CSV/Excel Importing, DataFrames, and GroupBy Operations

```
import pandas as pd

# 1.1 Load Salaries.csv from URL

url = "https://raw.githubusercontent.com/Apress/data-analysis-and-visualization-using-python/master/Ch07/Salaries.csv"

data = pd.read_csv(url)

print("First few rows of Salaries dataset:\n", data.head())


# 1.2 Load Cars.xlsx (from local file)

cars = pd.read_excel(r"C:\Users\j.KISHORERAM\Desktop\Agile Tribe\Sprint 3\Cars.xlsx")


print("\n First few rows of Cars dataset:\n", cars.head())


# Head

print("\n Head of Cars DataFrame:\n", cars.head())


# Tail

print("\n  Tail of Cars DataFrame:\n", cars.tail())


#  Shape

print("\n Shape of Cars DataFrame:", cars.shape)


#  Describe

print("\n Descriptive Statistics of Cars:\n", cars.describe())
```

```
#  TASK 2 - Dictionary to DataFrame

#  Step 1: Create DataFrame
data_dict = {
    'Name': ['John', 'Jane', 'Babu', 'Peter', 'Leju'],
    'Age': [25, 30, 35, 40, 55],
    'City': ['New York', 'London', 'Paris', 'UK', 'Germany']
}
df = pd.DataFrame(data_dict)
print("\n Dictionary Data:\n", data_dict)

#  🔸 Step 2: Display with label
print("\n DataFrame Creation:\n", data_dict)

#  🔸 Step 3: Head
print("\n First 5 Rows of df:\n", df.head())

#  🔸 Step 4: Info
print("\n DataFrame Info:")
df.info()
```

```
#  TASK 3 - GroupBy Operations

# Using the Salaries data from Task 1

#  1. Group by Department, mean Salary
grouped = data.groupby("Department")["Salary"].mean()
print("\n Mean Salary by Department:\n", grouped)
```

Output:

```
 Department
 Administration    47871.853659
Engineering       77478.688889
Finance           63692.857143
Name: Salary, dtype: float64
```

```
#  2. Group by multiple columns with multiple aggregations
multi_group = data.groupby(["Department", "Age"]).agg({
    "Salary": ['mean', 'max'],
    "YearsOfExperience": 'sum'
})
print("\n Group by Dept & Age with Aggregations:\n", multi_group)

#  3. Filter groups with mean Salary > 6000**0**
```

```
filtered = grouped[grouped > 60000]
print("\n Departments with Mean Salary > 60,000:\n", filtered)

#  4. Custom aggregation: salary range per department
def salary_range(x):
    return x.max() - x.min()

custom_agg = data.groupby("Department")["Salary"].agg(salary_range)
print("\n Salary Range per Department:\n", custom_agg)
```