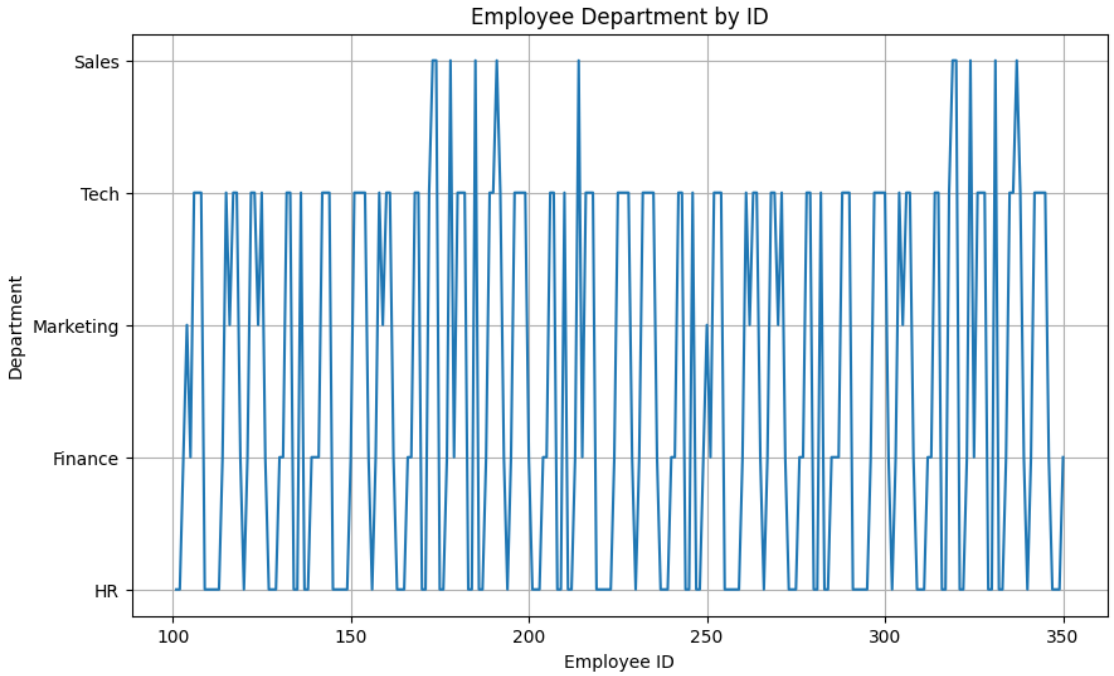
sagar-250-lab

September 21, 2023

[1]:
```
# Q1. Create a form using Tkinter on your Domain details
# Domain - Employee information system

import tkinter as tk

class EmployeeInformationForm:
    def __init__(self):
        self.root = tk.Tk()
        self.root.title("Employee Information System")

        # Creating labels and entry boxes for each field of employee information
        self.emp_id_label = tk.Label(self.root, text="Employee ID:")
        self.emp_id_entry = tk.Entry(self.root)

        self.first_name_label = tk.Label(self.root, text="First Name:")
        self.first_name_entry = tk.Entry(self.root)

        self.last_name_label = tk.Label(self.root, text="Last Name:")
        self.last_name_entry = tk.Entry(self.root)

        # Creating radio buttons for gender and department
        self.gender_label = tk.Label(self.root, text="Gender:")
        self.gender_variable = tk.StringVar(value="")
        self.male_radio_button = tk.Radiobutton(self.root, text="Male",␣
↪variable=self.gender_variable, value="Male")
        self.female_radio_button = tk.Radiobutton(self.root, text="Female",␣
↪variable=self.gender_variable, value="Female")

        self.department_label = tk.Label(self.root, text="Department:")
        self.department_variable = tk.StringVar(value="")
        self.engineering_radio_button = tk.Radiobutton(self.root,␣
↪text="Engineering", variable=self.department_variable, value="Engineering")
        self.sales_radio_button = tk.Radiobutton(self.root, text="Sales",␣
↪variable=self.department_variable, value="Sales")
        self.marketing_radio_button = tk.Radiobutton(self.root,␣
↪text="Marketing", variable=self.department_variable, value="Marketing")
```

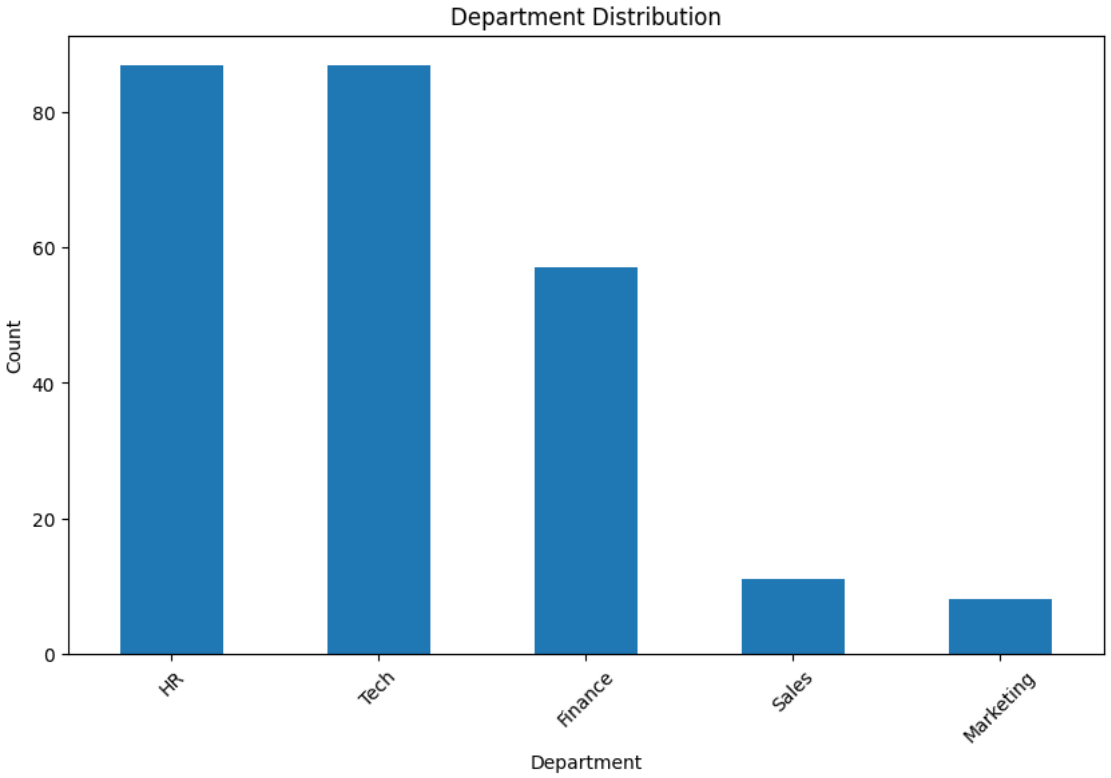
```
        # Adding the labels and entry boxes to the grid
        self.emp_id_label.grid(row=0, column=0)
        self.emp_id_entry.grid(row=0, column=1)

        self.first_name_label.grid(row=1, column=0)
        self.first_name_entry.grid(row=1, column=1)

        self.last_name_label.grid(row=2, column=0)
        self.last_name_entry.grid(row=2, column=1)

        # Adding the radio buttons to the grid
        self.gender_label.grid(row=3, column=0)
        self.male_radio_button.grid(row=3, column=1)
        self.female_radio_button.grid(row=3, column=2)

        self.department_label.grid(row=4, column=0)
        self.engineering_radio_button.grid(row=4, column=1)
        self.sales_radio_button.grid(row=4, column=2)
        self.marketing_radio_button.grid(row=4, column=3)

        # Adding a button to submit the form
        self.submit_button = tk.Button(self.root, text="Submit", command=self.
↪submit_form)
        self.submit_button.grid(row=5, column=1)

        # Starting the mainloop
        self.root.mainloop()

    def submit_form(self):
        # Getting the employee information from the entry boxes and radio␣
↪buttons
        emp_id = self.emp_id_entry.get()
        first_name = self.first_name_entry.get()
        last_name = self.last_name_entry.get()
        gender = self.gender_variable.get()
        department = self.department_variable.get()

        # Clearing the entry boxes and radio buttons after submit
        self.emp_id_entry.delete(0, tk.END)
        self.first_name_entry.delete(0, tk.END)
        self.last_name_entry.delete(0, tk.END)
        self.gender_variable.set("")
        self.department_variable.set("")

if __name__ == "__main__":
    EmployeeInformationForm()
```

[13]:
```
# Q2. Visualize your dataset using Matplotlib and write a program to create␣
↪line graph, bar graph, scatter plot and correlation graph or heat map.

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Loading the dataset
data = pd.read_csv("myDataset.csv")

# Line Graph
plt.figure(figsize=(10, 6))
plt.plot(data['Employee ID'], data['Department'])
plt.title('Employee Department by ID')
plt.xlabel('Employee ID')
plt.ylabel('Department')
plt.grid(True)
plt.show()
```



[11]:
```
#Bar graph

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Loading the dataset
data = pd.read_csv("myDataset.csv")

department_counts = data['Department'].value_counts()
plt.figure(figsize=(10, 6))
department_counts.plot(kind='bar')
plt.title('Department Distribution')
plt.xlabel('Department')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```

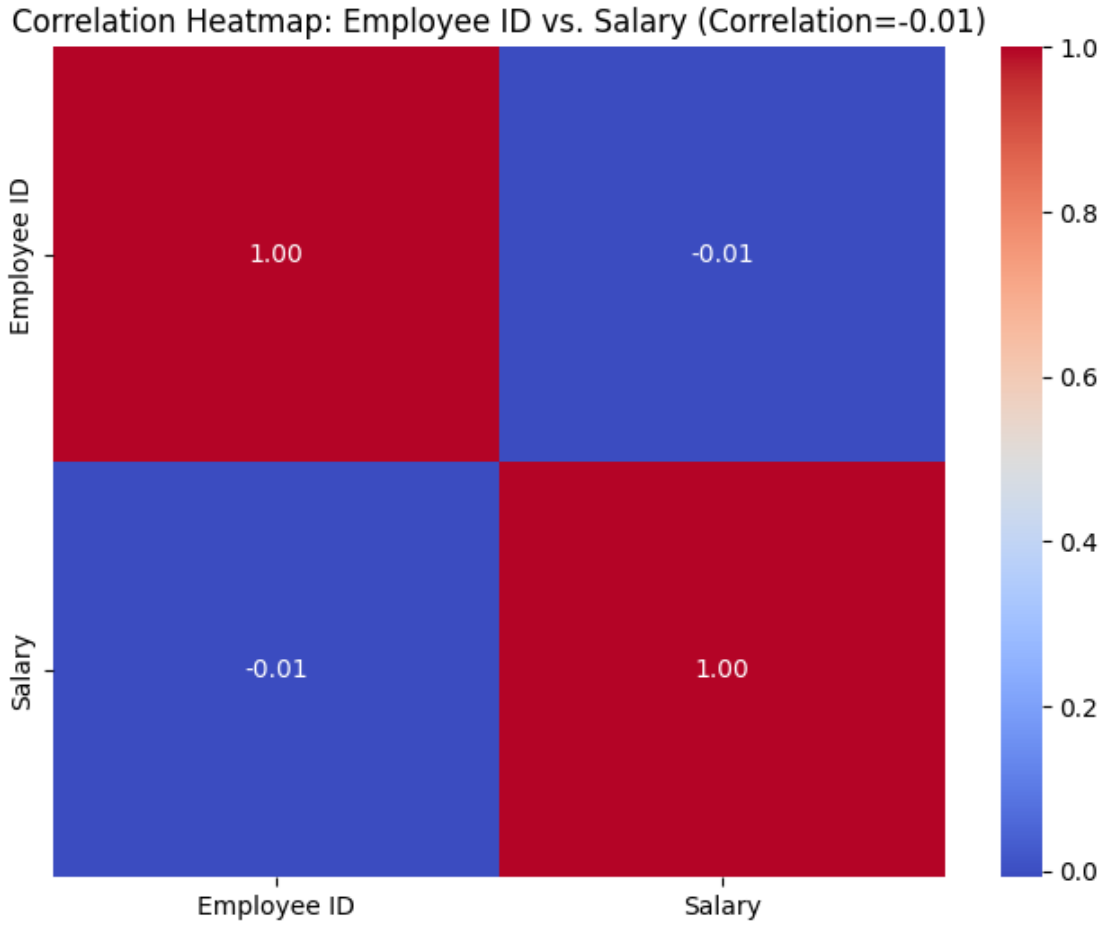


[12]:
```
#Scatter plot

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Loading the dataset
data = pd.read_csv("myDataset.csv")

plt.figure(figsize=(10, 6))
```

```
plt.scatter(data['Employee ID'], data['Department'])
plt.title('Scatter Plot of Employee ID vs. Department')
plt.xlabel('Employee ID')
plt.ylabel('Department')
plt.grid(True)
plt.show()
```

[16]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Loading the dataset (assuming it's in a CSV file)
data = pd.read_csv("myDataset.csv")

# Calculating the correlation between Employee ID and Salary
correlation = data['Employee ID'].corr(data['Salary'])

# Creating a correlation heatmap for Employee ID and Salary
correlation_matrix = data[['Employee ID', 'Salary']].corr()

# Creating the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title(f'Correlation Heatmap: Employee ID vs. Salary␣
 ↪(Correlation={correlation:.2f})')
```

```
plt.show()
```


Correlation Heatmap: Employee ID vs. Salary (Correlation=-0.01)
Employee ID
Salary
1.00
-0.01
-0.01
1.00
Employee ID
Salary
1.0
0.8
0.6
0.4
0.2
0.0