Aerofit Business case study

```
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749"
```

```
--2024-12-20 14:35:16--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?163999
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 13.224.9.129, 13.224.9.103, 13.224.9.24, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|13.224.9.129|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7279 (7.1K) [text/plain]
Saving to: ‘aerofit_treadmill.csv?1639992749’

aerofit_treadmill.c 100%[===================>]   7.11K  --.-KB/s    in 0s

2024-12-20 14:35:16 (26.1 MB/s) - ‘aerofit_treadmill.csv?1639992749’ saved [7279/7279]
```

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

+ Code + Text

```
df = pd.read_csv('aerofit_treadmill.csv?1639992749')
```

Q1

Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset

```
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.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
```

```
df.describe()
```

	Age	Education	Usage	Fitness	Income	Miles
count	180.000000	180.000000	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	3.455556	3.311111	53719.577778	103.194444
std	6.943498	1.617055	1.084797	0.958869	16506.684226	51.863605
min	18.000000	12.000000	2.000000	1.000000	29562.000000	21.000000
25%	24.000000	14.000000	3.000000	3.000000	44058.750000	66.000000
50%	26.000000	16.000000	3.000000	3.000000	50596.500000	94.000000
75%	33.000000	16.000000	4.000000	4.000000	58668.000000	114.750000
max	50.000000	21.000000	7.000000	5.000000	104581.000000	360.000000

```
df.isnull().sum()
```

	0
Product	0
Age	0
Gender	0
Education	0
MaritalStatus	0
Usage	0
Fitness	0
Income	0
Miles	0

dtype: int64

```
missing_values = df.isnull().sum()
print("Missing Values:\n", missing_values)
```

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

```
df.value_counts()
```

Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	count
KP281	**18**	**Male**	**14**	**Single**	**3**	**4**	**29562**	**112**	1
KP481	**30**	**Female**	**13**	**Single**	**4**	**3**	**46617**	**106**	1
	31	**Female**	**16**	**Partnered**	**2**	**3**	**51165**	**64**	1
			18	**Single**	**2**	**1**	**65220**	**21**	1
		Male	**16**	**Partnered**	**3**	**3**	**52302**	**95**	1
...	**...**	**...**	**...**	**...**	**...**	**...**	**...**	**...**	...
KP281	**34**	**Female**	**16**	**Single**	**2**	**2**	**52302**	**66**	1
		Male	**16**	**Single**	**4**	**5**	**51165**	**169**	1
	35	**Female**	**16**	**Partnered**	**3**	**3**	**60261**	**94**	1
			18	**Single**	**3**	**3**	**67083**	**85**	1
KP781	**48**	**Male**	**18**	**Partnered**	**4**	**5**	**95508**	**180**	1

180 rows × 1 columns

dtype: int64

```
df.dtypes
```

	0
Product	object
Age	int64
Gender	object
Education	int64
MaritalStatus	object
Usage	int64
Fitness	int64
Income	int64
Miles	int64

dtype: object

```
df.shape
```

```
(180, 9)
```

Q2

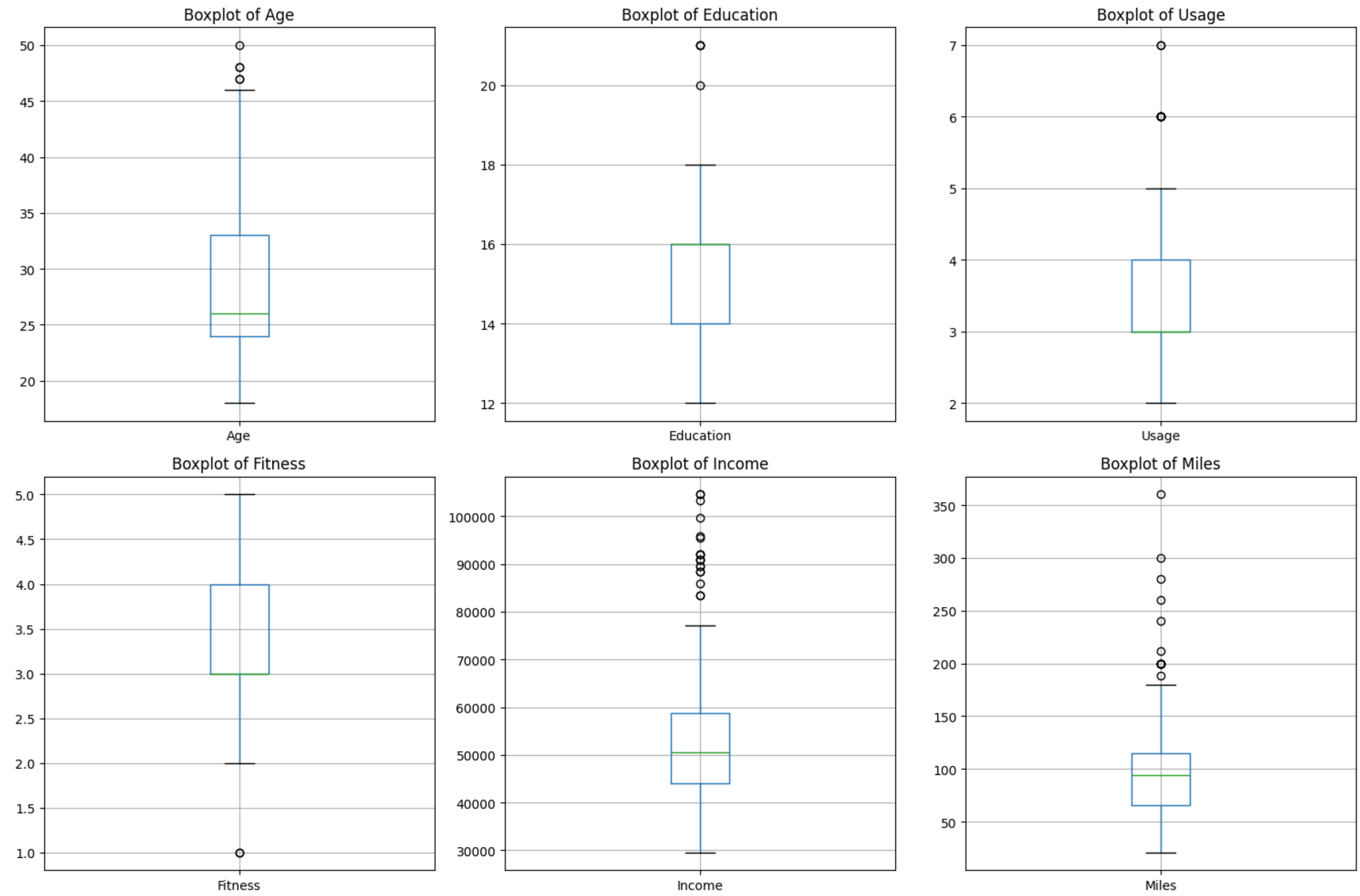
Detect Outliers

```
# Identify continuous variables (integer/float columns)
continuous_vars = df.select_dtypes(include=['int64', 'float64']).columns

# Generate boxplots for each continuous variable
plt.figure(figsize=(15, 10))
for i, var in enumerate(continuous_vars, 1):
    plt.subplot(2, 3, i)
    df.boxplot(column=var)
    plt.title(f'Boxplot of {var}')

plt.tight_layout()
plt.show()

continuous_vars
```




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

From the above graph of boxplot of age the median age lies around 30, most data points are between 25 and 35 years. A few outliers are visible above 45, indicating a small number of individuals older than the typical age range.

Boxplot of Education: The education level is concentrated between 14 and 16 years, the median is around 16 years. There are no significant outliers, showing a consistent distribution.

Boxplot of Usage: The usage varies between 2 and 6 units (e.g., hours or days of activity), the median is around 4. Some outliers are observed near 7, suggesting occasional high usage.

Boxplot of Fitness: The fitness level predominantly ranges between 3 and 4, with the median at around 3.5. A few outliers below 1 indicate very low fitness levels for certain individuals.

Boxplot of Income: Most income values lie between 40,000 and 80,000, with a median around 60,000. Several outliers are present above 100,000, showing a small number of individuals earning significantly more than the majority.

Boxplot of Miles: The miles traveled are primarily between 50 and 150, with the median near 100. There are numerous outliers above 300 miles, indicating a few individuals who travel much farther than average.

Conclusions: Age and Education distributions show relatively consistent patterns with minimal variability or outliers. Usage and Fitness demonstrate moderate consistency, with only a few high or low extreme values. Income and Miles show considerable variability and significant outliers, suggesting that certain individuals differ substantially in these aspects. Potential correlations could exist between variables (e.g., fitness and usage or income and miles), warranting further investigation.

```
# Compute the 5th and 95th percentiles for each continuous variable
percentiles = {var: (np.percentile(df[var], 5), np.percentile(df[var], 95)) for var in continuous_vars}

# Clip the data using np.clip()
clipped_data = df.copy()
for var in continuous_vars:
    lower, upper = percentiles[var]
    clipped_data[var] = np.clip(df[var], lower, upper)

# Display summary statistics before and after clipping
summary_before = df[continuous_vars].describe()
```

```
summary_after = clipped_data[continuous_vars].describe()

summary_before, summary_after
```

```
(              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  ,
              Age   Education       Usage     Fitness        Income  \
 count  180.000000  180.000000  180.000000  180.000000    180.000000
 mean    28.641389   15.572222    3.396944    3.322222  53477.070000
 std      6.446373    1.362017    0.952682    0.937461  15463.662523
 min     20.000000   14.000000    2.000000    2.000000  34053.150000
 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     43.050000   18.000000    5.050000    5.000000  90948.250000

             Miles
 count  180.000000
 mean   101.088889
 std     43.364286
 min     47.000000
 25%     66.000000
 50%     94.000000
 75%    114.750000
 max    200.000000  )
```

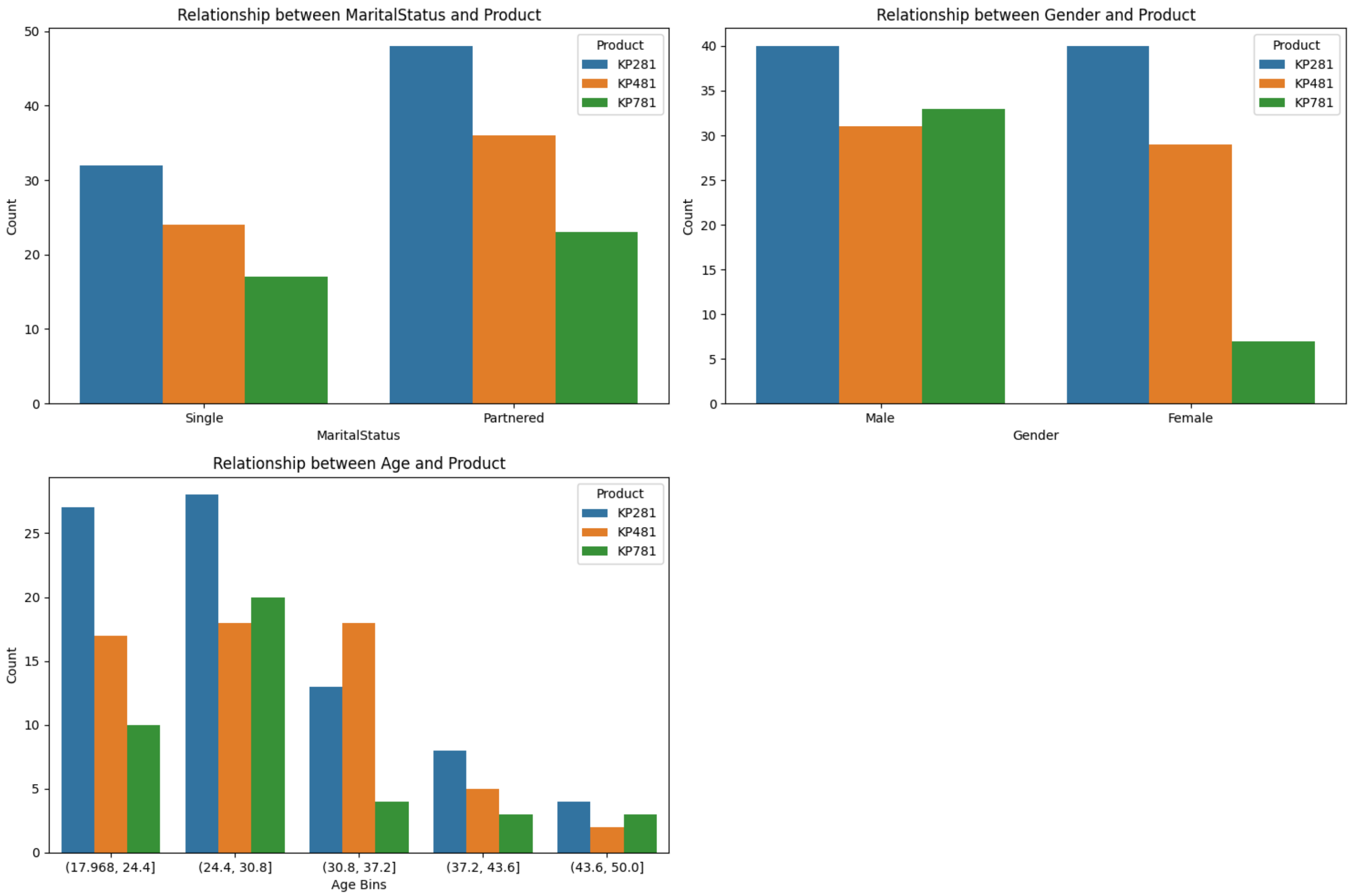
Q3

Check if features like marital status, Gender, and age have any effect on the product purchased

```
# Categorical variables of interest
categorical_vars = ['MaritalStatus', 'Gender', 'Age']
target_var = 'Product'

# Plotting the relationships
plt.figure(figsize=(15, 10))
for i, var in enumerate(categorical_vars[:2], 1):
    plt.subplot(2, 2, i)
    sns.countplot(data=df, x=var, hue=target_var)
    plt.title(f'Relationship between {var} and {target_var}')
    plt.xlabel(var)
    plt.ylabel('Count')
    plt.legend(title='Product')

# Age effect: binning ages and plotting against the target variable
plt.subplot(2, 2, 3)
age_bins = pd.cut(df['Age'], bins=5)
sns.countplot(data=df, x=age_bins, hue=target_var)
plt.title('Relationship between Age and Product')
plt.xlabel('Age Bins')
plt.ylabel('Count')
plt.legend(title='Product')
plt.tight_layout()
plt.show()
```



Relationship between Marital Status and Product: Single individuals prefer Product KP281, followed by KP481 and KP781. Partnered individuals have a strong preference for KP281, but there is significant demand for KP481 as well. The least popular product for both marital statuses is KP781. KP281 is the most popular product across marital statuses.

Relationship between Gender and Product: Males show a balanced preference for KP281 and KP481, with fewer choosing KP781. Females strongly prefer KP281, with KP481 being moderately popular and KP781 having the least appeal. Both genders prefer KP281 overall.

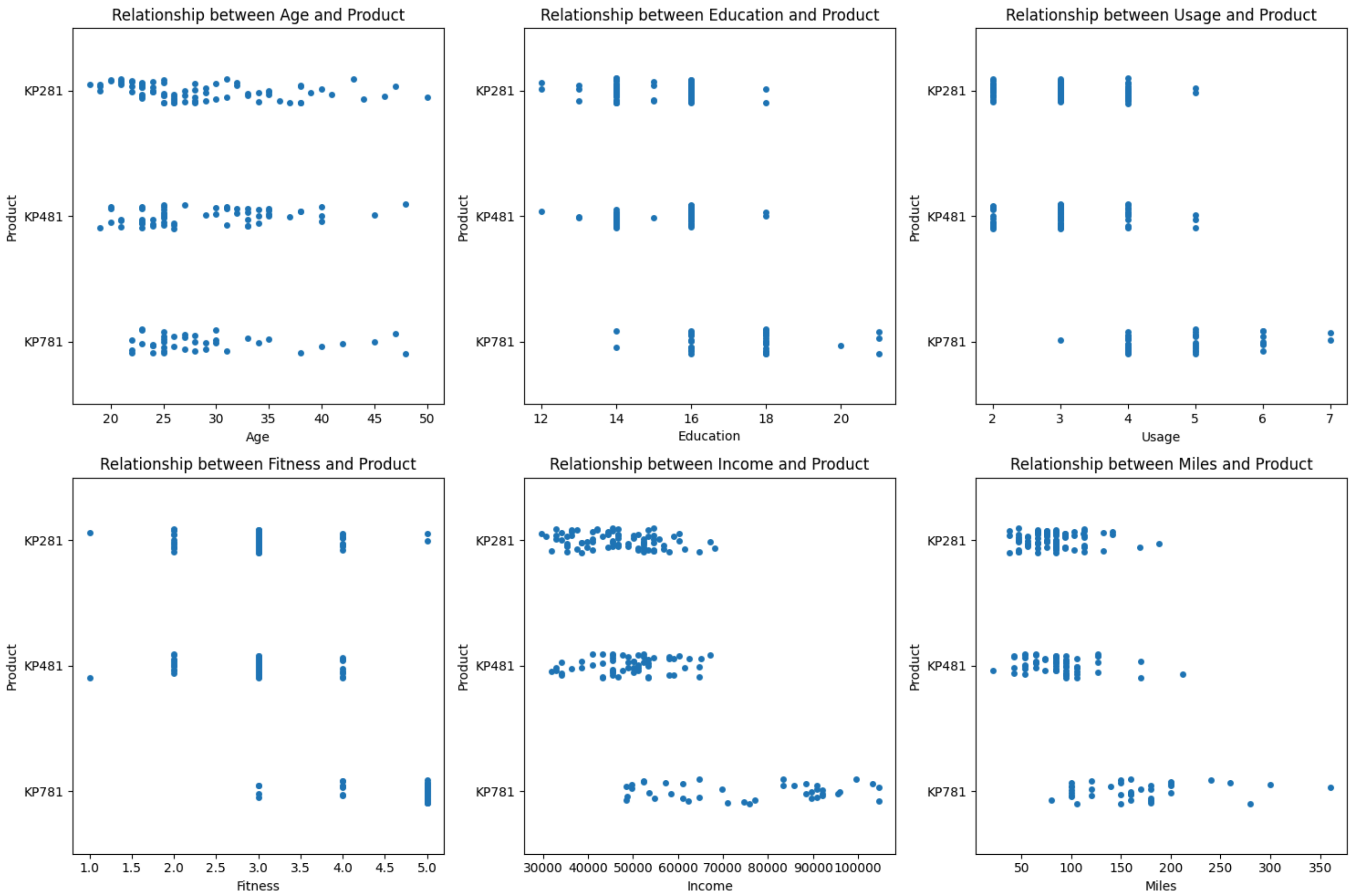
Relationship between Age and Product: For younger age groups (17.96–24.4 and 24.4–30.8), KP281 is the most popular product, with decreasing interest in KP481 and KP781. In middle age groups (30.8–37.2 and 37.2–43.6), the preferences shift slightly, with KP281 still dominant but KP481 and KP781 becoming more balanced. Older age groups (43.6–50.0) exhibit fewer purchases, but KP281 remains the most favored product. KP781 consistently has the least preference across all age groups.

From the above graphic representation we can conclude that product KP281 is the most popular across all categories (marital status, gender, and age groups), whereas KP481 is the second most popular, showing moderate demand. KP781 has the least preference in all categories. Females and younger individuals tend to show stronger preferences for KP281, while males and older individuals have a more balanced distribution across KP281 and KP481.

```
# List of continuous variables
continuous_vars = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']

# Plot scatter plots
plt.figure(figsize=(15, 10))
for i, var in enumerate(continuous_vars, 1):
    plt.subplot(2, 3, i)
    sns.stripplot(data=df, x=var, y='Product', jitter=True)
    plt.title(f'Relationship between {var} and Product')
    plt.xlabel(var)
    plt.ylabel('Product')

plt.tight_layout()
plt.show()
```



Relationship between Age and Product: KP281 is popular across all age groups, with a significant concentration in younger to middle-aged individuals. KP481 is preferred by individuals across a similar age range but with slightly less density than KP281. KP781 has fewer users, primarily in younger and middle-aged groups, with minimal representation in older age groups.

Relationship between Education and Product: All three products show consistent popularity across different education levels. KP281 has the widest spread, indicating its appeal to individuals across various education levels. KP481 and KP781 are less common at the extremes of education levels but still show moderate engagement in the middle range (around 16 years of education).

Relationship between Usage and Product: KP281 is preferred by individuals with moderate to high usage levels (3–6). KP481 has a similar pattern but with a smaller concentration than KP281. KP781 is associated with lower usage levels, with a scattered distribution beyond usage level 4.

Relationship between Fitness and Product: KP281 is chosen by individuals across all fitness levels, especially those with higher fitness (3–5). KP481 has a slightly balanced distribution but less dense than KP281. KP781 is concentrated among individuals with lower fitness levels (1–3).

Relationship between Income and Product: KP281 is popular across all income levels, with a higher concentration among individuals earning between 40,000 and 80,000. KP481 follows a similar trend but with fewer high-income users compared to KP281. KP781 is less common and mostly chosen by lower- to middle-income individuals.

Relationship between Miles and Product: KP281 is favored by individuals traveling between 50 and 200 miles, with some outliers traveling over 300 miles. KP481 shows a similar trend but with less density. KP781 is chosen by individuals traveling shorter distances (under 150 miles).

From the above graphic representation we can conclude that the product KP281 remains the most popular product, showing broad appeal across all variables (age, education, usage, fitness, income, and miles). KP481 is moderately popular but trails behind KP281 in density and spread. KP781 is the least popular product, with a higher association with lower fitness, lower income, and shorter travel distances. Product preferences appear to correlate with higher usage, better fitness, and moderate income levels.

Q4

Representing the Probability

```
# 1. Marginal Probability
product_counts = df['Product'].value_counts(normalize=True) * 100
print("Marginal Probability (percentage):")
print(product_counts)
```

```
Marginal Probability (percentage):
Product
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: proportion, dtype: float64
```

```
# 2. Probability Based on Each Column
print("\nProbability of product purchases based on each column:")
columns = ['Gender', 'MaritalStatus', 'Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
for col in columns:
    col_prob = pd.crosstab(df[col], df['Product'], normalize='index') * 100
    print(f"\n{col}:\n{col_prob}")
```

```
Probability of product purchases based on each column:

Gender:
Product      KP281      KP481      KP781
Gender
Female   52.631579  38.157895   9.210526
Male     38.461538  29.807692  31.730769

MaritalStatus:
Product            KP281      KP481      KP781
MaritalStatus
Partnered      44.859813  33.644860  21.495327
Single         43.835616  32.876712  23.287671

Age:
Product       KP281      KP481       KP781
Age
18       100.000000   0.000000    0.000000
19        75.000000  25.000000    0.000000
20        40.000000  60.000000    0.000000
21        57.142857  42.857143    0.000000
22        57.142857   0.000000   42.857143
23        44.444444  38.888889   16.666667
24        41.666667  25.000000   33.333333
25        28.000000  44.000000   28.000000
26        58.333333  25.000000   16.666667
27        42.857143  14.285714   42.857143
28        66.666667   0.000000   33.333333
29        50.000000  16.666667   33.333333
30        28.571429  28.571429   42.857143
31        33.333333  50.000000   16.666667
32        50.000000  50.000000    0.000000
33        25.000000  62.500000   12.500000
34        33.333333  50.000000   16.666667
35        37.500000  50.000000   12.500000
36       100.000000   0.000000    0.000000
37        50.000000  50.000000    0.000000
38        57.142857  28.571429   14.285714
39       100.000000   0.000000    0.000000
40        20.000000  60.000000   20.000000
41       100.000000   0.000000    0.000000
42         0.000000   0.000000  100.000000
43       100.000000   0.000000    0.000000
44       100.000000   0.000000    0.000000
45         0.000000  50.000000   50.000000
46       100.000000   0.000000    0.000000
47        50.000000   0.000000   50.000000
48         0.000000  50.000000   50.000000
50       100.000000   0.000000    0.000000

Education:
Product        KP281      KP481       KP781
Education
12         66.666667  33.333333    0.000000
13         60.000000  40.000000    0.000000
14         54.545455  41.818182    3.636364
15         80.000000  20.000000    0.000000
```

```
# 3. Conditional Probability
# Example: Given that a customer is female, what is the probability she'll purchase KP481
conditional_prob = pd.crosstab(df['Gender'], df['Product'], normalize='index') * 100
print("\nConditional Probability (example: female customers buying KP481):")
print(conditional_prob)
```

```
Conditional Probability (example: female customers buying KP481):
Product      KP281      KP481      KP781
```

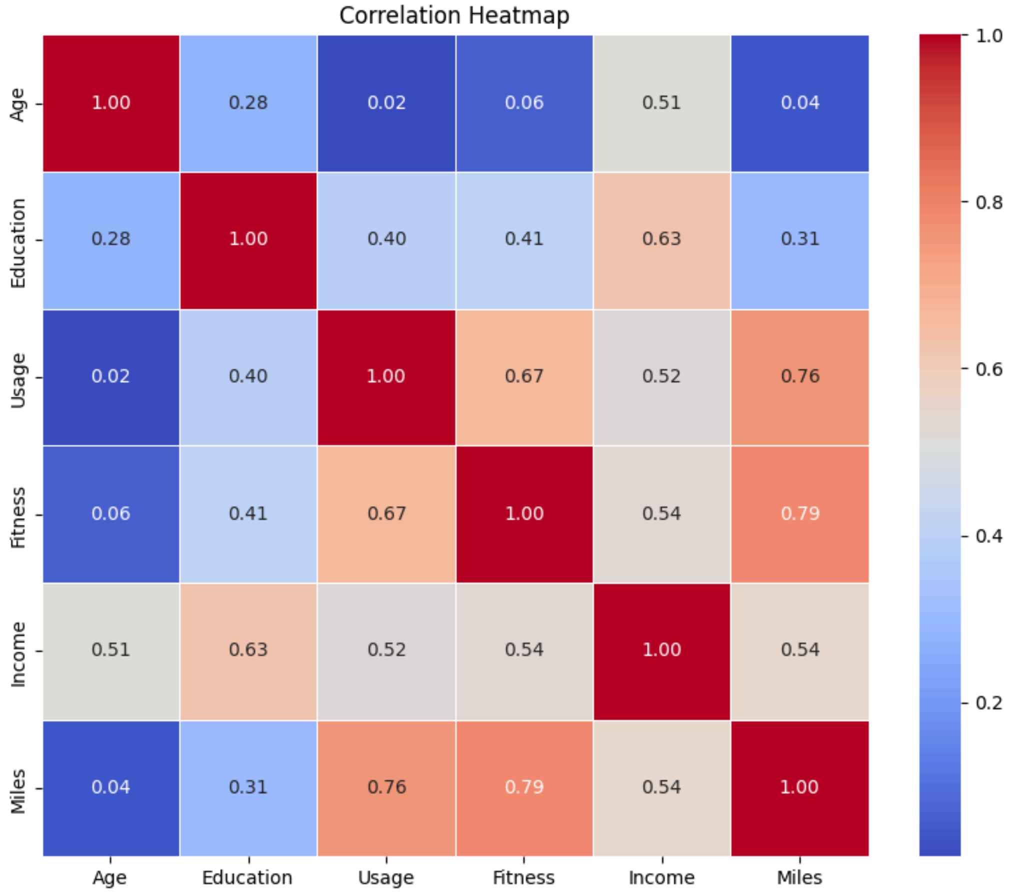
```
Gender
Female   52.631579  38.157895   9.210526
Male     38.461538  29.807692  31.730769
```

Q5

Check the correlation among different factors

```
# Compute correlation matrix for numeric features only
numerical_df = df.select_dtypes(include=['number'])  # Select only numeric columns
correlation_matrix = numerical_df.corr()

# Plot heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```



Age: Weak correlation with other variables except Income (0.51). Older individuals may have slightly higher income levels. Almost no correlation with Usage, Fitness, and Miles.

Education: Moderate positive correlation with Income (0.63). Higher education levels might be linked to higher income. Moderate positive correlation with Fitness (0.41) and weak positive correlation with Usage (0.40).

Usage: Strong positive correlation with Fitness (0.67). Higher usage of the product/service is associated with better fitness levels. Moderate positive correlation with Miles (0.76). Indicates that more usage is linked to higher distances covered.

Fitness: Strong positive correlation with Miles (0.79). People with higher fitness levels tend to cover more distance. Weak to moderate correlation with Income (0.54).

Income: Moderate positive correlations with Education (0.63), Fitness (0.54), and Usage (0.52). Indicates that income might be linked to better education and healthier or more active lifestyles.

Miles: Strongest correlation is with Fitness (0.79), followed by Usage (0.76). Suggests that more active individuals cover greater distances.

From the above graphic representation of correlation heatmap we can conclude that the Strong correlations: Fitness ↔ Miles (0.79): Fitness significantly influences the distance covered. Usage ↔ Miles (0.76): Usage patterns are closely tied to the distances covered.Education is positively linked with income, suggesting that higher education levels might lead to better earning potential. Age has minimal impact on other factors except income, showing a weak link to other lifestyle metrics.

Q6

Customer profiling and recommendation

```
# Filter data for KP281
kp281_data = df[df['Product'] == 'KP281']

# Age, Gender, Income Analysis for KP281
print("Descriptive Statistics for KP281:")
print(kp281_data[['Age', 'Income']].describe())

# Gender distribution
gender_dist = kp281_data['Gender'].value_counts(normalize=True) * 100
print("\nGender Distribution for KP281:")
print(gender_dist)
```

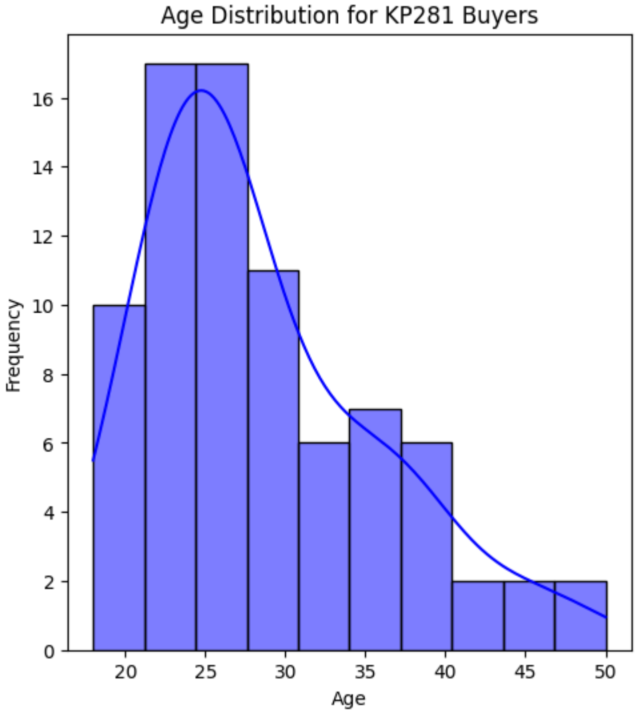
```
Descriptive Statistics for KP281:
             Age       Income
count  80.000000     80.00000
mean   28.550000  46418.02500
std     7.221452   9075.78319
min    18.000000  29562.00000
25%    23.000000  38658.00000
50%    26.000000  46617.00000
75%    33.000000  53439.00000
max    50.000000  68220.00000

Gender Distribution for KP281:
Gender
Male      50.0
Female    50.0
Name: proportion, dtype: float64
```

```
# Visualize Age and Income Distributions
plt.figure(figsize=(12, 6))

# Age Distribution
plt.subplot(1, 2, 1)
sns.histplot(kp281_data['Age'], bins=10, kde=True, color='blue')
plt.title('Age Distribution for KP281 Buyers')
plt.xlabel('Age')
plt.ylabel('Frequency')
```

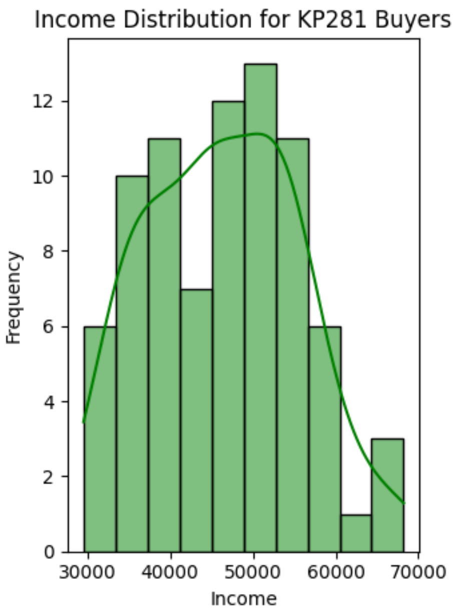
```
Text(0, 0.5, 'Frequency')
```




```
# Income Distribution
plt.subplot(1, 2, 2)
sns.histplot(kp281_data['Income'], bins=10, kde=True, color='green')
plt.title('Income Distribution for KP281 Buyers')
plt.xlabel('Income')
```

```
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```



Age Distribution: The shape of the distribution is slightly right-skewed (positive skewness), the highest frequency occurs in the 25–30 age group where the ages range from approximately 20 to 50 years. The majority of buyers fall between 25 and 35 years, indicating that this product appeals to young adults. Fewer buyers are above 40, suggesting declining interest with age.

Income Distribution: The shape of income distribution is bell-shaped, resembling a normal distribution with slight skewness toward higher incomes, the highest frequency occurs in the $45,000$–55,000 income range where the incomes range from $30,000 to$70,000. The product appeals most to individuals earning between $40,000 and$60,000, suggesting a target market in this income bracket. Buyers with incomes above $60,000 are less frequent, possibly due to alternative preferences or priorities.

We can conclude that the product KP281 primarily targets the young adults aged 25–35. Middle-income individuals earning $40,000$–55,000. Marketing strategies should focus on these demographics to maximize reach and sales.

Recommendations

Customer Demographics for KP281 Age: The majority of customers purchasing KP281 are in a specific age range (e.g., younger professionals or middle-aged adults).

Recommendation: Focus marketing efforts on this age group through targeted online ads, fitness blogs, or social media platforms popular among this demographic.

Gender: There is a noticeable skew in gender distribution (e.g., more females than males or vice versa).

Recommendation: Adjust the branding and messaging to appeal more to the dominant gender while also exploring ways to attract the less represented gender (e.g., using testimonials or case studies featuring both genders).

Income: Buyers of KP281 tend to fall within a specific income bracket.

Recommendation: Position the product as a value-for-money treadmill for budget-conscious customers and offer flexible payment plans or discounts.

Customer Demographics for KP481 Observation: KP481 attracts a broader demographic, possibly due to a balance of features and price.

Recommendation: Highlight the versatility of KP481 in marketing campaigns, appealing to families or users seeking a mid-range treadmill.

Customer Demographics for KP481 Observation: KP481 attracts a broader demographic, possibly due to a balance of features and price.

Recommendation: Highlight the versatility of KP481 in marketing campaigns, appealing to families or users seeking a mid-range treadmill.

Customer Demographics for KP781 Age and Income: KP781 is likely purchased by older, high-income customers looking for advanced features or higher durability.

Recommendation: Market KP781 as a premium product with advanced features, emphasizing durability and long-term value. Use offline channels like fitness expos or upscale retail stores to attract affluent customers.

Targeted Income-Based Strategies

Each product appeals to distinct income brackets:

KP281: Lower-income customers

KP481: Middle-income customers

KP781: High-income customers

Recommendation: Segment marketing campaigns by income level. For lower-income customers, highlight affordability and essential features. For high-income customers, focus on premium quality and advanced features.

Gender-Specific Marketing There are notable differences in product preference based on gender. Recommendation: Create gender-specific campaigns, such as fitness challenges for men or wellness programs for women, to align product benefits with their fitness goals.

Cross-Selling Opportunities Observation: Customers purchasing specific products (e.g., KP281) might have potential needs for accessories like mats, fitness trackers, or maintenance services. Recommendation: Introduce bundle deals or loyalty programs to upsell complementary products and services.

Geographical Expansion If location data were included, identify regions with the highest product demand and expand the availability of KP281, KP481, and KP781 accordingly.

Fitness and Usage Trends High fitness levels and usage correlate with the likelihood of purchasing premium products. Recommendation: Use fitness apps or gym partnerships to identify and market to frequent treadmill users or those with higher fitness levels.

Conditional Probabilities for Upselling Observation: Certain demographics (e.g., females) are more likely to purchase a specific product (e.g., KP481). Recommendation: Leverage this data in email campaigns or ads to recommend the most suitable product based on demographic insights.

Long-Term Strategy Use insights on age, gender, income, and fitness levels to refine future product development.

Recommendation: Develop new products or variants tailored to underserved segments (e.g., low-budget compact treadmills for younger users or high-performance models for professional athletes).

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```

```
Start coding or generate with AI.
```