Assignment Tasks

Using the dataframe below, answer the following questions.

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)

1. Sort the DataFrame df by the 'Name' column in ascending order.
2. Sort the DataFrame df by the 'Salary' column in descending order.
3. Create a new DataFrame that contains only the rows where 'Age' is greater than 25.
4. Create a new DataFrame that contains only the rows where 'Department' is 'Finance'.
5. Use the .where() method to create a new DataFrame where 'Salary' is greater than 55000, and replace the rest with NaN.
6. Use the .filter() method to filter the columns to include only 'Name' and 'Department'.
7. Calculate the mean age of employees in the DataFrame.
8. Calculate the maximum salary in the DataFrame.
9. Create a DataFrame where any rows with missing values (NaN) in any column are removed.
10. Fill the missing values in the 'Salary' column with the mean salary of the remaining employees.

Importing Libraries

```
import pandas as pd
import numpy as np
```

Creating DataFrame

```
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
```

	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. Sorting the DataFrame df by the 'Name' column in ascending order.

```
df.sort_values('Name') # By default ascending = True
```

	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. Sorting the DataFrame df by the 'Salary' column in descending order.

```
df.sort_values('Salary', ascending = False)
```

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

```
dAge = df[df['Age'] > 25]
dAge
```

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

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

```
dDept = df[df['Department'] == 'Finance']
dDept
```

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

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

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

	Name	Age	Salary	Department
0	Alice	25	NaN	HR
1	Bob	30	60000.0	Finance
2	Charlie	22	NaN	IT
3	David	35	70000.0	Finance
4	Eve	28	NaN	IT

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

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

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

7. Calculating the mean age of employees in the DataFrame.

```
df['Age'].mean()
```

```
28.0
```

8. Calculating the maximum salary in the DataFrame.

```
df['Salary'].max() # Skipping the NaN values
```

```
70000.0
```

```
df['Salary'].max(skipna = False)
```

```
nan
```

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

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

	Name	Age	Salary	Department
1	Bob	30	60000.0	Finance
3	David	35	70000.0	Finance

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

```
remaining_employees = df['Salary'].mean(skipna = True)
```

```
df.fillna(value = remaining_employees)
```

	Name	Age	Salary	Department
0	Alice	25	65000.0	HR
1	Bob	30	60000.0	Finance
2	Charlie	22	65000.0	IT
3	David	35	70000.0	Finance
4	Eve	28	65000.0	IT