In [3]:
```
!pip install pandas seaborn
```

```
Requirement already satisfied: pandas in /home/sargam/.conda/envs/myenv/li
b/python3.11/site-packages (2.2.3)
Requirement already satisfied: seaborn in /home/sargam/.conda/envs/myenv/l
ib/python3.11/site-packages (0.13.2)
Requirement already satisfied: numpy>=1.23.2 in /home/sargam/.conda/envs/m
yenv/lib/python3.11/site-packages (from pandas) (2.0.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/sargam/.con
da/envs/myenv/lib/python3.11/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /home/sargam/.conda/envs/
myenv/lib/python3.11/site-packages (from pandas) (2025.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /home/sargam/.co
nda/envs/myenv/lib/python3.11/site-packages (from seaborn) (3.10.1)
Requirement already satisfied: contourpy>=1.0.1 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.4->seabor
n) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/sargam/.conda/en
vs/myenv/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.4->seabo
rn) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /home/sargam/.conda/en
vs/myenv/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.4->seabo
rn) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.4->seabor
n) (24.2)
Requirement already satisfied: pillow>=8 in /home/sargam/.conda/envs/myen
v/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1
1.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib!=3.6.1,>=3.4->seabor
n) (3.2.3)
Requirement already satisfied: six>=1.5 in /home/sargam/.conda/envs/myenv/
lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.17.
0)
```

In [2]:
```
import pandas as pd;
df=pd.read_csv("/home/sargam/Downloads/Mall_Customers.csv")
df
```

Out[2]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
...	...	...	...	...	...
195	196	Female	35	120	79
196	197	Female	45	126	28
197	198	Male	32	126	74
198	199	Male	32	137	18
199	200	Male	30	137	83

200 rows × 5 columns

In [4]:
```
# Group by Genre and get descriptive statistics
grouped_stats = df.groupby('Genre').agg({
    'Age': ['mean', 'median', 'min', 'max', 'std'],
    'Annual Income (k$)': ['mean', 'median', 'min', 'max', 'std'],
    'Spending Score (1-100)': ['mean', 'median', 'min', 'max', 'std']
})
```

In [5]:
```
# Display the result
print("Grouped Summary Statistics by Genre:")
print(grouped_stats)
```

```
Grouped Summary Statistics by Genre:
              Age                                    Annual Income (k$)
\
             mean median min max        std                mean median min
Genre
Female  38.098214   35.0  18  68  12.644095           59.250000   60.0  16
Male    39.806818   37.0  18  70  15.514812           62.227273   62.5  15

                        Spending Score (1-100)
        max        std                   mean median min max       std
Genre
Female  126  26.011952              51.526786   50.0   5  99  24.11495
Male    137  26.638373              48.511364   50.0   1  97  27.89677
```

In [6]:
```
# Create a list of numeric values for each genre (for each categorical re
genre_age_list = df.groupby('Genre')['Age'].apply(list)
genre_income_list = df.groupby('Genre')['Annual Income (k$)'].apply(list)
genre_spending_score_list = df.groupby('Genre')['Spending Score (1-100)']
```

In [7]:
```
# Display the lists for each response to the categorical variable
print("\nList of numeric values for 'Age' grouped by 'Genre':")
print(genre_age_list)
```

```
List of numeric values for 'Age' grouped by 'Genre':
Genre
Female    [20, 23, 31, 22, 35, 23, 30, 35, 58, 24, 35, 3...
Male      [19, 21, 64, 67, 37, 22, 20, 52, 35, 25, 31, 2...
Name: Age, dtype: object
```

In [8]:
```
print("\nList of numeric values for 'Annual Income' grouped by 'Genre':")
print(genre_income_list)
```

```
List of numeric values for 'Annual Income' grouped by 'Genre':
Genre
Female    [16, 16, 17, 17, 18, 18, 19, 19, 20, 20, 21, 2...
Male      [15, 15, 19, 19, 20, 20, 21, 23, 24, 24, 25, 2...
Name: Annual Income (k$), dtype: object
```

In [9]:
```
print("\nList of numeric values for 'Spending Score' grouped by 'Genre':"
print(genre_spending_score_list)
```

```
List of numeric values for 'Spending Score' grouped by 'Genre':
Genre
Female    [6, 77, 40, 76, 6, 94, 72, 99, 15, 77, 35, 98,...
Male      [39, 81, 3, 14, 13, 79, 66, 29, 35, 73, 73, 82...
Name: Spending Score (1-100), dtype: object
```

In [ ]:
```

```

In [ ]:
```

```