ssignment-tanviredu2018-gmail-com

November 16, 2023

```
[1]: import numpy  as np
     import pandas as pd
```

```
[2]: data = {
     'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
     'Age': [25, 30, 22, 35, 28],
     'Salary': [50000, 60000, 45000, 70000, 55000],
     'Department': ['HR', 'Finance', 'IT', 'Finance', 'IT']
     }

     df = pd.DataFrame(data)
```

```
[3]: df
```

```
[3]:       Name  Age  Salary Department
     0    Alice   25   50000         HR
     1      Bob   30   60000    Finance
     2  Charlie   22   45000         IT
     3    David   35   70000    Finance
     4      Eve   28   55000         IT
```

0.1 1. Sort the DataFrame df by the 'Name' column in ascending order.

```
[6]: df.sort_values(by="Name",ascending=True)
```

```
[6]:       Name  Age  Salary Department
     0    Alice   25   50000         HR
     1      Bob   30   60000    Finance
     2  Charlie   22   45000         IT
     3    David   35   70000    Finance
     4      Eve   28   55000         IT
```

0.2 2.Sort the DataFrame df by the 'Salary' column in descending order.

```
[7]: df.sort_values(by="Salary",ascending=False)
```

[7]:
```
      Name  Age  Salary Department
3    David   35   70000    Finance
1      Bob   30   60000    Finance
4      Eve   28   55000         IT
0    Alice   25   50000         HR
2  Charlie   22   45000         IT
```

0.3 3.Create a new DataFrame that contains only the rows where 'Age' is greater than 25.

[11]:
```
filter = df['Age'] > 25
df_new = df[filter]
df_new
```

[11]:
```
    Name  Age  Salary Department
1    Bob   30   60000    Finance
3  David   35   70000    Finance
4    Eve   28   55000         IT
```

0.4 4. Create a new DataFrame that contains only the rows where 'Department' is 'Finance'.

[12]:
```
df_new = df[df['Department'] == 'Finance']
df_new
```

[12]:
```
    Name  Age  Salary Department
1    Bob   30   60000    Finance
3  David   35   70000    Finance
```

0.5 5. Use the .where() method to create a new DataFrame where 'Salary' is greater than 55000, and replace the rest with NaN.

[13]:
```
df.where(cond=df['Salary']>55000,other=np.nan)
```

[13]:
```
    Name   Age   Salary Department
0    NaN   NaN      NaN        NaN
1    Bob  30.0  60000.0    Finance
2    NaN   NaN      NaN        NaN
3  David  35.0  70000.0    Finance
4    NaN   NaN      NaN        NaN
```

0.6 6. Use the .filter() method to filter the columns to include only 'Name' and 'Department'.

```
[15]: df_new = df.filter(items=['Name','Department'])
      df_new
```

```
[15]:       Name Department
      0    Alice         HR
      1      Bob    Finance
      2  Charlie         IT
      3    David    Finance
      4      Eve         IT
```

0.7 7 . Calculate the mean age of employees in the DataFrame.

```
[17]: mean_value = df['Age'].mean()
      mean_value
```

```
[17]: 28.0
```

0.8 8. Calculate the maximum salary in the DataFrame.

```
[18]: max_salary = df['Salary'].max()
      max_salary
```

```
[18]: 70000
```

0.9 9 . Create a DataFrame where any rows with missing values (NaN) in any column are removed.

```
[19]: data = {
          'Name': ['Alice', 'Bob', np.nan, 'David', 'Eve'],
          'Age': [25, 30, 22, 35, np.nan],
          'Salary': [50000, 60000, 45000, 70000, 55000],
          'Department': ['HR', 'Finance', 'IT', 'Finance', 'IT']
      }

      df = pd.DataFrame(data)
```

```
[20]: df
```

```
[20]:     Name   Age  Salary Department
      0  Alice  25.0   50000         HR
      1    Bob  30.0   60000    Finance
      2    NaN  22.0   45000         IT
      3  David  35.0   70000    Finance
```

```
4    Eve   NaN   55000          IT
```

[21]:

```
new_df = df.dropna(axis=0)
new_df
```

[21]:

```
    Name   Age  Salary Department
0  Alice  25.0   50000         HR
1    Bob  30.0   60000    Finance
3  David  35.0   70000    Finance
```

0.10 10. Fill the missing values in the 'Salary' column with the mean salary of the remaining employees.

[22]:

```
data = {
    'Name': ['Alice', 'Bob', "Robin", 'David', 'Eve'],
    'Age': [25, 30, 22, 35, 60],
    'Salary': [50000, 60000, np.nan, 70000, 55000],
    'Department': ['HR', 'Finance', 'IT', 'Finance', 'IT']
}

df = pd.DataFrame(data)
df
```

[22]:

```
    Name  Age   Salary Department
0  Alice   25  50000.0         HR
1    Bob   30  60000.0    Finance
2  Robin   22      NaN         IT
3  David   35  70000.0    Finance
4    Eve   60  55000.0         IT
```

[24]:

```
mean_salary = df['Salary'].mean()
df['Salary'] = df['Salary'].fillna(mean_salary)
df
```

[24]:

```
    Name  Age   Salary Department
0  Alice   25  50000.0         HR
1    Bob   30  60000.0    Finance
2  Robin   22  58750.0         IT
3  David   35  70000.0    Finance
4    Eve   60  55000.0         IT
```

[ ]: