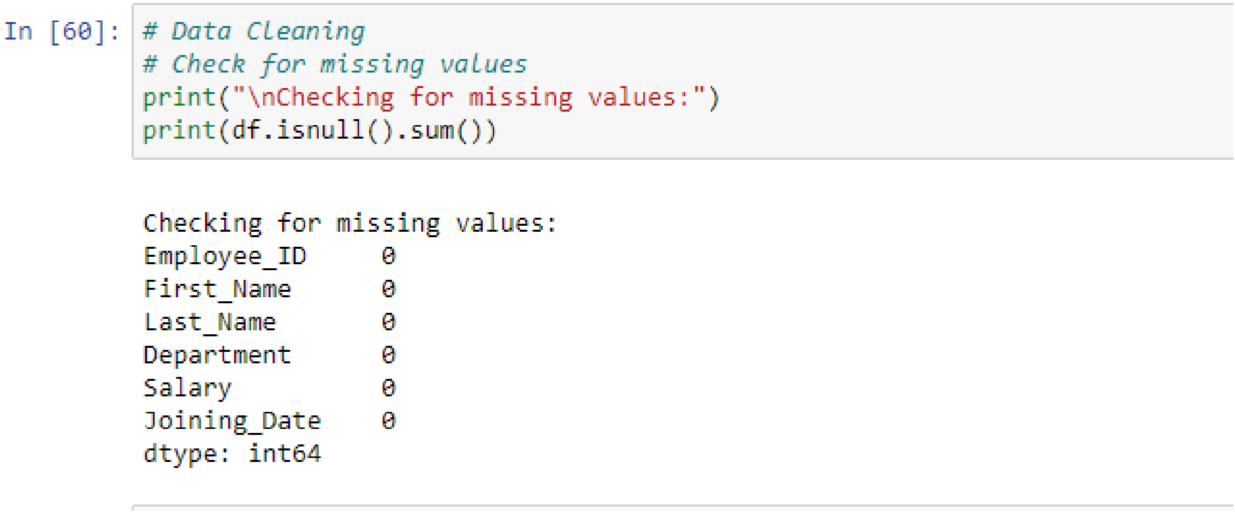
Time-based Analysis: Create a new column Years_Worked representing the number of years each employee has worked in the company. Calculate the average salary for employees based on the number of years they have worked (grouped by years).

Data Visualization: Use Matplotlib or Seaborn to create a bar chart showing the average salary for each department. Create a histogram of the distribution of employee salaries.



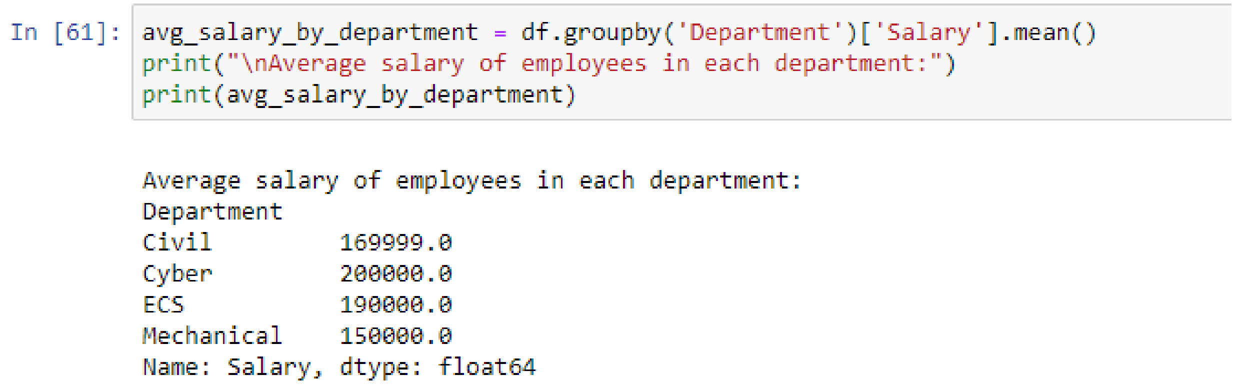
```
In [59]: import pandas as pd
         import matplotlib.pyplot as plt

         # Data Loading
         df = pd.read_csv(r"C:\Users\ACER\Desktop\employee_data.csv")
         print("First 5 rows of the dataset:")
         print(df.head())
```

```
First 5 rows of the dataset:
   Employee_ID First_Name Last_Name  Department  Salary Joining_Date
0          123    Devyani    Thakre         ECS  190000   09-09-2023
1          124    Darshan   Langade       Cyber  200000   15-05-2023
2          124     Riddhi   Deogade  Mechanical  150000   03-04-2024
3          125      Nayan  Pillewar       Civil  169999   03-06-2025
```



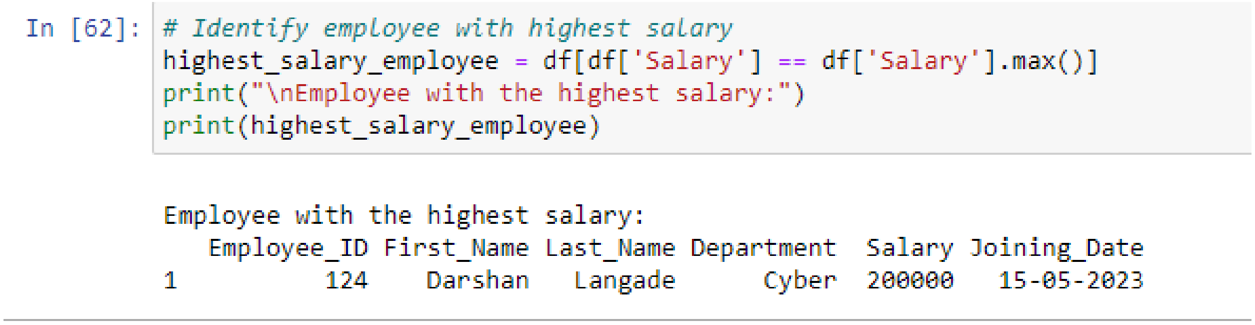
```
In [60]: # Data Cleaning
         # Check for missing values
         print("\nChecking for missing values:")
         print(df.isnull().sum())
```

```
Checking for missing values:
Employee_ID     0
First_Name      0
Last_Name       0
Department      0
Salary          0
Joining_Date    0
dtype: int64
```



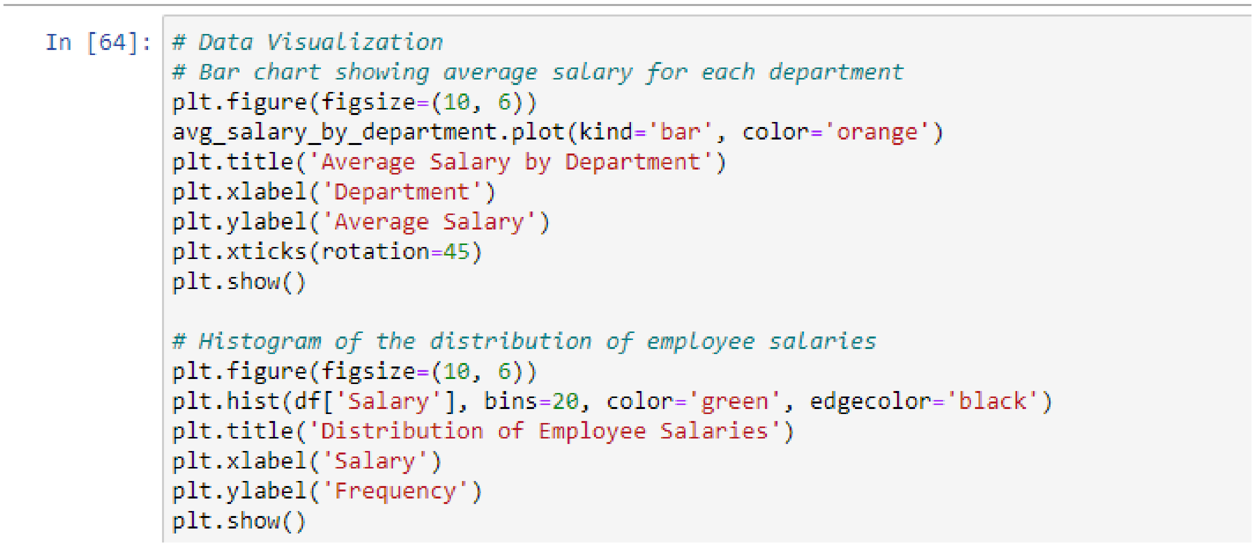
```
In [61]: avg_salary_by_department = df.groupby('Department')['Salary'].mean()
         print("\nAverage salary of employees in each department:")
         print(avg_salary_by_department)
```

```

Average salary of employees in each department:
Department
Civil         169999.0
Cyber         200000.0
ECS           190000.0
Mechanical    150000.0
Name: Salary, dtype: float64
```

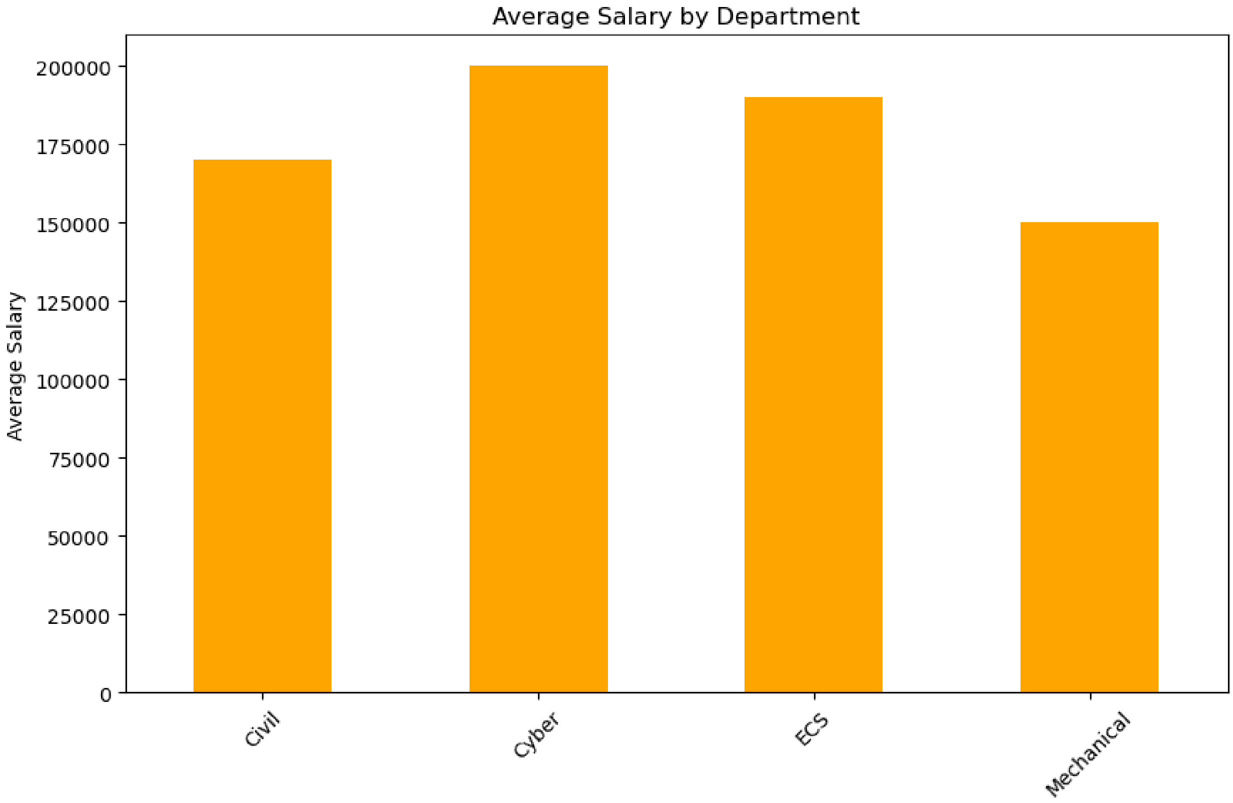


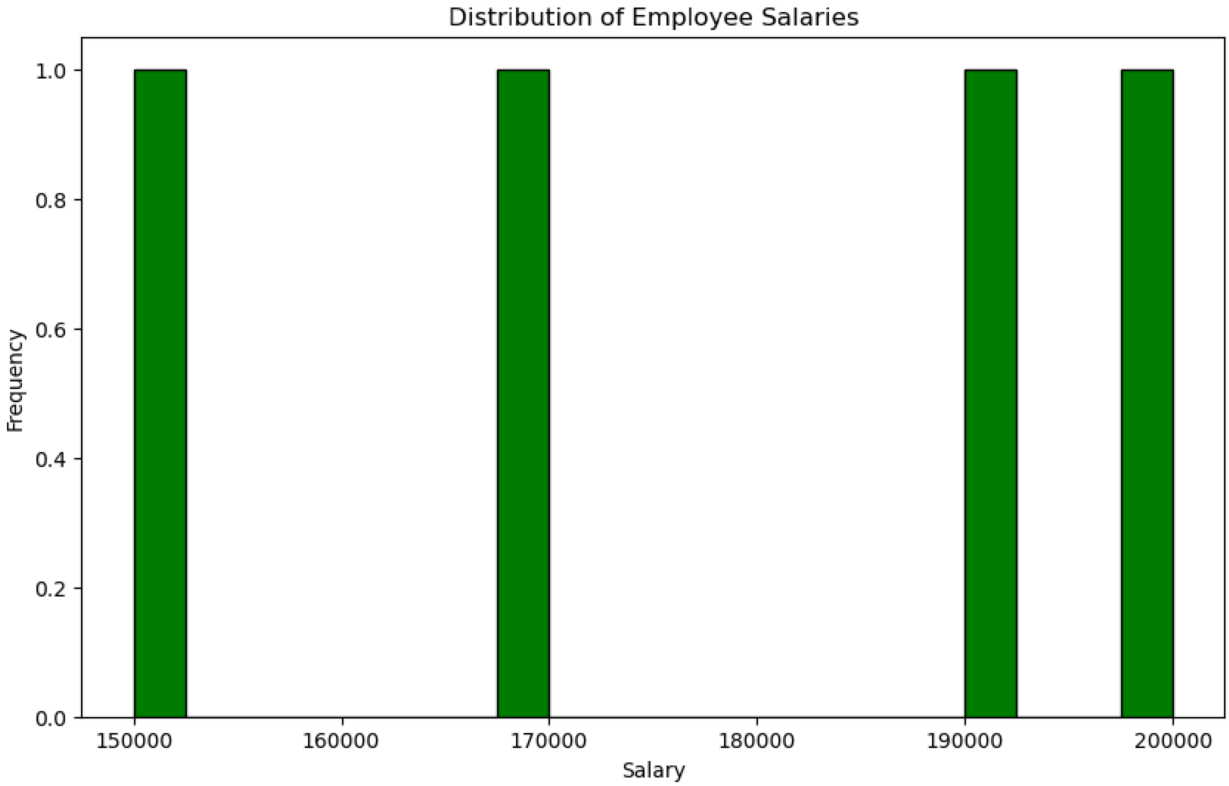
```
In [62]: # Identify employee with highest salary
         highest_salary_employee = df[df['Salary'] == df['Salary'].max()]
         print("\nEmployee with the highest salary:")
         print(highest_salary_employee)
```

```

Employee with the highest salary:
   Employee_ID First_Name Last_Name Department  Salary Joining_Date
1          124    Darshan    Langade      Cyber  200000   15-05-2023
```



```
In [64]: # Data Visualization
         # Bar chart showing average salary for each department
         plt.figure(figsize=(10, 6))
         avg_salary_by_department.plot(kind='bar', color='orange')
         plt.title('Average Salary by Department')
         plt.xlabel('Department')
         plt.ylabel('Average Salary')
         plt.xticks(rotation=45)
         plt.show()

         # Histogram of the distribution of employee salaries
         plt.figure(figsize=(10, 6))
         plt.hist(df['Salary'], bins=20, color='green', edgecolor='black')
         plt.title('Distribution of Employee Salaries')
         plt.xlabel('Salary')
         plt.ylabel('Frequency')
         plt.show()
```


Average Salary by Department
Average Salary
200000
175000
150000
125000
100000
75000
50000
25000
0
Civil
Cyber
ECS
Mechanical
Department


Distribution of Employee Salaries
Frequency
1.0
0.8
0.6
0.4
0.2
0.0
150000
160000
170000
180000
190000
200000
Salary