

Lesson18_Assignment

(nahiyanhkhan@gmail.com)

Using the dataframe below, answer the following questions.

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

```
In [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)
df
```

Out[2]:

	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

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

```
In [3]: df.sort_values('Name')
```

Out[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

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

```
In [4]: df.sort_values('Salary', ascending=False)
```

Out[4]:

	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

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

```
In [5]: df_age25 = df[df['Age']>25]
df_age25
```

Out[5]:

	Name	Age	Salary	Department
1	Bob	30	60000	Finance
3	David	35	70000	Finance
4	Eve	28	55000	IT

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

```
In [6]: df_finance = df[df['Department']=='Finance']
df_finance
```

Out[6]:

	Name	Age	Salary	Department
1	Bob	30	60000	Finance
3	David	35	70000	Finance

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

```
In [7]: df_salary_55k = df.where(cond=df['Salary']>55000)
df_salary_55k
```

Out[7]:

	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

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

```
In [8]: df.filter(items=['Name', 'Department'])
```

Out[8]:

	Name	Department
0	Alice	HR
1	Bob	Finance
2	Charlie	IT
3	David	Finance
4	Eve	IT

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

```
In [9]: df['Age'].mean()
```

Out[9]: 28.0

8) Calculate the maximum salary in the DataFrame.

```
In [10]: df['Salary'].max()
```

Out[10]: 70000

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

```
In [11]: df.loc[len(df)] = ['Habib', np.nan, 44000, 'IT']  
df.loc[len(df)] = ['Sakib', 25, np.nan, 'HR']  
df
```

Out[11]:

	Name	Age	Salary	Department
0	Alice	25.0	50000.0	HR
1	Bob	30.0	60000.0	Finance
2	Charlie	22.0	45000.0	IT
3	David	35.0	70000.0	Finance
4	Eve	28.0	55000.0	IT
5	Habib	NaN	44000.0	IT
6	Sakib	25.0	NaN	HR

```
In [12]: df_no_nan = df.copy()
df_no_nan
```

Out[12]:

	Name	Age	Salary	Department
0	Alice	25.0	50000.0	HR
1	Bob	30.0	60000.0	Finance
2	Charlie	22.0	45000.0	IT
3	David	35.0	70000.0	Finance
4	Eve	28.0	55000.0	IT
5	Habib	NaN	44000.0	IT
6	Sakib	25.0	NaN	HR

```
In [13]: df_no_nan.dropna(inplace=True)
df_no_nan
```

Out[13]:

	Name	Age	Salary	Department
0	Alice	25.0	50000.0	HR
1	Bob	30.0	60000.0	Finance
2	Charlie	22.0	45000.0	IT
3	David	35.0	70000.0	Finance
4	Eve	28.0	55000.0	IT

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

```
In [14]: df
```

Out[14]:

	Name	Age	Salary	Department
0	Alice	25.0	50000.0	HR
1	Bob	30.0	60000.0	Finance
2	Charlie	22.0	45000.0	IT
3	David	35.0	70000.0	Finance
4	Eve	28.0	55000.0	IT
5	Habib	NaN	44000.0	IT
6	Sakib	25.0	NaN	HR

```
In [15]: values = {'Salary':df['Salary'].mean()}  
df.fillna(value=values, inplace=True)  
df
```

Out[15]:

	Name	Age	Salary	Department
0	Alice	25.0	50000.0	HR
1	Bob	30.0	60000.0	Finance
2	Charlie	22.0	45000.0	IT
3	David	35.0	70000.0	Finance
4	Eve	28.0	55000.0	IT
5	Habib	NaN	44000.0	IT
6	Sakib	25.0	54000.0	HR