BUSINESS CASE: AEROFIT- DESCRIPTIVE STATISTICS AND PROBABILITY

PROBLEM STATEMENT: AeroFit, a premier brand in the fitness equipment industry, is committed to elevating its market presence and product portfolio through strategic insights and customer-centric approaches. To achieve this objective, the esteemed market research team at AeroFit has embarked on a comprehensive analysis endeavor. This initiative aims to delve deep into the intricate dynamics of customer demographics, usage behaviors, fitness inclinations, and purchasing tendencies across its diverse range of treadmill offerings.

By meticulously unraveling these multifaceted dimensions, AeroFit endeavors to sculpt a nuanced understanding of its clientele. Such discernment will not only refine its marketing strategies but also invigorate its product lineup. Ultimately, AeroFit aspires to curate bespoke recommendations for prospective clientele, thereby fostering enhanced customer engagement, fortifying its market position, and heralding a new era of tailored excellence in fitness solutions.

The **key components** of the problem statement are:

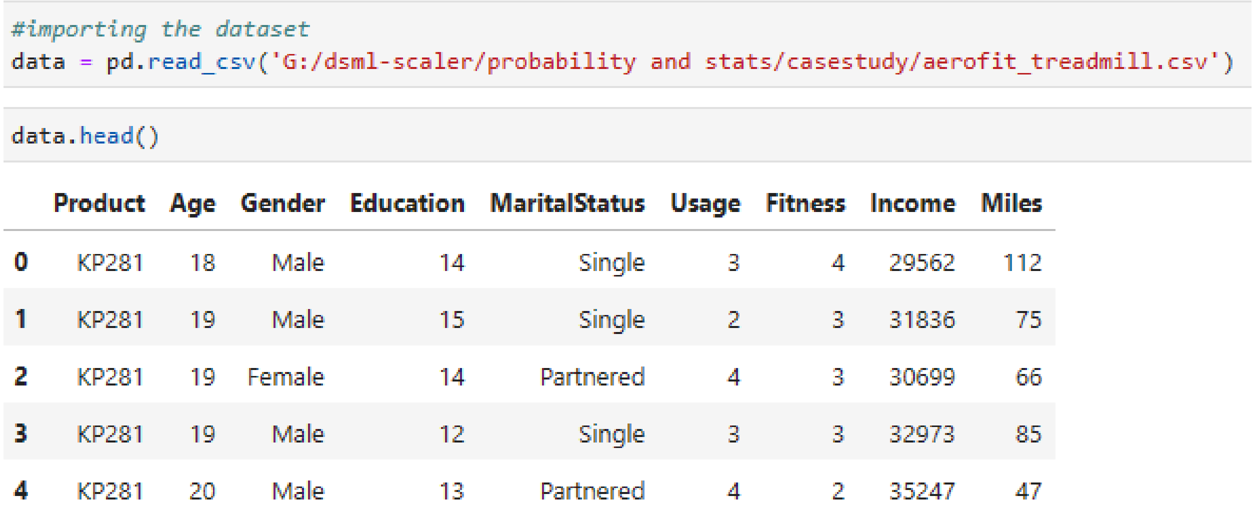
- **Objective:** AeroFit aims to enhance its marketing strategies and product offerings.
- **Goal:** Identify the characteristics of the target audience for each type of treadmill offered by AeroFit.
- **Analysis Scope:** Understanding demographics, usage patterns, fitness preferences, and purchasing behaviors of customers across different treadmill products.
- **Purpose:** The purpose is as follows:
  - ➢ Provide better recommendations of treadmills to new customers.
  - ➢ Optimize marketing strategies.
  - ➢ Tailor product offerings to meet customer preferences effectively.
- **Methodology:** Methodology opted will be as follows:
  - ➢ Utilize market research and analysis techniques to delve into customer data.
  - ➢ Employ statistical analysis and data visualization to uncover insights.
- **Outcome:** The outcome is as follows:
  - ➢ Informed decision making to refine marketing strategies and product offerings.
  - ➢ Enhanced customer engagement and satisfaction.
  - ➢ Strengthened market position for Aerofit in the fitness equipment industry.

ANALYZING BASIC METRICS:

- **Demographic Characteristics Analysis:** We will analyze key demographic variables such as age, gender, education, marital status, and income for each treadmill product to understand the profile of customers purchasing AeroFit treadmills.
- **Usage Patterns Variation:** By examining the average usage frequency per week for each treadmill product, we will identify variations in usage behavior among customers.
- **Self-Rated Fitness Level Analysis:** We will determine the distribution of self-rated fitness levels among customers for each treadmill product to understand their fitness preferences.
- **Expected Mileage Differences:** Comparing the average expected mileage per week for different treadmill products will help us assess variations in usage expectations among customers.
- **Customer Preferences Distribution:** We will calculate the proportion of customers purchasing each treadmill product and visualize the distribution to identify popular choices.
- **Influence of Demographic Characteristics:** Analyzing the relationship between demographic variables and treadmill product preferences will help us understand how demographic characteristics influence customer choices.
- **Association Between Customer Characteristics and Product Preferences:** Constructing contingency tables and computing conditional probabilities will help us identify notable associations between customer characteristics and product preferences.
- **Marginal Probabilities Calculations:** Calculating the marginal probabilities of purchasing each treadmill product within different customer segments will provide insights into product popularity among different demographics.
- **Distribution of Customer Preferences Across Aerofit Treadmill Product Portfolio:** Calculate the proportion of customers purchasing each treadmill product. Visualize the distribution to identify popular choices among customers.
- **Insights From Customer Profiling:** Utilizing customer profiles developed for each treadmill product, we will derive insights to make better recommendations for new customers based on their demographic characteristics and preferences.
- **Actionable Insights and Recommendations:** Finally, we will provide actionable recommendations based on the analysis insights, enabling AeroFit to optimize marketing strategies and product offerings effectively.

By addressing these questions and analyzing basic metrics, we will gain comprehensive insights into the characteristics of the target audience for each AeroFit treadmill product, enabling AeroFit to make data-driven decisions and enhance its business strategies accordingly.

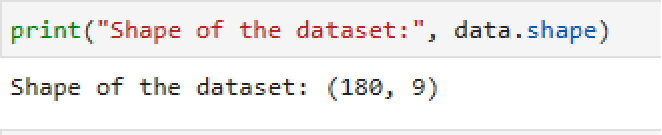
OBSERVATIONS OF DATASET: In this step, we will begin by importing the dataset and conducting a series of operations to gain fundamental insights into its structure and contents. Through these operations, we aim to extract basic details about the dataset, such as its size, the types of data it contains, and summary statistics that provide a snapshot of the dataset's characteristics.



```
#importing the dataset
data = pd.read_csv('G:/dsml-scaler/probability and stats/casestudy/aerofit_treadmill.csv')
```

```
data.head()
```

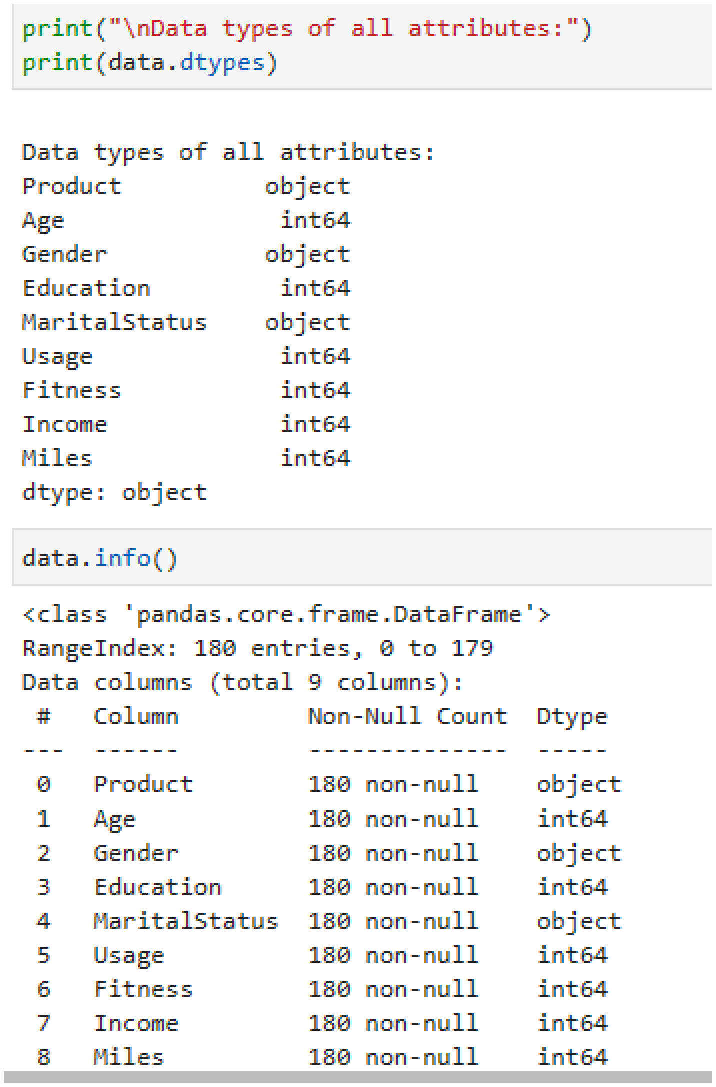
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47



```
print("Shape of the dataset:", data.shape)
```

```
Shape of the dataset: (180, 9)
```

The dataset consists of 180 rows and 9 columns, encapsulating a comprehensive range of data points for analysis.



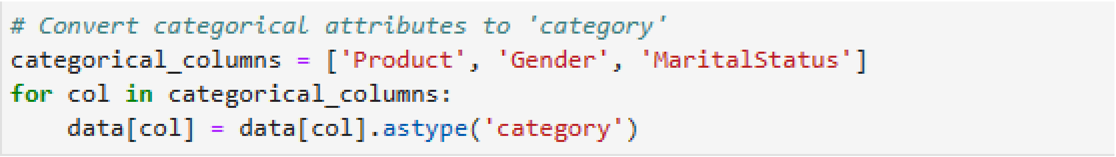
```
print("\nData types of all attributes:")
print(data.dtypes)
```

```
Data types of all attributes:
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

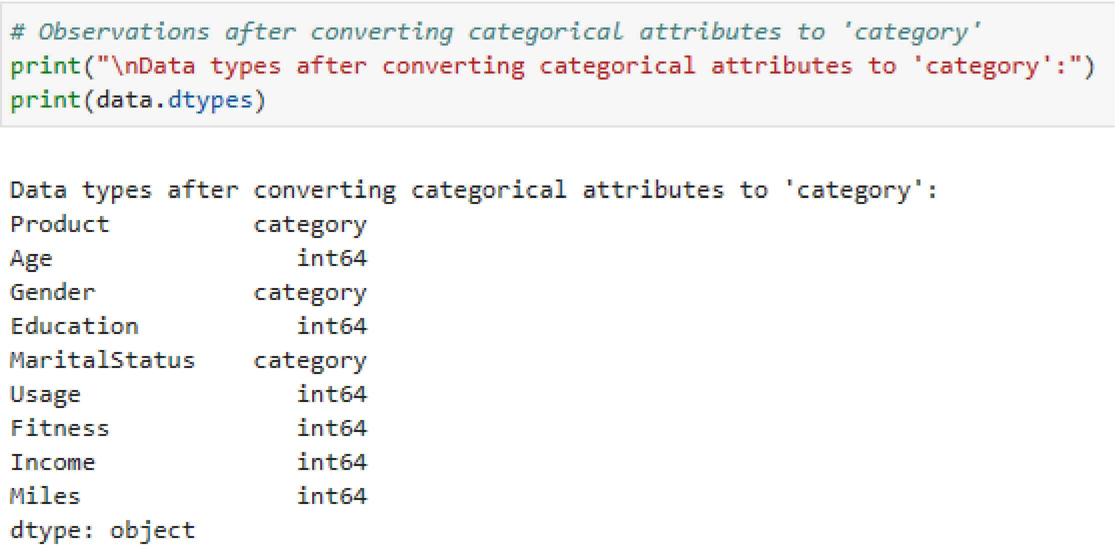
```
data.info()
```

```
<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 
```

The dataset comprises numerical values across most columns, with 'Product', 'Gender', and 'Marital Status' represented as categorical variables in object format.



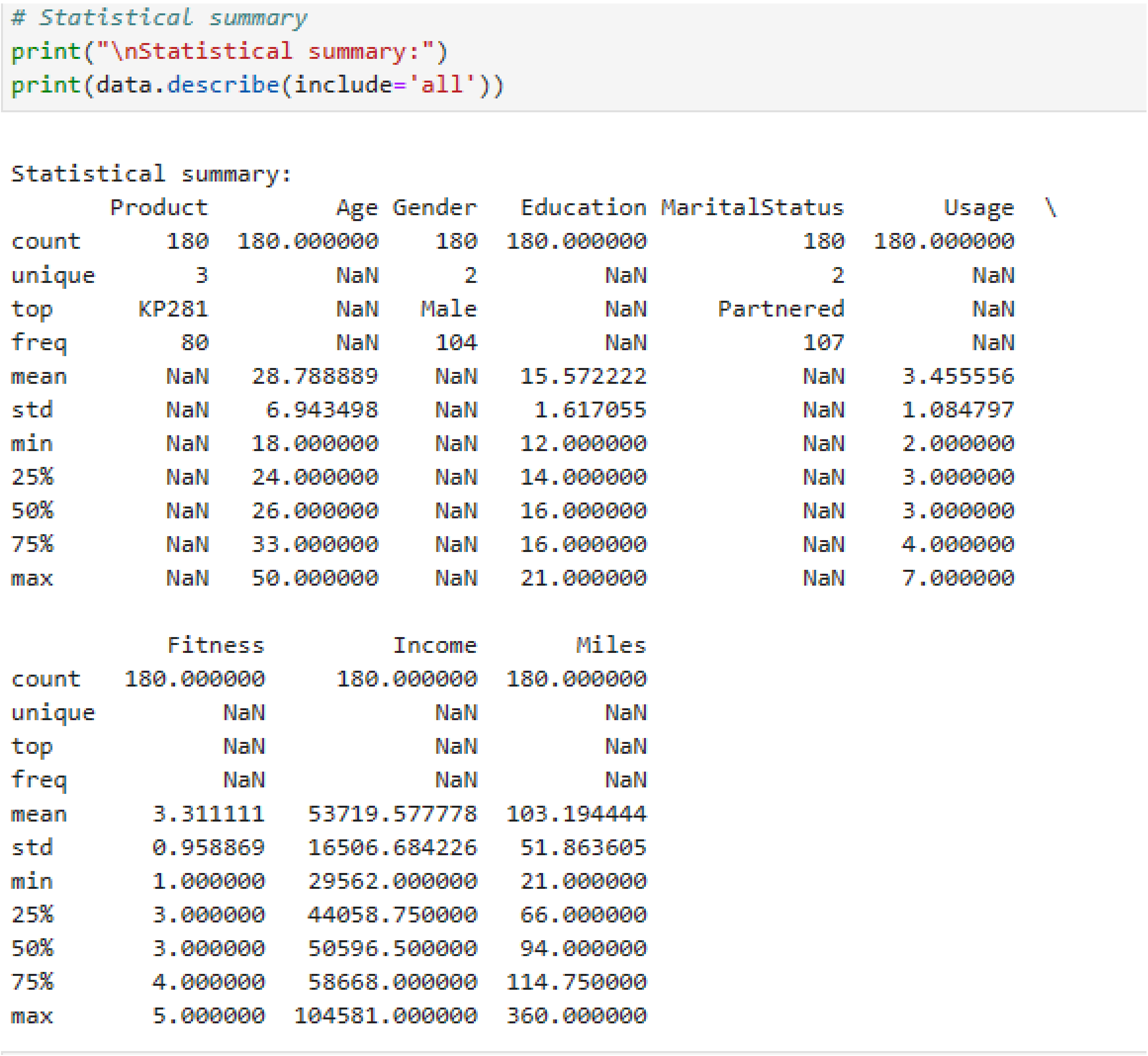
```
# Convert categorical attributes to 'category'
categorical_columns = ['Product', 'Gender', 'MaritalStatus']
for col in categorical_columns:
    data[col] = data[col].astype('category')
```



```
# Observations after converting categorical attributes to 'category'
print("\nData types after converting categorical attributes to 'category':")
print(data.dtypes)
```

```
Data types after converting categorical attributes to 'category':
Product          category
Age                 int64
Gender           category
Education           int64
MaritalStatus    category
Usage               int64
Fitness             int64
Income              int64
Miles               int64
dtype: object
```

Following the conversion of categorical attributes to the 'category' data type, a thorough assessment of all attribute data types was conducted to ensure uniformity and accuracy.

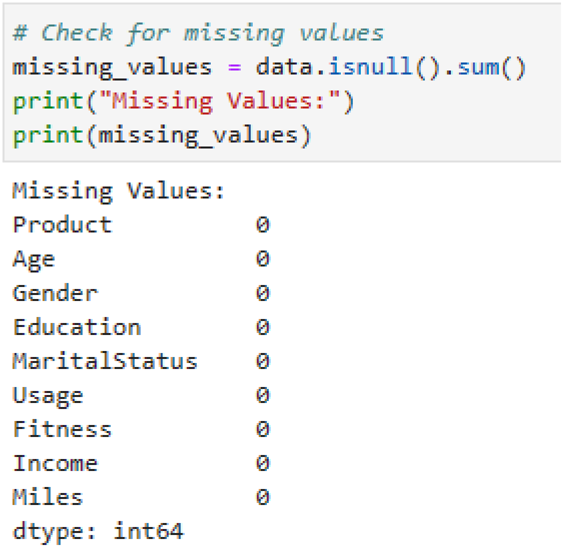


```
# Statistical summary
print("\nStatistical summary:")
print(data.describe(include='all'))
```

```
Statistical summary:
       Product         Age Gender   Education MaritalStatus       Usage  \
count      180  180.000000    180  180.000000           180  180.000000   
unique       3         NaN      2         NaN             2         NaN   
top      KP281         NaN   Male         NaN     Partnered         NaN   
freq        80         NaN    104         NaN           107         NaN   
mean       NaN   28.788889    NaN   15.572222           NaN    3.455556   
std        NaN    6.943498    NaN    1.617055           NaN    1.084797   
min        NaN   18.000000    NaN   12.000000           NaN    2.000000   
25%        NaN   24.000000    NaN   14.000000           NaN    3.000000   
50%        NaN   26.000000    NaN   16.000000           NaN    3.000000   
75%        NaN   33.000000    NaN   16.000000           NaN    4.000000   
max        NaN   50.000000    NaN   21.000000           NaN    7.000000   

           Fitness         Income       Miles  
count   180.000000     180.000000  180.000000  
unique         NaN            NaN         NaN  
top            NaN            NaN         NaN  
freq           NaN            NaN         NaN  
mean      3.311111   53719.577778  103.194444  
std       0.958869   16506.684226   51.863605  
min       1.000000   29562.000000   21.000000  
25%       3.000000   44058.750000   66.000000  
50%       3.000000   50596.500000   94.000000  
75%       4.000000   58668.000000  114.750000  
max       5.000000  104581.000000  360.000000  
```

A comprehensive statistical summary of the dataset was generated to provide a detailed overview of its key characteristics and distributional properties.

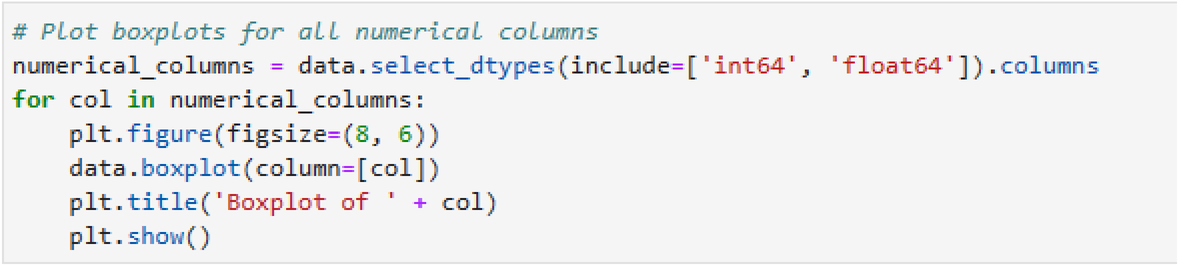
MISSING VALUES AND OUTLIERS DETECTION: In this step, our focus will be on identifying and addressing missing values and outliers within the dataset. By systematically examining the data, we aim to rectify any discrepancies and ensure the integrity and reliability of our analysis.



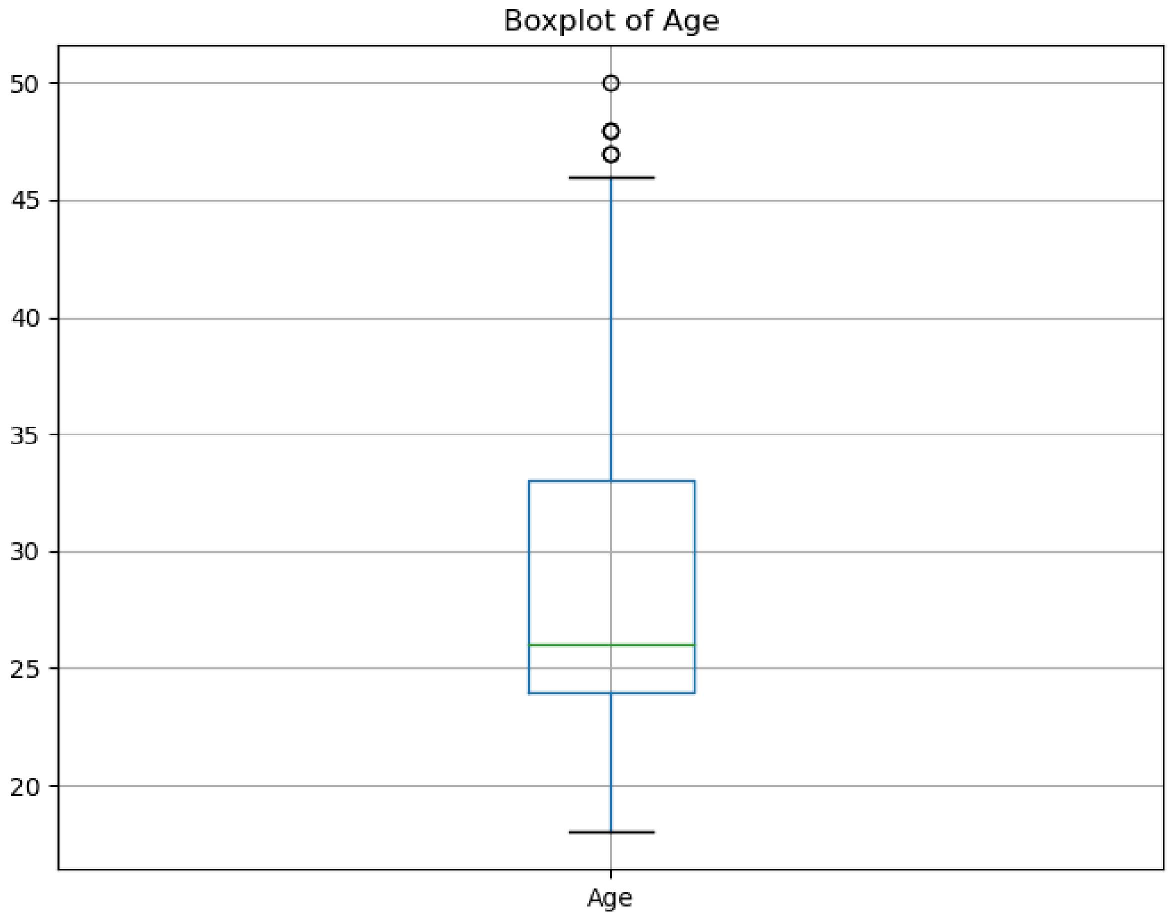
```python
# Check for missing values
missing_values = data.isnull().sum()
print("Missing Values:")
print(missing_values)
```

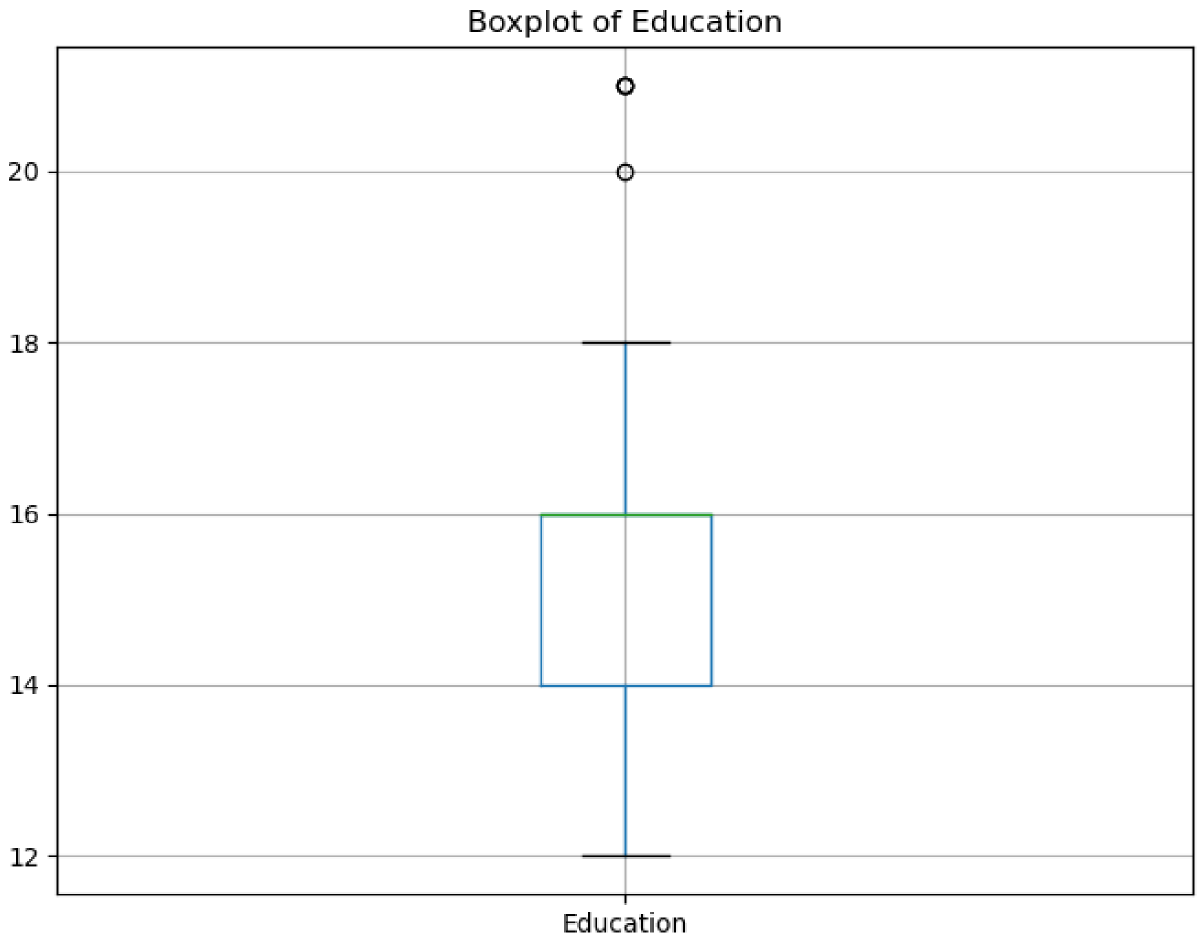
```
Missing Values:
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

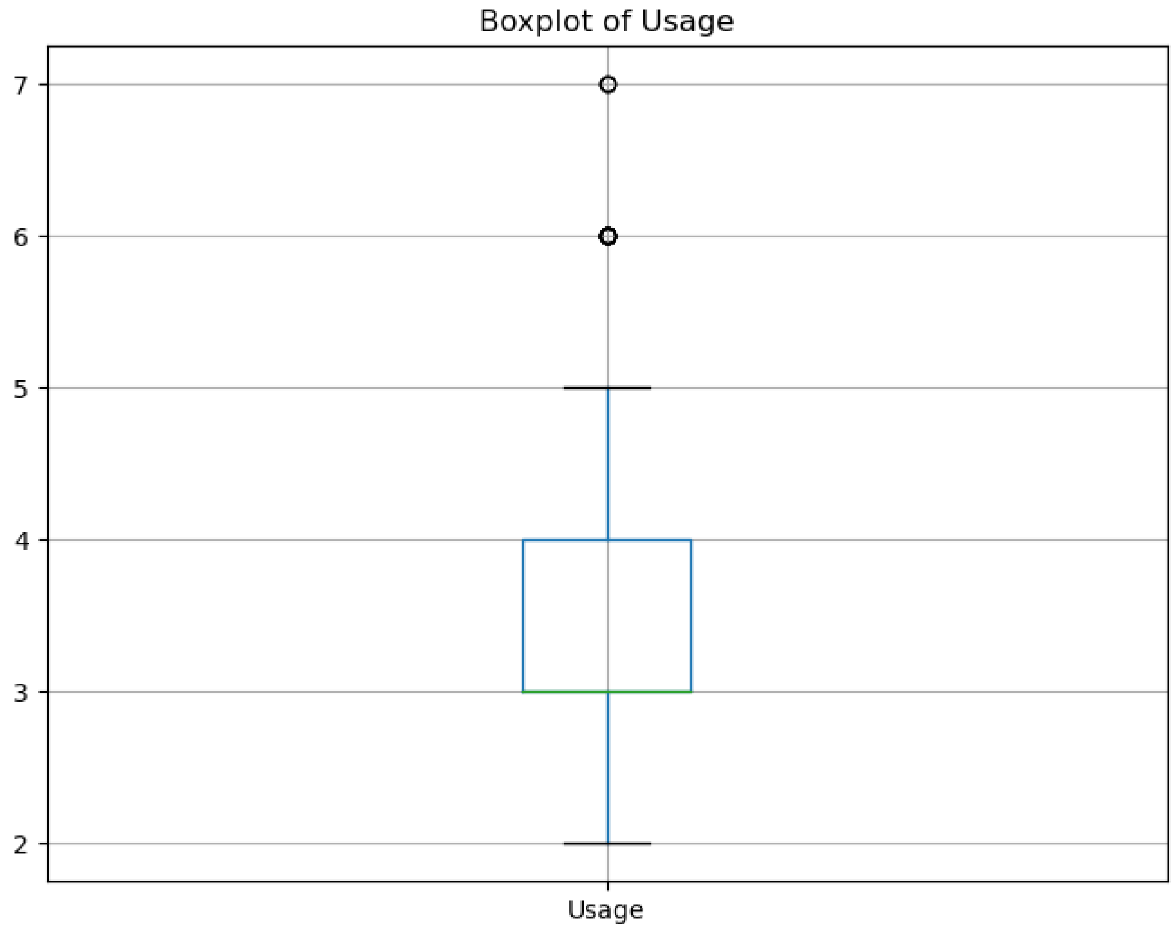
Since there are no missing values in the dataset, our attention will now shift towards detecting and managing outliers, if any. This process will involve scrutinizing the data for any unusual or extreme observations that may impact the robustness of our analysis.

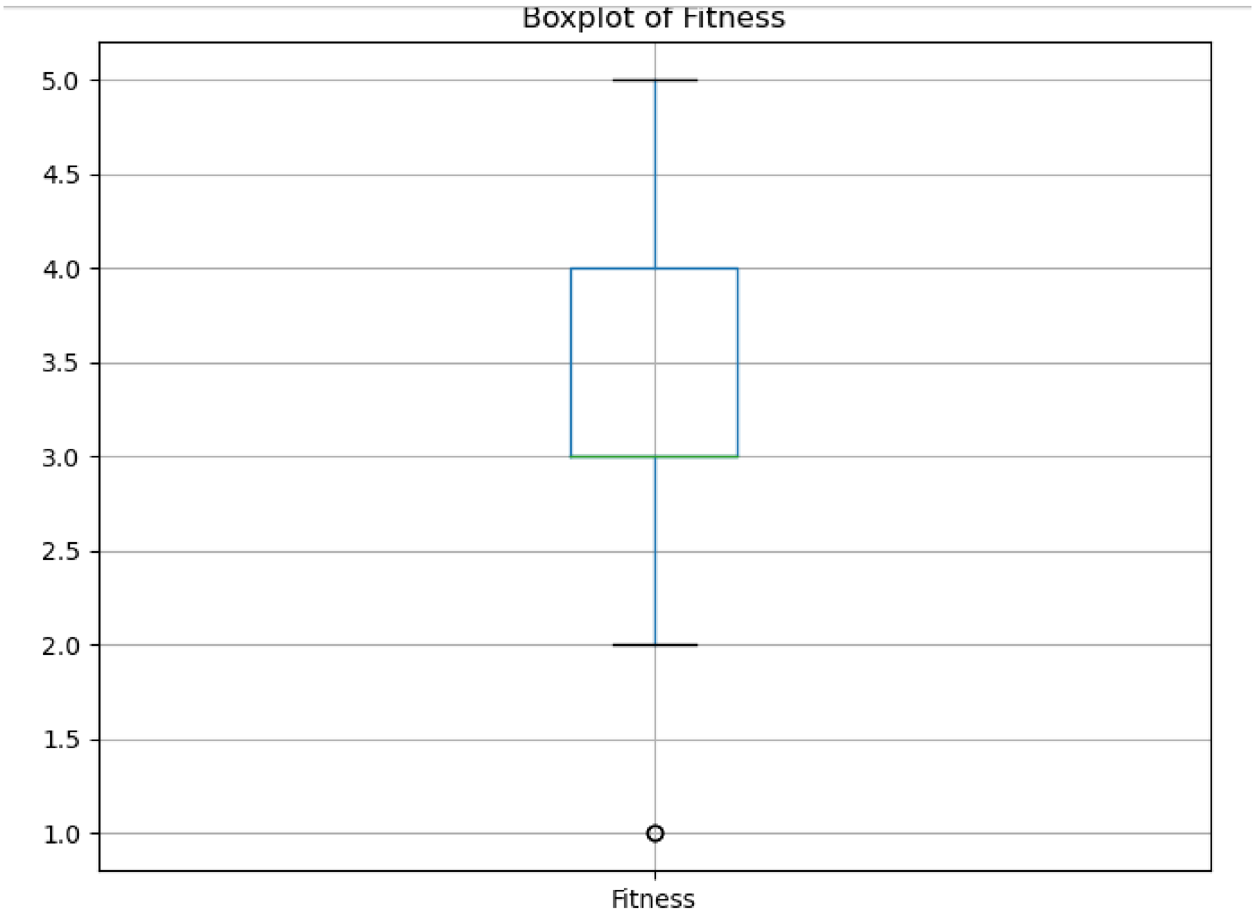


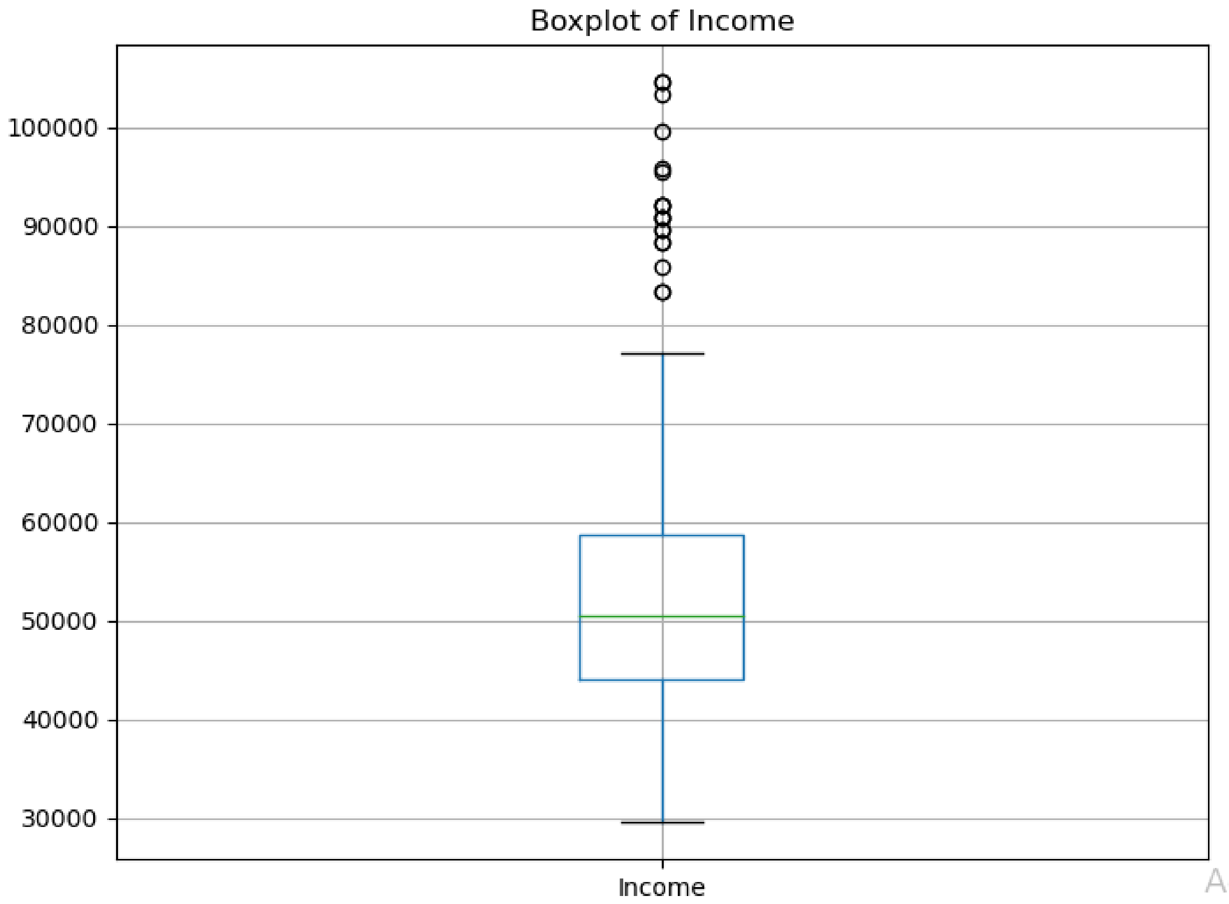
```python
# Plot boxplots for all numerical columns
numerical_columns = data.select_dtypes(include=['int64', 'float64']).columns
for col in numerical_columns:
    plt.figure(figsize=(8, 6))
    data.boxplot(column=[col])
    plt.title('Boxplot of ' + col)
    plt.show()
```

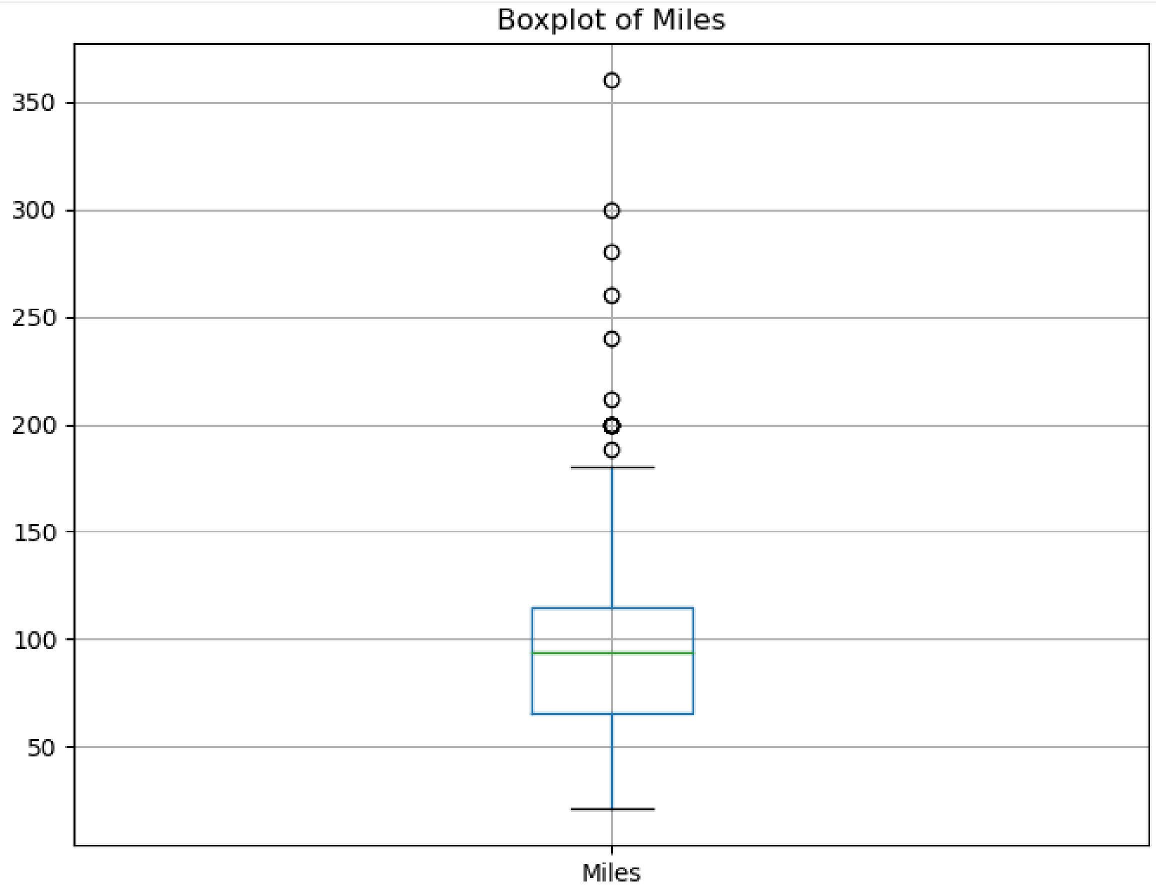

Boxplot of Age
50
45
40
35
30
25
20
Age


Boxplot of Education
20
18
16
14
12
Education


Boxplot of Usage
7
6
5
4
3
2
Usage

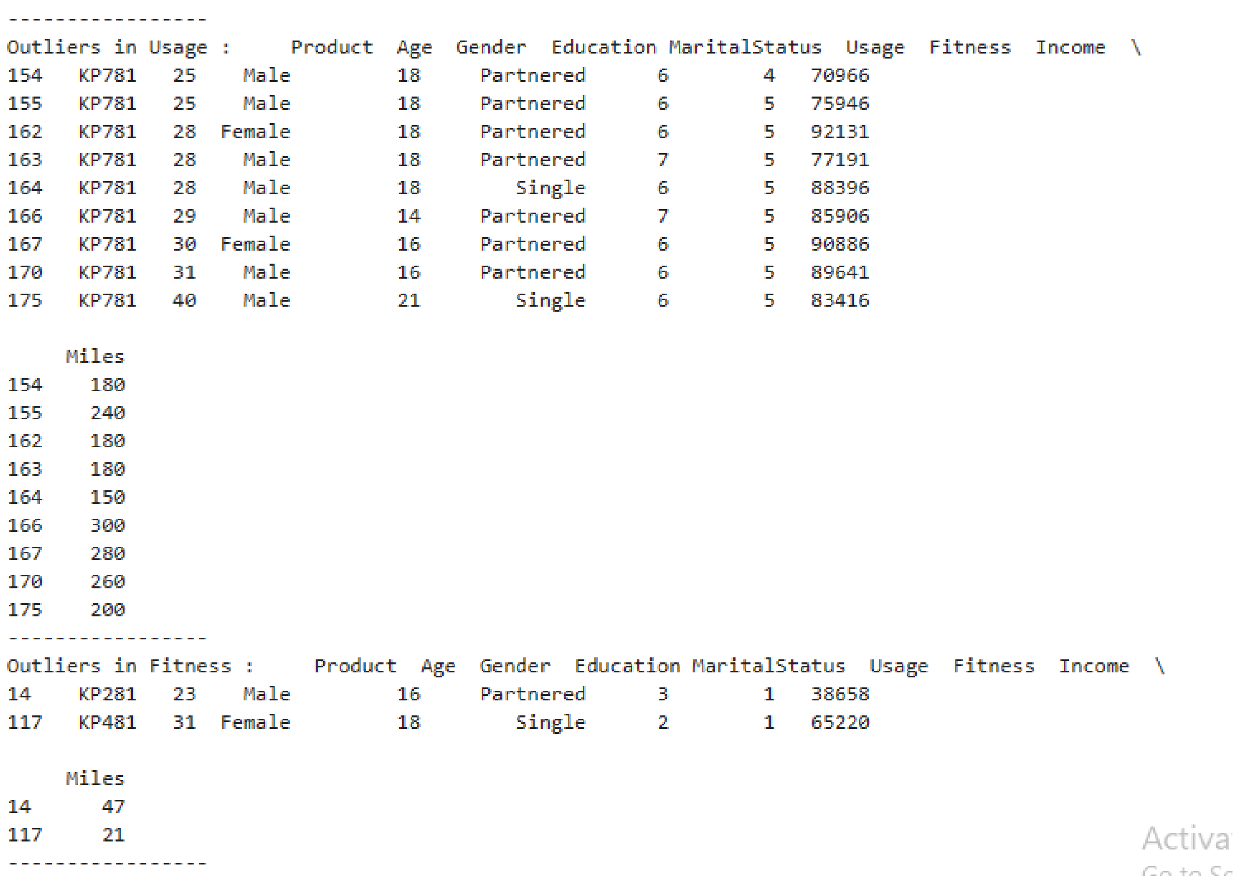

Boxplot of Fitness
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Fitness


Boxplot of Income
100000
90000
80000
70000
60000
50000
40000
30000
Income

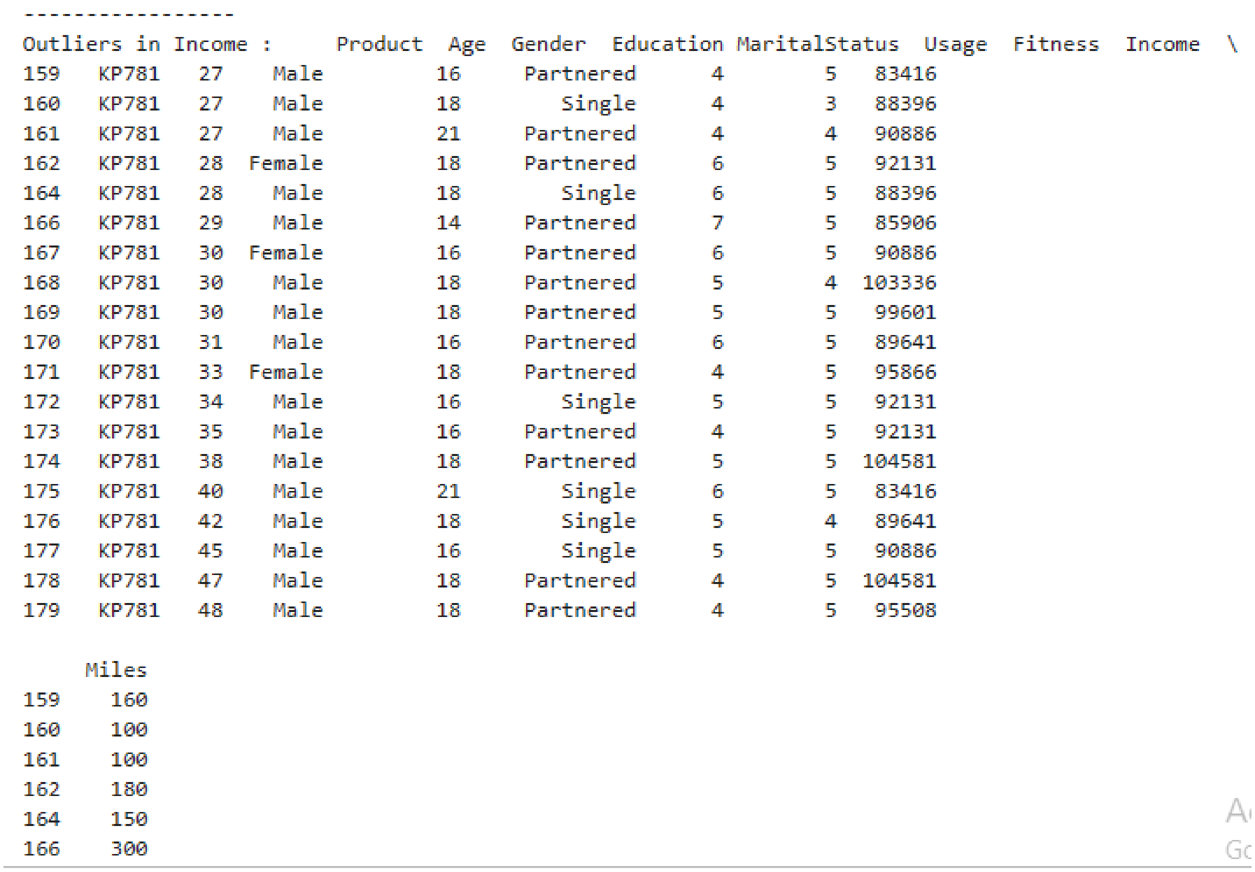


```
for col in numerical_columns:
    Q1 = data[col].quantile(0.25)
    Q3 = data[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = data[(data[col] < lower_bound) | (data[col] > upper_bound)]
    print("Outliers in", col, ":", outliers)
    print("-----------------")
```

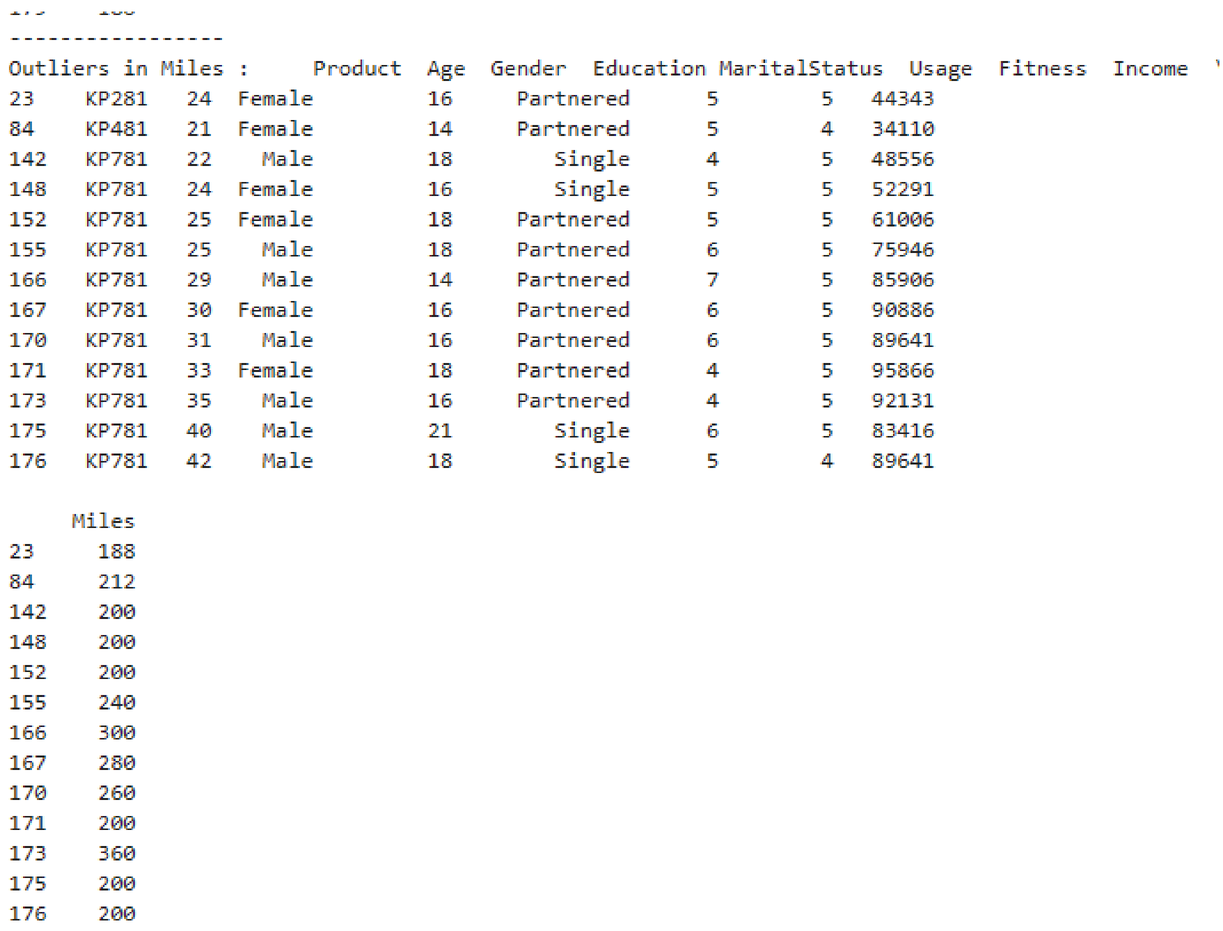
```
Outliers in Age :     Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
78     KP281   47    Male         16     Partnered      4        3   56850
79     KP281   50  Female         16     Partnered      3        3   64809
139    KP481   48    Male         16     Partnered      2        3   57987
178    KP781   47    Male         18     Partnered      4        5  104581
179    KP781   48    Male         18     Partnered      4        5   95508

     Miles
78      94
79      66
139     64
178    120
179    180
-----------------
```



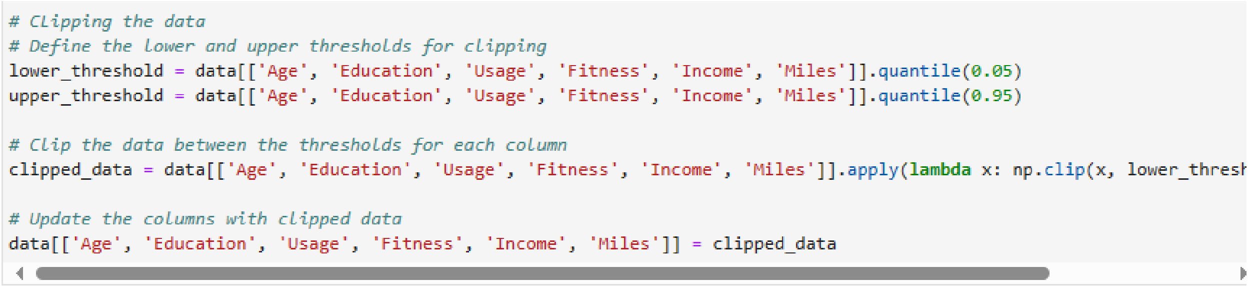
```
-----------------
Outliers in Usage :     Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
154    KP781   25    Male         18     Partnered      6        4   70966
155    KP781   25    Male         18     Partnered      6        5   75946
162    KP781   28  Female         18     Partnered      6        5   92131
163    KP781   28    Male         18     Partnered      7        5   77191
164    KP781   28    Male         18        Single      6        5   88396
166    KP781   29    Male         14     Partnered      7        5   85906
167    KP781   30  Female         16     Partnered      6        5   90886
170    KP781   31    Male         16     Partnered      6        5   89641
175    KP781   40    Male         21        Single      6        5   83416

     Miles
154    180
155    240
162    180
163    180
164    150
166    300
167    280
170    260
175    200
-----------------
Outliers in Fitness :     Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
14     KP281   23    Male         16     Partnered      3        1   38658
117    KP481   31  Female         18        Single      2        1   65220

     Miles
14      47
117     21
-----------------
```



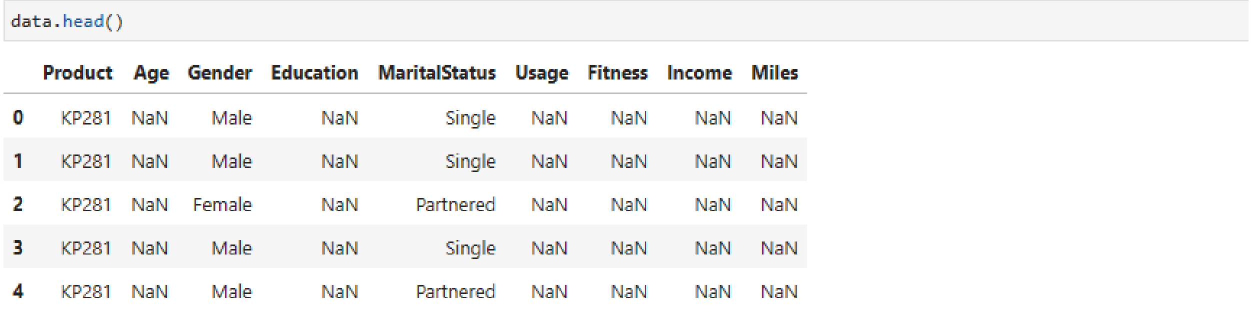
```
-----------------
Outliers in Income :     Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
159    KP781   27    Male         16     Partnered      4        5   83416
160    KP781   27    Male         18        Single      4        3   88396
161    KP781   27    Male         21     Partnered      4        4   90886
162    KP781   28  Female         18     Partnered      6        5   92131
164    KP781   28    Male         18        Single      6        5   88396
166    KP781   29    Male         14     Partnered      7        5   85906
167    KP781   30  Female         16     Partnered      6        5   90886
168    KP781   30    Male         18     Partnered      5        4  103336
169    KP781   30    Male         18     Partnered      5        5   99601
170    KP781   31    Male         16     Partnered      6        5   89641
171    KP781   33  Female         18     Partnered      4        5   95866
172    KP781   34    Male         16        Single      5        5   92131
173    KP781   35    Male         16     Partnered      4        5   92131
174    KP781   38    Male         18     Partnered      5        5  104581
175    KP781   40    Male         21        Single      6        5   83416
176    KP781   42    Male         18        Single      5        4   89641
177    KP781   45    Male         16        Single      5        5   90886
178    KP781   47    Male         18     Partnered      4        5  104581
179    KP781   48    Male         18     Partnered      4        5   95508

     Miles
159    160
160    100
161    100
162    180
164    150
166    300
```



```
-----------------
Outliers in Miles :     Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  '
23     KP281    24   Female         16      Partnered       5        5    44343
84     KP481    21   Female         14      Partnered       5        4    34110
142    KP781    22     Male         18         Single       4        5    48556
148    KP781    24   Female         16         Single       5        5    52291
152    KP781    25   Female         18      Partnered       5        5    61006
155    KP781    25     Male         18      Partnered       6        5    75946
166    KP781    29     Male         14      Partnered       7        5    85906
167    KP781    30   Female         16      Partnered       6        5    90886
170    KP781    31     Male         16      Partnered       6        5    89641
171    KP781    33   Female         18      Partnered       4        5    95866
173    KP781    35     Male         16      Partnered       4        5    92131
175    KP781    40     Male         21         Single       6        5    83416
176    KP781    42     Male         18         Single       5        4    89641

     Miles
23     188
84     212
142    200
148    200
152    200
155    240
166    300
167    280
170    260
171    200
173    360
175    200
176    200
```



```
# Clipping the data
# Define the lower and upper thresholds for clipping
lower_threshold = data[['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']].quantile(0.05)
upper_threshold = data[['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']].quantile(0.95)

# Clip the data between the thresholds for each column
clipped_data = data[['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']].apply(lambda x: np.clip(x, lower_thresh

# Update the columns with clipped data
data[['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']] = clipped_data
```



```
data.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	NaN	Male	NaN	Single	NaN	NaN	NaN	NaN
1	KP281	NaN	Male	NaN	Single	NaN	NaN	NaN	NaN
2	KP281	NaN	Female	NaN	Partnered	NaN	NaN	NaN	NaN
3	KP281	NaN	Male	NaN	Single	NaN	NaN	NaN	NaN
4	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN

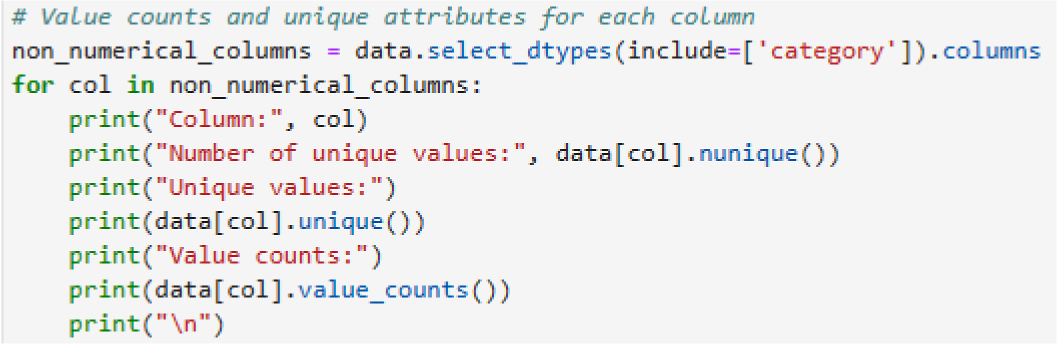
Clipping of data is giving us NaN values so skipped this step.

The presence of outliers aligns with the goals and objectives of our research, as they represent rare or extreme events that are integral to understanding the variability within the dataset. By retaining these outliers, we ensure that our analysis captures the full spectrum of potential outcomes and phenomena, allowing for a more comprehensive understanding of the data.

Additionally, we have verified the authenticity and accuracy of outliers within the dataset to confirm that they are genuine data points and not the result of errors, measurement inaccuracies, or data entry mistakes. This validation process assures us that the outliers hold meaningful insights and contribute valuable information to our analysis.

Given these considerations, we have chosen not to remove the outliers from the dataset. Doing so would risk overlooking important nuances and potentially biasing our analysis towards more conventional or central tendencies, thereby limiting the richness of our findings and compromising the integrity of our research outcomes. By retaining the outliers, we maintain the fidelity and completeness of our dataset, enabling us to explore and interpret the full range of possibilities within the data.

NON GRAPHICAL ANALYSIS: In this section, we will analyze the value counts and uniqueness of the data. This step is crucial for understanding the distribution and frequency of values within each attribute and identifying any patterns or inconsistencies in the dataset.

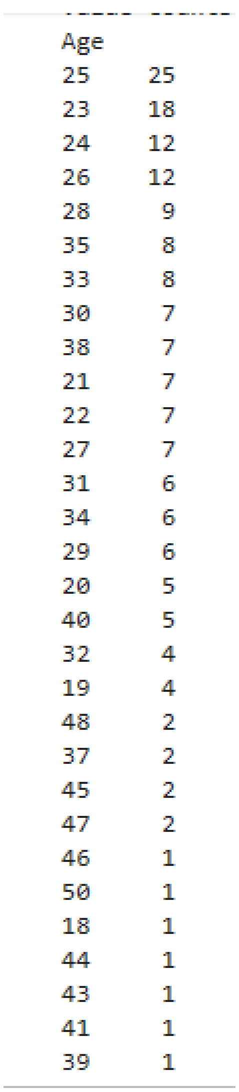


```
# Value counts and unique attributes for each column
non_numerical_columns = data.select_dtypes(include=['category']).columns
for col in non_numerical_columns:
    print("Column:", col)
    print("Number of unique values:", data[col].nunique())
    print("Unique values:")
    print(data[col].unique())
    print("Value counts:")
    print(data[col].value_counts())
    print("\n")
```

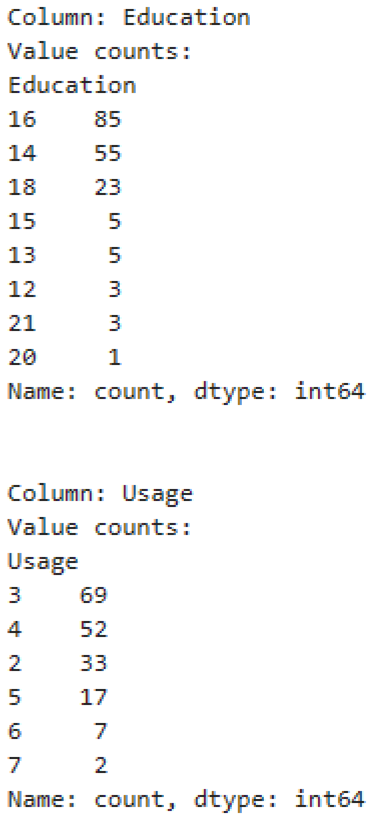
```
Column: Product
Number of unique values: 3
Unique values:
['KP281', 'KP481', 'KP781']
Categories (3, object): ['KP281', 'KP481', 'KP781']
Value counts:
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64


Column: Gender
Number of unique values: 2
Unique values:
['Male', 'Female']
Categories (2, object): ['Female', 'Male']
Value counts:
Gender
Male      104
Female     76
Name: count, dtype: int64
```

```
Column: MaritalStatus
Number of unique values: 2
Unique values:
['Single', 'Partnered']
Categories (2, object): ['Partnered', 'Single']
Value counts:
MaritalStatus
Partnered    107
Single        73
Name: count, dtype: int64
```

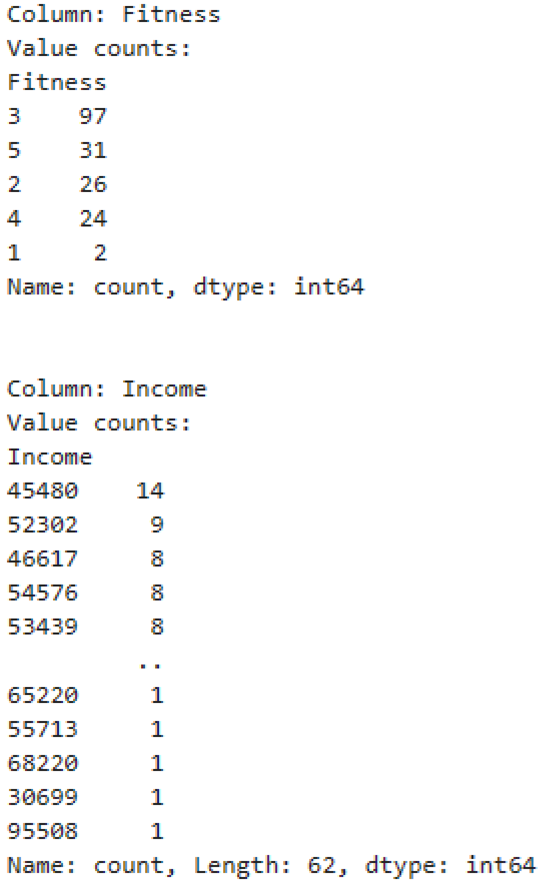
```python
for col in numerical_columns:
    print("Column:", col)
    print("Value counts:")
    print(data[col].value_counts())
    print("\n")
```



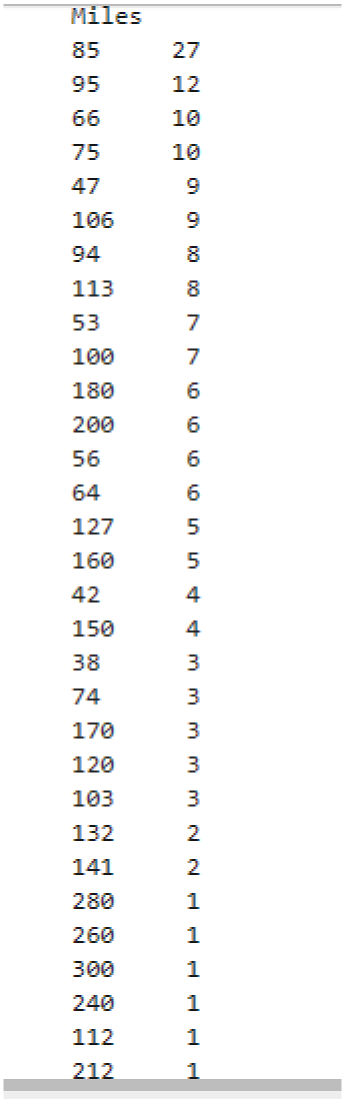
```
Age
25    25
23    18
24    12
26    12
28     9
35     8
33     8
30     7
38     7
21     7
22     7
27     7
31     6
34     6
29     6
20     5
40     5
32     4
19     4
48     2
37     2
45     2
47     2
46     1
50     1
18     1
44     1
43     1
41     1
39     1
```



```
Column: Education
Value counts:
Education
16    85
14    55
18    23
15     5
13     5
12     3
21     3
20     1
Name: count, dtype: int64


Column: Usage
Value counts:
Usage
3    69
4    52
2    33
5    17
6     7
7     2
Name: count, dtype: int64
```



```
Column: Fitness
Value counts:
Fitness
3    97
5    31
2    26
4    24
1     2
Name: count, dtype: int64


Column: Income
Value counts:
Income
45480    14
52302     9
46617     8
54576     8
53439     8
         ..
65220     1
55713     1
68220     1
30699     1
95508     1
Name: count, Length: 62, dtype: int64
```

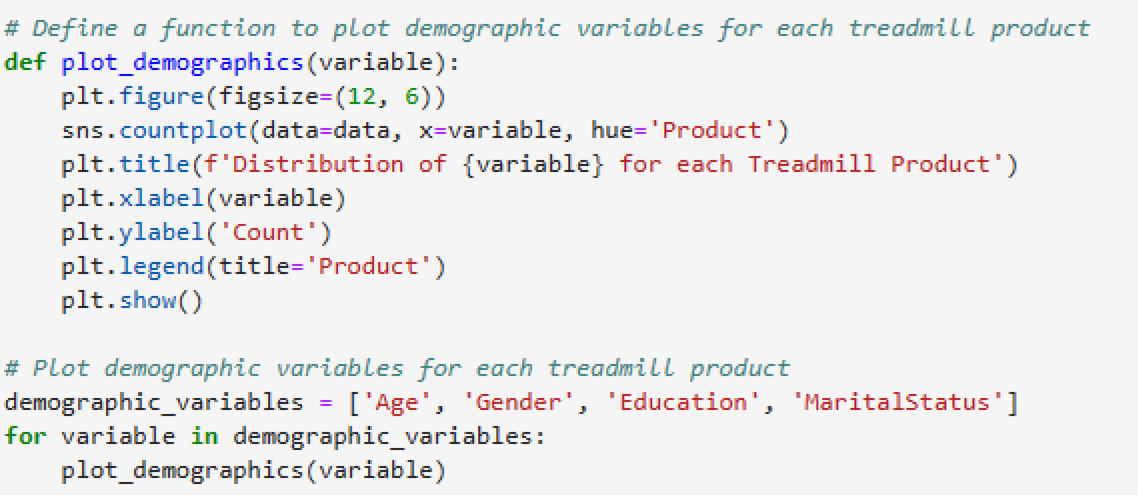


```
Miles
85     27
95     12
66     10
75     10
47      9
106     9
94      8
113     8
53      7
100     7
180     6
200     6
56      6
64      6
127     5
160     5
42      4
150     4
38      3
74      3
170     3
120     3
103     3
132     2
141     2
280     1
260     1
300     1
240     1
112     1
212     1
```

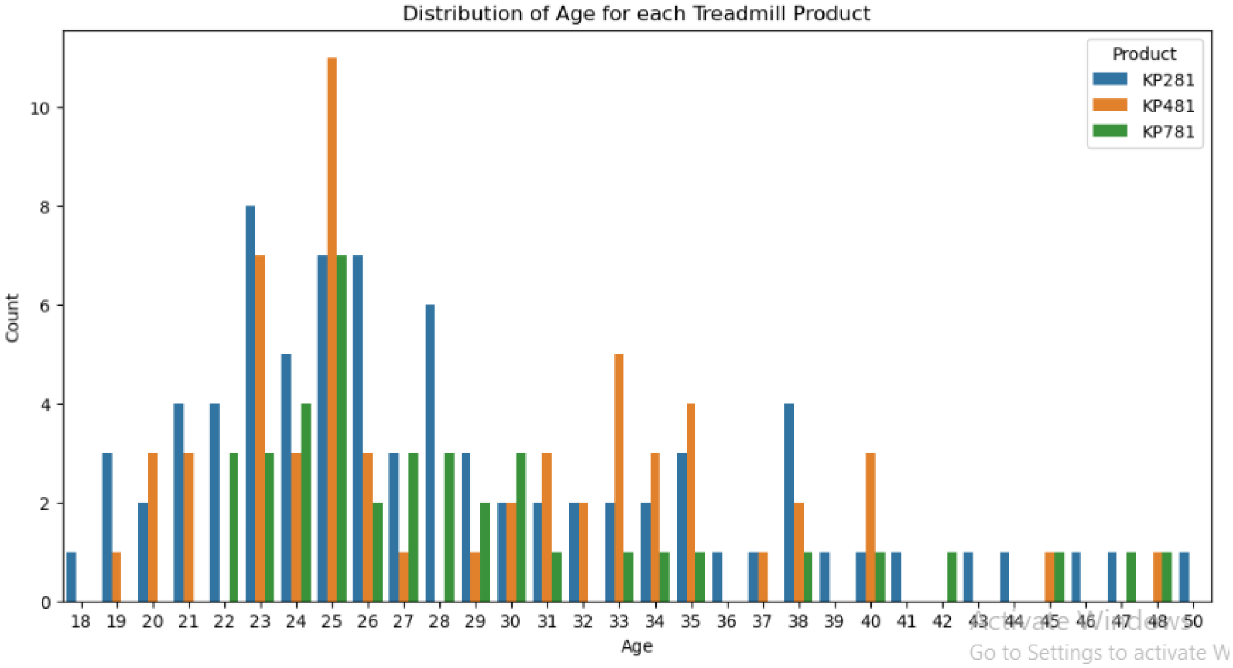
Upon analyzing the dataset, it's evident that the data pertains to three models of treadmills, with KP281 being the most commonly sold model.

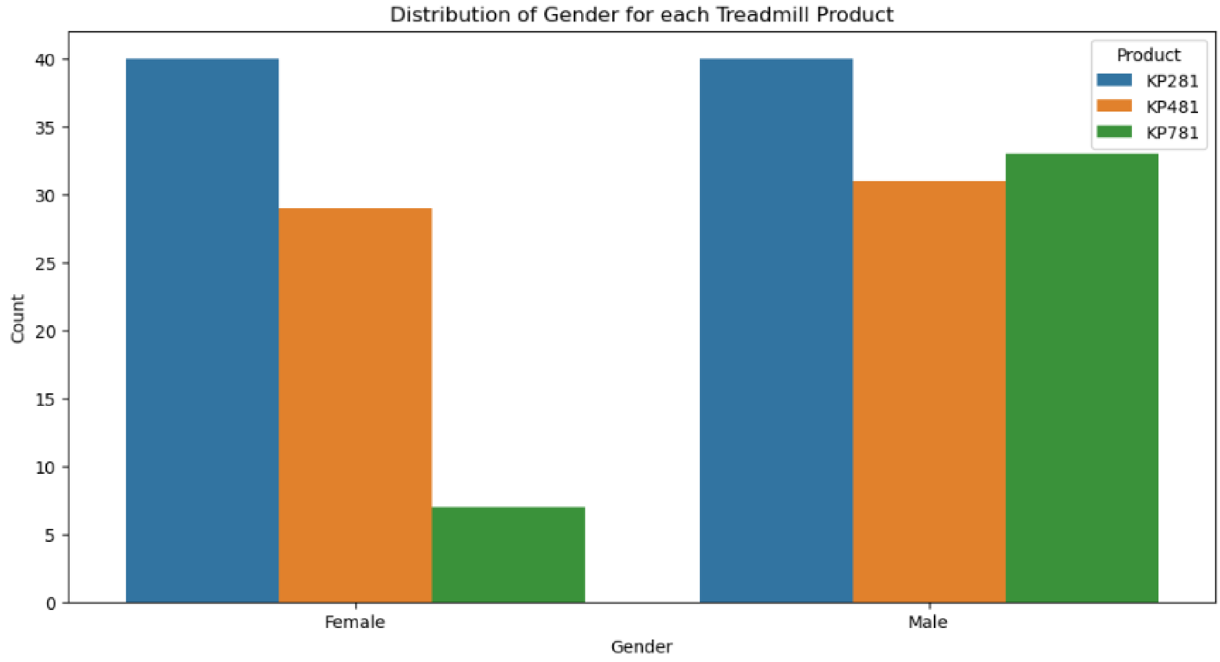
VISUAL ANALYSIS: Let's delve into visual analysis to address the questions we've outlined while analyzing the basic metrics of the dataset.

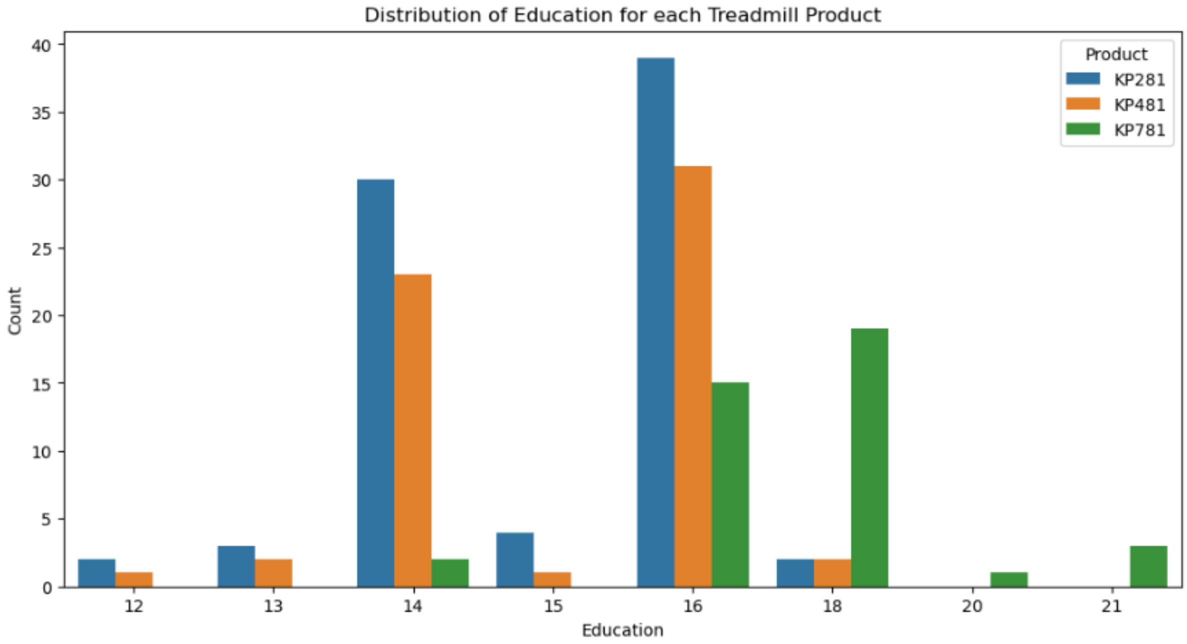
- **Demographic Characteristics Analysis:** We will analyze key demographic variables such as age, gender, education, marital status, and income for each treadmill product to understand the profile of customers purchasing AeroFit treadmills.

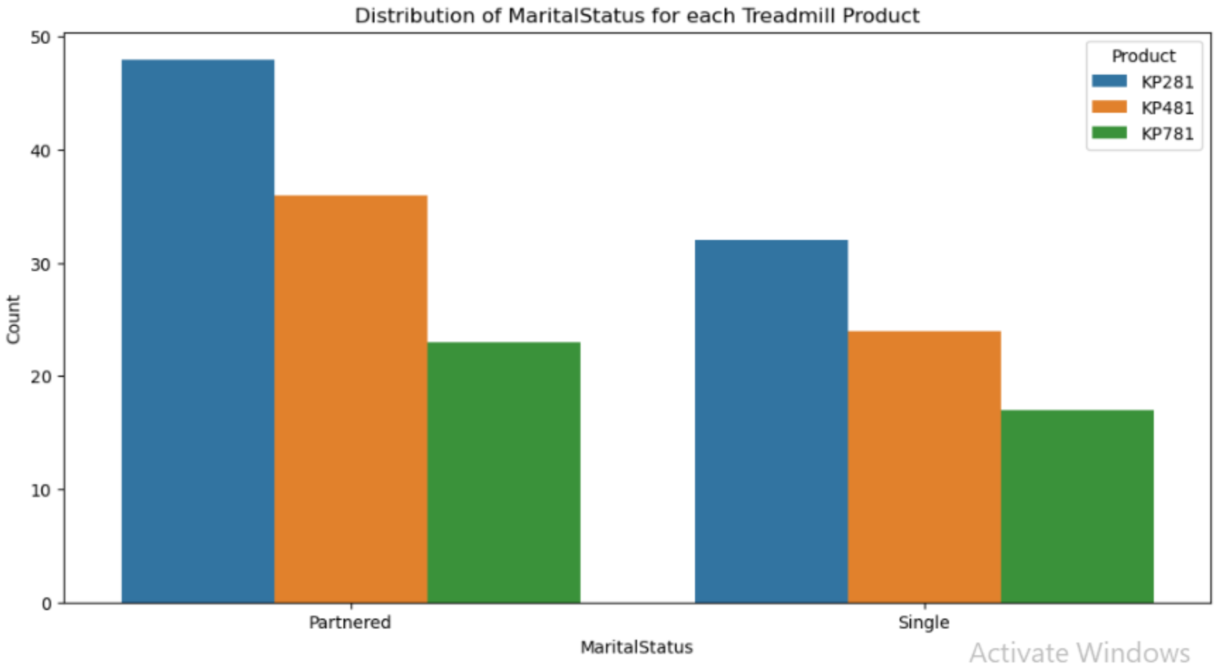


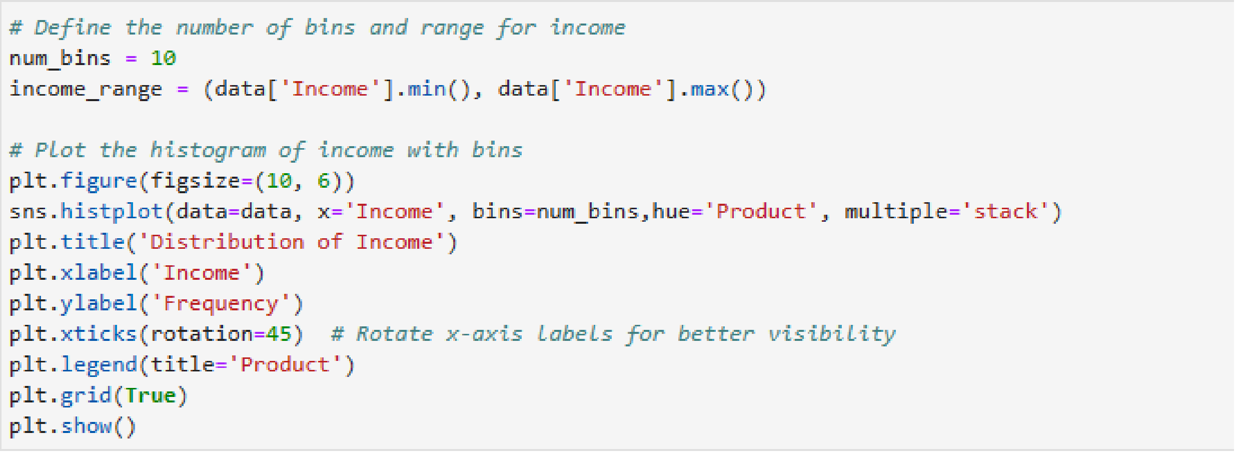
```
# Define a function to plot demographic variables for each treadmill product
def plot_demographics(variable):
    plt.figure(figsize=(12, 6))
    sns.countplot(data=data, x=variable, hue='Product')
    plt.title(f'Distribution of {variable} for each Treadmill Product')
    plt.xlabel(variable)
    plt.ylabel('Count')
    plt.legend(title='Product')
    plt.show()

# Plot demographic variables for each treadmill product
demographic_variables = ['Age', 'Gender', 'Education', 'MaritalStatus']
for variable in demographic_variables:
    plot_demographics(variable)
```



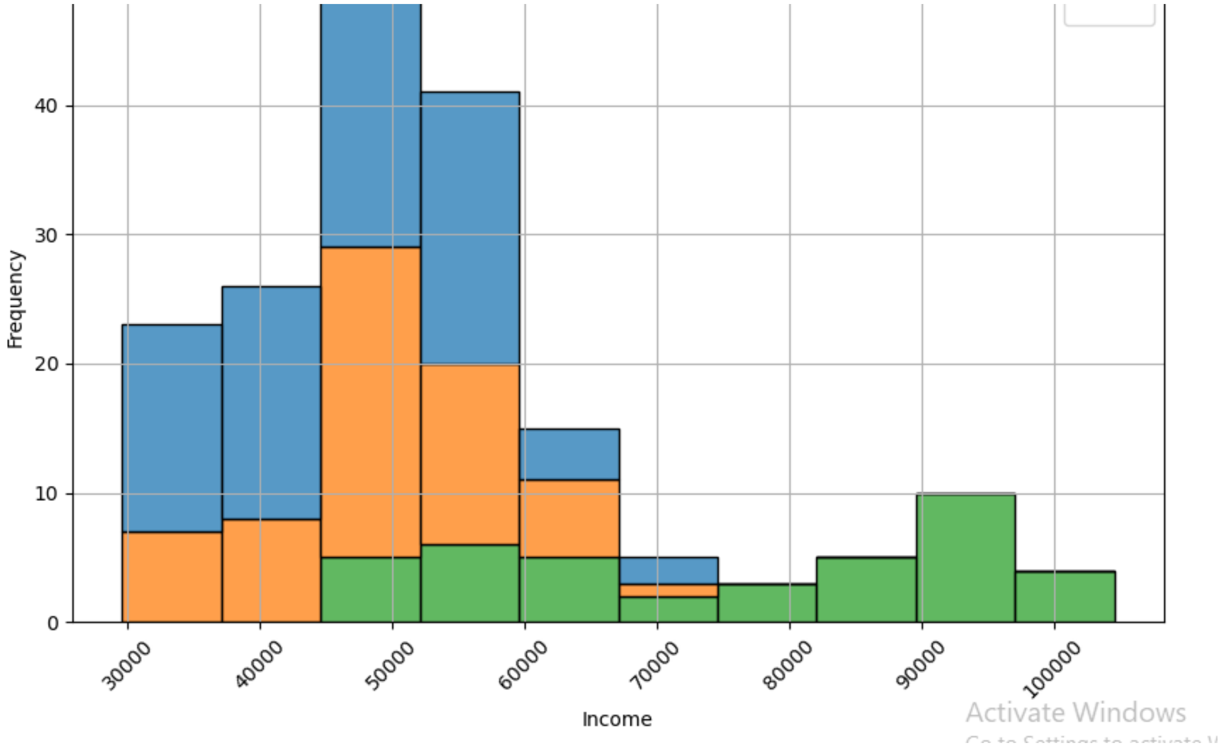

Distribution of Gender for each Treadmill Product


Distribution of Education for each Treadmill Product





```
# Define the number of bins and range for income
num_bins = 10
income_range = (data['Income'].min(), data['Income'].max())

# Plot the histogram of income with bins
plt.figure(figsize=(10, 6))
sns.histplot(data=data, x='Income', bins=num_bins,hue='Product', multiple='stack')
plt.title('Distribution of Income')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.xticks(rotation=45)  # Rotate x-axis labels for better visibility
plt.legend(title='Product')
plt.grid(True)
plt.show()
```



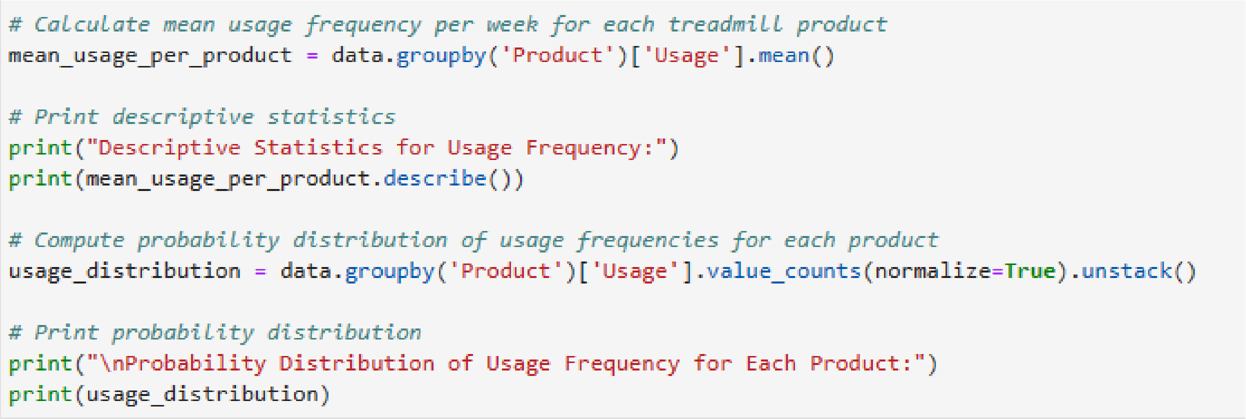
<u>Insights:</u> On average, the highest number of treadmill sales across all three models is observed in the age group of 23-26. Specifically, for KP481, the maximum sales were recorded among individuals aged 25, totaling approximately 12 units. The second-highest sales were attributed to KP281, with approximately 8 units sold among the 23-year-olds. Meanwhile, for KP781, the peak sales occurred in the 25-year-old age group, with around 7 units sold.

In terms of gender distribution, KP281 emerges as the most popular choice, with approximately 40 units sold to both males and females. KP481 follows as the second most favored model, with approximately 28 units sold to both genders. Interestingly, KP781 exhibits a gender preference, with around 32 units sold to males and only 5 units sold to females.

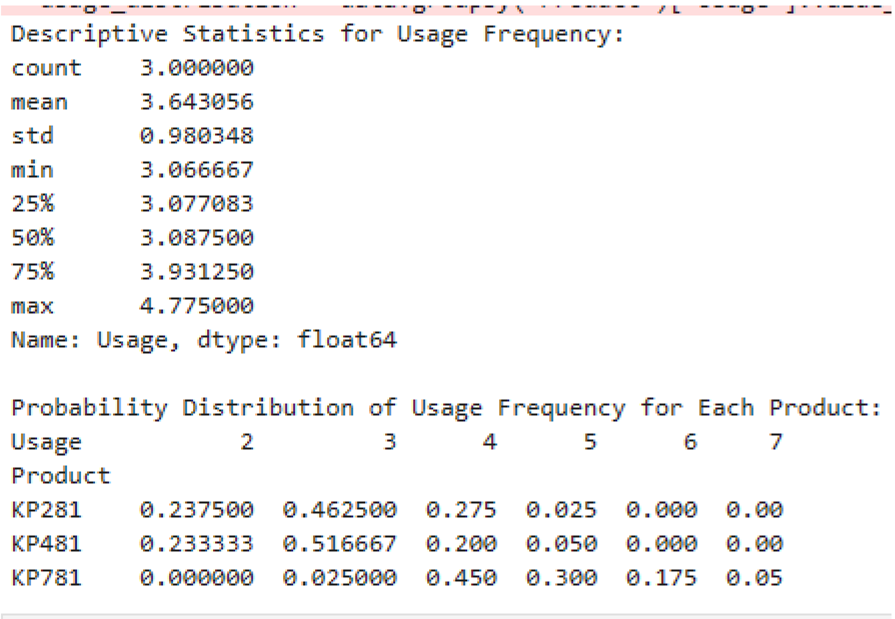
Regarding education level, individuals with 16 years of education show the highest purchasing trend across all models, with KP281 leading with approximately 40 units sold. However, for KP781, the maximum sales were observed among those with 18 years of education, totaling around 17 units. Additionally, individuals with 20-21 years of education exclusively prefer KP781.

In terms of marital status, partnered individuals exhibit a higher likelihood of purchasing treadmills, with KP281 being the most favored model, recording approximately 48 units sold. Conversely, individuals with a salary ranging from $30,000 to $70,000 tend to prefer KP281 and then KP481. On the other hand, those with a salary between $75,000 and $150,000 show a preference for KP781.

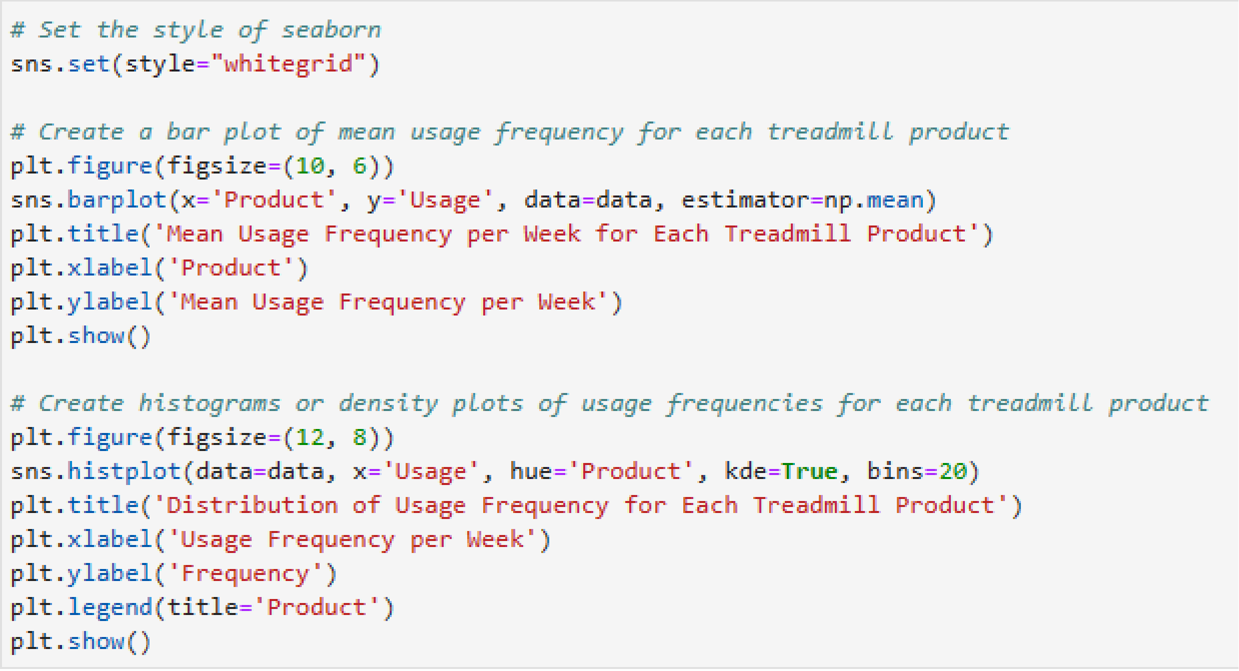
- **Usage Patterns Variation:** Examine the average usage frequency per week for each treadmill product to identify variations in usage behavior.



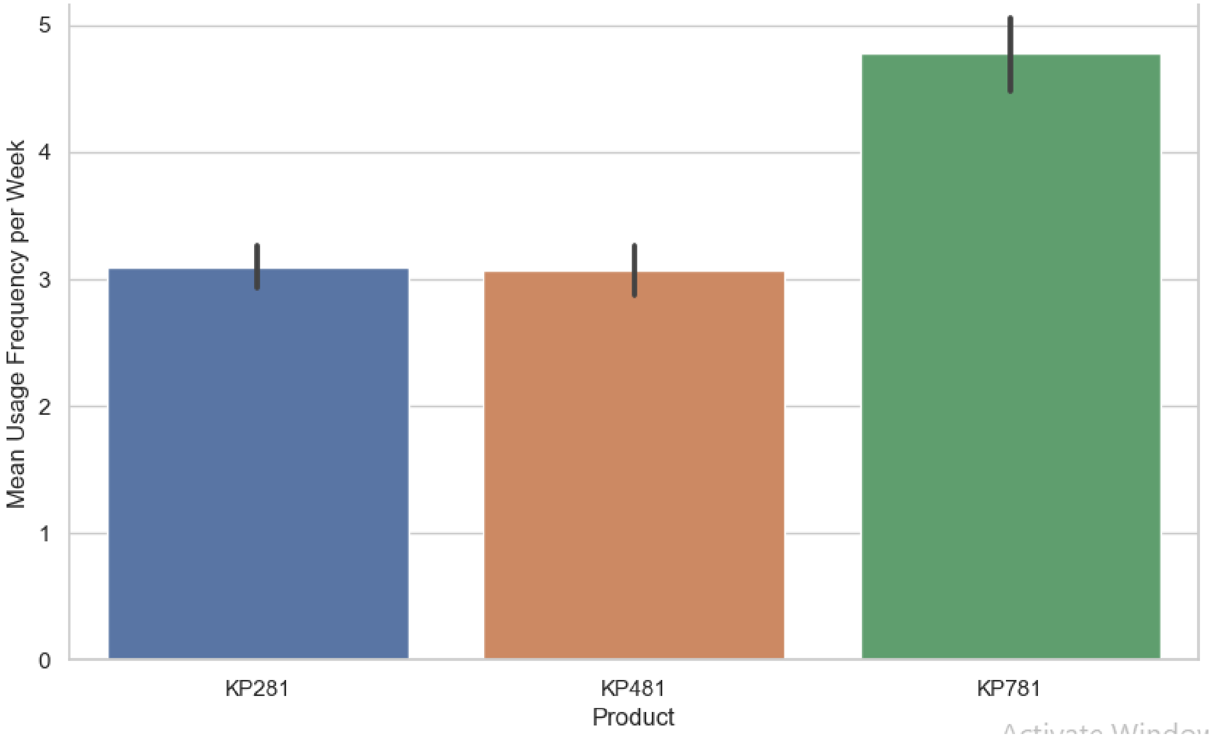
```python
# Calculate mean usage frequency per week for each treadmill product
mean_usage_per_product = data.groupby('Product')['Usage'].mean()

# Print descriptive statistics
print("Descriptive Statistics for Usage Frequency:")
print(mean_usage_per_product.describe())

# Compute probability distribution of usage frequencies for each product
usage_distribution = data.groupby('Product')['Usage'].value_counts(normalize=True).unstack()

# Print probability distribution
print("\nProbability Distribution of Usage Frequency for Each Product:")
print(usage_distribution)
```

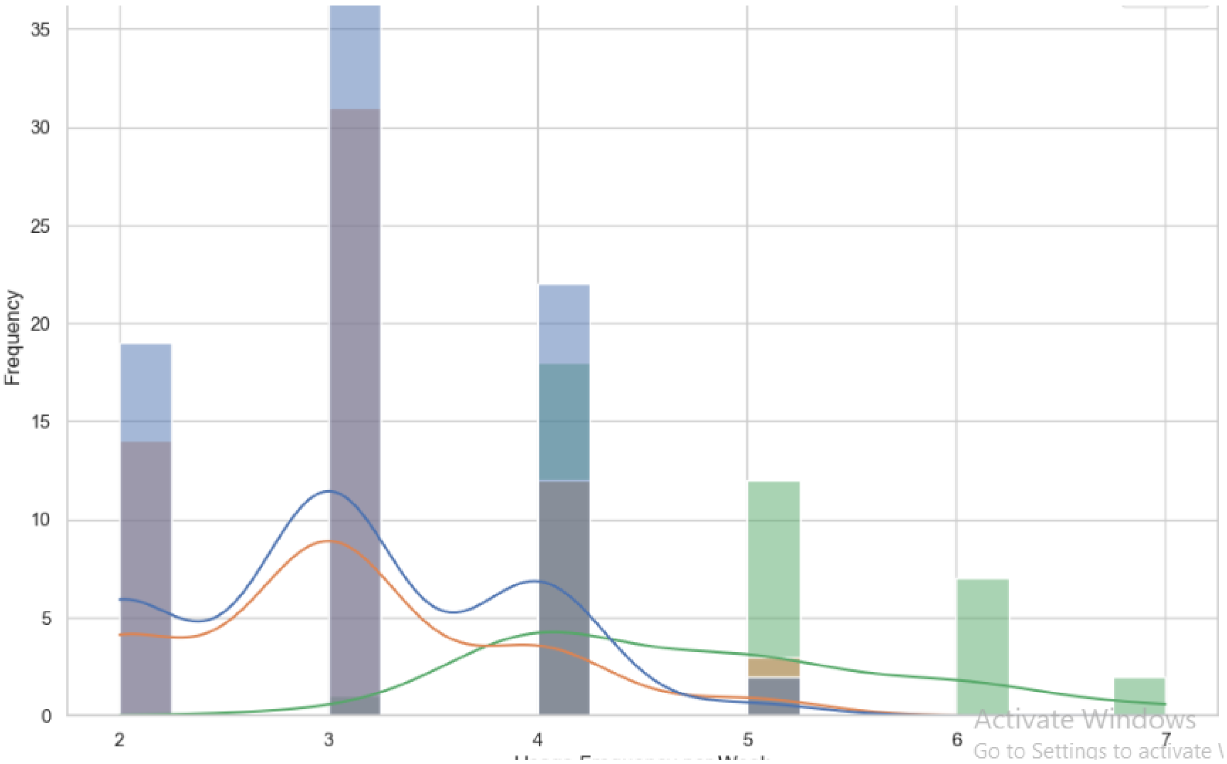


```
Descriptive Statistics for Usage Frequency:
count    3.000000
mean     3.643056
std      0.980348
min      3.066667
25%      3.077083
50%      3.087500
75%      3.931250
max      4.775000
Name: Usage, dtype: float64

Probability Distribution of Usage Frequency for Each Product:
Usage           2         3      4      5      6     7
Product
KP281    0.237500  0.462500  0.275  0.025  0.000  0.00
KP481    0.233333  0.516667  0.200  0.050  0.000  0.00
KP781    0.000000  0.025000  0.450  0.300  0.175  0.05
```



```
# Set the style of seaborn
sns.set(style="whitegrid")

# Create a bar plot of mean usage frequency for each treadmill product
plt.figure(figsize=(10, 6))
sns.barplot(x='Product', y='Usage', data=data, estimator=np.mean)
plt.title('Mean Usage Frequency per Week for Each Treadmill Product')
plt.xlabel('Product')
plt.ylabel('Mean Usage Frequency per Week')
plt.show()

# Create histograms or density plots of usage frequencies for each treadmill product
plt.figure(figsize=(12, 8))
sns.histplot(data=data, x='Usage', hue='Product', kde=True, bins=20)
plt.title('Distribution of Usage Frequency for Each Treadmill Product')
plt.xlabel('Usage Frequency per Week')
plt.ylabel('Frequency')
plt.legend(title='Product')
plt.show()
```





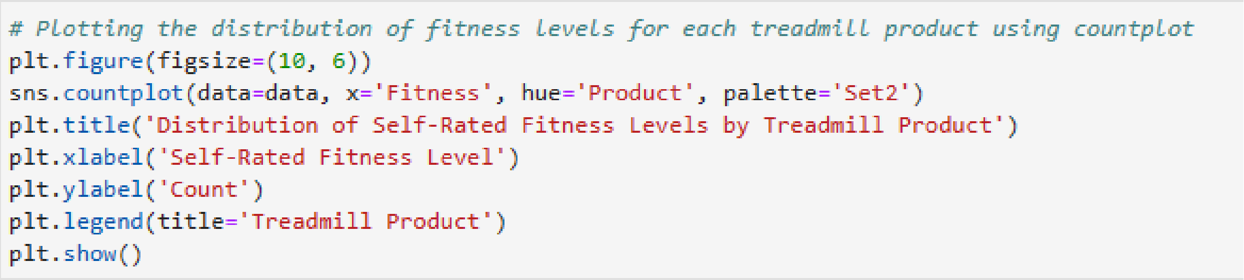
Insights: The mean usage per week for KP781 stands out as the highest among all models, averaging around 5, while the other two models maintain a similar mean usage of 3.

When considering the probabilities of usage, KP281 and KP481 show their highest probabilities during the third week, with KP481 at 51% and KP281 at 46%. In contrast, KP781 exhibits its highest probability of usage during the fourth week, amounting to 45%. Notably, KP481 maintains the highest probability of being used during the third week.

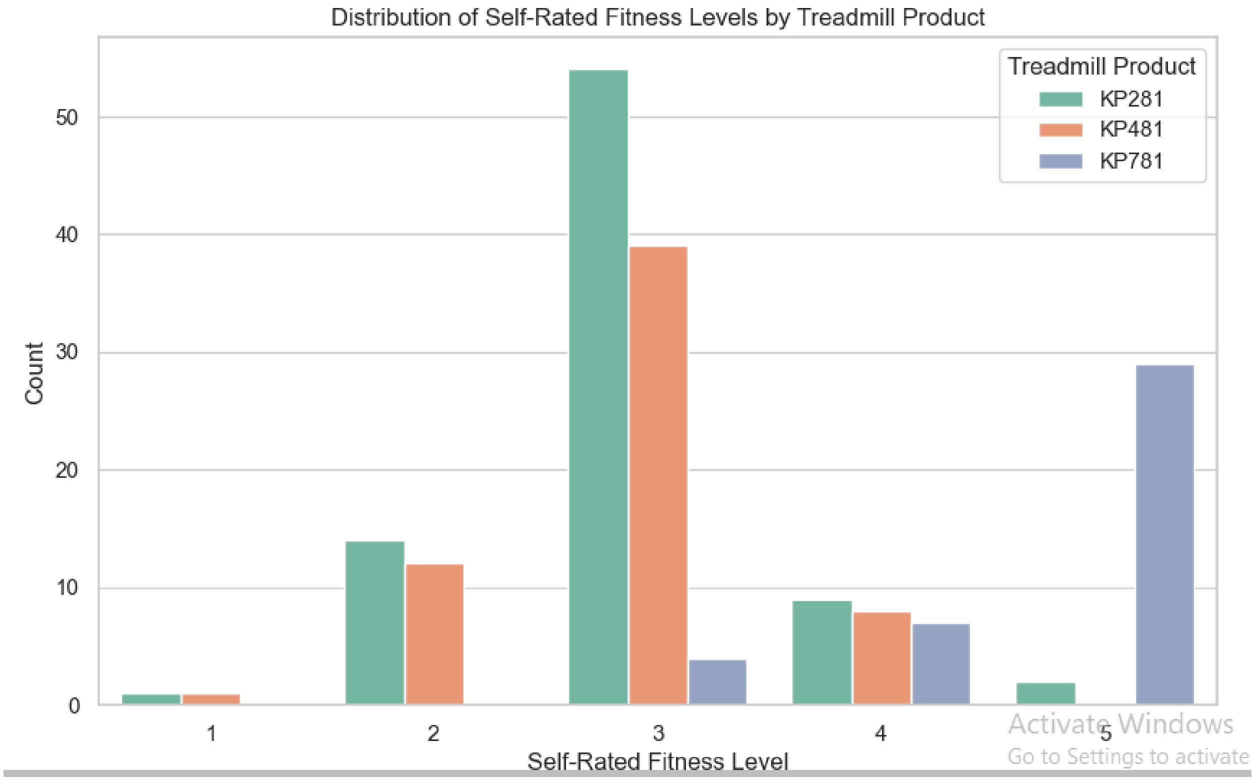
However, during the sixth and seventh weeks, only KP781 retains any probability of usage, while the probabilities for the other two models drop to 0%.

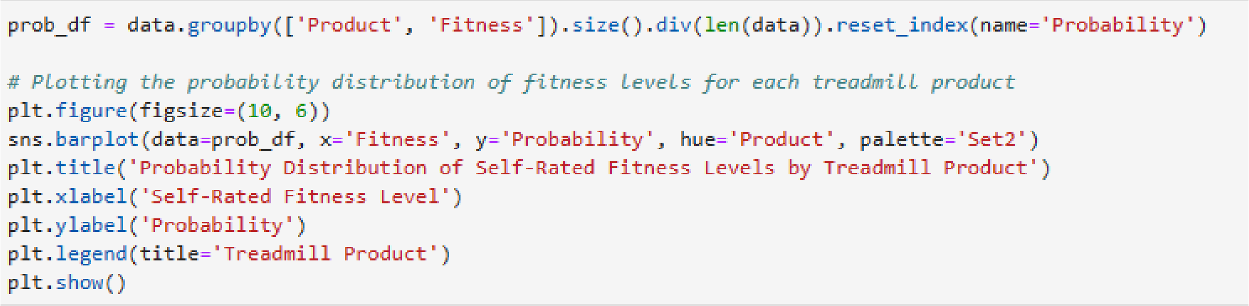
Examining the highest frequency of usage per week reveals that KP281 and KP481 peak during the third week, followed by a gradual decline to zero by the sixth and seventh weeks. In contrast, KP781 shows a distinct pattern with a sudden rise in usage observed around the fourth week, which then remains relatively stable with a slight decrease until the seventh week.

- **Self – Rated Fitness Level Analysis:** Determine the distribution of self-rated fitness levels among customers for each treadmill product to understand fitness preferences.

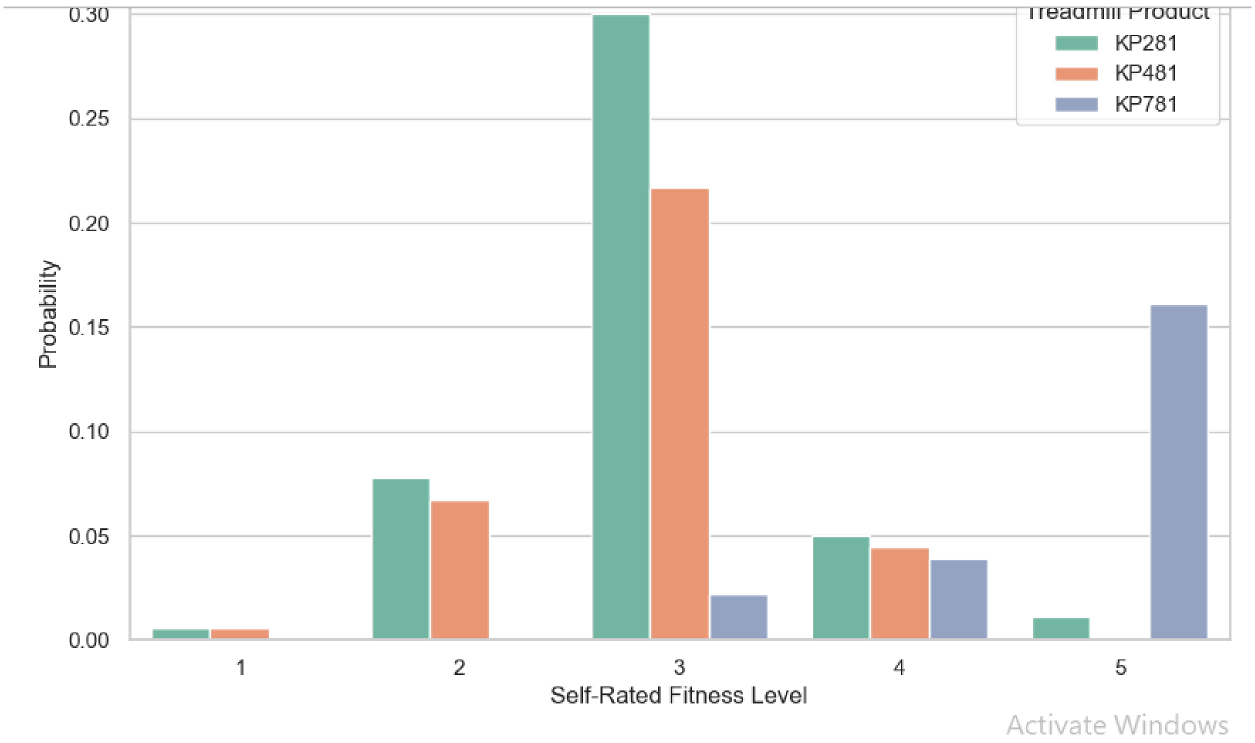


```
# Plotting the distribution of fitness levels for each treadmill product using countplot
plt.figure(figsize=(10, 6))
sns.countplot(data=data, x='Fitness', hue='Product', palette='Set2')
plt.title('Distribution of Self-Rated Fitness Levels by Treadmill Product')
plt.xlabel('Self-Rated Fitness Level')
plt.ylabel('Count')
plt.legend(title='Treadmill Product')
plt.show()
```





```
prob_df = data.groupby(['Product', 'Fitness']).size().div(len(data)).reset_index(name='Probability')

# Plotting the probability distribution of fitness levels for each treadmill product
plt.figure(figsize=(10, 6))
sns.barplot(data=prob_df, x='Fitness', y='Probability', hue='Product', palette='Set2')
plt.title('Probability Distribution of Self-Rated Fitness Levels by Treadmill Product')
plt.xlabel('Self-Rated Fitness Level')
plt.ylabel('Probability')
plt.legend(title='Treadmill Product')
plt.show()
```

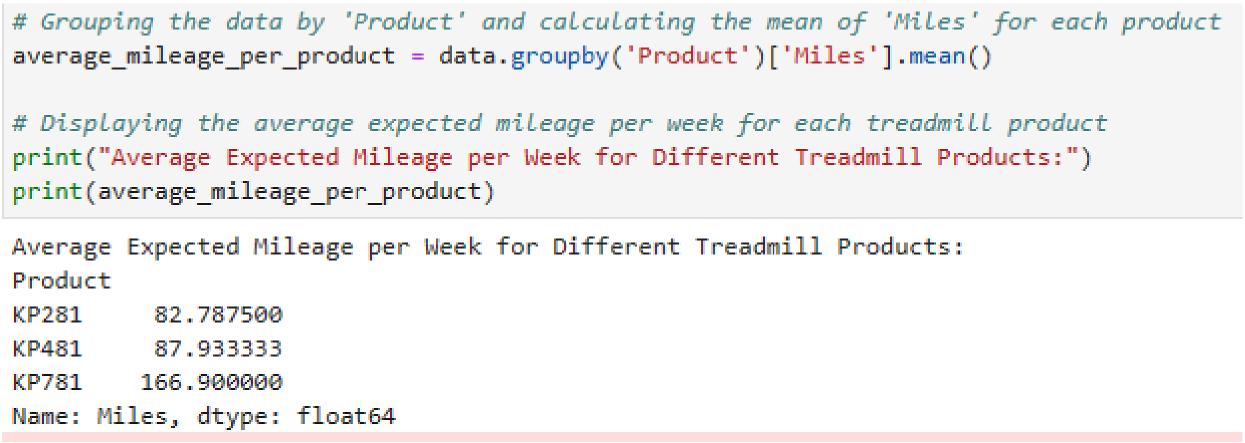


Insights: Based on the data analysis, we observe that the highest rating of 5 is obtained by KP781, with approximately 28 ratings and a probability of around 0.16. However, if we consider the maximum number of ratings, KP281 emerges as the leader with around 55 ratings and a probability of approximately 0.30, despite receiving an average rating of 3. This suggests that while KP781 excels in receiving the highest individual ratings, KP281 remains the preferred choice in terms of the sheer number of users.

Moreover, at the rating of 4, KP281 again demonstrates the highest ratings and probability, although the other two treadmill models also receive considerable ratings, closely competing with KP281.

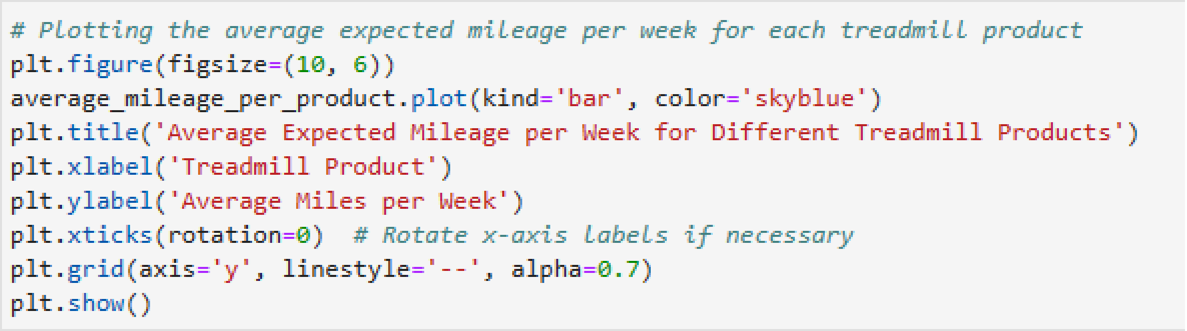
In conclusion, if a customer prioritizes the highest rating, KP781 stands out as the preferred choice. However, if a rating of 4 is also considered satisfactory, all three treadmill options become viable. This indicates that while KP781 may excel in achieving top ratings, KP281 remains a popular choice among customers, considering both ratings and user preferences.

- **Expected Mileage Differences:** Compare the average expected mileage per week for different treadmill products to assess variations in usage expectations.

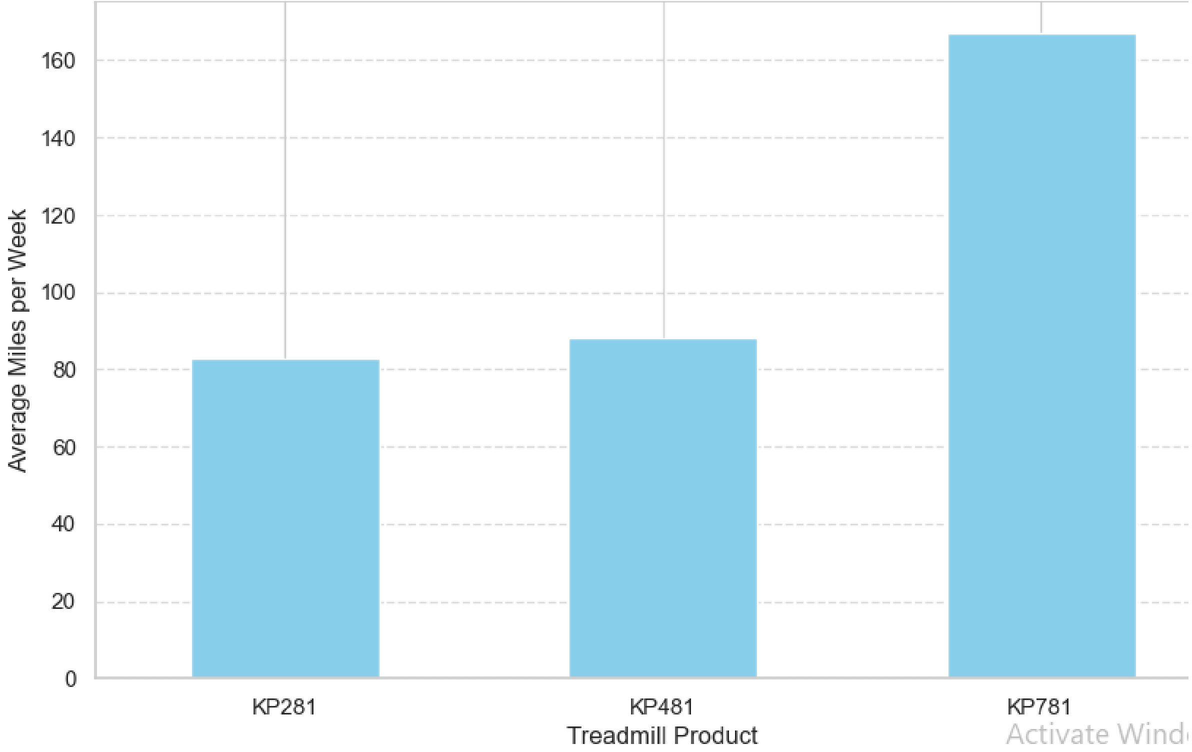


```
# Grouping the data by 'Product' and calculating the mean of 'Miles' for each product
average_mileage_per_product = data.groupby('Product')['Miles'].mean()

# Displaying the average expected mileage per week for each treadmill product
print("Average Expected Mileage per Week for Different Treadmill Products:")
print(average_mileage_per_product)
```

```
Average Expected Mileage per Week for Different Treadmill Products:
Product
KP281     82.787500
KP481     87.933333
KP781    166.900000
Name: Miles, dtype: float64
```

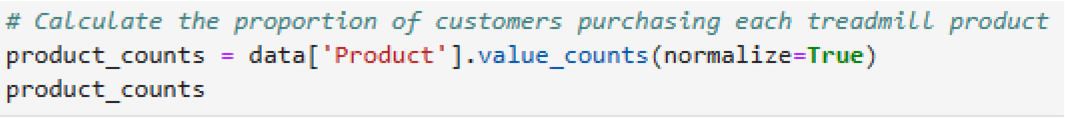


```
# Plotting the average expected mileage per week for each treadmill product
plt.figure(figsize=(10, 6))
average_mileage_per_product.plot(kind='bar', color='skyblue')
plt.title('Average Expected Mileage per Week for Different Treadmill Products')
plt.xlabel('Treadmill Product')
plt.ylabel('Average Miles per Week')
plt.xticks(rotation=0)  # Rotate x-axis labels if necessary
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

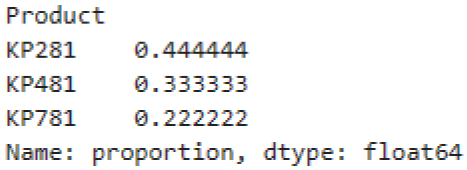


Insights: The highest average mileage per week is observed for KP781, standing at 166.90 miles per week, nearly double the average mileage of both other models. Conversely, both other models, KP281 and KP481, exhibit a comparable average mileage of approximately 80 miles per week.

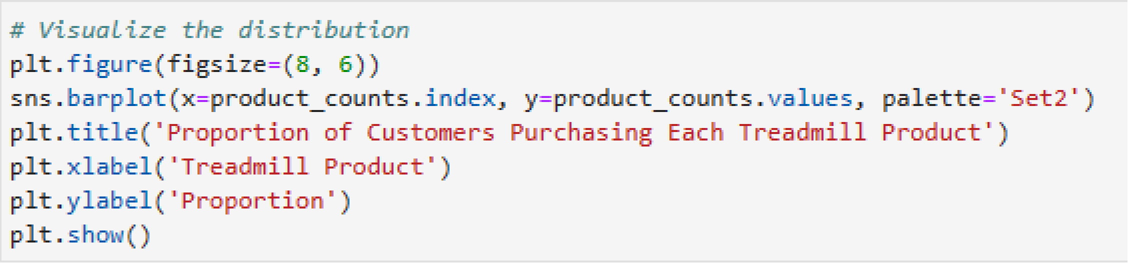
- **Customer Preferences Distribution:** Calculate the proportion of customers purchasing each treadmill product and visualize the distribution to identify popular choices.



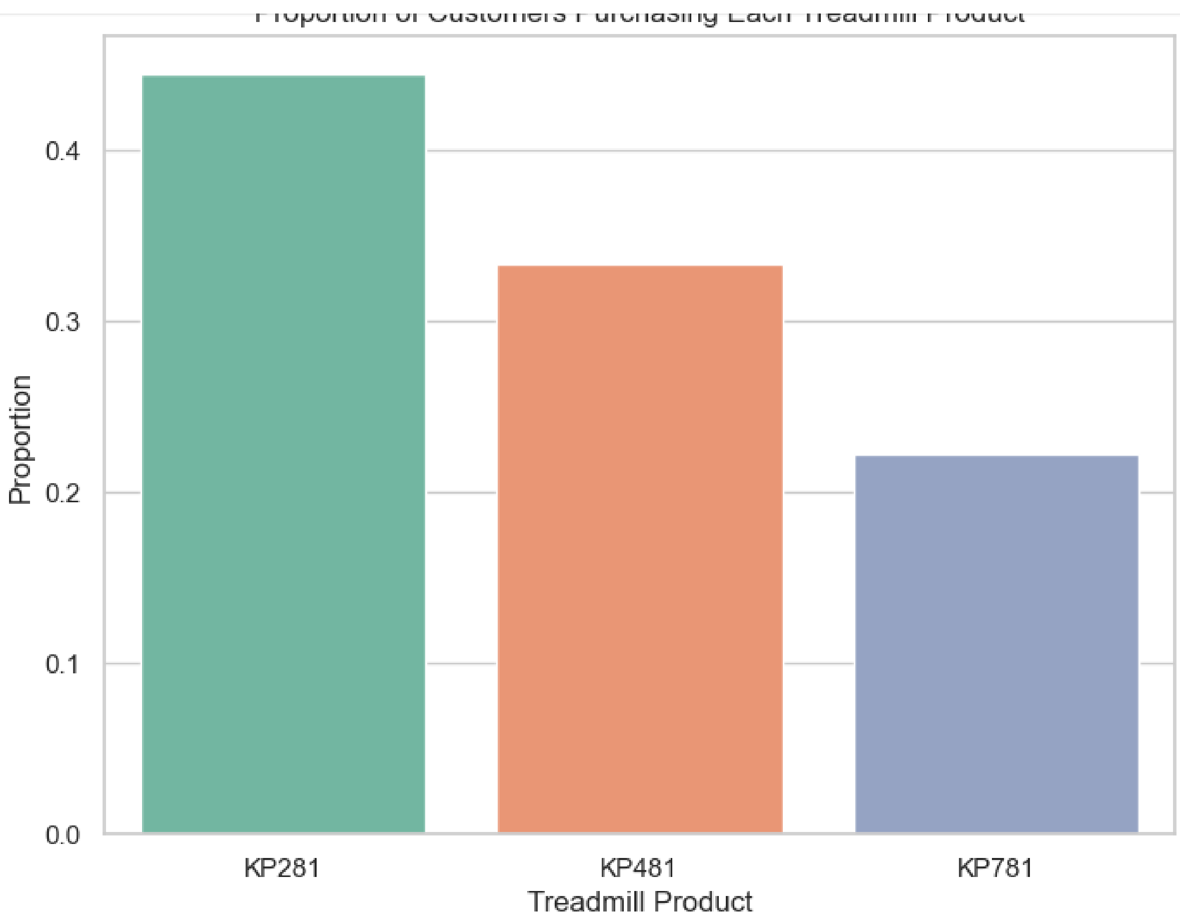
```
# Calculate the proportion of customers purchasing each treadmill product
product_counts = data['Product'].value_counts(normalize=True)
product_counts
```



```
Product
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: proportion, dtype: float64
```

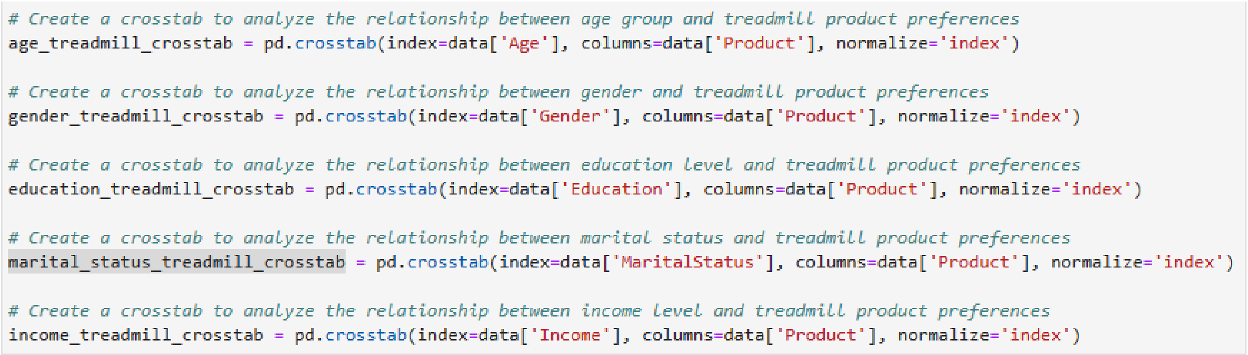


```
# Visualize the distribution
plt.figure(figsize=(8, 6))
sns.barplot(x=product_counts.index, y=product_counts.values, palette='Set2')
plt.title('Proportion of Customers Purchasing Each Treadmill Product')
plt.xlabel('Treadmill Product')
plt.ylabel('Proportion')
plt.show()
```

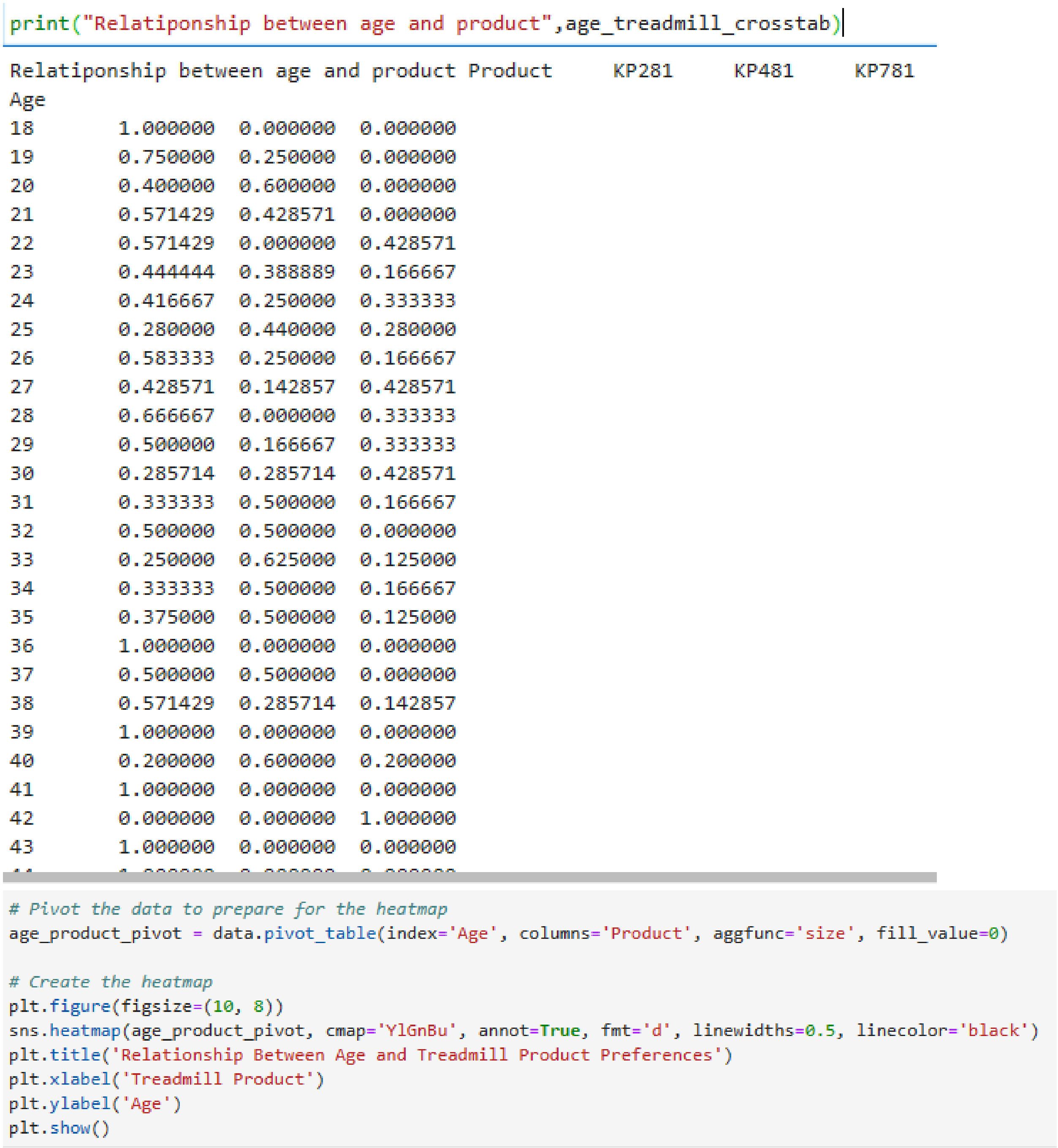


Insights: Based on customer preferences, the data indicates that KP281 is the most favored treadmill model, comprising approximately 44% of the total purchases, followed closely by KP481, which accounts for nearly 33% of the total purchases.

- **Influence of Demographic Characteristics:**
  a) Analyze the relationship between demographic variables and treadmill product preferences to understand demographic influences.
  b) Construct contingency tables



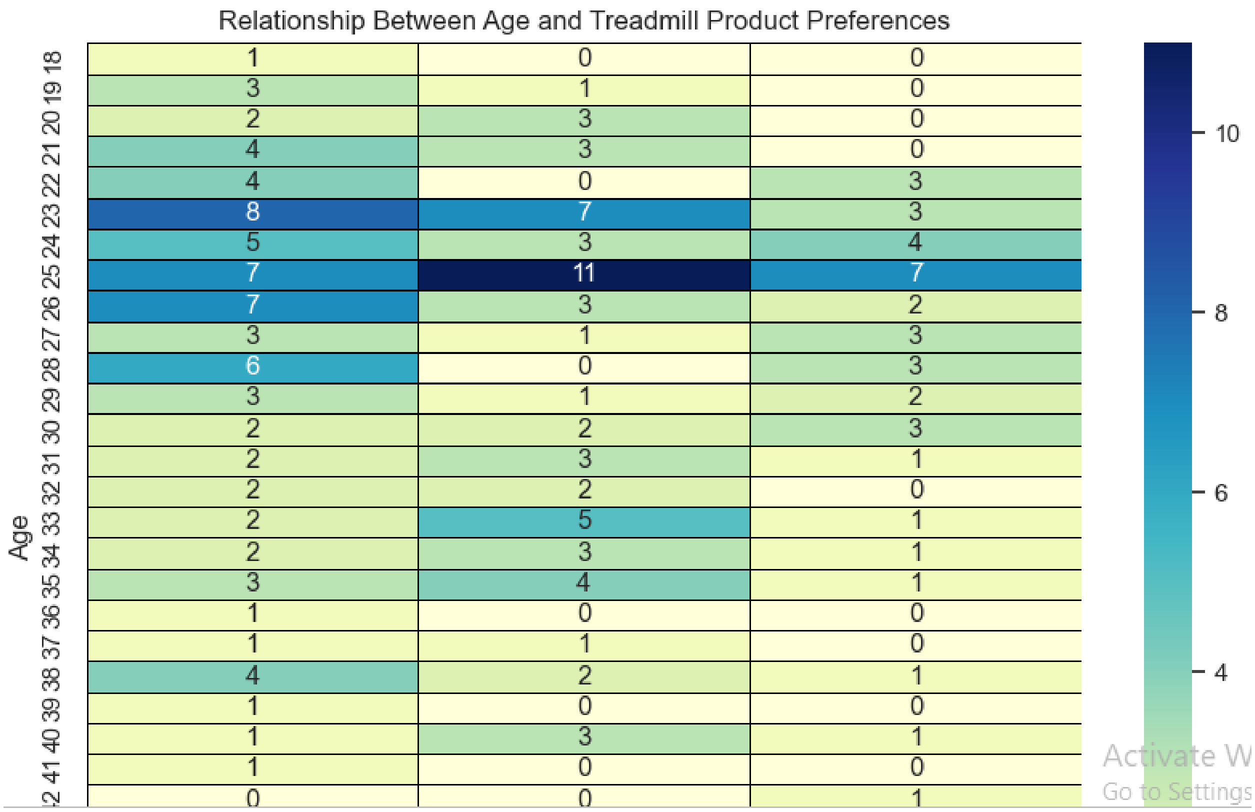
```
# Create a crosstab to analyze the relationship between age group and treadmill product preferences
age_treadmill_crosstab = pd.crosstab(index=data['Age'], columns=data['Product'], normalize='index')

# Create a crosstab to analyze the relationship between gender and treadmill product preferences
gender_treadmill_crosstab = pd.crosstab(index=data['Gender'], columns=data['Product'], normalize='index')

# Create a crosstab to analyze the relationship between education level and treadmill product preferences
education_treadmill_crosstab = pd.crosstab(index=data['Education'], columns=data['Product'], normalize='index')

# Create a crosstab to analyze the relationship between marital status and treadmill product preferences
marital_status_treadmill_crosstab = pd.crosstab(index=data['MaritalStatus'], columns=data['Product'], normalize='index')

# Create a crosstab to analyze the relationship between income level and treadmill product preferences
income_treadmill_crosstab = pd.crosstab(index=data['Income'], columns=data['Product'], normalize='index')
```

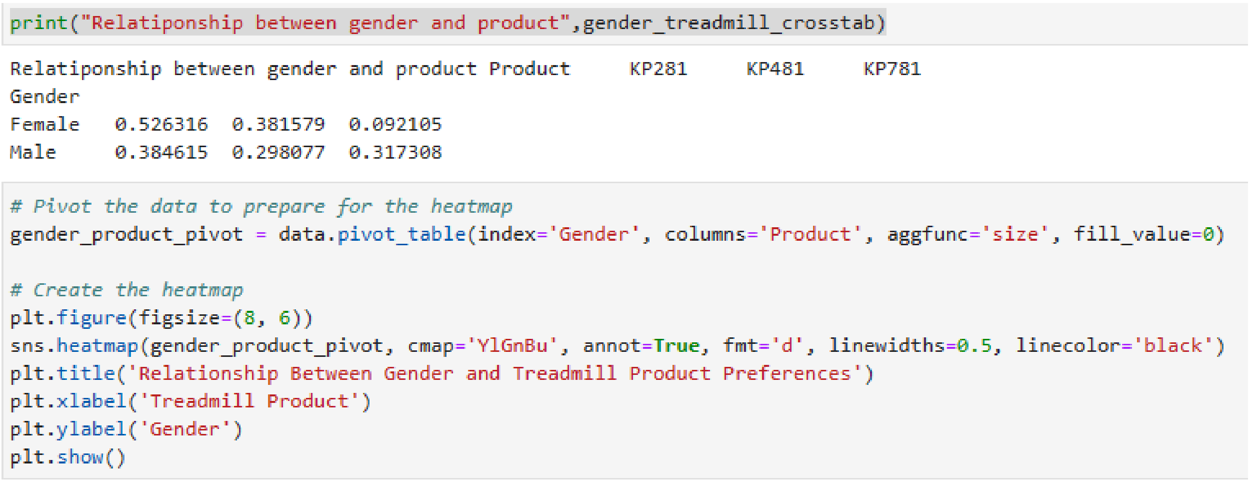


```
print("Relatiponship between age and product",age_treadmill_crosstab)
```

```
Relatiponship between age and product Product     KP281     KP481     KP781
Age
18        1.000000  0.000000  0.000000
19        0.750000  0.250000  0.000000
20        0.400000  0.600000  0.000000
21        0.571429  0.428571  0.000000
22        0.571429  0.000000  0.428571
23        0.444444  0.388889  0.166667
24        0.416667  0.250000  0.333333
25        0.280000  0.440000  0.280000
26        0.583333  0.250000  0.166667
27        0.428571  0.142857  0.428571
28        0.666667  0.000000  0.333333
29        0.500000  0.166667  0.333333
30        0.285714  0.285714  0.428571
31        0.333333  0.500000  0.166667
32        0.500000  0.500000  0.000000
33        0.250000  0.625000  0.125000
34        0.333333  0.500000  0.166667
35        0.375000  0.500000  0.125000
36        1.000000  0.000000  0.000000
37        0.500000  0.500000  0.000000
38        0.571429  0.285714  0.142857
39        1.000000  0.000000  0.000000
40        0.200000  0.600000  0.200000
41        1.000000  0.000000  0.000000
42        0.000000  0.000000  1.000000
43        1.000000  0.000000  0.000000
```

```
# Pivot the data to prepare for the heatmap
age_product_pivot = data.pivot_table(index='Age', columns='Product', aggfunc='size', fill_value=0)

# Create the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(age_product_pivot, cmap='YlGnBu', annot=True, fmt='d', linewidths=0.5, linecolor='black')
plt.title('Relationship Between Age and Treadmill Product Preferences')
plt.xlabel('Treadmill Product')
plt.ylabel('Age')
plt.show()
```

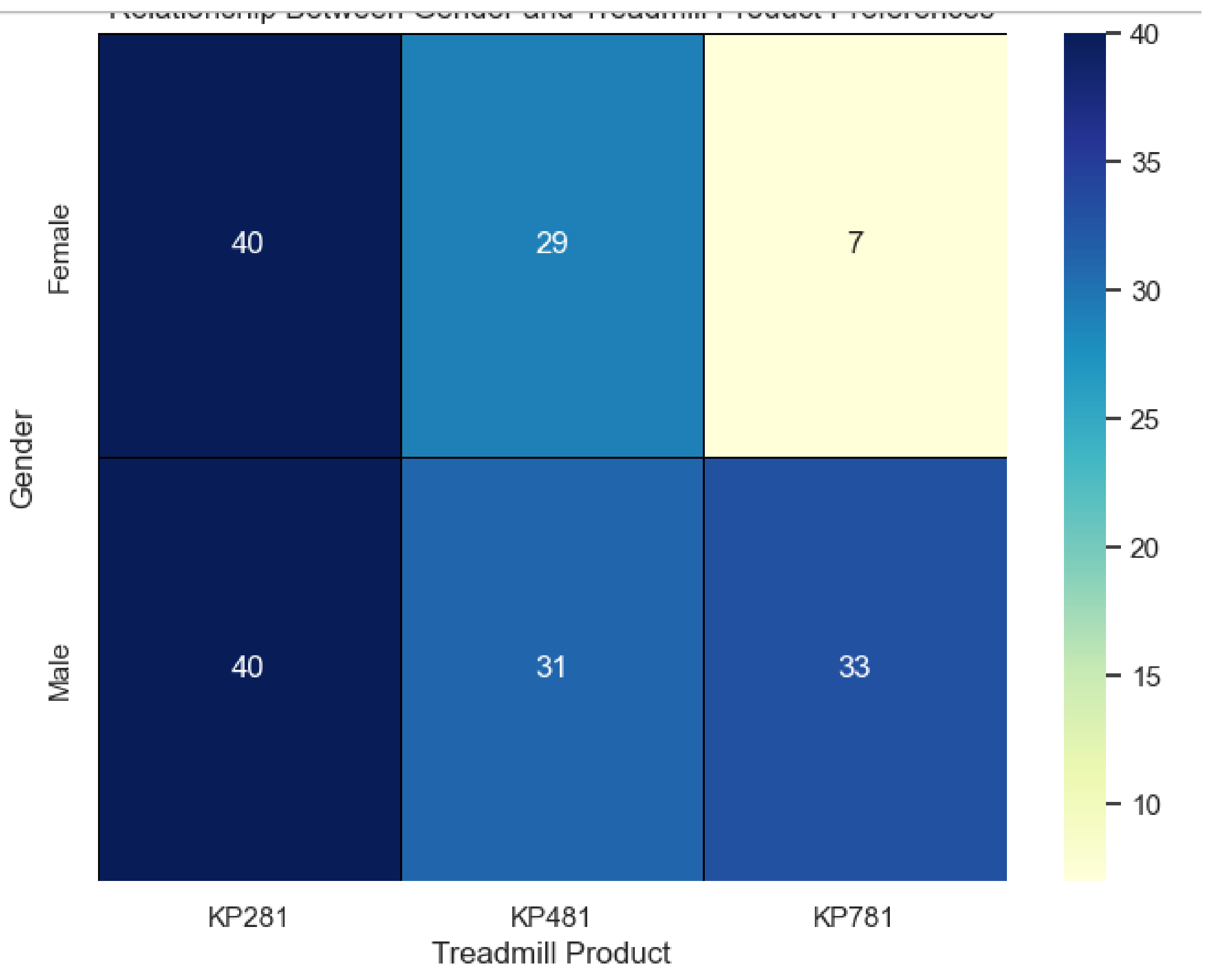




```
print("Relatiponship between gender and product",gender_treadmill_crosstab)
```

```
Relatiponship between gender and product Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
```

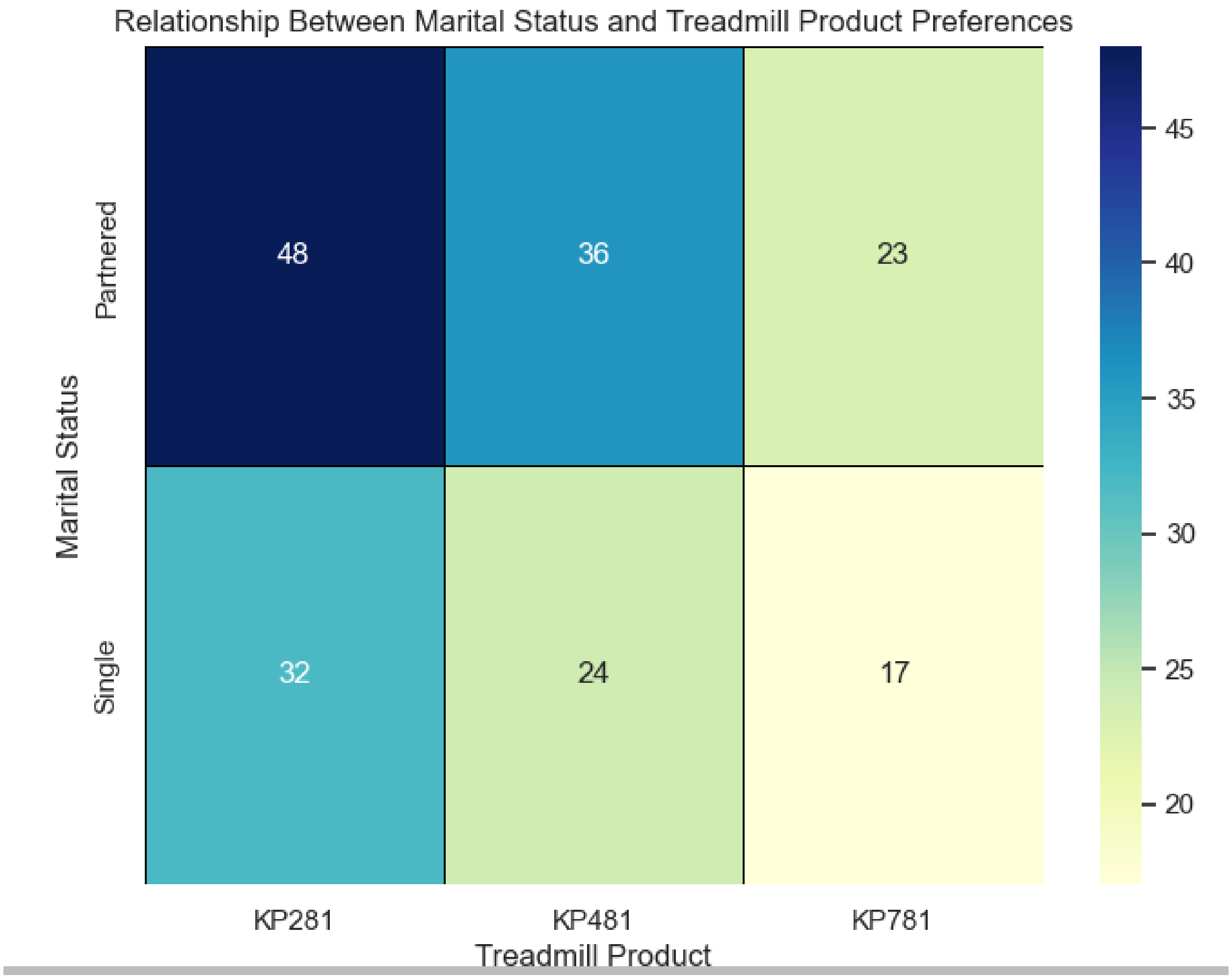
```
# Pivot the data to prepare for the heatmap
gender_product_pivot = data.pivot_table(index='Gender', columns='Product', aggfunc='size', fill_value=0)

# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(gender_product_pivot, cmap='YlGnBu', annot=True, fmt='d', linewidths=0.5, linecolor='black')
plt.title('Relationship Between Gender and Treadmill Product Preferences')
plt.xlabel('Treadmill Product')
plt.ylabel('Gender')
plt.show()
```



```
print("Relatiponship between education and product",education_treadmill_crosstab)
```

```
Relatiponship between education and product Product        KP281      KP481      KP781
Education
12         0.666667  0.333333  0.000000
13         0.600000  0.400000  0.000000
14         0.545455  0.418182  0.036364
15         0.800000  0.200000  0.000000
16         0.458824  0.364706  0.176471
18         0.086957  0.086957  0.826087
20         0.000000  0.000000  1.000000
21         0.000000  0.000000  1.000000
```

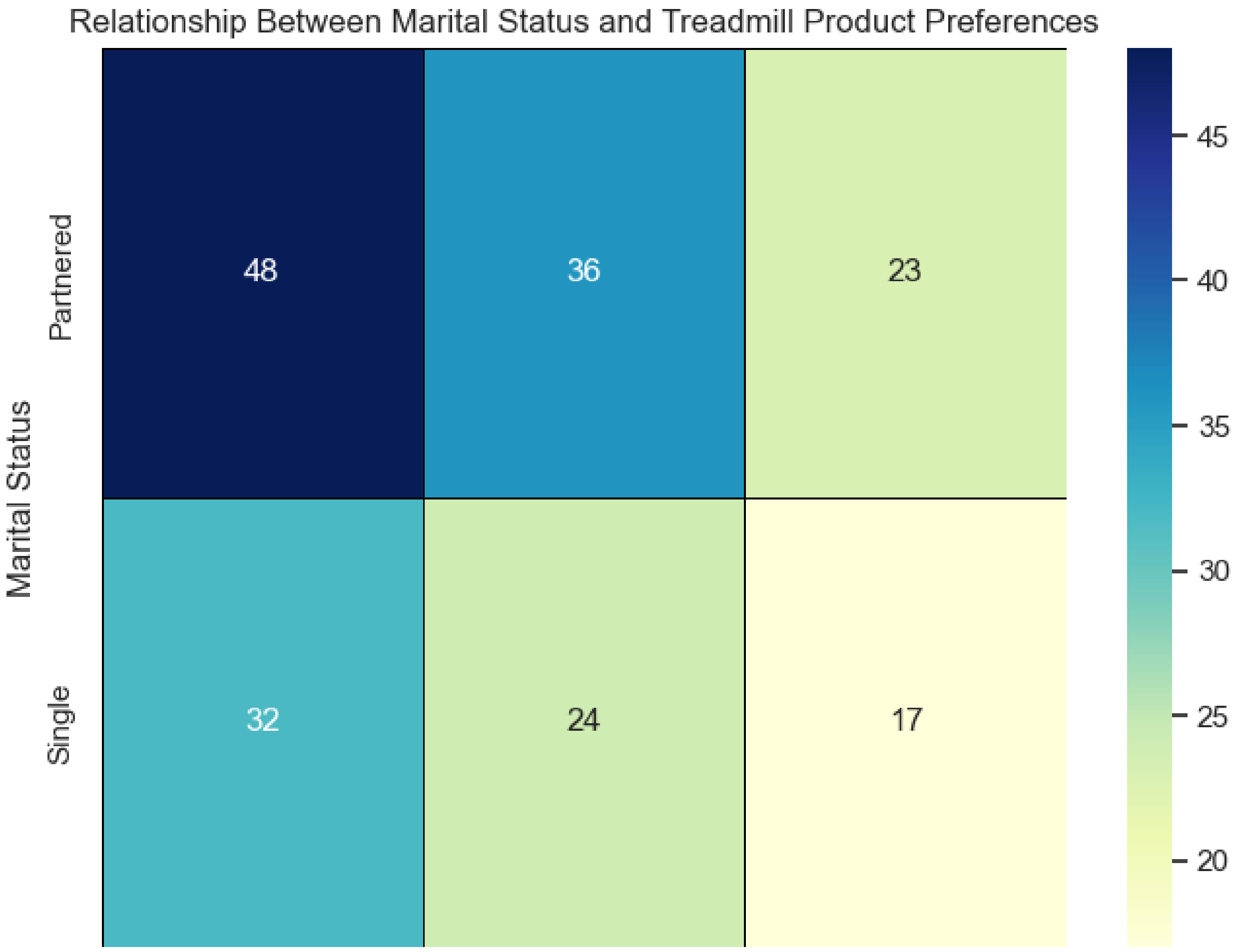
```
# Pivot the data to prepare for the heatmap
marital_product_pivot = data.pivot_table(index='MaritalStatus', columns='Product', aggfunc='size', fill_value=0)

# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(marital_product_pivot, cmap='YlGnBu', annot=True, fmt='d', linewidths=0.5, linecolor='black')
plt.title('Relationship Between Marital Status and Treadmill Product Preferences')
plt.xlabel('Treadmill Product')
plt.ylabel('Marital Status')
plt.show()
```

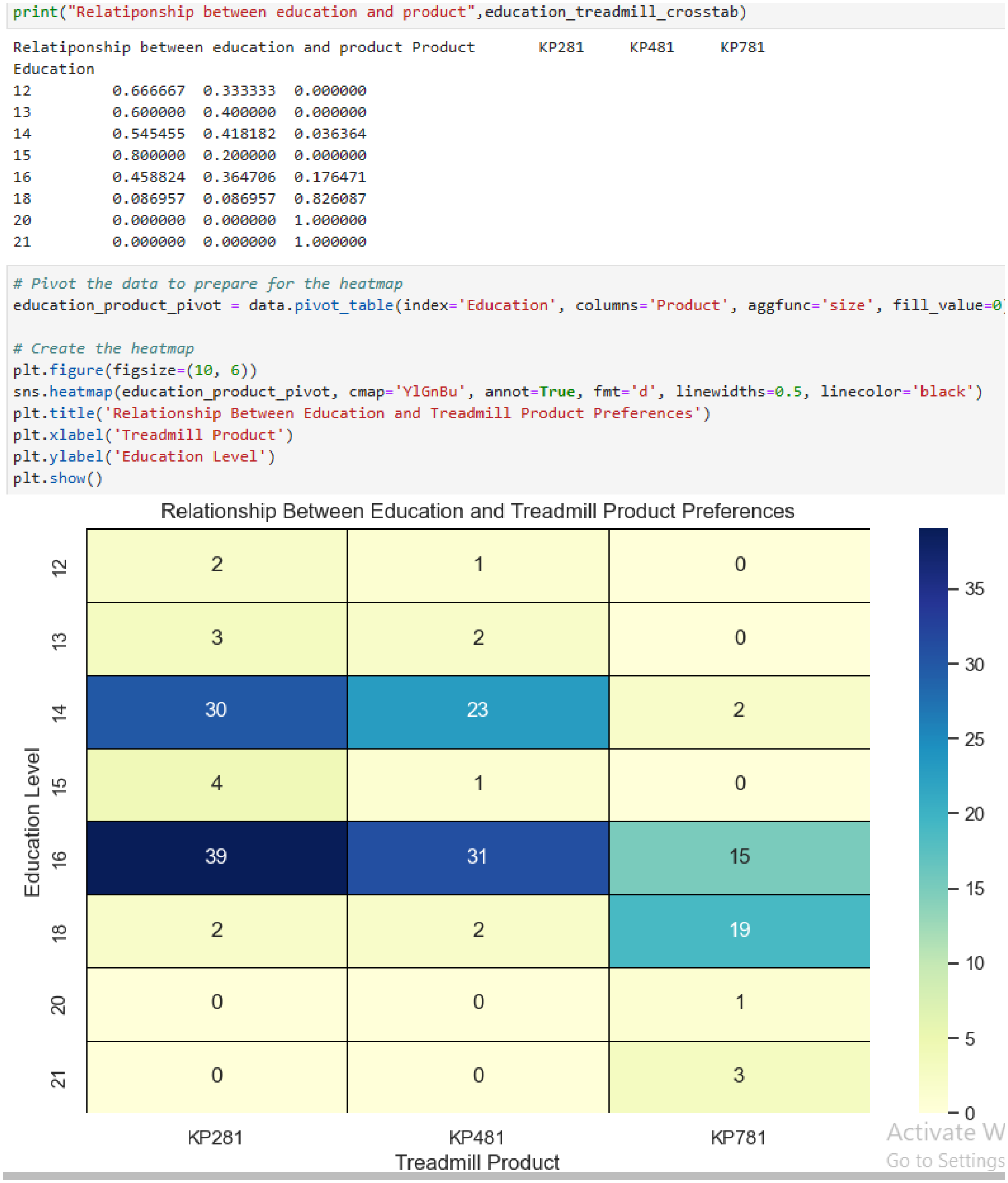

Relationship Between Marital Status and Treadmill Product Preferences
Marital Status
Partnered
Single
48
36
23
32
24
17
KP281
KP481
KP781
Treadmill Product
45
40
35
30
25
20

```
print("Relatiponship between marital status and product",marital_status_treadmill_crosstab)
```

```
Relatiponship between marital status and product Product          KP281     KP481     KP781
MaritalStatus
Partnered      0.448598  0.336449  0.214953
Single         0.438356  0.328767  0.232877
```

```
# Pivot the data to prepare for the heatmap
marital_product_pivot = data.pivot_table(index='MaritalStatus', columns='Product', aggfunc='size', fill_value=0)

# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(marital_product_pivot, cmap='YlGnBu', annot=True, fmt='d', linewidths=0.5, linecolor='black')
plt.title('Relationship Between Marital Status and Treadmill Product Preferences')
plt.xlabel('Treadmill Product')
plt.ylabel('Marital Status')
plt.show()
```





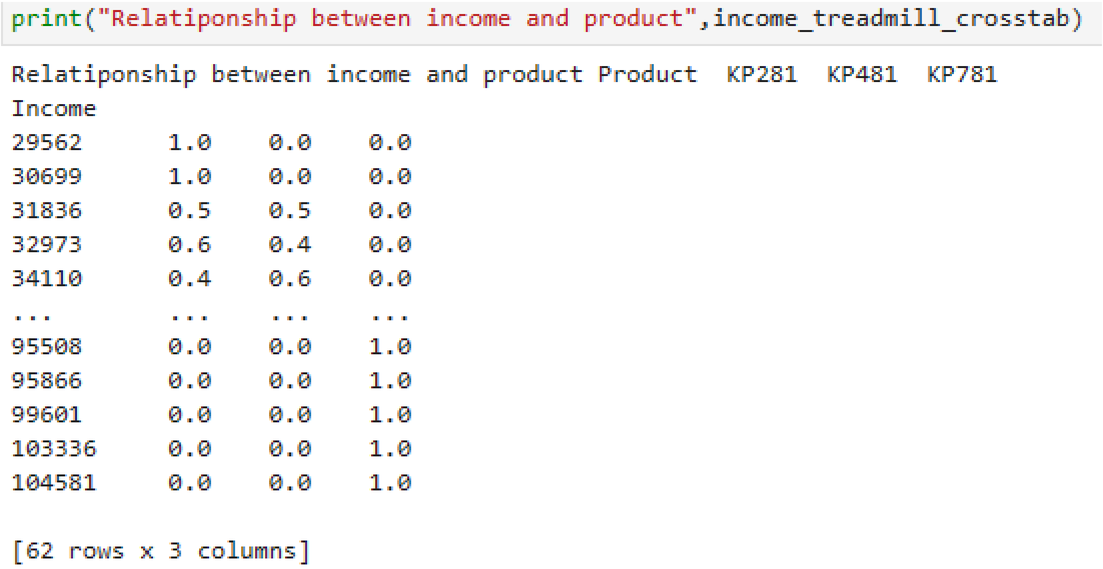
```
print("Relatiponship between education and product",education_treadmill_crosstab)
```

```
Relatiponship between education and product Product       KP281     KP481     KP781
Education
12         0.666667  0.333333  0.000000
13         0.600000  0.400000  0.000000
14         0.545455  0.418182  0.036364
15         0.800000  0.200000  0.000000
16         0.458824  0.364706  0.176471
18         0.086957  0.086957  0.826087
20         0.000000  0.000000  1.000000
21         0.000000  0.000000  1.000000
```

```
# Pivot the data to prepare for the heatmap
education_product_pivot = data.pivot_table(index='Education', columns='Product', aggfunc='size', fill_value=0

# Create the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(education_product_pivot, cmap='YlGnBu', annot=True, fmt='d', linewidths=0.5, linecolor='black')
plt.title('Relationship Between Education and Treadmill Product Preferences')
plt.xlabel('Treadmill Product')
plt.ylabel('Education Level')
plt.show()
```

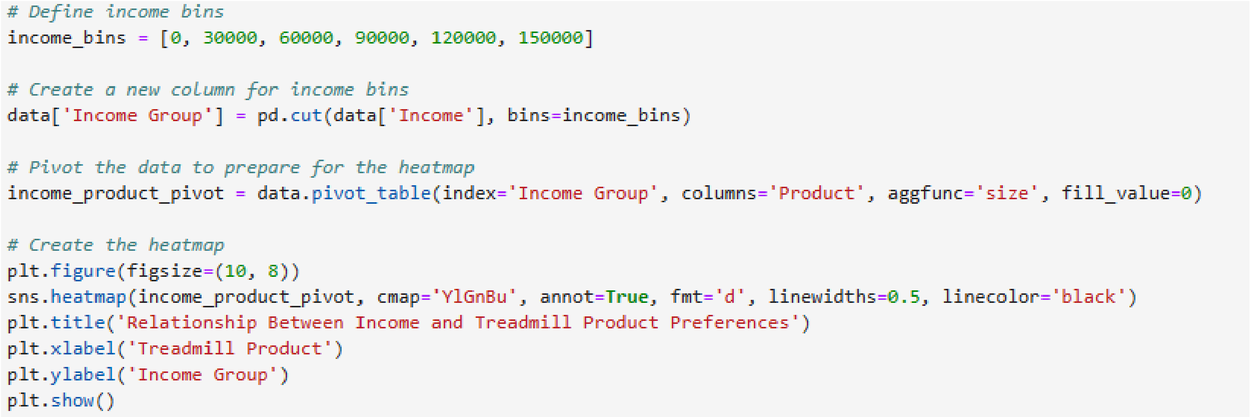
Relationship Between Education and Treadmill Product Preferences

Education Level	KP281	KP481	KP781
12	2	1	0
13	3	2	0
14	30	23	2
15	4	1	0
16	39	31	15
18	2	2	19
20	0	0	1
21	0	0	3

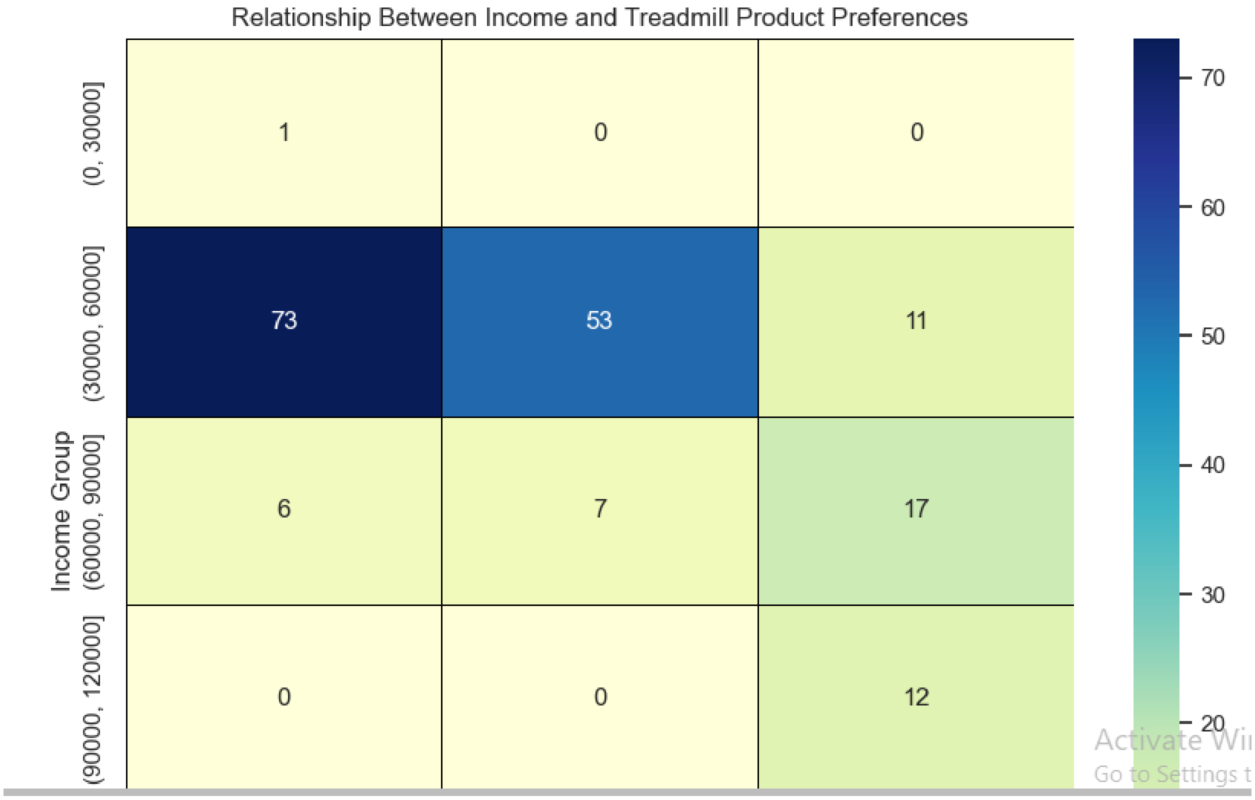


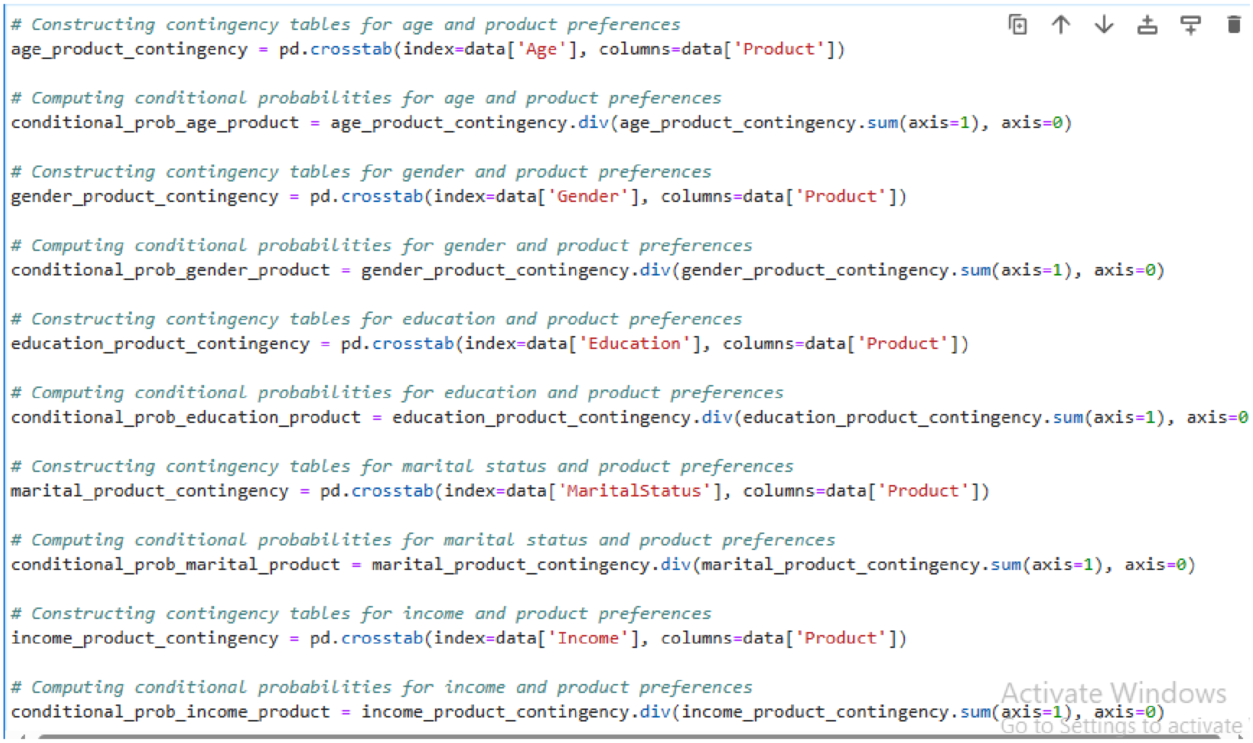
```
print("Relatiponship between income and product",income_treadmill_crosstab)
```

```
Relatiponship between income and product Product  KP281  KP481  KP781
Income
29562       1.0    0.0    0.0
30699       1.0    0.0    0.0
31836       0.5    0.5    0.0
32973       0.6    0.4    0.0
34110       0.4    0.6    0.0
...         ...    ...    ...
95508       0.0    0.0    1.0
95866       0.0    0.0    1.0
99601       0.0    0.0    1.0
103336      0.0    0.0    1.0
104581      0.0    0.0    1.0

[62 rows x 3 columns]
```

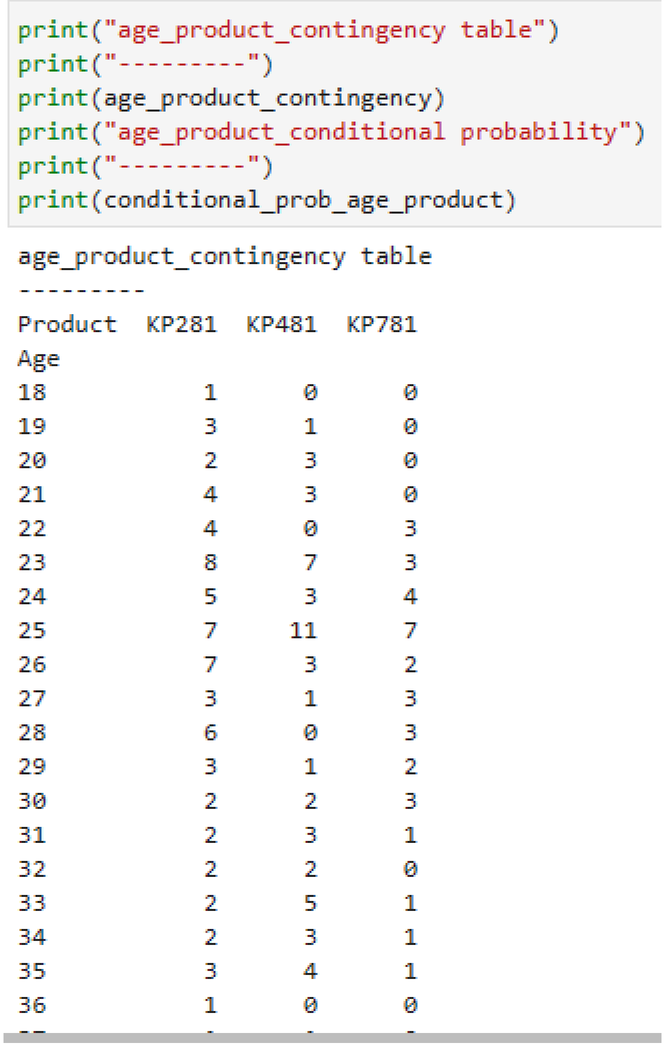


```
# Define income bins
income_bins = [0, 30000, 60000, 90000, 120000, 150000]

# Create a new column for income bins
data['Income Group'] = pd.cut(data['Income'], bins=income_bins)

# Pivot the data to prepare for the heatmap
income_product_pivot = data.pivot_table(index='Income Group', columns='Product', aggfunc='size', fill_value=0)

# Create the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(income_product_pivot, cmap='YlGnBu', annot=True, fmt='d', linewidths=0.5, linecolor='black')
plt.title('Relationship Between Income and Treadmill Product Preferences')
plt.xlabel('Treadmill Product')
plt.ylabel('Income Group')
plt.show()
```



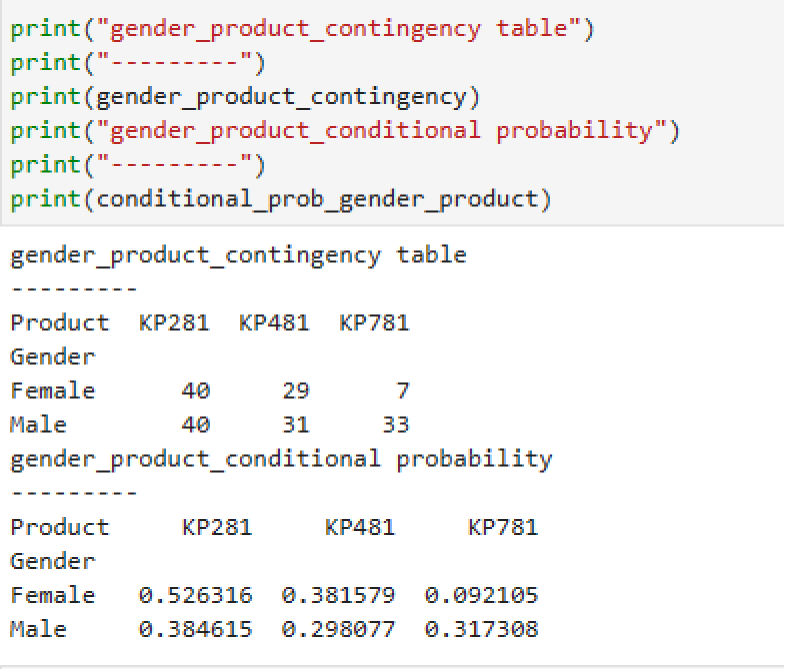


```
# Constructing contingency tables for age and product preferences
age_product_contingency = pd.crosstab(index=data['Age'], columns=data['Product'])

# Computing conditional probabilities for age and product preferences
conditional_prob_age_product = age_product_contingency.div(age_product_contingency.sum(axis=1), axis=0)

# Constructing contingency tables for gender and product preferences
gender_product_contingency = pd.crosstab(index=data['Gender'], columns=data['Product'])

# Computing conditional probabilities for gender and product preferences
conditional_prob_gender_product = gender_product_contingency.div(gender_product_contingency.sum(axis=1), axis=0)

# Constructing contingency tables for education and product preferences
education_product_contingency = pd.crosstab(index=data['Education'], columns=data['Product'])

# Computing conditional probabilities for education and product preferences
conditional_prob_education_product = education_product_contingency.div(education_product_contingency.sum(axis=1), axis=0

# Constructing contingency tables for marital status and product preferences
marital_product_contingency = pd.crosstab(index=data['MaritalStatus'], columns=data['Product'])

# Computing conditional probabilities for marital status and product preferences
conditional_prob_marital_product = marital_product_contingency.div(marital_product_contingency.sum(axis=1), axis=0)

# Constructing contingency tables for income and product preferences
income_product_contingency = pd.crosstab(index=data['Income'], columns=data['Product'])

# Computing conditional probabilities for income and product preferences
conditional_prob_income_product = income_product_contingency.div(income_product_contingency.sum(axis=1), axis=0)
```



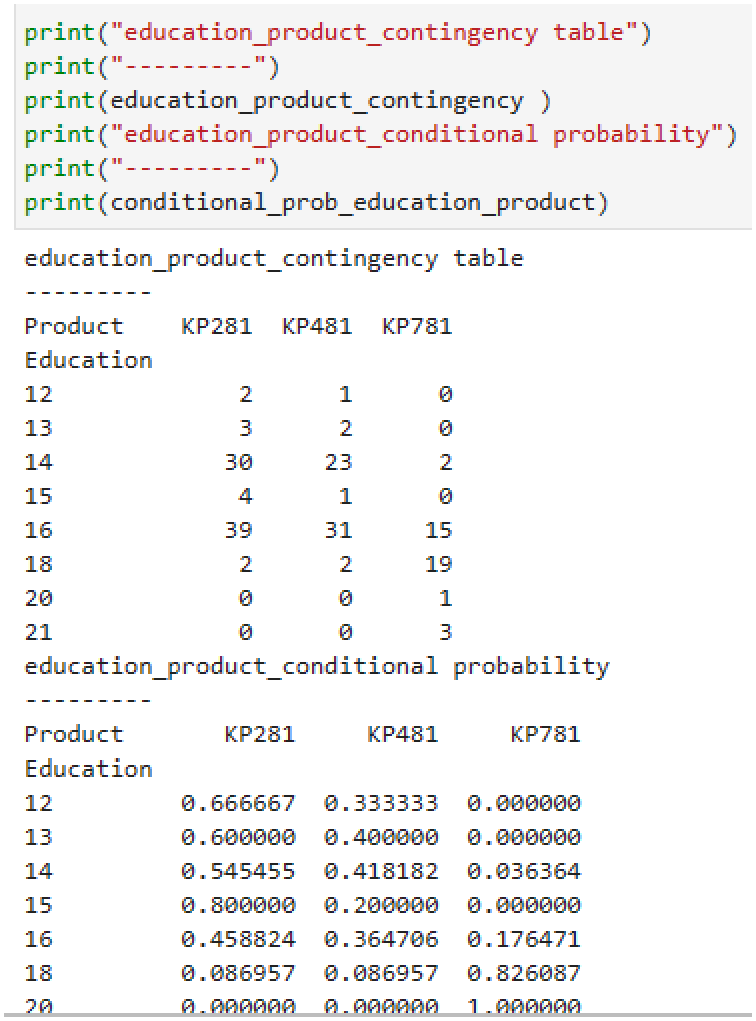
```
print("age_product_contingency table")
print("---------")
print(age_product_contingency)
print("age_product_conditional probability")
print("---------")
print(conditional_prob_age_product)
```

```
age_product_contingency table
---------
Product  KP281  KP481  KP781
Age
18           1      0      0
19           3      1      0
20           2      3      0
21           4      3      0
22           4      0      3
23           8      7      3
24           5      3      4
25           7     11      7
26           7      3      2
27           3      1      3
28           6      0      3
29           3      1      2
30           2      2      3
31           2      3      1
32           2      2      0
33           2      5      1
34           2      3      1
35           3      4      1
36           1      0      0
```



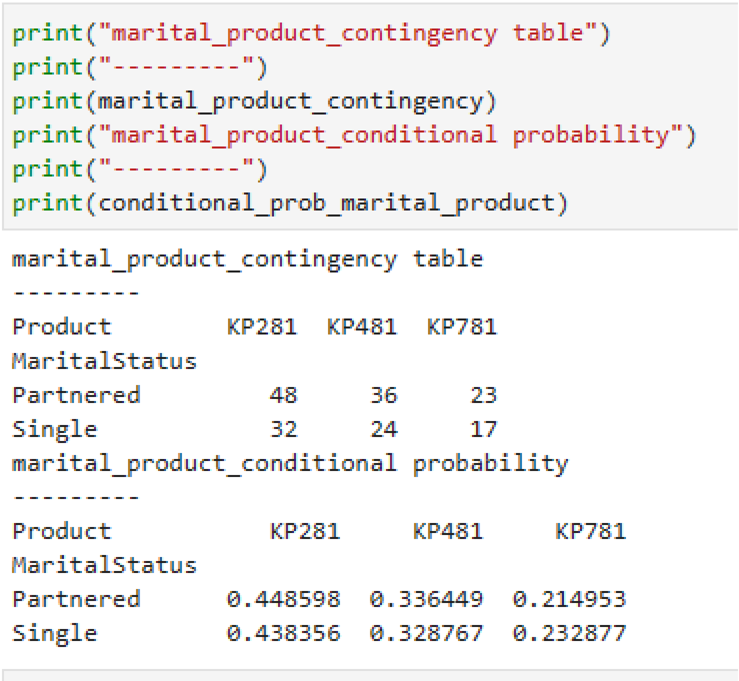
```
print("gender_product_contingency table")
print("---------")
print(gender_product_contingency)
print("gender_product_conditional probability")
print("---------")
print(conditional_prob_gender_product)
```

```
gender_product_contingency table
---------
Product  KP281  KP481  KP781
Gender
Female      40     29      7
Male        40     31     33
gender_product_conditional probability
---------
Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
```



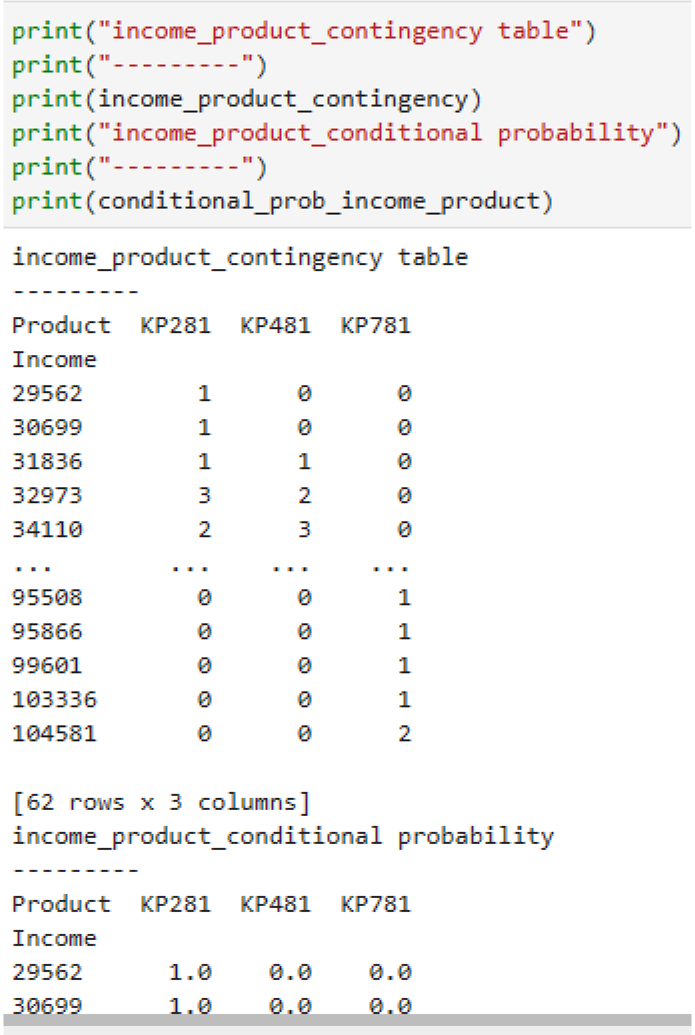
```
print("education_product_contingency table")
print("---------")
print(education_product_contingency )
print("education_product_conditional probability")
print("---------")
print(conditional_prob_education_product)
```

```
education_product_contingency table
---------
Product    KP281  KP481  KP781
Education
12             2      1      0
13             3      2      0
14            30     23      2
15             4      1      0
16            39     31     15
18             2      2     19
20             0      0      1
21             0      0      3
education_product_conditional probability
---------
Product       KP281     KP481     KP781
Education
12         0.666667  0.333333  0.000000
13         0.600000  0.400000  0.000000
14         0.545455  0.418182  0.036364
15         0.800000  0.200000  0.000000
16         0.458824  0.364706  0.176471
18         0.086957  0.086957  0.826087
20         0.000000  0.000000  1.000000
```



```
print("marital_product_contingency table")
print("---------")
print(marital_product_contingency)
print("marital_product_conditional probability")
print("---------")
print(conditional_prob_marital_product)
```

```
marital_product_contingency table
---------
Product        KP281  KP481  KP781
MaritalStatus
Partnered         48     36     23
Single            32     24     17
marital_product_conditional probability
---------
Product           KP281     KP481     KP781
MaritalStatus
Partnered      0.448598  0.336449  0.214953
Single         0.438356  0.328767  0.232877
```



```
print("income_product_contingency table")
print("---------")
print(income_product_contingency)
print("income_product_conditional probability")
print("---------")
print(conditional_prob_income_product)
```

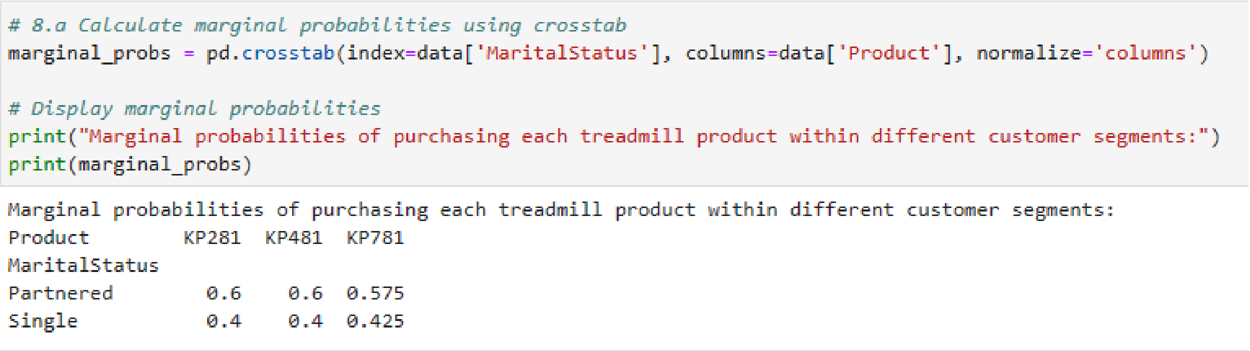
```
income_product_contingency table
---------
Product  KP281  KP481  KP781
Income
29562        1      0      0
30699        1      0      0
31836        1      1      0
32973        3      2      0
34110        2      3      0
...        ...    ...    ...
95508        0      0      1
95866        0      0      1
99601        0      0      1
103336       0      0      1
104581       0      0      2

[62 rows x 3 columns]
income_product_conditional probability
---------
Product  KP281  KP481  KP781
Income
29562      1.0    0.0    0.0
30699      1.0    0.0    0.0
```

Insights: The age group of 21-26 demonstrates the highest propensity for purchasing treadmills, with a declining trend observed in older age groups. Across all age demographics, KP281 emerges as the most preferred treadmill model, boasting varying probabilities ranging from 0.75 to 0.57 across different age groups. Interestingly, KP281 garners preference from both genders, with a slightly higher probability among females (0.52) compared to males (0.38). Conversely, males exhibit a preference for KP781, with a probability of 0.31.

Education-wise, individuals with 12-16 years of education lean towards KP281, whereas those with 18-21 years of education are inclined towards KP781. The likelihood of purchasing a treadmill product remains relatively consistent between single and partnered individuals, albeit slightly higher among partnered individuals.

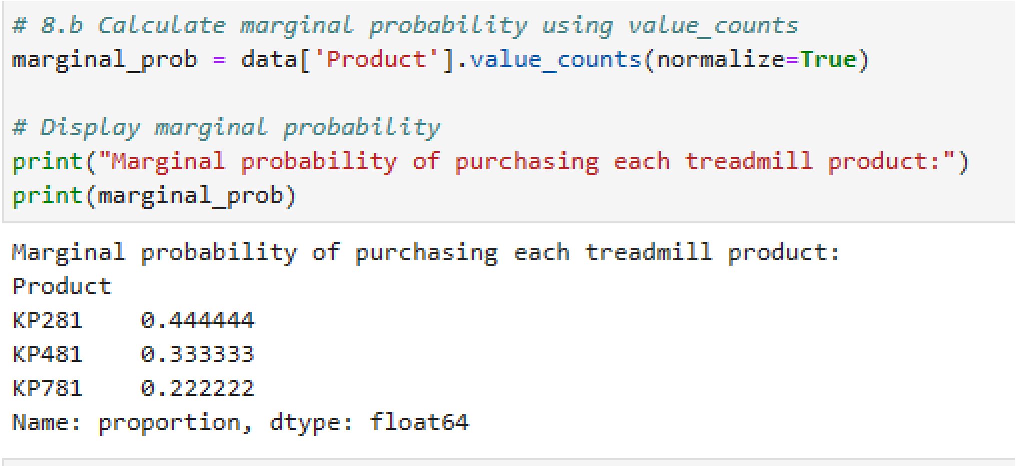
Moreover, customers with higher incomes tend to opt for KP781, while those with average incomes are more inclined towards KP281. These insights provide valuable information for AeroFit to tailor its marketing strategies and product offerings to cater to specific demographic segments effectively.

- **Marginal Probabilities Calculations:** a) Calculate the marginal probabilities of purchasing each treadmill product within different customer segments to understand product popularity among demographics. b) Find the marginal probability (what percent of customers have purchased KP281, KP481, or KP781)



```
# 8.a Calculate marginal probabilities using crosstab
marginal_probs = pd.crosstab(index=data['MaritalStatus'], columns=data['Product'], normalize='columns')

# Display marginal probabilities
print("Marginal probabilities of purchasing each treadmill product within different customer segments:")
print(marginal_probs)
```

```
Marginal probabilities of purchasing each treadmill product within different customer segments:
Product        KP281  KP481  KP781
MaritalStatus
Partnered        0.6    0.6  0.575
Single           0.4    0.4  0.425
```

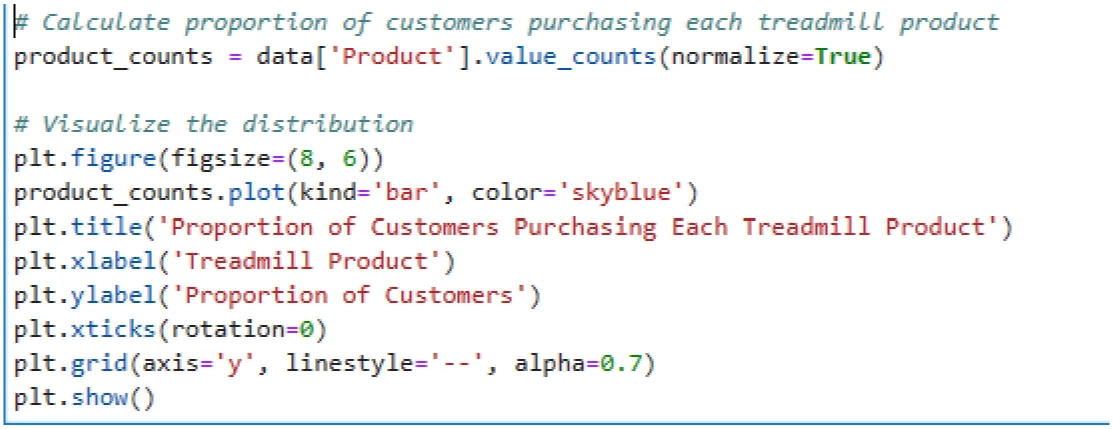


```
# 8.b Calculate marginal probability using value_counts
marginal_prob = data['Product'].value_counts(normalize=True)

# Display marginal probability
print("Marginal probability of purchasing each treadmill product:")
print(marginal_prob)
```

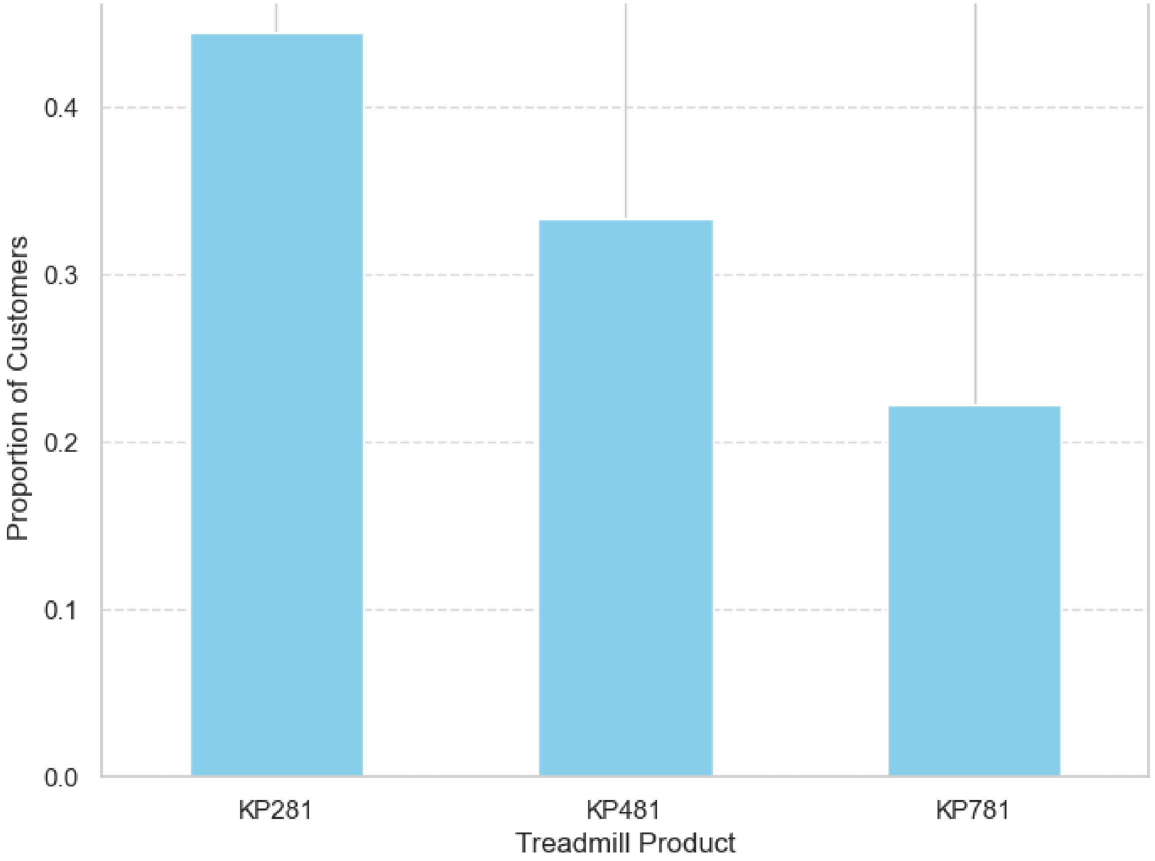
```
Marginal probability of purchasing each treadmill product:
Product
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: proportion, dtype: float64
```

Insights: For all three treadmill models, the probability of purchasing the product is higher among partnered customers compared to single customers. Among the three models, KP281 has the highest overall probability of purchase, with a probability of 0.44.

- **Distribution of Customer Preferences Across Aerofit Treadmill Product Portfolio:** Calculate the proportion of customers purchasing each treadmill product and visualize the distribution to identify popular choices among customers.



```
# Calculate proportion of customers purchasing each treadmill product
product_counts = data['Product'].value_counts(normalize=True)

# Visualize the distribution
plt.figure(figsize=(8, 6))
product_counts.plot(kind='bar', color='skyblue')
plt.title('Proportion of Customers Purchasing Each Treadmill Product')
plt.xlabel('Treadmill Product')
plt.ylabel('Proportion of Customers')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



Insights: The majority of customers prefer buying KP281 followed by KP481.

<u>INSIGHTS FROM CUSTOMER PROFILING:</u> Based on the analysis of customer profiles for each treadmill product, the following insights can be derived:

- **Demographic Analysis:**
  - Age: Customers in the age group of 20-30 show the highest preference for all treadmill products, with KP281 being the most popular choice across all age groups. From the age 42-50 KP281 and KP&81 are equally preferred.
  - Gender: Both males and females exhibit a strong preference for KP281, although males show slightly higher preference for KP781 compared to females.
  - Education: Customers with education levels ranging from 12 to 16 years tend to prefer KP281, while those with higher education levels (18-21 years) show a preference for KP781.
  - Marital Status: Partnered individuals are more likely to purchase all treadmill products compared to single individuals, with KP281 being the most popular choice among both groups.
- **Preference Analysis:**
  - KP281 is the most preferred treadmill product across all demographic segments, indicating its widespread popularity among customers.

   - KP781 tends to attract customers with higher income levels, suggesting that it may be perceived as a premium offering.

- **Segmentation:**
   - Based on demographic characteristics and preferences, customers can be segmented into different groups, such as age cohorts, gender, education levels, income and marital status.
   - Tailoring recommendations to each segment can help personalize the shopping experience and increase customer satisfaction.
- **Feedback Loop:**
   - Continuously gather feedback from customers through surveys, reviews, and social media channels to understand their evolving needs and preferences.
   - Use data analytics tools to monitor customer behavior and track the effectiveness of recommendations over time.
   - Adjust marketing strategies and product offerings based on insights gained from customer feedback to ensure alignment with customer preferences and market trends.

By leveraging insights from customer profiles, AeroFit can enhance its marketing strategies, optimize product offerings, and improve customer engagement to drive business growth and customer satisfaction.

RECOMMENDATIONS: Based on the insights derived from customer profiles, AeroFit can develop targeted marketing strategies and promotional initiatives to effectively reach different demographic segments and drive sales:

- **Targeted Marketing Campaigns:**
   - Focus marketing efforts on customers in the age group of 20-26, as they show the highest preference for treadmill products.
   - Highlight the features and benefits of each treadmill product in marketing materials, emphasizing KP781's advanced features, high mileage capacity, and superior ratings.
   - Tailor messaging to appeal to specific demographic segments, such as emphasizing the affordability and versatility of KP281 for younger customers, and highlighting the premium quality and performance of KP781 for customers with higher income levels.
- **Promotions and Discounts:**
   - Offer promotions or discounts on KP481 to attract customers who are looking for a mid-range option with a balance of features and affordability.
   - Provide exclusive discounts or incentives for partnered individuals to encourage purchases, leveraging their higher likelihood of buying treadmill products.
- **Education Content and Resources:**
   - Create educational content and resources that guide customers in making informed decisions based on their fitness goals and lifestyle preferences.
   - Offer online guides, videos, and articles that provide tips on selecting the right treadmill based on factors such as fitness level, workout preferences, and space constraints.
- **Product Portfolio Expansion:**

- Consider expanding the product portfolio to include options that cater to specific demographic segments or emerging trends identified in the analysis.
- Explore opportunities to introduce new treadmill models targeting niche markets, such as compact treadmills for urban dwellers or advanced performance models for fitness enthusiasts.

By implementing these targeted strategies, AeroFit can increase customer satisfaction, drive sales, and strengthen its position in the fitness equipment market. Additionally, by focusing on customer demographics and preferences, AeroFit can build brand loyalty and establish itself as a trusted provider of quality treadmill products.

CONCLUSION: In conclusion, the analysis revealed that KP281 is the most preferred treadmill model among customers, especially those aged 20-26, while KP781 is favored by individuals with higher incomes. Partnered customers showed a higher likelihood of purchasing, suggesting targeted marketing efforts towards this segment. AeroFit should capitalize on KP781's high ratings and mileage capacity, while offering discounts on KP481 to attract mid-range buyers. Expanding the product line to cater to niche segments could further drive growth. Overall, AeroFit can enhance its marketing strategies and product offerings to better meet customer needs and preferences, ensuring continued success in the fitness equipment market.