Aerofit Business Case

```
[ ]: import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
```

0.1 Defining Problem Statement

Aerofit is a company that sells high-end treadmills. Even though their products are of good quality, their sales are not consistent, and they don't clearly understand why some customers buy their products while others don't.

They want to find out what factors—like a person's age, income, gender, or how often they exercise—affect their decision to buy a treadmill. By studying customer data, Aerofit hopes to understand their buyers better, improve their marketing, and increase sales.

##Analysing basic matrics

```
[ ]: df = pd.read_csv('aerofit.csv')
     df.head()
```

```
[ ]:   Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  Miles
     0   KP281   18    Male         14        Single      3        4   29562    112
     1   KP281   19    Male         15        Single      2        3   31836     75
     2   KP281   19  Female         14     Partnered      4        3   30699     66
     3   KP281   19    Male         12        Single      3        3   32973     85
     4   KP281   20    Male         13     Partnered      4        2   35247     47
```

```
[ ]: df.shape
```

```
[ ]: (180, 9)
```

```
[ ]: df.describe()
```

```
[ ]:               Age   Education       Usage     Fitness         Income  \
     count  180.000000  180.000000  180.000000  180.000000     180.000000
     mean    28.788889   15.572222    3.455556    3.311111   53719.577778
     std      6.943498    1.617055    1.084797    0.958869   16506.684226
     min     18.000000   12.000000    2.000000    1.000000   29562.000000
     25%     24.000000   14.000000    3.000000    3.000000   44058.750000
     50%     26.000000   16.000000    3.000000    3.000000   50596.500000
```

```
75%      33.000000   16.000000    4.000000    4.000000   58668.000000
max      50.000000   21.000000    7.000000    5.000000  104581.000000

            Miles
count  180.000000
mean   103.194444
std     51.863605
min     21.000000
25%     66.000000
50%     94.000000
75%    114.750000
max    360.000000
```

[ ]:
```
df.dtypes
```

[ ]:
```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

[ ]:
```
df.columns
```

[ ]:
```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Product        180 non-null    object
 1   Age            180 non-null    int64
 2   Gender         180 non-null    object
 3   Education      180 non-null    int64
 4   MaritalStatus  180 non-null    object
 5   Usage          180 non-null    int64
 6   Fitness        180 non-null    int64
 7   Income         180 non-null    int64
```

```
 8   Miles          180 non-null    int64
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

[ ]:
```
#Check for missing values
df.isnull().sum()
```

[ ]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

[ ]:
```
df.head()
```

[ ]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

###Conversion of categorical data to categories

[ ]:
```
#We found Gender,MaritalStatus,Product as categorical data
df['Product'] = df['Product'].astype('category')
df['MaritalStatus'] = df['MaritalStatus'].astype('category')
df['Gender'] = df['Gender'].astype('category')
```

[ ]:
```
#Check if data got converted to categories
df.dtypes
```

[ ]:
```
Product          category
Age                 int64
Gender           category
Education           int64
MaritalStatus    category
Usage               int64
Fitness             int64
Income              int64
Miles               int64
dtype: object
```

##Non Graphical Analysis

[ ]:
```
df.value_counts()
```

[ ]:
```
Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  Miles
KP281    18   Male    14         Single         3      4        29562   112
1
         19   Female  14         Partnered      4      3        30699   66
1
              Male    12         Single         3      3        32973   85
1
                      15         Single         2      3        31836   75
1
         20   Female  14         Partnered      3      3        32973   66
1
..
KP781    40   Male    21         Single         6      5        83416   200
1
         42   Male    18         Single         5      4        89641   200
1
         45   Male    16         Single         5      5        90886   160
1
         47   Male    18         Partnered      4      5        104581  120
1
         48   Male    18         Partnered      4      5        95508   180
1
Name: count, Length: 180, dtype: int64
```

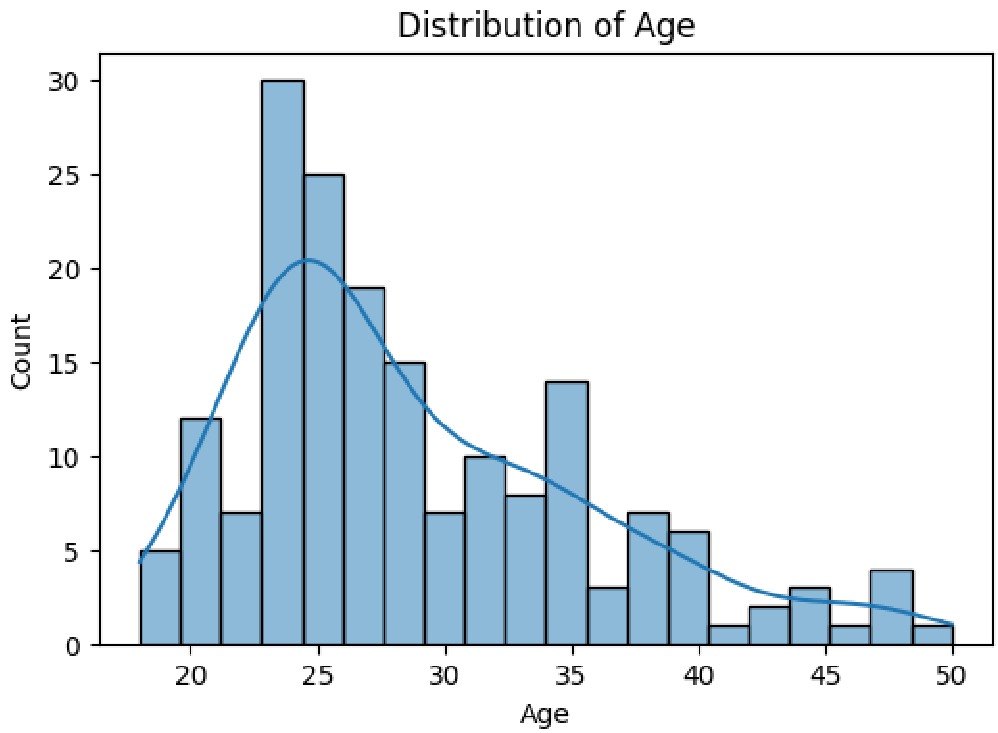
[ ]:
```
df.nunique()
```

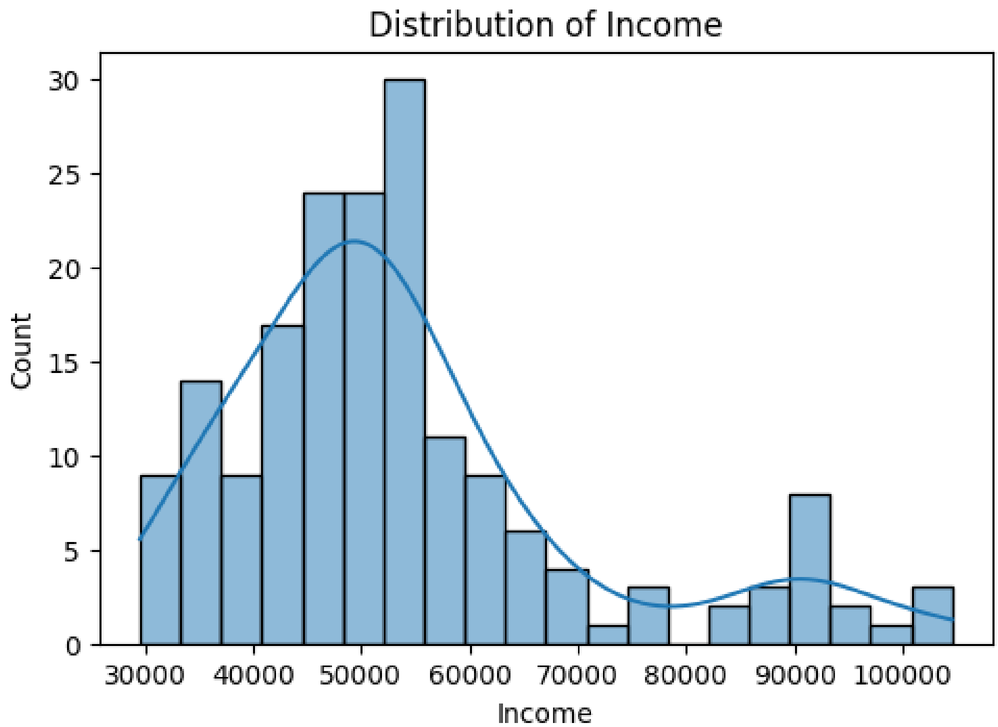
[ ]:
```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

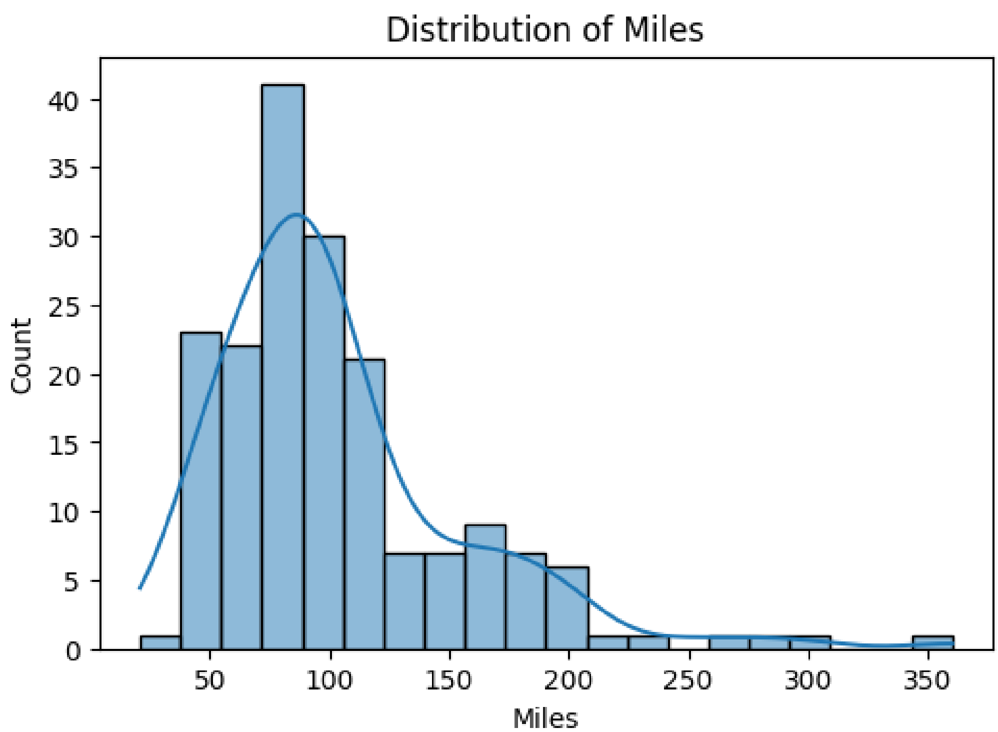
Univariate Analysis

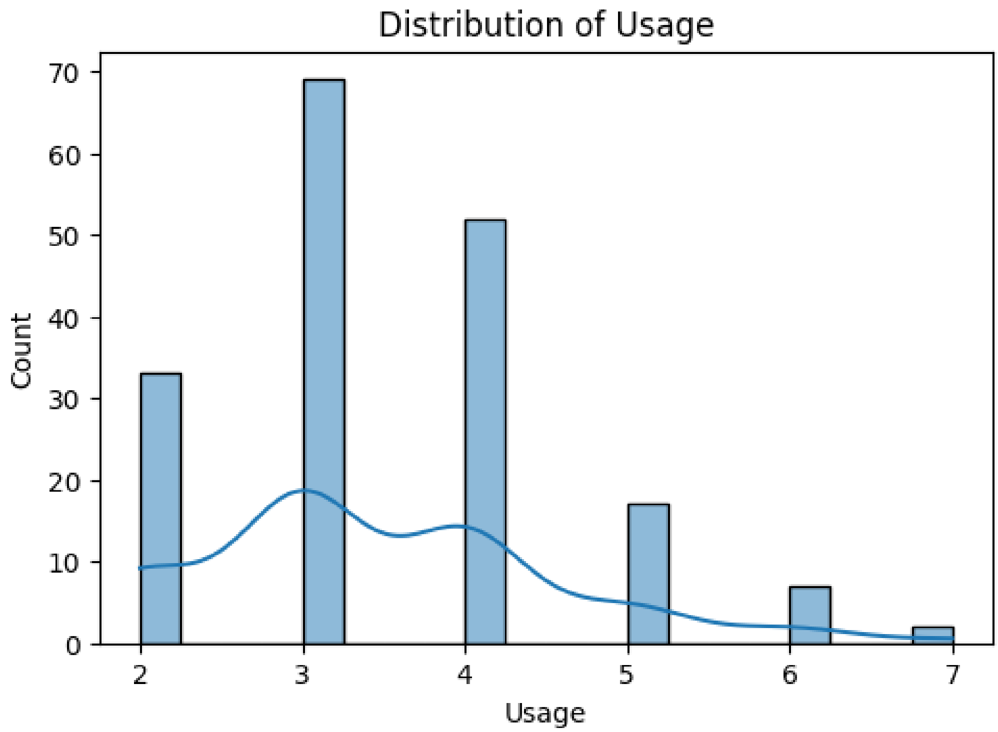
[ ]:
```
cont_cols = ['Age', 'Income', 'Miles', 'Usage', 'Fitness']

for col in cont_cols:
    plt.figure(figsize=(6, 4))
    sns.histplot(df[col], kde=True, bins=20)
```

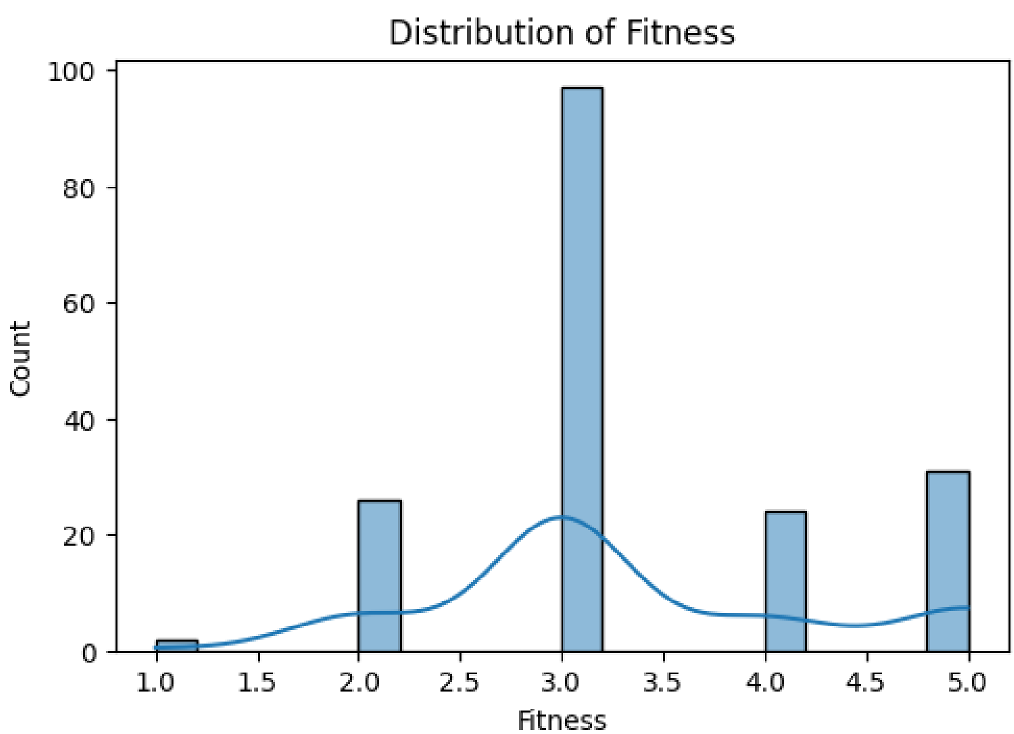
```
    plt.title(f'Distribution of {col}')
    plt.show()
```




Distribution of Income
Count
Income
0
5
10
15
20
25
30
30000
40000
50000
60000
70000
80000
90000
100000

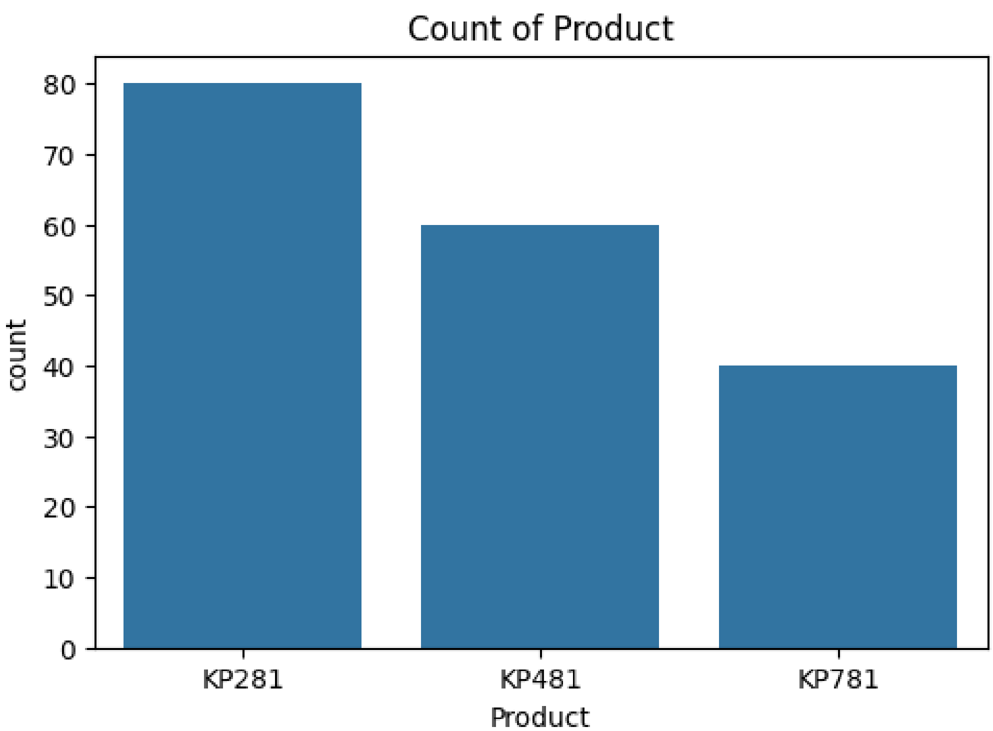

Distribution of Miles
Count
Miles
0
5
10
15
20
25
30
35
40
50
100
150
200
250
300
350

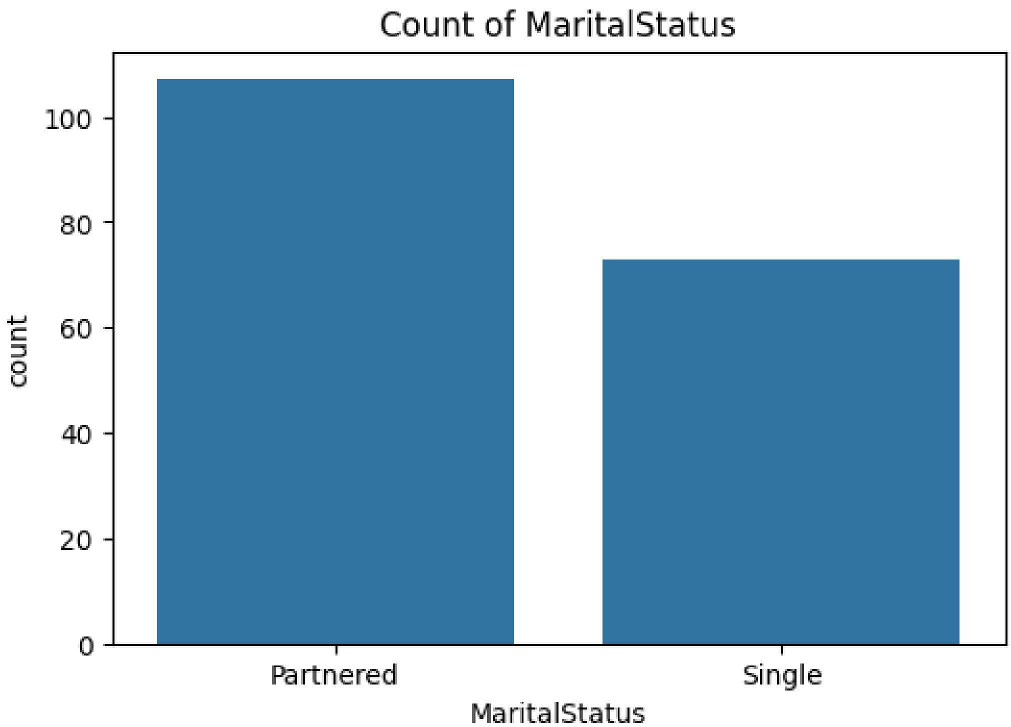

Distribution of Usage
Count
Usage
70
60
50
40
30
20
10
0
2
3
4
5
6
7

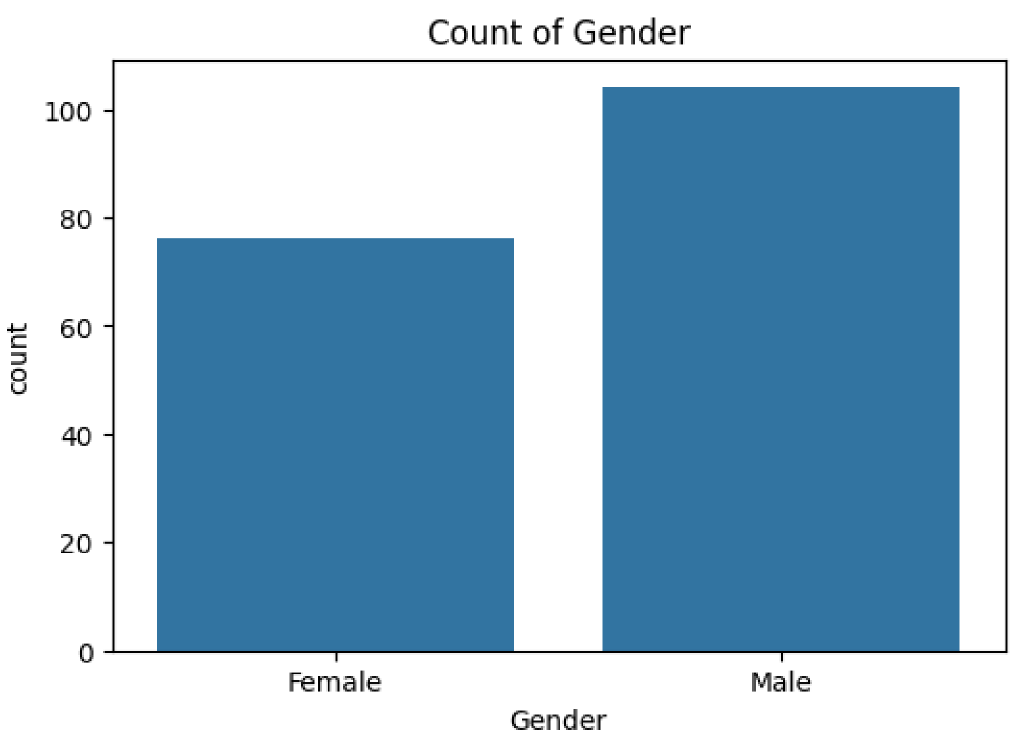

Distribution of Fitness
Count
Fitness
100
80
60
40
20
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

0.1.1 Count plots for categorical data

[ ]:
```
cat_cols = ['Product', 'MaritalStatus', 'Gender']

for col in cat_cols:
    plt.figure(figsize=(6, 4))
    sns.countplot(data=df, x=col)
    plt.title(f'Count of {col}')
    plt.show()
```

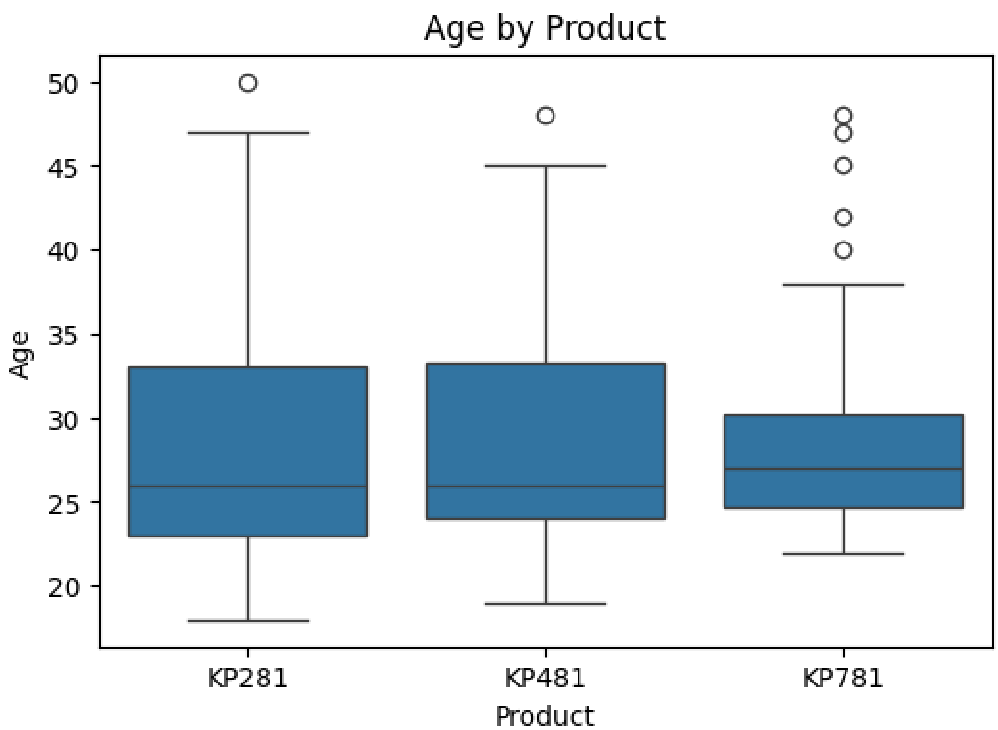


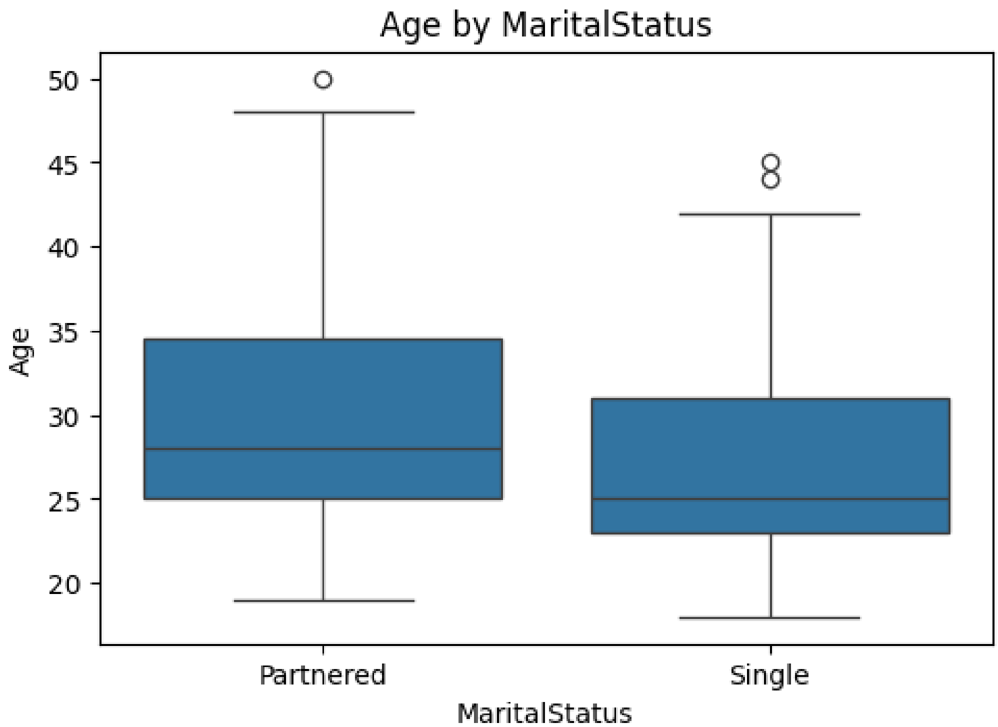

Count of MaritalStatus
count
100
80
60
40
20
0
Partnered
Single
MaritalStatus

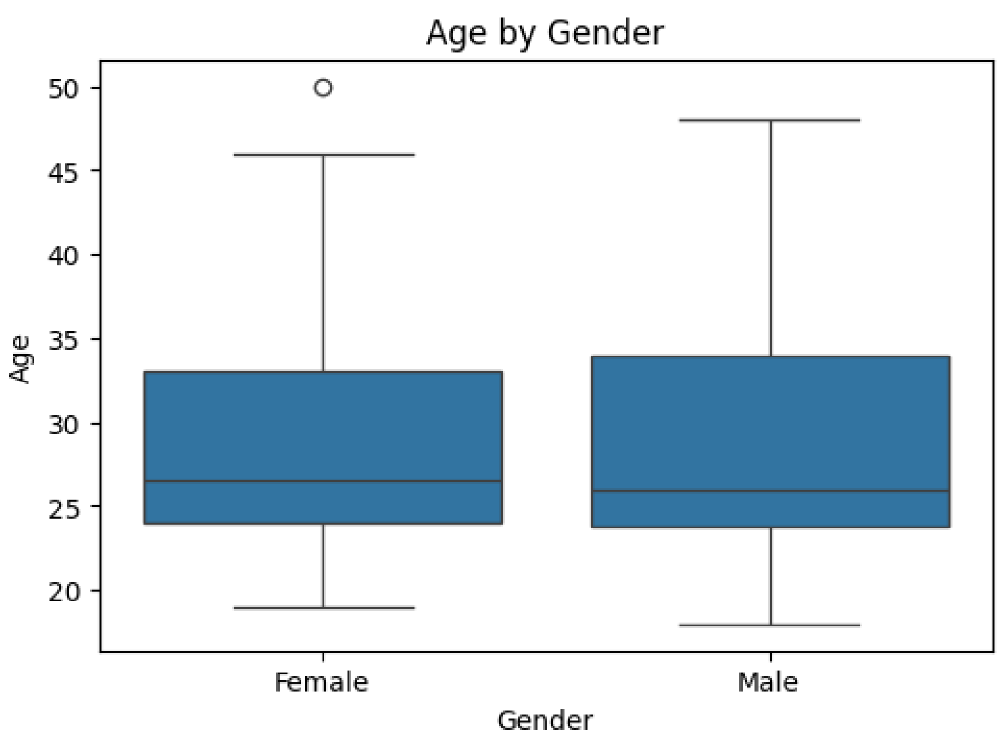

Count of Gender
count
100
80
60
40
20
0
Female
Male
Gender

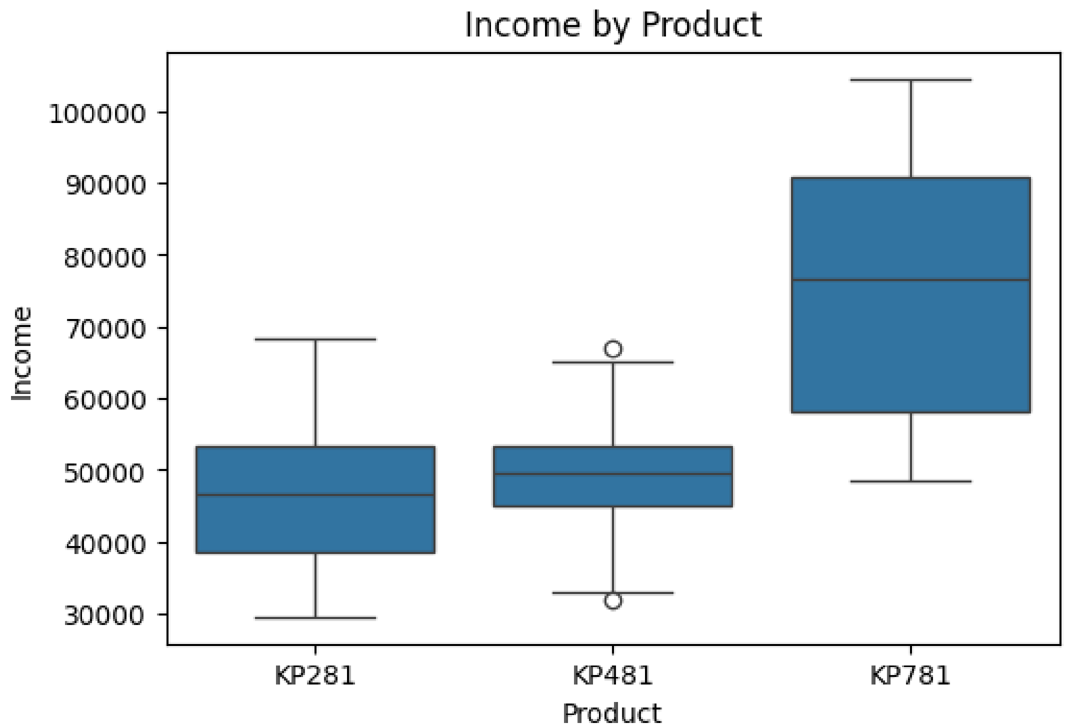
##Bivariate Analysis Boxplot for Continuous vs Categorical

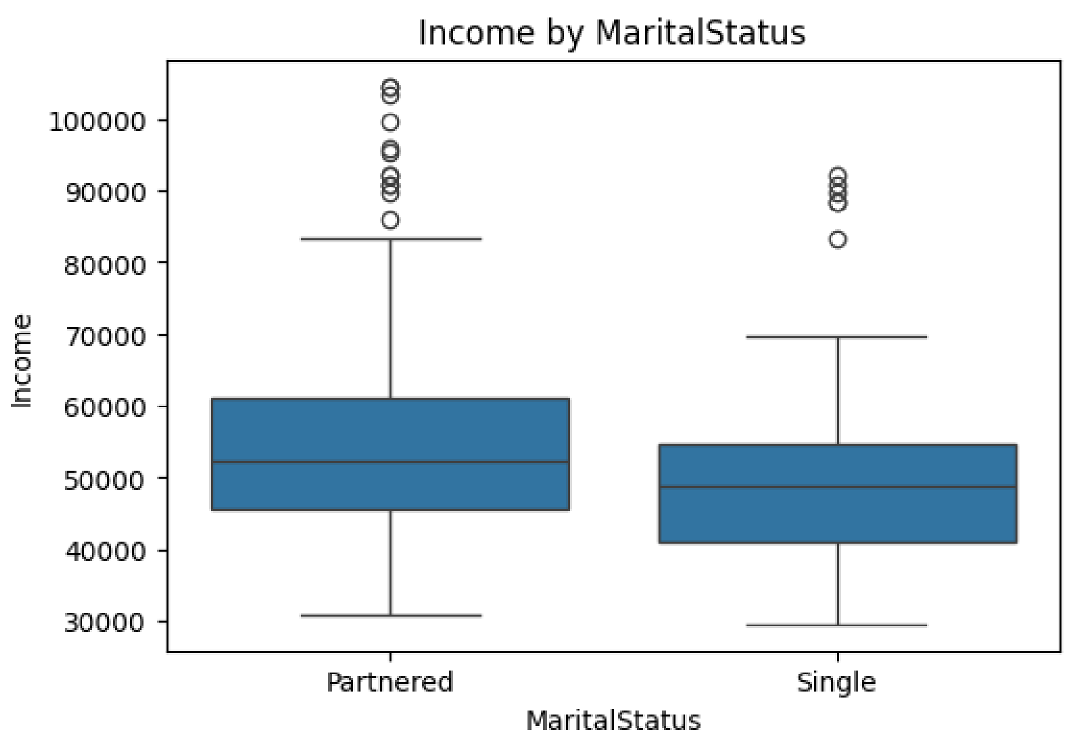
[ ]:
```
for col in cont_cols:
    for cat in cat_cols:
        plt.figure(figsize=(6, 4))
        sns.boxplot(data=df, x=cat, y=col)
        plt.title(f'{col} by {cat}')
        plt.show()
```

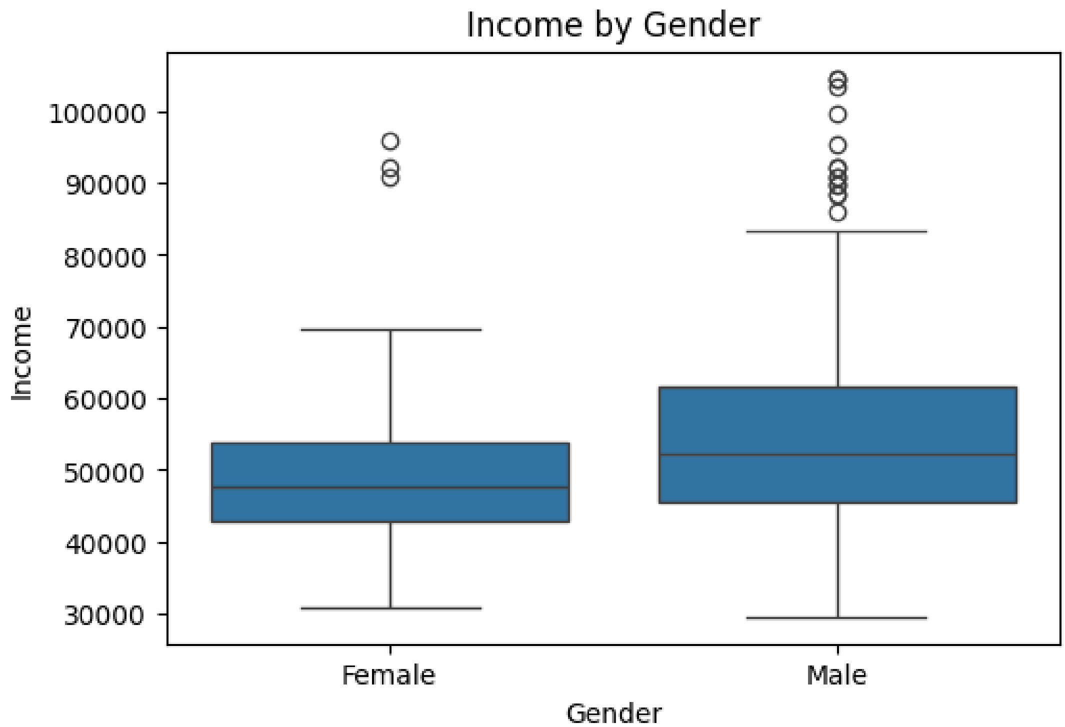


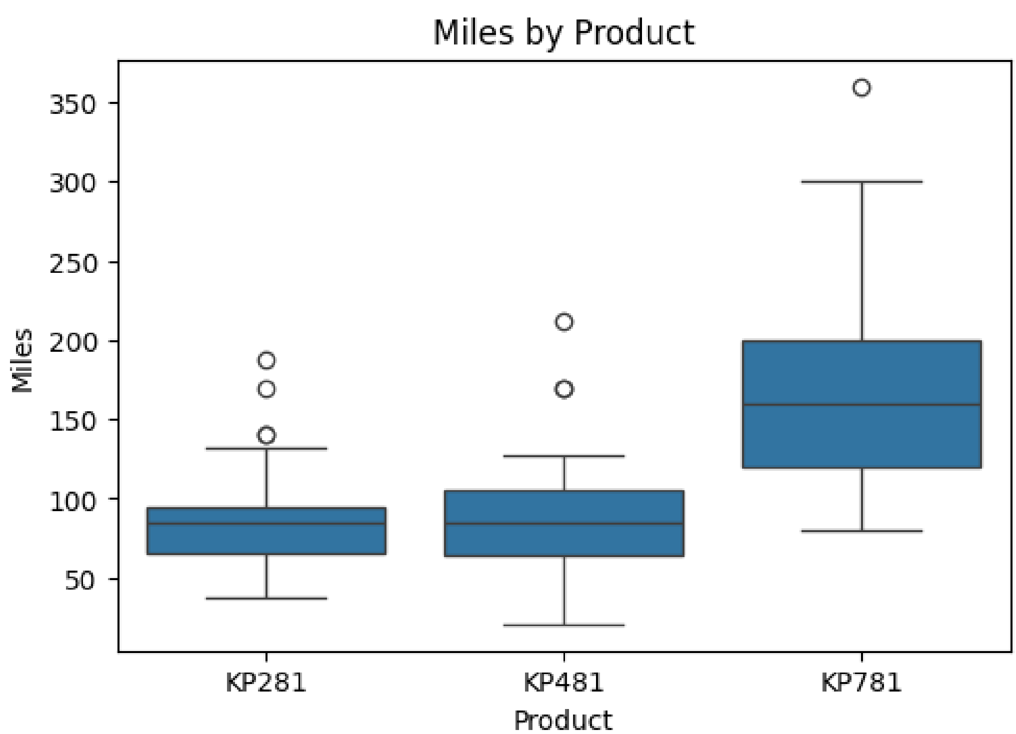

Age by MaritalStatus
Age
50
45
40
35
30
25
20
Partnered
Single
MaritalStatus

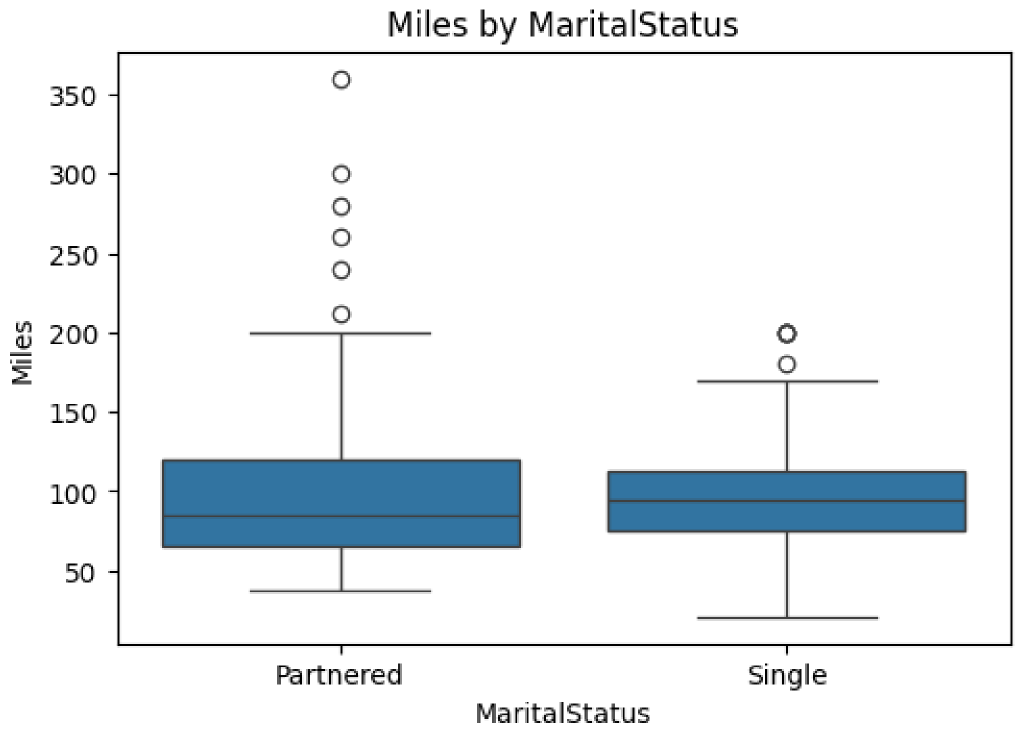

Age by Gender
Age
50
45
40
35
30
25
20
Female
Male
Gender

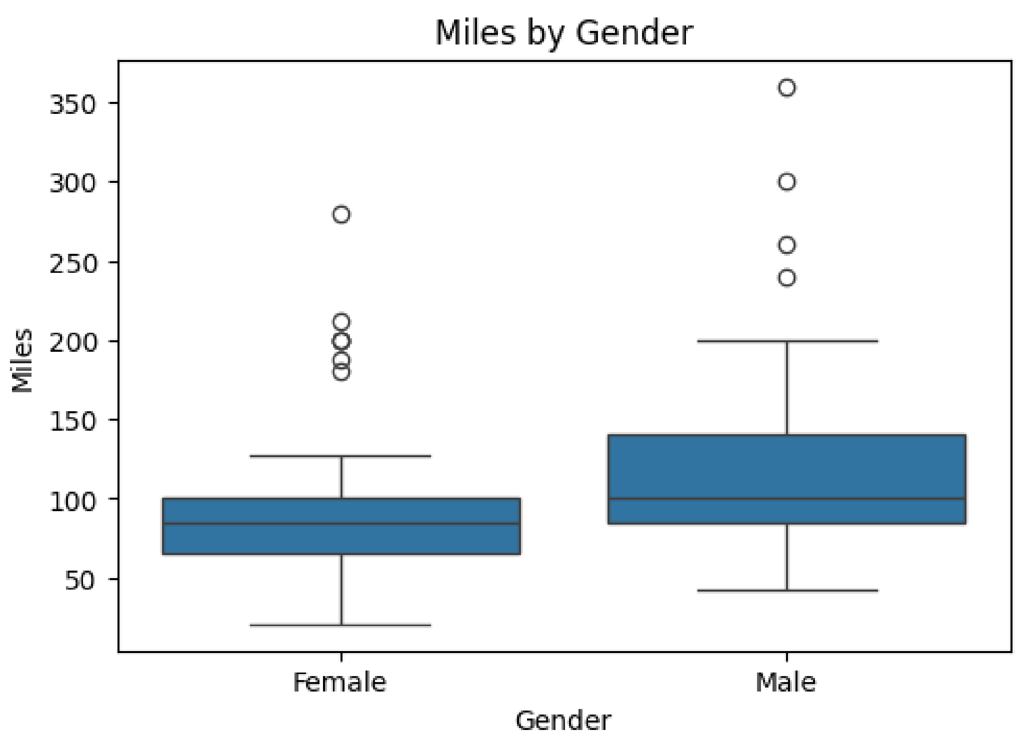

Income by Product
100000
90000
80000
70000
60000
50000
40000
30000
Income
KP281
KP481
KP781
Product

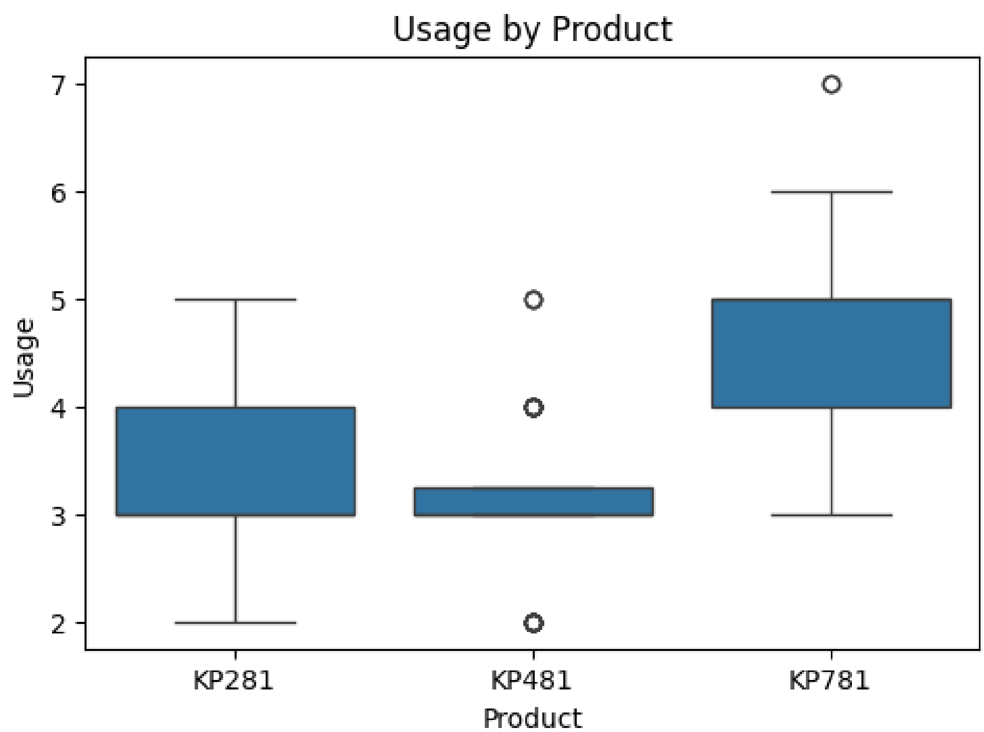

Income by MaritalStatus
100000
90000
80000
70000
60000
50000
40000
30000
Income
Partnered
Single
MaritalStatus

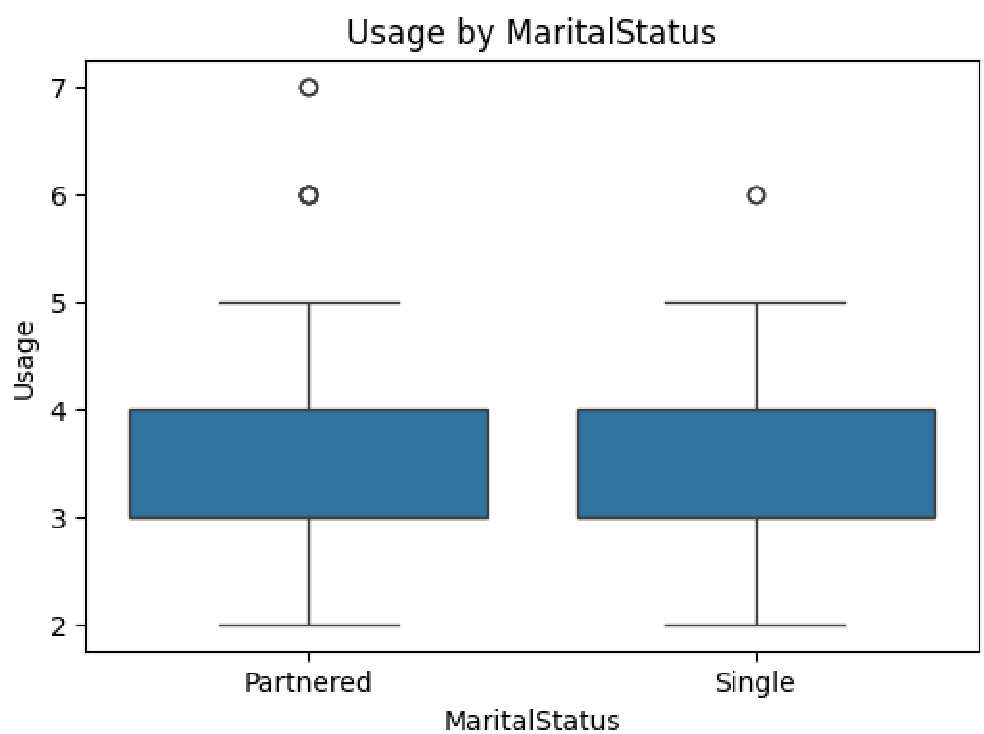

Income by Gender
100000
90000
80000
70000
60000
50000
40000
30000
Income
Female
Male
Gender

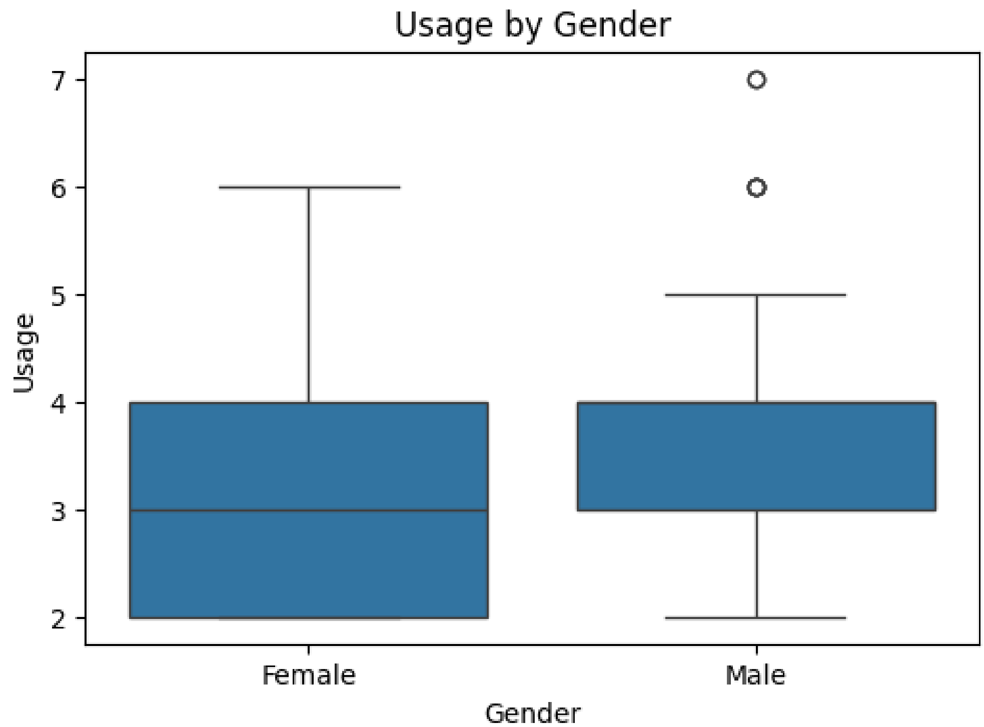

Miles by Product
350
300
250
200
150
100
50
Miles
KP281
KP481
KP781
Product

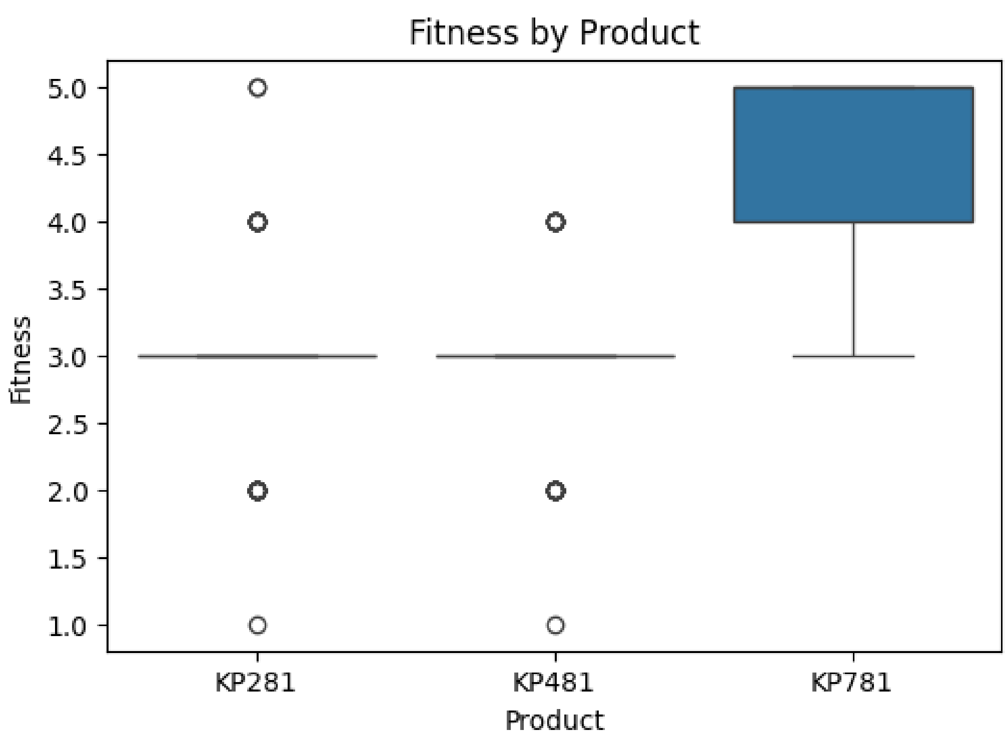

Miles by MaritalStatus
Miles
350
300
250
200
150
100
50
Partnered
Single
MaritalStatus

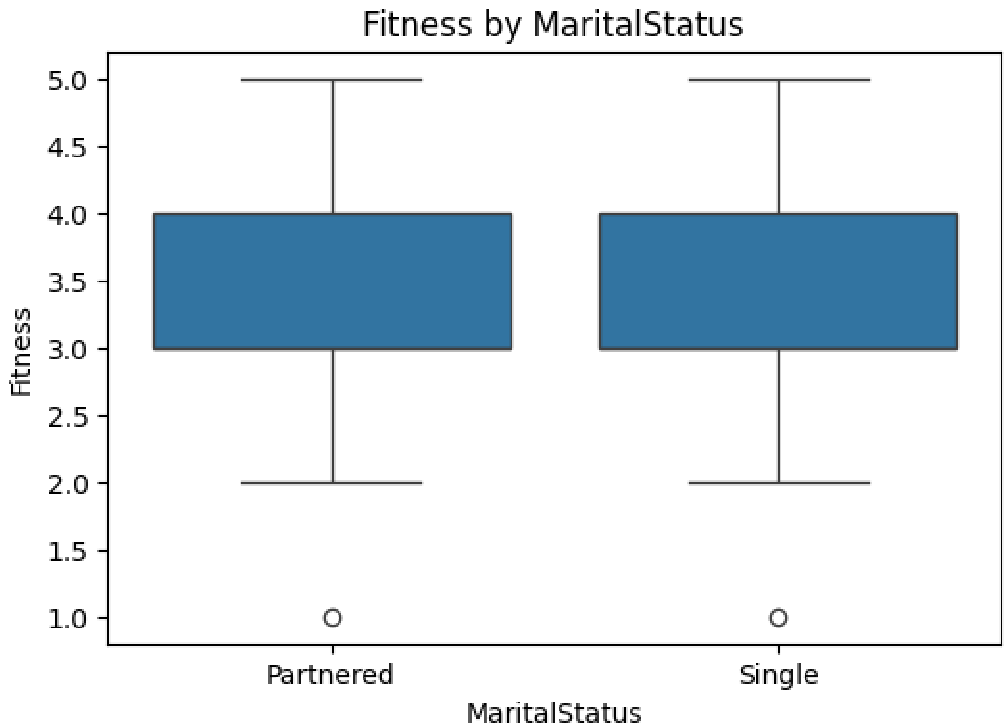

Miles by Gender
Miles
350
300
250
200
150
100
50
Female
Male
Gender

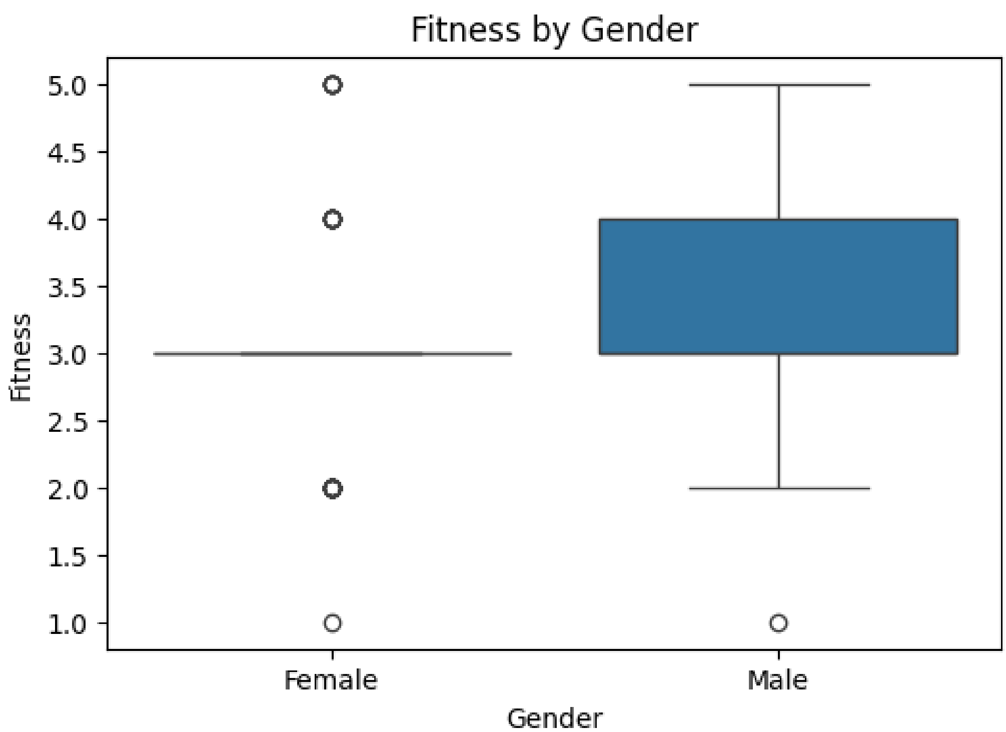

Usage by Product
Usage
7
6
5
4
3
2
KP281
KP481
KP781
Product


Usage by MaritalStatus
Usage
7
6
5
4
3
2
Partnered
Single
MaritalStatus


Usage by Gender
7
6
5
4
3
2
Usage
Female
Male
Gender

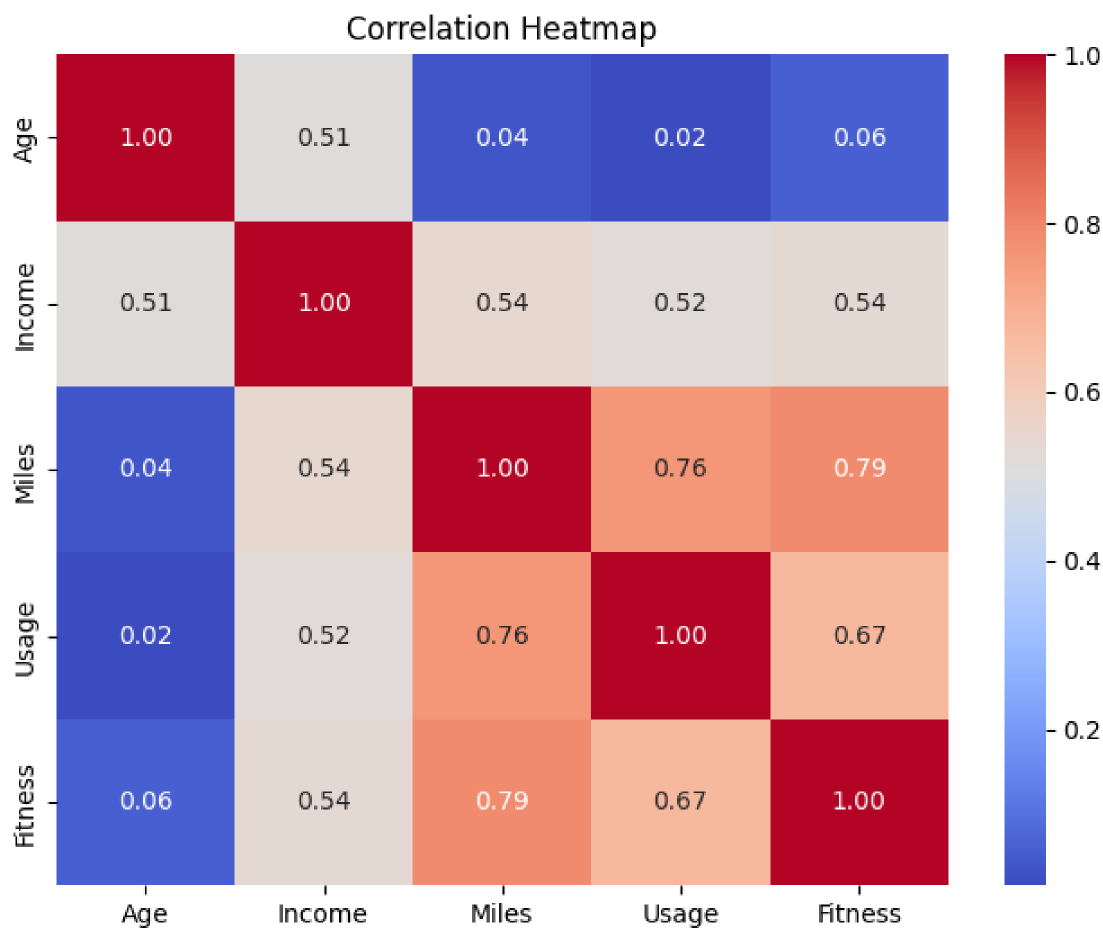

Fitness by Product
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Fitness
KP281
KP481
KP781
Product


Fitness by MaritalStatus
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Fitness
Partnered
Single
MaritalStatus


Fitness by Gender
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Fitness
Female
Male
Gender

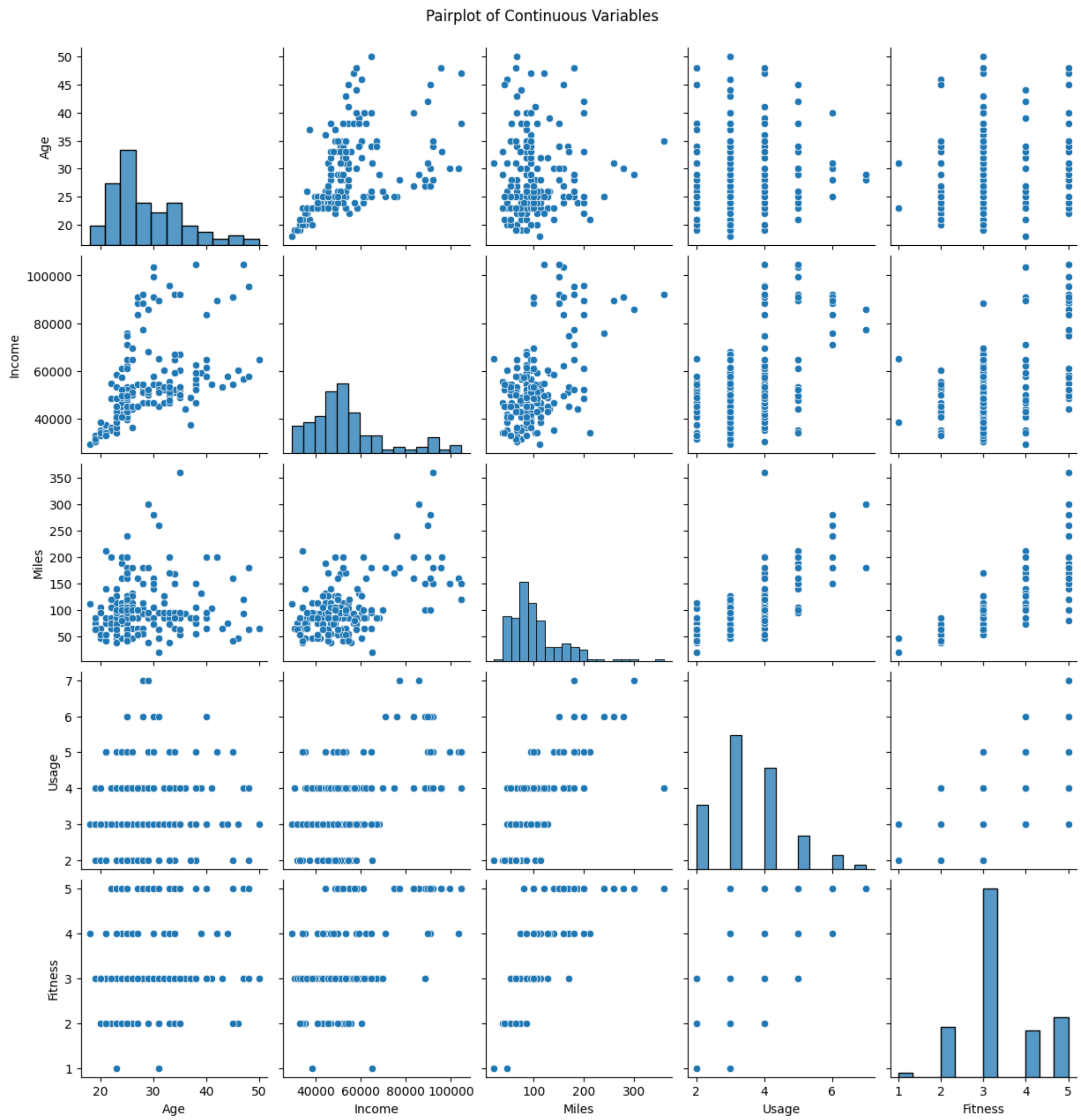
0.1.2 Correlation Analysis

[ ]:
```
# Correlation matrix
plt.figure(figsize=(8, 6))
corr = df[cont_cols].corr()
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```



0.1.3 Pairplot

[ ]:
```
sns.pairplot(df[cont_cols])
plt.suptitle("Pairplot of Continuous Variables", y=1.02)
plt.show()
```

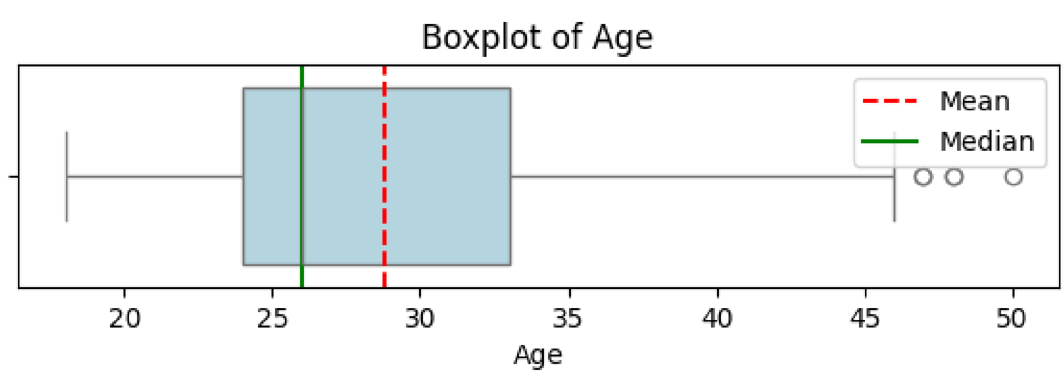


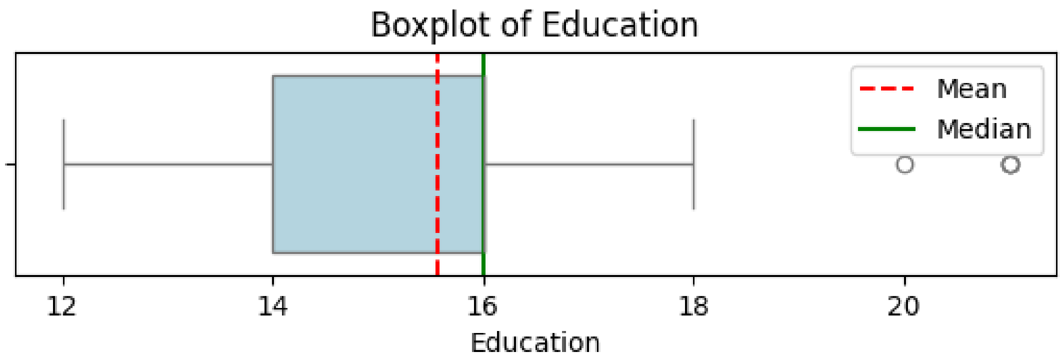
##Missing Values & Outlier Detection

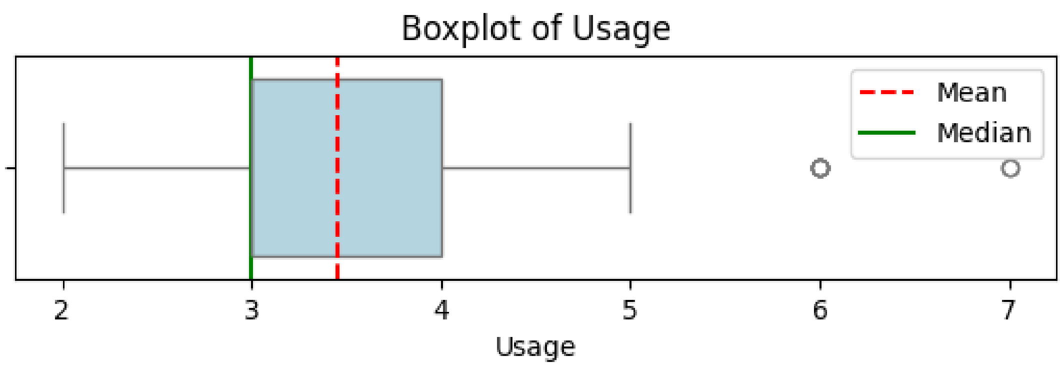
```
[ ]: for col in df.select_dtypes(include='number').columns:
         mean_val = df[col].mean()
         median_val = df[col].median()
         diff = abs(mean_val - median_val)
         print(f"\nColumn: {col}")
         print(f"Mean: {mean_val:.2f}, Median: {median_val:.2f}, Difference: {diff:.
     ↪2f}")
         if diff > 0.5:  # You can choose your own threshold based on data scale
             print("  Possible outliers detected based on mean-median difference.")
```

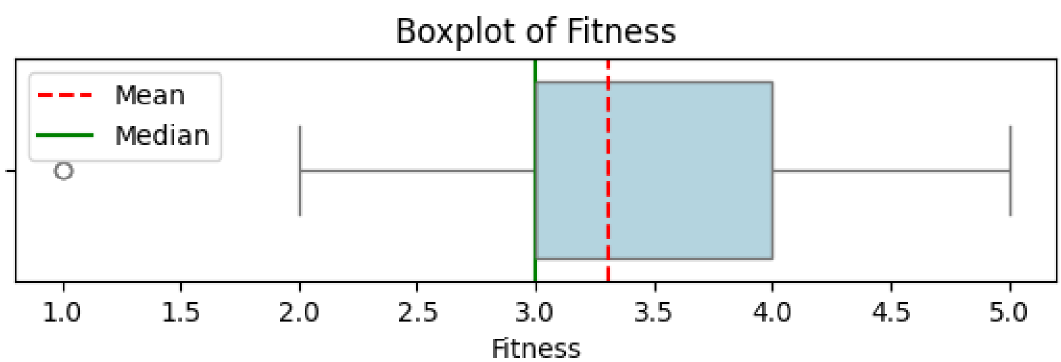
```
Column: Age
Mean: 28.79, Median: 26.00, Difference: 2.79
 Possible outliers detected based on mean-median difference.

Column: Education
Mean: 15.57, Median: 16.00, Difference: 0.43

Column: Usage
Mean: 3.46, Median: 3.00, Difference: 0.46

Column: Fitness
Mean: 3.31, Median: 3.00, Difference: 0.31

Column: Income
Mean: 53719.58, Median: 50596.50, Difference: 3123.08
 Possible outliers detected based on mean-median difference.

Column: Miles
Mean: 103.19, Median: 94.00, Difference: 9.19
 Possible outliers detected based on mean-median difference.
```

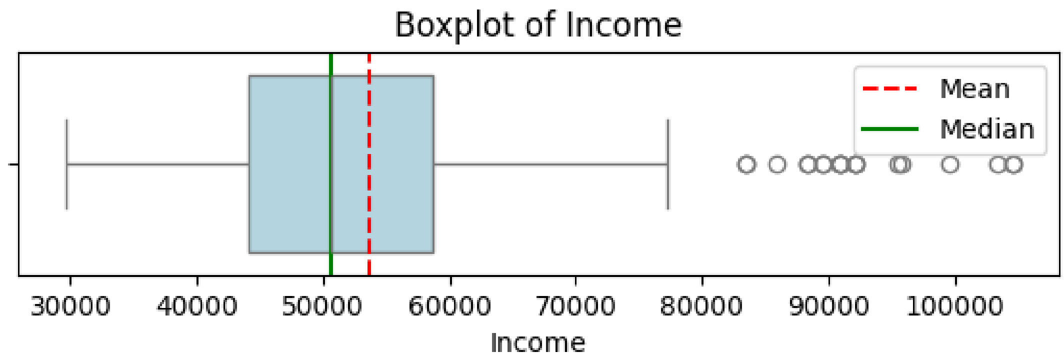
[ ]:
```python
import matplotlib.pyplot as plt
import seaborn as sns
num_cols = df.select_dtypes(include='number').columns

for col in num_cols:
    plt.figure(figsize=(7, 1.5))
    sns.boxplot(x=df[col], color="lightblue")
    plt.title(f'Boxplot of {col}')
    plt.axvline(df[col].mean(), color='red', linestyle='--', label='Mean')
    plt.axvline(df[col].median(), color='green', linestyle='-', label='Median')
    plt.legend()
    plt.show()
```

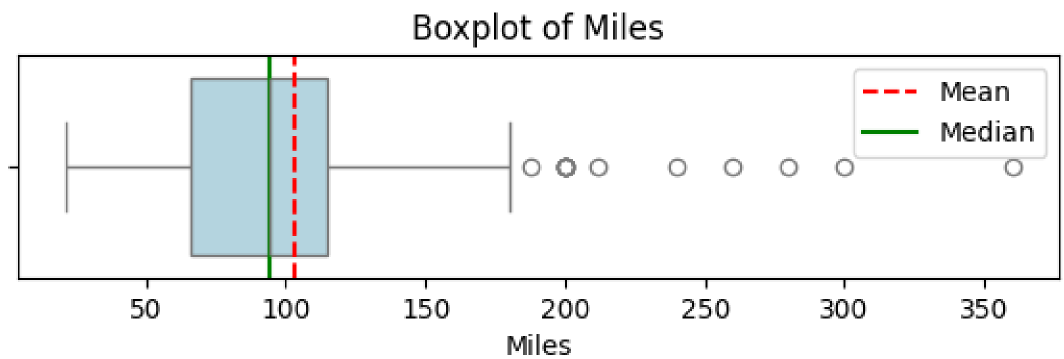



Boxplot of Education
Mean
Median
12
14
16
18
20
Education


Boxplot of Usage
Mean
Median
2
3
4
5
6
7
Usage

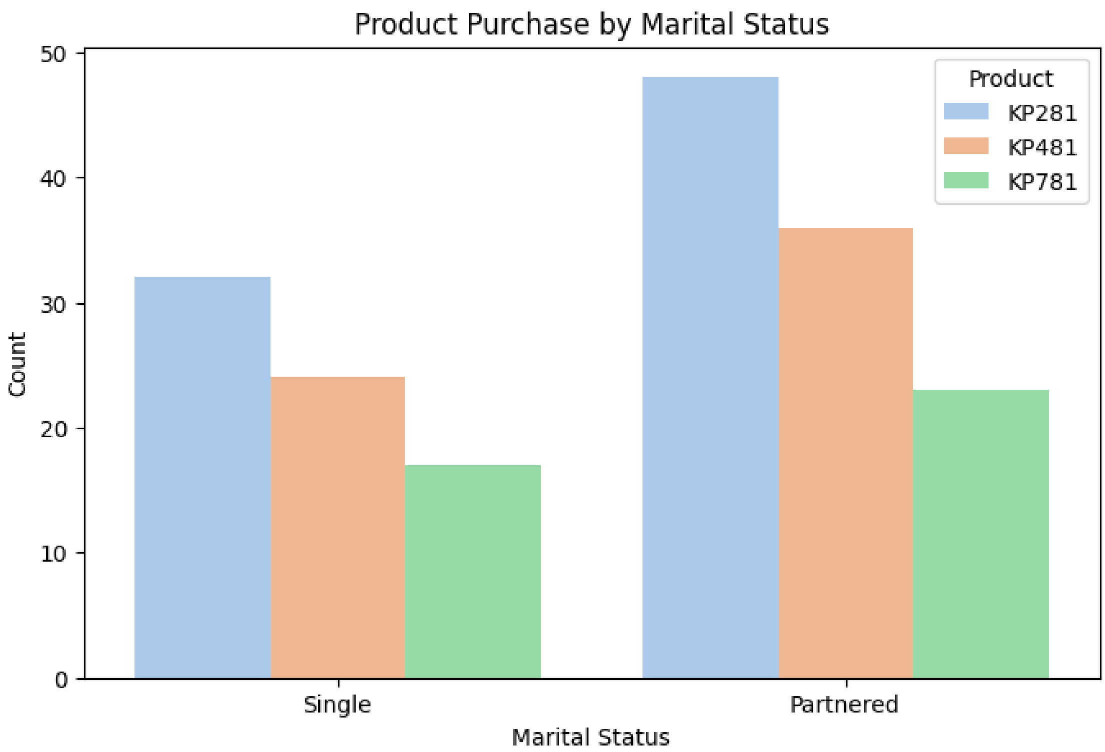

Boxplot of Fitness
Mean
Median
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Fitness





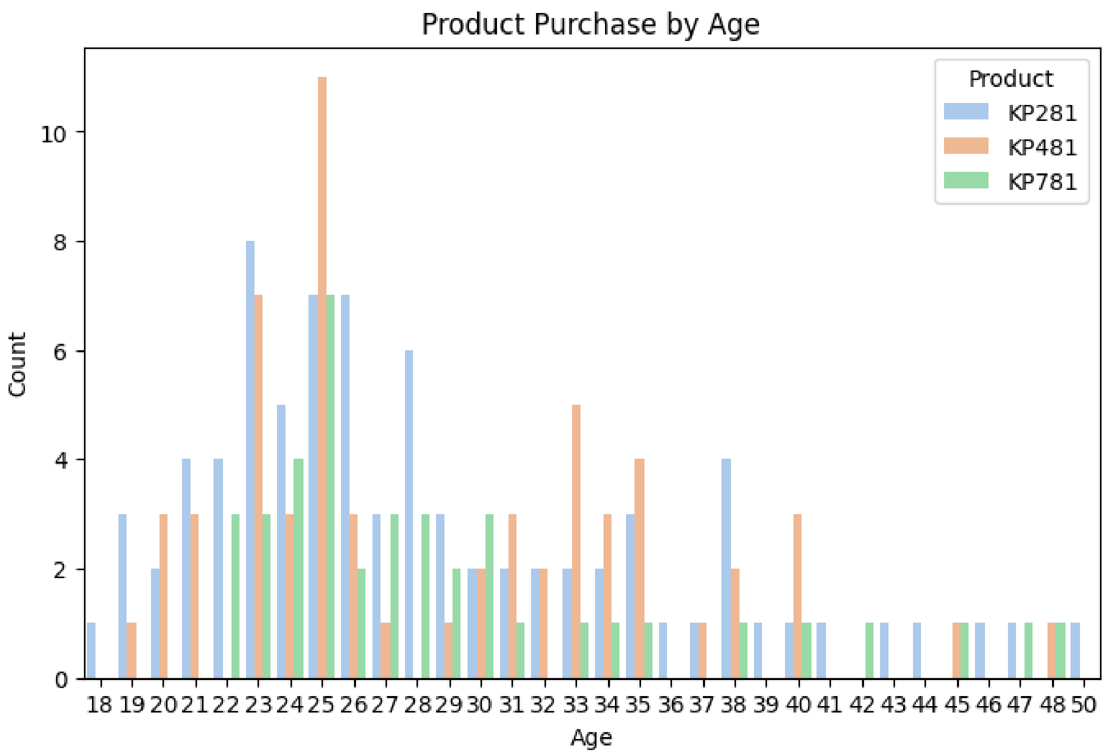
[ ]:
```
# MaritalStatus  its effect on product purchase using countplot
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='MaritalStatus', hue='Product', palette='pastel')
plt.title('Product Purchase by Marital Status')
plt.xlabel('Marital Status')
plt.ylabel('Count')
```

[ ]:
```
Text(0, 0.5, 'Count')
```



[ ]:
```
# age  its effect on product purchase using countplot
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='Age', hue='Product', palette='pastel')
plt.title('Product Purchase by Age')
plt.xlabel('Age')
plt.ylabel('Count')
```

[ ]:
```
Text(0, 0.5, 'Count')
```



[ ]:
```
# Let's say df is your Aerofit dataset
product_counts = pd.crosstab(index=df['Product'], columns='count')

# Convert to percentage (marginal probability)
product_percentage = product_counts / product_counts.sum() * 100

# Rename column for clarity
product_percentage.columns = ['Percentage']

# Optional: round for readability
product_percentage = product_percentage.round(2)

print(product_percentage)
```

```
         Percentage
Product
KP281         44.44
KP481         33.33
KP781         22.22
```

0.2 Business Insights

1.The KP281 treadmill seems to be the most popular choice among customers. This might be because it fits well with people's budgets or needs.

2.There's a pretty even mix of men and women buying treadmills, so the marketing strategy doesn't need to target one gender more than the other.

3.A large number of buyers are single, which could mean they're more focused on personal health or fitness goals.

4.Most customers are in the 25 to 40 age range, showing that younger adults are more health-conscious or willing to invest in fitness equipment.

5.People with higher incomes usually go for the KP781, which suggests it's the premium model and appeals to those who can afford more features.

6.Those who rated themselves higher in fitness tend to use the treadmill more frequently, especially if they own the higher-end models.

7.Customers who plan to walk or run more miles every week are the ones who usually go for better or advanced treadmills.

8.It looks like people with higher education also tend to use the treadmill more regularly, possibly because they're more aware of fitness benefits.

9.Most people earn between $50,000 and $100,000, although a few high-income customers are present too.

10.Variables like Miles, Usage, and Fitness level are somewhat connected. It makes sense—those who are more fit usually use the treadmill more and cover more distance.

0.3 Range of Attributes

1.Age ranges mostly from around 18 to 50 years. It's mostly younger to middle-aged people.

2.Income goes from roughly 30K to 150K. Most people fall between 50K and 100K.

3.Miles per week range from 20 to 150. There's a good mix of light and heavy users.

4.Usage (days per week) ranges from 1 to 7. Most people use it about 3–4 times a week.

5.Fitness level is self-rated from 1 to 5. Most customers rate themselves at level 3 or 4.

0.3.1 Distribution & Relationships Between Variables

1.Age, Income, and Miles are all skewed toward the lower side. That means most people are in the younger age group, have mid-level income, and run moderate miles.

2.The categorical values like Gender and Marital Status have clear groups with fairly even counts.

3.There are some meaningful relationships

4.People who use the treadmill more often also cover more miles.

5.Those who think they are more fit also tend to use it more.

6.Higher income has some influence on which product they buy, especially when it comes to premium models.

##Observations from the Plots

0.3.2 Univariate

Age: Most buyers are in their late 20s to 30s.

Income: A lot of people earn around 60K–90K per year.

Miles: Many use the treadmill for 40–60 miles weekly.

Usage: 3–4 times a week seems to be the average.

Fitness: Most people rate themselves as moderately fit (level 3 or 4).

0.3.3 Categorical Variables

Product: KP281 is the best-seller, while KP781 is more exclusive.

Gender: Fairly balanced between men and women.

Marital Status: More single people seem to buy these treadmills.

0.3.4 Bivariate

1.People with higher income tend to buy KP781.

2.Older customers lean slightly toward the mid or premium models.

3.Men and women show slight differences in miles covered, but it's not very significant.

4.Both single and partnered people show a variety of fitness levels.

5.KP781 users use the treadmill more often on average.

0.4 Correlation Heatmap and Pairplot

1.There's a clear connection between how much people say they're fit, how often they use the treadmill, and how far they go each week.

2.These plots also help identify small clusters—like higher-income buyers leaning toward premium models.

0.5 Recommendations

1.KP281 Top selling Model

Since most people buy the KP281 model, show it more in ads and give discounts to attract even more buyers.

2.Based on Age Most buyers are between 25 and 40 years old. Show young people in your ads and post on different platforms

3.Based on Budget

For Low budget Show KP281.

For Medium budget Suggest KP481.

For High budget Highlight KP781.

4.**Based on type of user bold text**

For Beginners recommend KP281

For Regular Users suggest KP481

For Fitness Lovers athlete or runners suggest KP781

5.Production

Make more of KP281 because it's everyone's favorite. But still make some KP781 for those who want the fancy one — just don't make too many.