In [14]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sb
```

In [15]:

```
df = pd.read_csv("Mall_Customers.csv")
df.head()
```

Out[15]:

	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

In [16]:

```
df.mean(numeric_only=True)
```

Out[16]:

```
CustomerID                100.50
Age                        38.85
Annual Income (k$)         60.56
Spending Score (1-100)     50.20
dtype: float64
```

In [17]:

```
df.mode(numeric_only=True)
```

Out[17]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
0	1	32.0	54.0	42.0
1	2	NaN	78.0	NaN
2	3	NaN	NaN	NaN
3	4	NaN	NaN	NaN
4	5	NaN	NaN	NaN
...	...	...	...	...
195	196	NaN	NaN	NaN
196	197	NaN	NaN	NaN
197	198	NaN	NaN	NaN
198	199	NaN	NaN	NaN
199	200	NaN	NaN	NaN

200 rows × 4 columns

In [18]:

```
df.median(numeric_only=True)
```

Out[18]:

```
CustomerID                100.5
Age                        36.0
Annual Income (k$)         61.5
Spending Score (1-100)     50.0
dtype: float64
```

In [19]:

```
df.min(numeric_only=True)
```

Out[19]:

```
CustomerID                  1
Age                        18
Annual Income (k$)         15
Spending Score (1-100)      1
dtype: int64
```

In [20]:

```
df.max(numeric_only=True)
```

Out[20]:

```
CustomerID                200
Age                        70
Annual Income (k$)        137
Spending Score (1-100)     99
dtype: int64
```

In [21]:

```
df.std(numeric_only=True)
```

Out[21]:

```
CustomerID                57.879185
Age                       13.969007
Annual Income (k$)        26.264721
Spending Score (1-100)    25.823522
dtype: float64
```

In [22]:

```
grouped = df.groupby('Age')['Annual Income (k$)'].describe()
print(grouped)
```

```
     count        mean        std   min     25%    50%     75%    max
Age                                                                   
18     4.0   51.250000  14.056434  33.0   44.25   53.5   60.50   65.0
19     8.0   57.000000  20.632845  15.0   47.50   63.5   67.25   81.0
20     5.0   41.600000  24.815318  16.0   21.00   37.0   61.00   73.0
21     5.0   38.800000  19.018412  15.0   30.00   33.0   54.00   62.0
22     3.0   31.333333  22.278540  17.0   18.50   20.0   38.50   57.0
23     6.0   41.500000  23.441416  16.0   20.75   41.5   60.00   70.0
24     4.0   39.250000  16.357975  20.0   33.50   38.5   44.25   60.0
25     3.0   57.666667  29.263174  24.0   48.00   72.0   74.50   77.0
26     2.0   58.000000   5.656854  54.0   56.00   58.0   60.00   62.0
27     6.0   63.166667  18.400181  40.0   49.50   63.5   75.25   88.0
28     4.0   85.250000  11.615363  76.0   76.75   82.0   90.50  101.0
29     5.0   63.600000  28.866936  28.0   40.00   73.0   79.00   98.0
30     7.0   76.142857  39.612408  19.0   56.00   78.0   93.50  137.0
31     8.0   48.375000  23.323425  17.0   35.50   41.5   70.50   81.0
32    11.0   87.181818  26.809768  48.0   74.00   77.0  100.00  137.0
33     3.0   80.333333  35.837597  42.0   64.00   86.0   99.50  113.0
34     5.0   79.000000  15.968719  58.0   78.00   78.0   78.00  103.0
35     9.0   46.666667  38.632888  18.0   21.00   24.0   74.00  120.0
36     6.0   81.000000  22.556595  37.0   85.50   87.0   87.00  103.0
37     3.0   65.000000  40.112342  20.0   49.00   78.0   87.50   97.0
38     6.0   74.500000  20.462160  54.0   64.75   69.0   76.25  113.0
39     3.0   72.666667   4.725816  69.0   70.00   71.0   74.50   78.0
40     6.0   61.666667  19.551641  29.0   55.50   64.5   70.50   87.0
41     2.0  101.000000   2.828427  99.0  100.00  101.0  102.00  103.0
42     2.0   60.000000  36.769553  34.0   47.00   60.0   73.00   86.0
43     3.0   65.666667  15.695010  48.0   59.50   71.0   74.50   78.0
44     2.0   75.500000   3.535534  73.0   74.25   75.5   76.75   78.0
45     3.0   69.333333  50.767444  28.0   41.00   54.0   90.00  126.0
46     3.0   59.000000  36.755952  25.0   39.50   54.0   76.00   98.0
47     6.0   70.166667  27.694163  43.0   51.75   65.5   76.25  120.0
48     5.0   58.200000  13.700365  39.0   54.00   60.0   61.00   77.0
49     7.0   51.000000  12.884099  33.0   40.50   54.0   62.00   65.0
50     5.0   58.600000  18.420098  40.0   43.00   58.0   67.00   85.0
51     2.0   55.500000  16.263456  44.0   49.75   55.5   61.25   67.0
52     2.0   55.500000  45.961941  23.0   39.25   55.5   71.75   88.0
53     2.0   39.500000   9.192388  33.0   36.25   39.5   42.75   46.0
54     4.0   59.750000  30.998656  28.0   42.25   55.0   72.50  101.0
55     1.0   57.000000        NaN  57.0   57.00   57.0   57.00   57.0
56     1.0   79.000000        NaN  79.0   79.00   79.0   79.00   79.0
57     2.0   64.500000  14.849242  54.0   59.25   64.5   69.75   75.0
58     2.0   54.000000  48.083261  20.0   37.00   54.0   71.00   88.0
59     4.0   65.250000  21.792583  43.0   51.25   62.5   76.50   93.0
60     3.0   43.333333  11.547005  30.0   40.00   50.0   50.00   50.0
63     2.0   56.500000  12.020815  48.0   52.25   56.5   60.75   65.0
64     1.0   19.000000        NaN  19.0   19.00   19.0   19.00   19.0
65     2.0   50.500000  17.677670  38.0   44.25   50.5   56.75   63.0
66     2.0   63.000000   0.000000  63.0   63.00   63.0   63.00   63.0
67     4.0   45.500000  18.699376  19.0   40.00   50.5   56.00   62.0
68     3.0   56.666667   7.767453  48.0   53.50   59.0   61.00   63.0
69     1.0   44.000000        NaN  44.0   44.00   44.0   44.00   44.0
70     2.0   47.500000   2.121320  46.0   46.75   47.5   48.25   49.0
```

In [23]:

```
income_by_group = df.groupby('Age')['Annual Income (k$)'].apply(list)
print(income_by_group)
```

```
Age
18                                   [33, 48, 59, 65]
19                   [15, 46, 48, 63, 64, 65, 74, 81]
20                               [16, 21, 37, 61, 73]
21                               [15, 30, 33, 54, 62]
22                                       [17, 20, 57]
23                           [16, 18, 29, 54, 62, 70]
24                                   [20, 38, 39, 60]
25                                       [24, 72, 77]
26                                           [54, 62]
27                           [40, 46, 60, 67, 78, 88]
28                                  [76, 77, 87, 101]
29                               [28, 40, 73, 79, 98]
30                      [19, 34, 78, 78, 88, 99, 137]
31                   [17, 25, 39, 40, 43, 70, 72, 81]
32    [48, 60, 73, 75, 76, 77, 87, 97, 103, 126, 137]
33                                      [42, 86, 113]
34                              [58, 78, 78, 78, 103]
35              [18, 19, 21, 23, 24, 28, 74, 93, 120]
36                          [37, 85, 87, 87, 87, 103]
37                                       [20, 78, 97]
38                          [54, 64, 67, 71, 78, 113]
39                                       [69, 71, 78]
40                           [29, 54, 60, 69, 71, 87]
41                                          [99, 103]
42                                           [34, 86]
43                                       [48, 71, 78]
44                                           [73, 78]
45                                      [28, 54, 126]
46                                       [25, 54, 98]
47                          [43, 49, 60, 71, 78, 120]
48                               [39, 54, 60, 61, 77]
49                       [33, 39, 42, 54, 62, 62, 65]
50                               [40, 43, 58, 67, 85]
51                                           [44, 67]
52                                           [23, 88]
53                                           [33, 46]
54                                  [28, 47, 63, 101]
55                                               [57]
56                                               [79]
57                                           [54, 75]
58                                           [20, 88]
59                                   [43, 54, 71, 93]
60                                       [30, 50, 50]
63                                           [48, 65]
64                                               [19]
65                                           [38, 63]
66                                           [63, 63]
67                                   [19, 47, 54, 62]
68                                       [48, 59, 63]
69                                               [44]
70                                           [46, 49]
Name: Annual Income (k$), dtype: object
```