In [1]:
```python
#Data Wrangling
import pandas as pd
import numpy as np

# Sample data creation with specified names
data = {
    'Name': ['Dharun', 'Shriya', 'Devil', 'Wizard', 'Warrior'],
    'Age': [25, np.nan, 30, 35, np.nan],
    'City': ['New York', 'Los Angeles', 'Chicago', 'New York', 'Los Angeles'],
    'Salary': [70000, 80000, 120000, 95000, np.nan]
}

# Create a DataFrame
df = pd.DataFrame(data)
print("Initial DataFrame:\n", df)
```

```
Initial DataFrame:
       Name   Age         City    Salary
0   Dharun  25.0     New York   70000.0
1   Shriya   NaN  Los Angeles   80000.0
2    Devil  30.0      Chicago  120000.0
3   Wizard  35.0     New York   95000.0
4  Warrior   NaN  Los Angeles       NaN
```

In [2]:
```python
# Handling missing values
# Fill missing values in 'Age' with the mean age
mean_age = df['Age'].mean()
df['Age'].fillna(mean_age, inplace=True)
print("\nDataFrame after Filling Missing Ages:\n", df)
```

```
DataFrame after Filling Missing Ages:
       Name   Age         City    Salary
0   Dharun  25.0     New York   70000.0
1   Shriya  30.0  Los Angeles   80000.0
2    Devil  30.0      Chicago  120000.0
3   Wizard  35.0     New York   95000.0
4  Warrior  30.0  Los Angeles       NaN
```

In [3]:
```
# Fill missing values in 'Salary' with the median salary
median_salary = df['Salary'].median()
df['Salary'].fillna(median_salary, inplace=True)
print("\nDataFrame after Filling Missing Salaries:\n", df)
```

```
DataFrame after Filling Missing Salaries:
       Name   Age         City    Salary
0   Dharun  25.0     New York   70000.0
1   Shriya  30.0  Los Angeles   80000.0
2    Devil  30.0      Chicago  120000.0
3   Wizard  35.0     New York   95000.0
4  Warrior  30.0  Los Angeles   87500.0
```

In [4]:
```
# Filtering data: Select only rows where Age is greater than 30
filtered_df = df[df['Age'] > 30]
print("\nFiltered DataFrame (Age > 30):\n", filtered_df)
```

```
Filtered DataFrame (Age > 30):
      Name   Age      City   Salary
3  Wizard  35.0  New York  95000.0
```

In [5]:
```
# Adding a new column: Calculate a bonus (10% of Salary)
df['Bonus'] = df['Salary'] * 0.10
print("\nDataFrame after Adding Bonus Column:\n", df)
```

```
DataFrame after Adding Bonus Column:
       Name   Age         City    Salary    Bonus
0   Dharun  25.0     New York   70000.0   7000.0
1   Shriya  30.0  Los Angeles   80000.0   8000.0
2    Devil  30.0      Chicago  120000.0  12000.0
3   Wizard  35.0     New York   95000.0   9500.0
4  Warrior  30.0  Los Angeles   87500.0   8750.0
```

In [6]:
```python
# Grouping data by 'City' and calculating average salary
grouped_df = df.groupby('City')['Salary'].mean().reset_index()
print("\nAverage Salary by City:\n", grouped_df)
```

```
Average Salary by City:
          City    Salary
0      Chicago  120000.0
1  Los Angeles   83750.0
2     New York   82500.0
```

In [7]:
```python
# Reshaping data: Pivoting the DataFrame to see average salaries by City
pivot_df = df.pivot_table(values='Salary', index='City', aggfunc='mean')
print("\nPivot Table of Average Salary by City:\n", pivot_df)
```

```
Pivot Table of Average Salary by City:
                Salary
City
Chicago      120000.0
Los Angeles   83750.0
New York      82500.0
```

In [8]:
```python
# Sorting the DataFrame by Salary
sorted_df = df.sort_values(by='Salary', ascending=False)
print("\nDataFrame Sorted by Salary:\n", sorted_df)
```

```
DataFrame Sorted by Salary:
       Name   Age         City    Salary    Bonus
2    Devil  30.0      Chicago  120000.0  12000.0
3   Wizard  35.0     New York   95000.0   9500.0
4  Warrior  30.0  Los Angeles   87500.0   8750.0
1   Shriya  30.0  Los Angeles   80000.0   8000.0
0   Dharun  25.0     New York   70000.0   7000.0
```

In [9]:
```
# Dropping the 'Bonus' column
df.drop(columns=['Bonus'], inplace=True)
print("\nDataFrame after Dropping Bonus Column:\n", df)
```

```
DataFrame after Dropping Bonus Column:
       Name   Age         City    Salary
0   Dharun  25.0     New York   70000.0
1   Shriya  30.0  Los Angeles   80000.0
2    Devil  30.0      Chicago  120000.0
3   Wizard  35.0     New York   95000.0
4  Warrior  30.0  Los Angeles   87500.0
```

In [10]:
```
# Resetting the index
df.reset_index(drop=True, inplace=True)
print("\nDataFrame after Resetting Index:\n", df)
```

```
DataFrame after Resetting Index:
       Name   Age         City    Salary
0   Dharun  25.0     New York   70000.0
1   Shriya  30.0  Los Angeles   80000.0
2    Devil  30.0      Chicago  120000.0
3   Wizard  35.0     New York   95000.0
4  Warrior  30.0  Los Angeles   87500.0
```

In [ ]:
```
220901020 - DHARUN J
```