Aerofit
Fitness Store

:

About Aerofit:

Aerofit is a leading brand offering fitness equipment such as treadmills, exercise bikes, gym equipment, and accessories catering to diverse customer needs.

Objective:

The objective is to create customer profiles for each Aerofit treadmill product through descriptive analytics. Additionally, construct two-way tables to compute conditional and marginal probabilities, providing insights for business decisions.

Product Portfolio:

1. KP281: Entry-level treadmill priced at USD 1,500.
2. KP481: Mid-level treadmill priced at USD 1,750.
3. KP781: Advanced treadmill with premium features priced at USD 2,500.

Features of the Dataset:

- Product: Purchased treadmill model (KP281, KP481, or KP781).
- Age: Customer's age in years.
- Gender: Male/Female.
- Education: Customer's education level in years.
- MaritalStatus: Customer's marital status (Single or partnered).

- Usage: Average weekly usage of the treadmill.
- Income: Annual income in USD.
- Fitness: Self-rated fitness level on a scale of 1 to 5.
- Miles: Average weekly distance expected to walk/run on the treadmill.

1. Exploratory Data Analysis

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

In [ ]:
```
# Importing the data set
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv
```

```
--2024-03-23 11:16:05--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_trea
dmill.csv
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.172.139.46, 18.172.139.61, 18.172.139.9
4, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.172.139.46|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7279 (7.1K) [text/plain]
Saving to: 'aerofit_treadmill.csv'

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

2024-03-23 11:16:05 (1.36 GB/s) - 'aerofit_treadmill.csv' saved [7279/7279]
```

In [ ]:
```
# Read the CSV file
df = pd.read_csv('aerofit_treadmill.csv')
df.head()
```

Out[ ]:

	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

In [ ]:

```
# To check the number of rows and column given in the dataset.
df.shape
```

Out[ ]:

```
(180, 9)
```

In [ ]:

```
# To check the missing value present in the dataset.
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
```

Insights:

Based on the analysis, it's evident that the dataset doesn't contain any missing values.

In [ ]:
```
# To check the Datatype of all the columns present in a dataset.
df.dtypes
```

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

1.1 Statistical Summary

In [ ]:
```
# Statistical Summary(object type columns):
df.describe(include = 'object')
```

Out[ ]:

	Product	Gender	MaritalStatus
count	180	180	180
unique	3	2	2
top	KP281	Male	Partnered
freq	80	104	107

In [ ]:
```
# Statistical summary of numeric data type columns
df.describe()
```

Out[ ]:

	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

1.2 Duplicate Detection

In [ ]:

```
df.duplicated().value_counts()
```

Out[ ]:

```
False    180
dtype: int64
```

Insights:

The dataset does not contain any duplicate entries.

1.3 Sanity check for columns

In [ ]:

```
# Checking unique values for all columns

for column in df.columns:
    unique_values = df[column].unique()
    print(f'Unique values in {column} column are:')
    print(unique_values)
    print('-' * 50)
```

```
Unique values in Product column are:
['KP281' 'KP481' 'KP781']
--------------------------------------------------
Unique values in Age column are:
[18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
 43 44 46 47 50 45 48 42]
--------------------------------------------------
Unique values in Gender column are:
['Male' 'Female']
--------------------------------------------------
Unique values in Education column are:
[14 15 12 13 16 18 20 21]
--------------------------------------------------
Unique values in MaritalStatus column are:
['Single' 'Partnered']
--------------------------------------------------
Unique values in Usage column are:
[3 2 4 5 6 7]
--------------------------------------------------
Unique values in Fitness column are:
[4 3 2 1 5]
--------------------------------------------------
Unique values in Income column are:
[ 29562  31836  30699  32973  35247  37521  36384  38658  40932  34110
  39795  42069  44343  45480  46617  48891  53439  43206  52302  51165
  50028  54576  68220  55713  60261  67083  56850  59124  61398  57987
  64809  47754  65220  62535  48658  54781  48556  58516  53536  61006
  57271  52291  49801  62251  64741  70966  75946  74701  69721  83416
  88396  90886  92131  77191  52290  85906 103336  99601  89641  95866
 104581  95508]
--------------------------------------------------
Unique values in Miles column are:
[112  75  66  85  47 141 103  94 113  38 188  56 132 169  64  53 106  95
 212  42 127  74 170  21 120 200 140 100  80 160 180 240 150 300 280 260
 360]
--------------------------------------------------
```

In [ ]:

```
# Checking the number of unique values for columns

for i in df.columns:
    print('Unique values in',i,'column are :')
    print(df[i].nunique())
    print('-'*50)
```

```
Unique values in Product column are :
3
--------------------------------------------------
Unique values in Age column are :
32
--------------------------------------------------
Unique values in Gender column are :
2
--------------------------------------------------
Unique values in Education column are :
8
--------------------------------------------------
Unique values in MaritalStatus column are :
2
--------------------------------------------------
Unique values in Usage column are :
6
--------------------------------------------------
Unique values in Fitness column are :
5
--------------------------------------------------
Unique values in Income column are :
62
--------------------------------------------------
Unique values in Miles column are :
37
--------------------------------------------------
```

Insights:

The dataset does not contain any outliers or abnormal values.

2. Detect Outliers

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

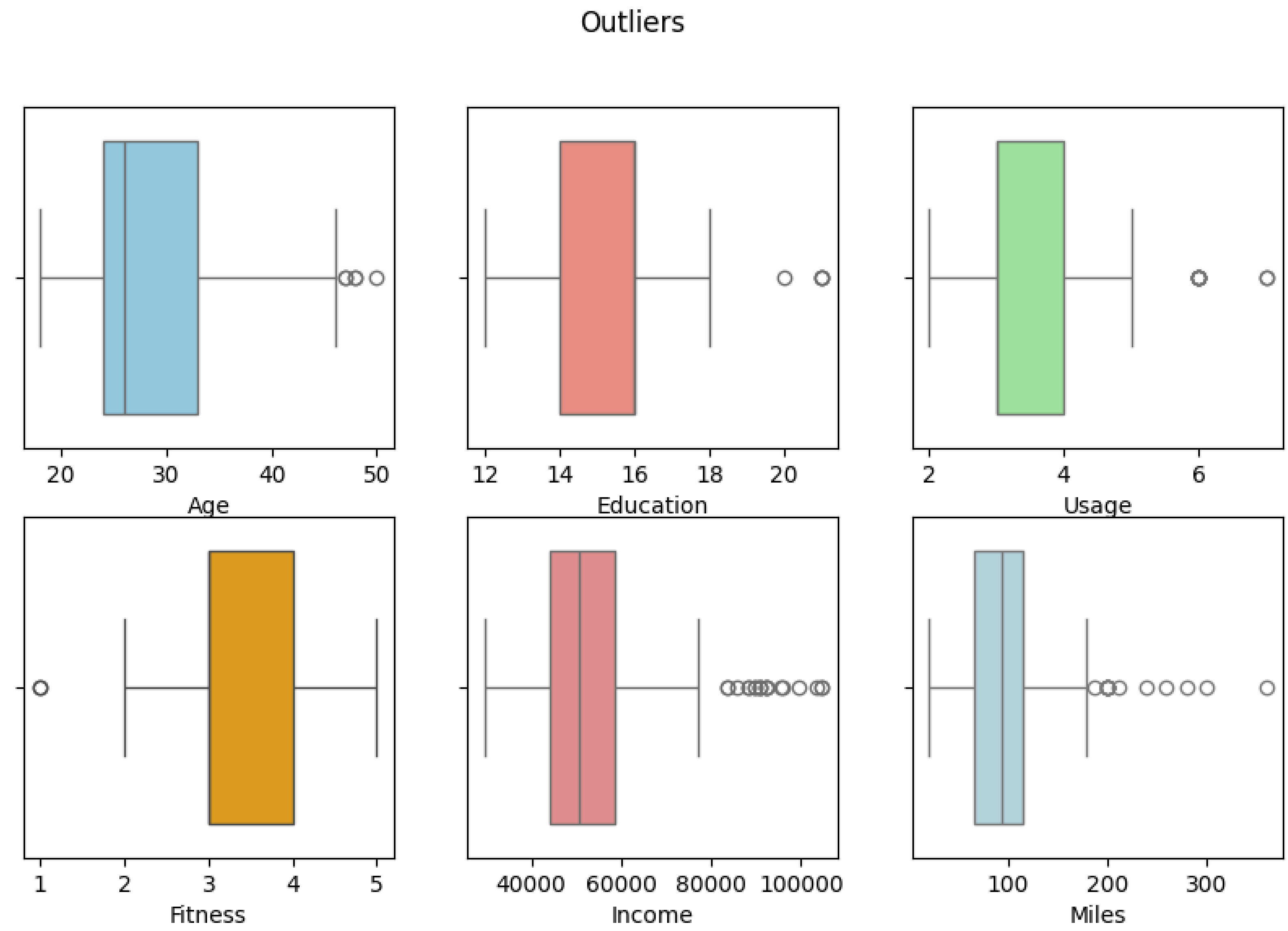
Out[ ]:

	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

2.1 Finding outliers using boxplots

In [ ]:

```
fig, ax = plt.subplots(2,3, figsize = (10,6))
fig.suptitle("Outliers")

sns.boxplot(data=df, x = "Age", ax=ax[0,0], color='skyblue')
sns.boxplot(data=df, x = "Education", ax=ax[0,1], color='salmon')
sns.boxplot(data=df, x = "Usage", ax=ax[0,2], color='lightgreen')
sns.boxplot(data=df, x = "Fitness", ax=ax[1,0], color='orange')
sns.boxplot(data=df, x = "Income", ax=ax[1,1], color='lightcoral')
sns.boxplot(data=df, x = "Miles", ax=ax[1,2], color='lightblue')

plt.show()
```

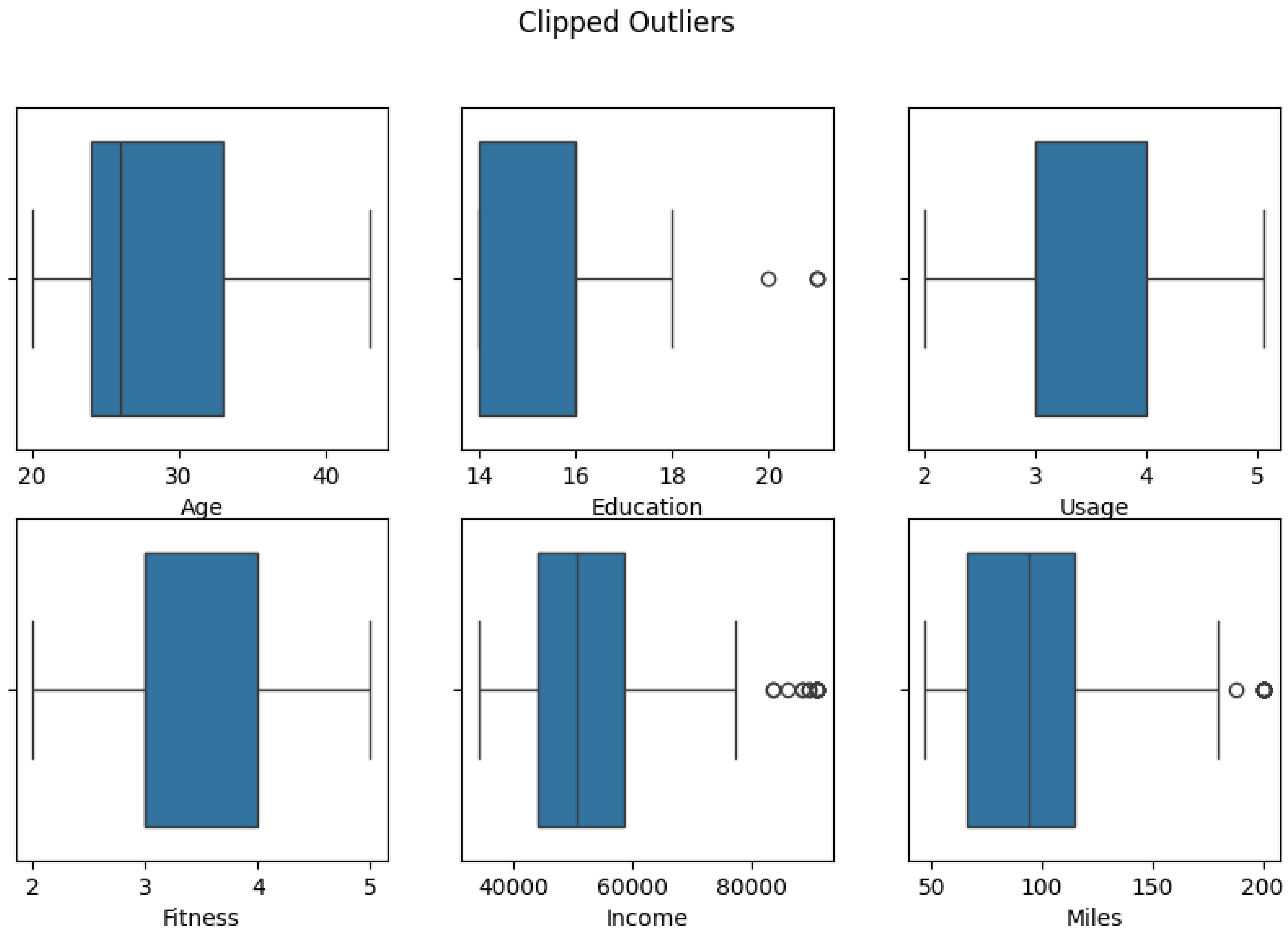


Insights:

Upon examining the graphs, it's evident that Income and Miles display a significant abundance of outliers, while the other variables show a relatively lower presence of outliers.

2.2 Remove/clip the data between the 5 percentile and 95 percentile.

In [ ]:
```
# Cliping the data between the 5 percentile and 95 percentile
clipped_Age = np.clip(df['Age'], np.percentile(df['Age'],5), np.percentile(df['Age'],95))
clipped_Education = np.clip(df['Education'], np.percentile(df['Education'],5), np.percentile(df['Age'],95))
clipped_Usage = np.clip(df['Usage'], np.percentile(df['Usage'],5), np.percentile(df['Usage'],95))
clipped_Fitness = np.clip(df['Fitness'], np.percentile(df['Fitness'],5), np.percentile(df['Fitness'],95))
clipped_Income = np.clip(df['Income'], np.percentile(df['Income'],5), np.percentile(df['Income'],95))
clipped_Miles = np.clip(df['Miles'], np.percentile(df['Miles'],5), np.percentile(df['Miles'],95))

fig, ax = plt.subplots(2,3, figsize = (10,6))
fig.suptitle("Clipped Outliers")

plt.subplot(2,3,1)
sns.boxplot(data=df, x = clipped_Age)

plt.subplot(2,3,2)
sns.boxplot(data=df, x = clipped_Education)

plt.subplot(2,3,3)
sns.boxplot(data=df, x = clipped_Usage)

plt.subplot(2,3,4)
sns.boxplot(data=df, x = clipped_Fitness)

plt.subplot(2,3,5)
sns.boxplot(data=df, x = clipped_Income)

plt.subplot(2,3,6)
sns.boxplot(data=df, x = clipped_Miles)

plt.show()
```



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

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

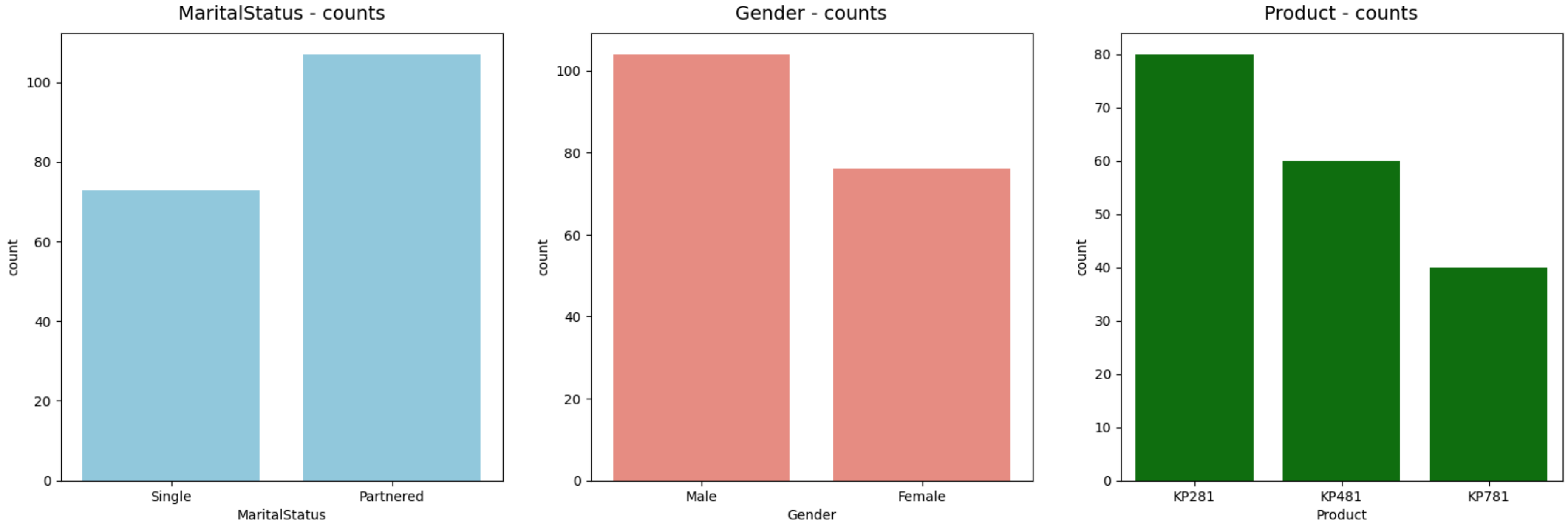
Out[ ]:

	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

3.1 Univariate Analysis

In [ ]:

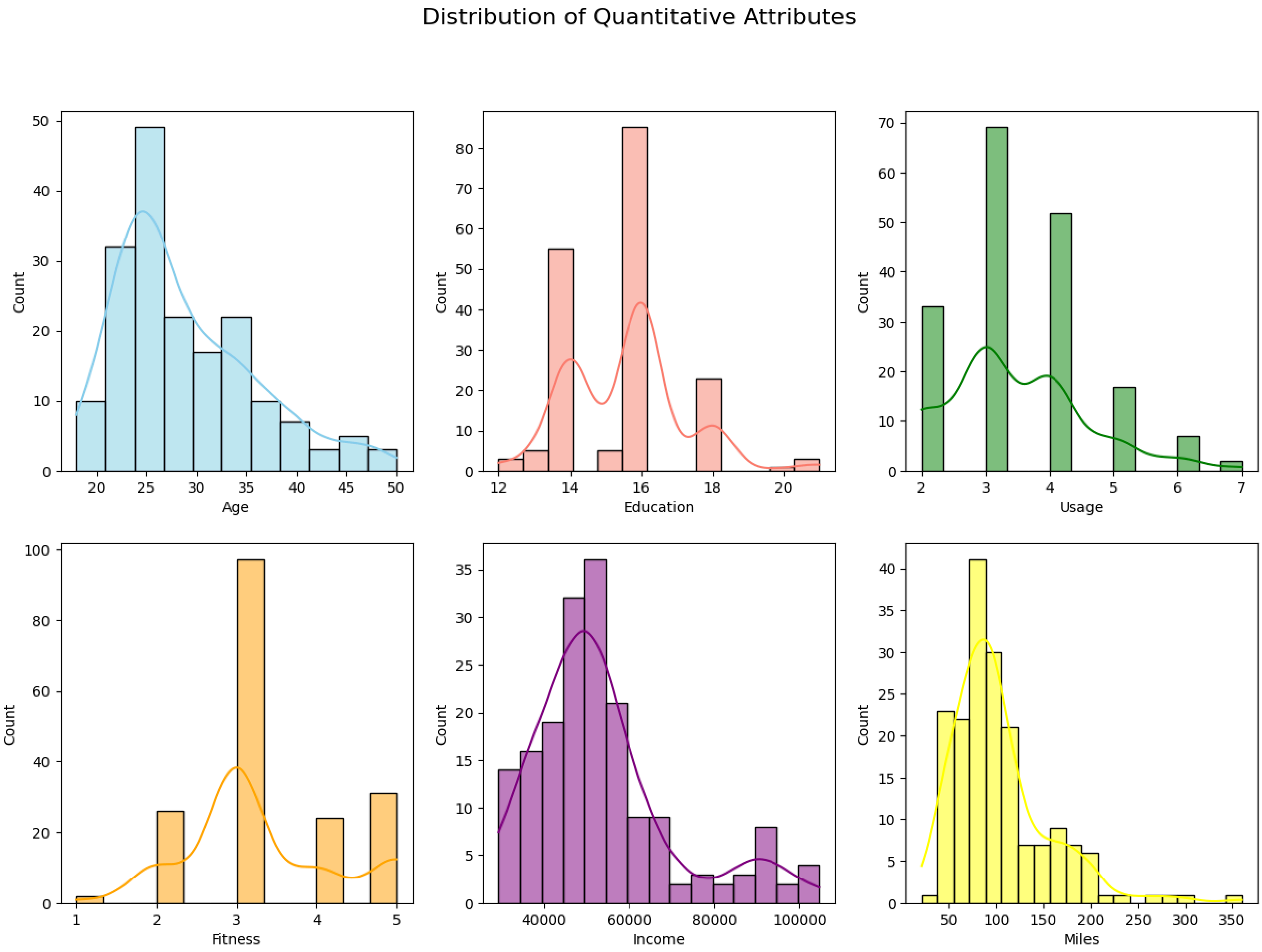
```
fig, axs = plt.subplots(1, 3, figsize=(20, 6))
sns.countplot(data=df, x='MaritalStatus', ax=axs[0], color='skyblue')
sns.countplot(data=df, x='Gender', ax=axs[1], color='salmon')
sns.countplot(data=df, x='Product', ax=axs[2], color='green')

axs[0].set_title('MaritalStatus - counts', pad=10, fontsize=14)
axs[1].set_title('Gender - counts', pad=10, fontsize=14)
axs[2].set_title('Product - counts', pad=10, fontsize=14)
plt.show()
```



Insights:

The dataset predominantly consists of male customers, with a higher representation than females. Partnered customers appear to outnumber single individuals. Moreover, the product KP281 emerges as the most frequently purchased item among customers.

In [ ]:
```
fig, ax = plt.subplots(2, 3, figsize=(15, 10))
fig.suptitle('Distribution of Quantitative Attributes', fontsize=16)

# Plotting with different colors
sns.histplot(data=df, x='Age', kde=True, color='skyblue', ax=ax[0, 0])
sns.histplot(data=df, x='Education', kde=True, color='salmon', ax=ax[0, 1])
sns.histplot(data=df, x='Usage', kde=True, color='green', ax=ax[0, 2])
sns.histplot(data=df, x='Fitness', kde=True, color='orange', ax=ax[1, 0])
sns.histplot(data=df, x='Income', kde=True, color='purple', ax=ax[1, 1])
sns.histplot(data=df, x='Miles', kde=True, color='yellow', ax=ax[1, 2])

plt.show()
```


Distribution of Quantitative Attributes
Count
Age
Education
Usage
Fitness
Income
Miles

3.2 Bivariate Analysis

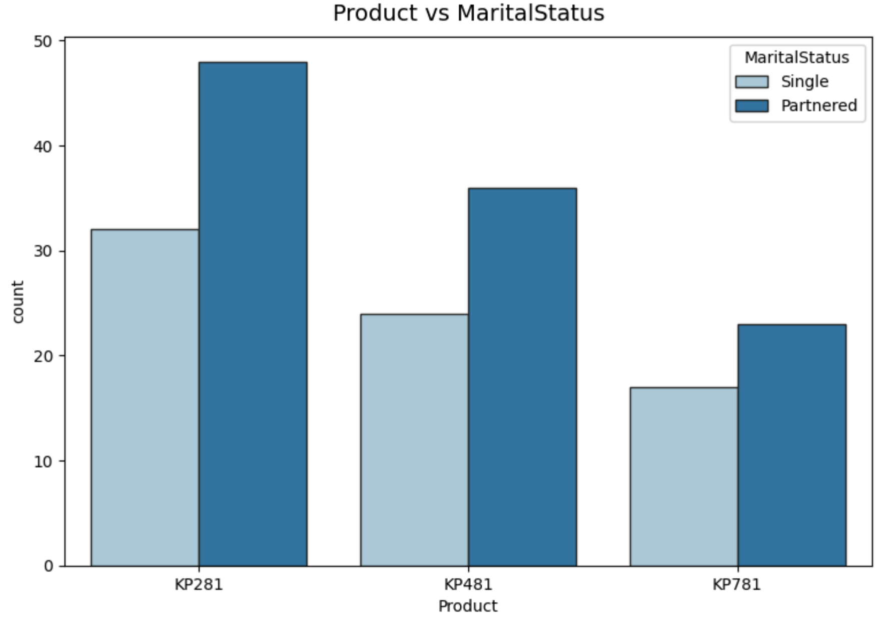
To be precise - normalize count for each variable is shown below

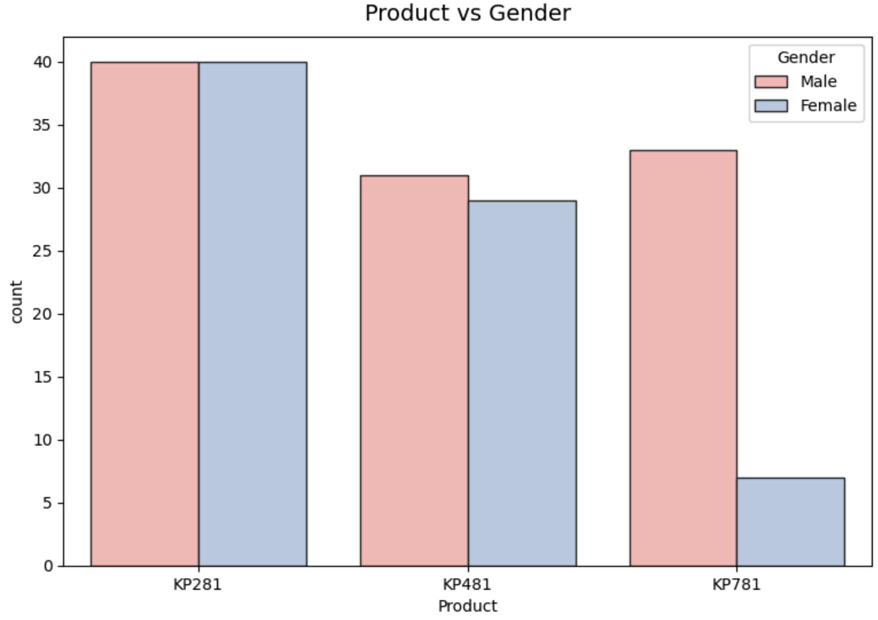
In [ ]:
```
df1 = df[['MaritalStatus', 'Gender', 'Product']].melt()
df1.groupby(['variable', 'value'])[['value']].count() / len(df)
```

Out[ ]:

variable	value	value
Gender	Female	0.422222
	Male	0.577778
MaritalStatus	Partnered	0.594444
	Single	0.405556
Product	KP281	0.444444
	KP481	0.333333
	KP781	0.222222

In [ ]:
```
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(20, 6))
sns.countplot(data=df, x='Product', hue='MaritalStatus', edgecolor='0.15', palette='Paired', ax=axs[0])
sns.countplot(data=df, x='Product', hue='Gender', edgecolor='0.15', palette='Pastel1', ax=axs[1])

axs[0].set_title('Product vs MaritalStatus', pad=10, fontsize=14)
axs[1].set_title('Product vs Gender', pad=10, fontsize=14)
plt.show()
```



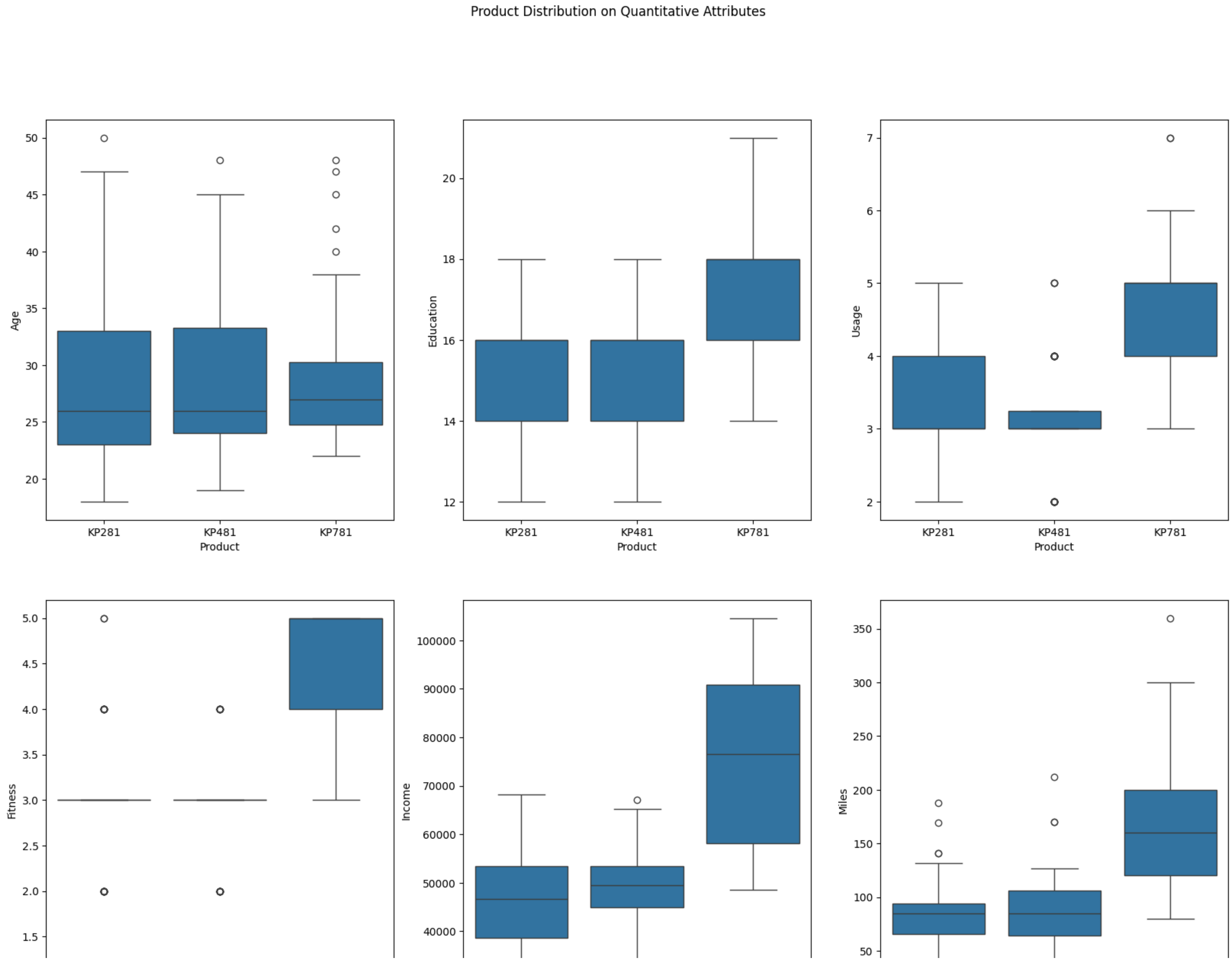


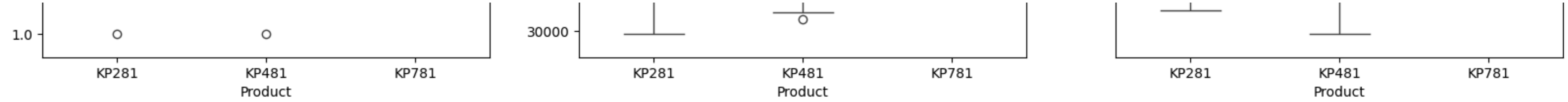
Insights:

The countplot above illustrates that both males and females use the product KP281 in nearly equal proportions, with a majority of users being partnered. Conversely, KP781 and other products seem to be predominantly favored by males.

In [ ]:
```
fig, ax = plt.subplots(2, 3, figsize=(20, 15))
fig.suptitle('Product Distribution on Quantitative Attributes')

plt.subplot(2, 3, 1)
sns.boxplot(data=df, x='Product', y='Age')

plt.subplot(2, 3, 2)
sns.boxplot(data=df, x='Product', y='Education')

plt.subplot(2, 3, 3)
sns.boxplot(data=df, x='Product', y='Usage')

plt.subplot(2, 3, 4)
sns.boxplot(data=df, x='Product', y='Fitness')

plt.subplot(2, 3, 5)
```

```
sns.boxplot(data=df, x='Product', y='Income')

plt.subplot(2, 3, 6)
sns.boxplot(data=df, x='Product', y='Miles')

plt.show()
```


Product Distribution on Quantitative Attributes
Age
Education
Usage
Fitness
Income
Miles
Product
KP281
KP481
KP781



Insights:

1. **Product vs Age:**

   - KP281 and KP481 attract customers aged between 22 and 33, while KP781 gains popularity among those aged 25 to 30, with additional traction observed among individuals over 40.

2. **Product vs Education:**

   - Customers opting for KP281 and KP481 typically have a maximum education level of 16 years, while those choosing KP781 tend to have pursued higher education, reaching 18 years or more.

3. **Product vs Usage:**

   - KP781 is preferred by customers anticipating frequent treadmill usage, exceeding four times a week, while KP281 and KP481 are chosen by customers with varying usage patterns.

4. **Product vs Fitness:**

   - KP781 customers are perceived to be in better physical fitness compared to those choosing other products, indicating a preference among fitness-conscious individuals.

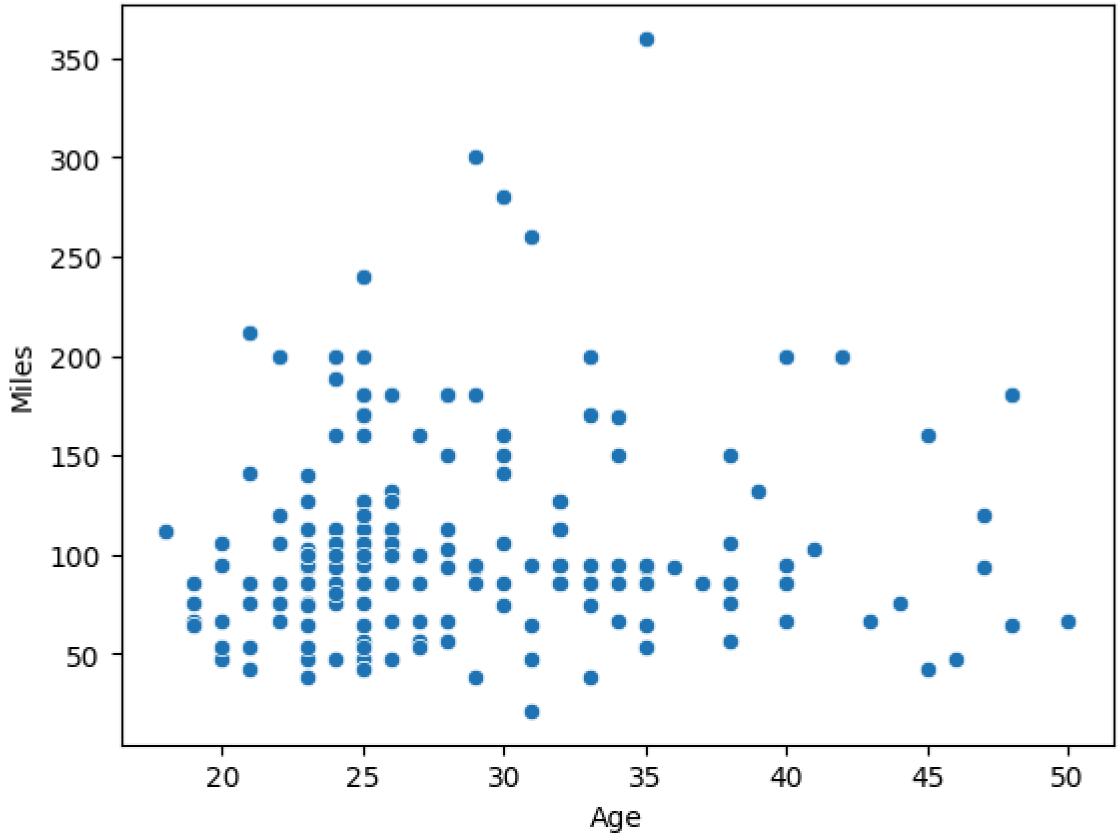
5. **Product vs Income:**

   - Higher-income customers favor KP781, while middle-income customers show a preference for KP281, with slightly higher middle-income individuals opting for KP481.

6. **Product vs Miles:**

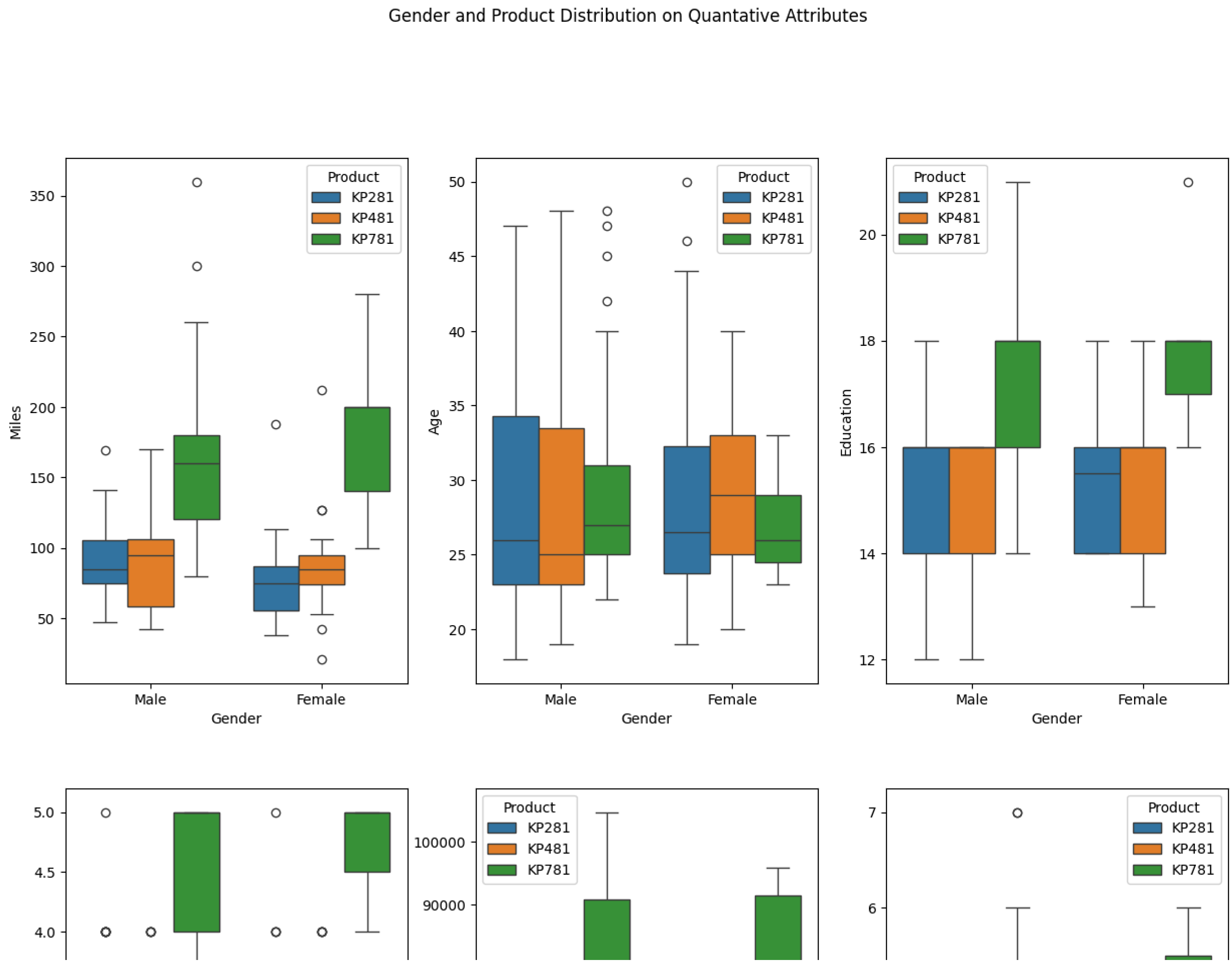
   - KP781 boasts the highest mileage range, indicating its suitability for intense workouts, whereas KP281 and KP481 cater more to moderate exercise, aligning with customers' diverse fitness goals.

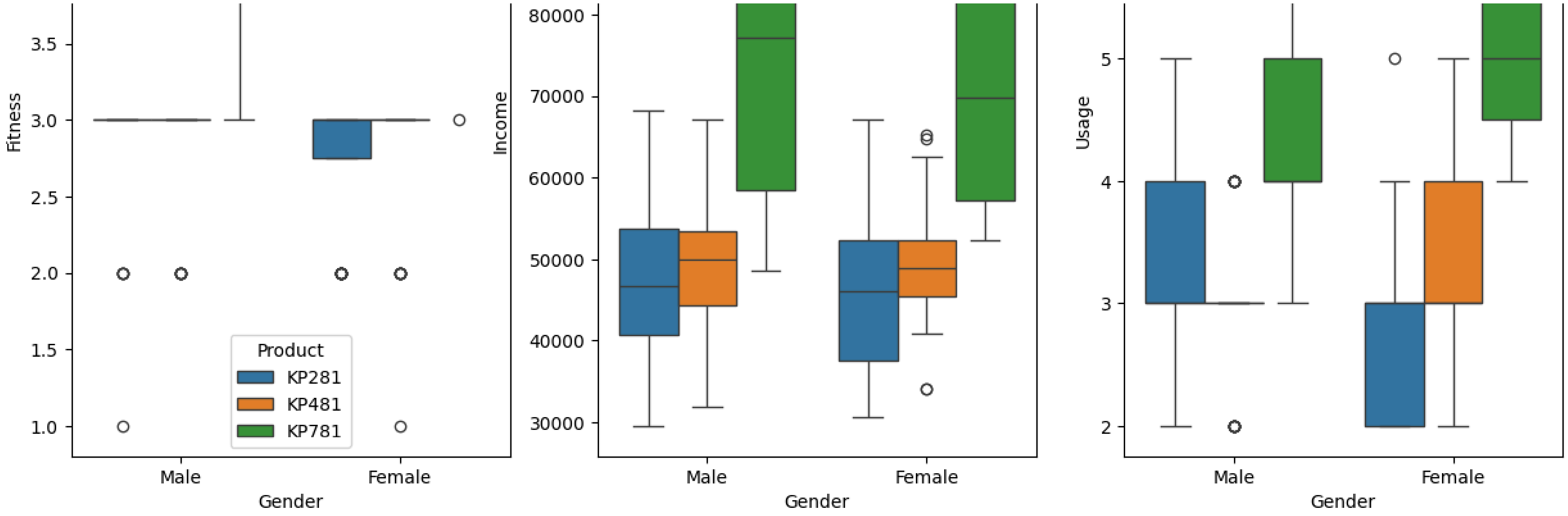
In [ ]:
```
sns.scatterplot(data = df, x = 'Age', y = 'Miles')
plt.show()
```



In [ ]:
```
# Multivariate Analysis
fig, ax = plt.subplots(2,3,figsize = (15,15))
fig.suptitle('Gender and Product Distribution on Quantative Attributes')

plt.subplot(2,3,1)
sns.boxplot(data = df, x = 'Gender', y = 'Miles', hue = 'Product')
plt.subplot(2,3,2)
sns.boxplot(data = df, x = 'Gender', y = 'Age', hue = 'Product')
plt.subplot(2,3,3)
sns.boxplot(data = df, x = 'Gender', y = 'Education', hue = 'Product')
plt.subplot(2,3,4)
sns.boxplot(data = df, x = 'Gender', y = 'Fitness', hue = 'Product')
plt.subplot(2,3,5)
sns.boxplot(data = df, x = 'Gender', y = 'Income', hue = 'Product')
plt.subplot(2,3,6)
```

```
sns.boxplot(data = df, x = 'Gender', y = 'Usage', hue = 'Product')
plt.show()
```


Gender and Product Distribution on Quantative Attributes
Product
KP281
KP481
KP781
Miles
Age
Education
Gender
Male
Female



4. Representing the probability

4.1 Find the marginal probability(what percent of customers have purchased KP281, KP481, or KP781)

4.1 Adding New Columns for Enhanced Analysis

4.1.1 Age Column

Categorizing age values into four distinct buckets:

1. Young Adults: 18-25 years
2. Adults: 26-35 years
3. Middle-Aged Adults: 36-45 years
4. Elder: 46 years and above

4.1.2 Education Column

Grouping education values into three categories:

1. Primary Education: Up to 12 years
2. Secondary Education: 13 to 15 years
3. Higher Education: 16 years and above

4.1.3 Income Column

Dividing income values into four segments:

1. Low Income: Up to $40,000
2. Moderate Income: $40,000 to $60,000
3. High Income: $60,000 to $80,000
4. Very High Income: Above $80,000

4.1.4 Miles Column

Segmenting miles values into four categories:

1. Light Activity: Up to 50 miles
2. Moderate Activity: 51 to 100 miles
3. Active Lifestyle: 101 to 200 miles
4. Fitness Enthusiast: Above 200 miles

In [ ]:
```
# Binning age values into categories
age_bins = [17, 25, 35, 45, float('inf')]
age_labels = ['Young Adults', 'Adults', 'Middle-Aged Adults', 'Elder']
df['age_group'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels)

# Binning education values into categories
edu_bins = [0, 12, 15, float('inf')]
edu_labels = ['Primary Education', 'Secondary Education', 'Higher Education']
df['edu_group'] = pd.cut(df['Education'], bins=edu_bins, labels=edu_labels)

# Binning income values into categories
income_bins = [0, 40000, 60000, 80000, float('inf')]
income_labels = ['Low Income', 'Moderate Income', 'High Income', 'Very High Income']
df['income_group'] = pd.cut(df['Income'], bins=income_bins, labels=income_labels)
```

```
# Binning miles values into categories
miles_bins = [0, 50, 100, 200, float('inf')]
miles_labels = ['Light Activity', 'Moderate Activity', 'Active Lifestyle', 'Fitness Enthusiast']
df['miles_group'] = pd.cut(df['Miles'], bins=miles_bins, labels=miles_labels)
```

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

Out[ ]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	age_group	edu_group	income_group	miles_group
0	KP281	18	Male	14	Single	3	4	29562	112	Young Adults	Secondary Education	Low Income	Active Lifestyle
1	KP281	19	Male	15	Single	2	3	31836	75	Young Adults	Secondary Education	Low Income	Moderate Activity
2	KP281	19	Female	14	Partnered	4	3	30699	66	Young Adults	Secondary Education	Low Income	Moderate Activity
3	KP281	19	Male	12	Single	3	3	32973	85	Young Adults	Primary Education	Low Income	Moderate Activity
4	KP281	20	Male	13	Partnered	4	2	35247	47	Young Adults	Secondary Education	Low Income	Light Activity

In [ ]:
```
# Probability of product purchase w.r.t Gender
pd.crosstab(index = df['Product'], columns = df['Gender'], margins = True, normalize = True).round(2)
```

Out[ ]:

Gender	Female	Male	All
Product			
KP281	0.22	0.22	0.44
KP481	0.16	0.17	0.33
KP781	0.04	0.18	0.22
All	0.42	0.58	1.00

Insights:

1. Females account for 42% of treadmill purchases. Among females:

- 22% opt for KP281
- 16% prefer KP481
- 4% choose KP781

2. Males constitute 58% of treadmill purchases. Among males:

   - 22% opt for KP281
   - 17% prefer KP481
   - 18% choose KP781

These insights reveal the distribution of treadmill purchases based on gender, providing valuable information for targeted marketing strategies.

In [ ]:
```
# Probability of product purchase w.r.t Age
pd.crosstab(index = df['Product'], columns = df['age_group'], margins = True, normalize = True).round(2)
```

Out[ ]:

age_group / Product	Young Adults	Adults	Middle-Aged Adults	Elder	All
KP281	0.19	0.18	0.06	0.02	0.44
KP481	0.16	0.13	0.04	0.01	0.33
KP781	0.09	0.09	0.02	0.01	0.22
All	0.44	0.41	0.12	0.03	1.00

Insights:

1. Among customers aged 18-25 (Young Adults), the probability of purchasing a treadmill is 44%. The conditional probabilities for each treadmill model are:

   - KP281: 19%
   - KP481: 16%
   - KP781: 9%

2. For customers aged 26-35 (Adults), the probability of purchasing a treadmill is 41%. The conditional probabilities for each treadmill model are:

- KP281: 18%
- KP481: 13%
- KP781: 9%

3. Middle-aged customers (36-45) have a lower probability of purchasing a treadmill, at 12%. Conditional probabilities for this age group are not provided, indicating the need for further analysis.

4. Customers above 45 years old (Elder) have the lowest probability of treadmill purchase, at only 3%. No conditional probabilities are given for this group, highlighting the necessity for additional investigation into their preferences.

These insights shed light on the likelihood of purchasing each treadmill model based on the age group of the customer.

In [ ]:
```
# Probability of product purchase w.r.t education
pd.crosstab(index = df['Product'], columns = df['edu_group'], margins = True, normalize = True).round(2)
```

Out[ ]:

edu_group Product	Primary Education	Secondary Education	Higher Education	All
KP281	0.01	0.21	0.23	0.44
KP481	0.01	0.14	0.18	0.33
KP781	0.00	0.01	0.21	0.22
All	0.02	0.36	0.62	1.00

Insights:

1. Customers with Higher Education (Above 15 Years) have a 62% probability of purchasing a treadmill. The conditional probabilities for each treadmill model given Higher Education are:

   - KP281: 23%
   - KP481: 18%
   - KP781: 21%

2. Customers with Secondary Education (13-15 yrs) show a 36% probability of purchasing a treadmill. The conditional probabilities for each treadmill model given Secondary Education are:

- KP281: 21%
- KP481: 14%
- KP781: 1%

3. Customers with Primary Education (0 to 12 yrs) exhibit only a 2% probability of purchasing a treadmill. No specific conditional probabilities are provided for this education level, suggesting a need for further analysis to understand their preferences for each treadmill model.

These insights shed light on the purchasing probabilities for each treadmill model based on the education level of customers.

In [ ]:
```
# Probability of product purchase w.r.t income
pd.crosstab(index = df['Product'], columns = df['income_group'], margins = True, normalize = True).round(2)
```

Out[ ]:

income_group	Low Income	Moderate Income	High Income	Very High Income	All
Product					
KP281	0.13	0.28	0.03	0.00	0.44
KP481	0.05	0.24	0.04	0.00	0.33
KP781	0.00	0.06	0.06	0.11	0.22
All	0.18	0.59	0.13	0.11	1.00

Insights:

1. Customers with Low Income (<40k) have a probability of 18% of purchasing a treadmill. Within this income group:
   - The conditional probability of purchasing KP281 is 13%
   - The conditional probability of purchasing KP481 is 5%
   - The conditional probability of purchasing KP781 is 0%
2. Customers with Moderate Income (40k - 60k) show a probability of 59% of purchasing a treadmill. Among this income segment:
   - The conditional probability of purchasing KP281 is 28%
   - The conditional probability of purchasing KP481 is 24%
   - The conditional probability of purchasing KP781 is 6%
3. Customers with High Income (60k - 80k) exhibit a probability of 13% of purchasing a treadmill. Within this income bracket:

- The conditional probability of purchasing KP281 is 3%
- The conditional probability of purchasing KP481 is 4%
- The conditional probability of purchasing KP781 is 6%

4. Customers with Very High Income (>80k) have an 11% probability of purchasing a treadmill. Within this income category:

   - The conditional probability of purchasing KP281 is 0%
   - The conditional probability of purchasing KP481 is 0%
   - The conditional probability of purchasing KP781 is 11%

In [ ]:
```
# Probability of product purchase w.r.t miles
pd.crosstab(index = df['Product'], columns = df['miles_group'], margins = True, normalize = True).round(2)
```

Out[ ]:

miles_group	Light Activity	Moderate Activity	Active Lifestyle	Fitness Enthusiast	All
Product					
KP281	0.07	0.28	0.10	0.00	0.44
KP481	0.03	0.22	0.08	0.01	0.33
KP781	0.00	0.04	0.15	0.03	0.22
All	0.09	0.54	0.33	0.03	1.00

Insights:

1. For customers with a Light Activity lifestyle (0 to 50 miles/week), the probability of purchasing a treadmill is 9%. Among these customers:

   - The conditional probability of purchasing KP281 is 7%.
   - The conditional probability of purchasing KP481 is 3%.
   - The conditional probability of purchasing KP781 is 0%.

2. Customers with a Moderate Activity lifestyle (51 to 100 miles/week) have a 54% probability of purchasing a treadmill. Within this group:

   - The conditional probability of purchasing KP281 is 28%.
   - The conditional probability of purchasing KP481 is 22%.
   - The conditional probability of purchasing KP781 is 4%.

3. For customers with an Active Lifestyle (100 to 200 miles/week), the probability of purchasing a treadmill is 33%. Among these customers:
    - The conditional probability of purchasing KP281 is 10%.
    - The conditional probability of purchasing KP481 is 8%.
    - The conditional probability of purchasing KP781 is 15%.

In [ ]:
```
# Probability of product purchase w.r.t maritalstatus
pd.crosstab(index = df['Product'], columns = df['MaritalStatus'], margins = True, normalize = True).round(2)
```

Out[ ]:

MaritalStatus	Partnered	Single	All
Product			
KP281	0.27	0.18	0.44
KP481	0.20	0.13	0.33
KP781	0.13	0.09	0.22
All	0.59	0.41	1.00

Insights:

1. Married customers are more likely to purchase a treadmill, with a probability of 59%. When considering married customers:
    - The probability of purchasing KP281 is 27%
    - The probability of purchasing KP481 is 20%
    - The probability of purchasing KP781 is 13%.
2. Unmarried customers have a probability of 41% of purchasing a treadmill. When considering unmarried customers:
    - The probability of purchasing KP281 is 18%
    - The probability of purchasing KP481 is 13%
    - The probability of purchasing KP781 is 9%.

In [ ]:
```
# Probability of product purchase w.r.t usage
pd.crosstab(index = df['Product'], columns = df['Usage'], margins = True, normalize = True).round(2)
```

Out[ ]:

Usage	2	3	4	5	6	7	All
Product							
KP281	0.11	0.21	0.12	0.01	0.00	0.00	0.44
KP481	0.08	0.17	0.07	0.02	0.00	0.00	0.33
KP781	0.00	0.01	0.10	0.07	0.04	0.01	0.22
All	0.18	0.38	0.29	0.09	0.04	0.01	1.00

Insights:

1. For customers with a usage of 3 times per week, the probability of purchasing a treadmill is 38%. The conditional probabilities of purchasing each treadmill model given this usage frequency are:
   - KP281: 21%
   - KP481: 17%
   - KP781: 1%
2. When customers use the treadmill 4 times per week, the probability of a purchase is 29%. The conditional probabilities for each treadmill model under this usage frequency are:
   - KP281: 12%
   - KP481: 7%
   - KP781: 10%
3. Customers using the treadmill 2 times per week have a purchasing probability of 18%. The conditional probabilities for each treadmill model given this usage frequency are:
   - KP281: 11%
   - KP481: 8%
   - KP781: 0%

In [ ]:

```
# Probability of product purchase w.r.t fitness
pd.crosstab(index = df['Product'], columns = df['Fitness'], margins = True, normalize = True).round(2)
```

Out[ ]:

Fitness	1	2	3	4	5	All
Product						
KP281	0.01	0.08	0.30	0.05	0.01	0.44
KP481	0.01	0.07	0.22	0.04	0.00	0.33
KP781	0.00	0.00	0.02	0.04	0.16	0.22
All	0.01	0.14	0.54	0.13	0.17	1.00

Insights

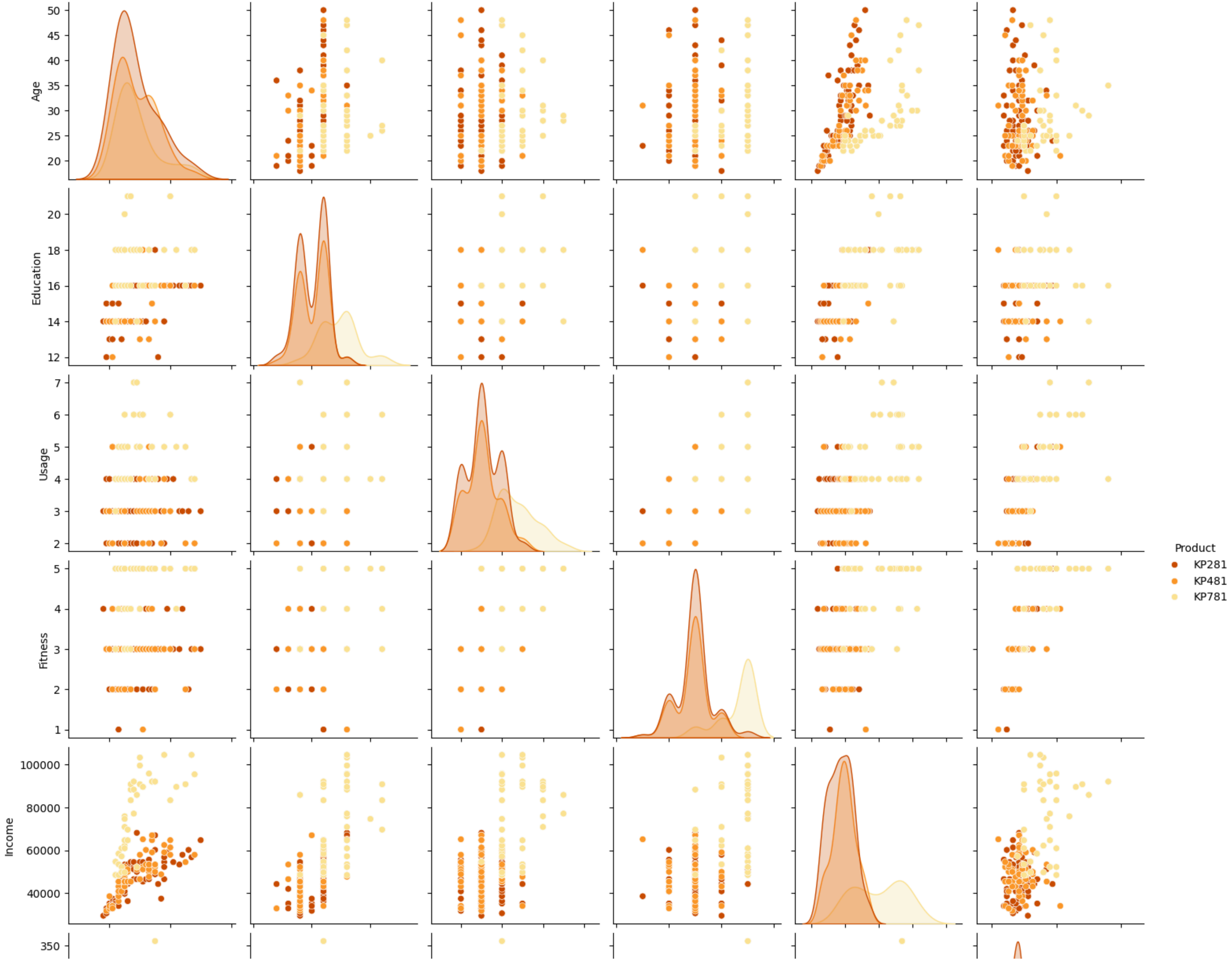
1. The probability of a treadmill being purchased by a customer with average (3) fitness is 54%. Additionally, the conditional probability of purchasing each treadmill model given that the customer has average fitness is as follows:
   - KP281: 30%
   - KP481: 22%
   - KP781: 2%
2. The probability of a treadmill being purchased by a customer with fitness levels of 2, 4, or 5 is approximately 15%.
3. The probability of a treadmill being purchased by a customer with very low (1) fitness is only 1%.

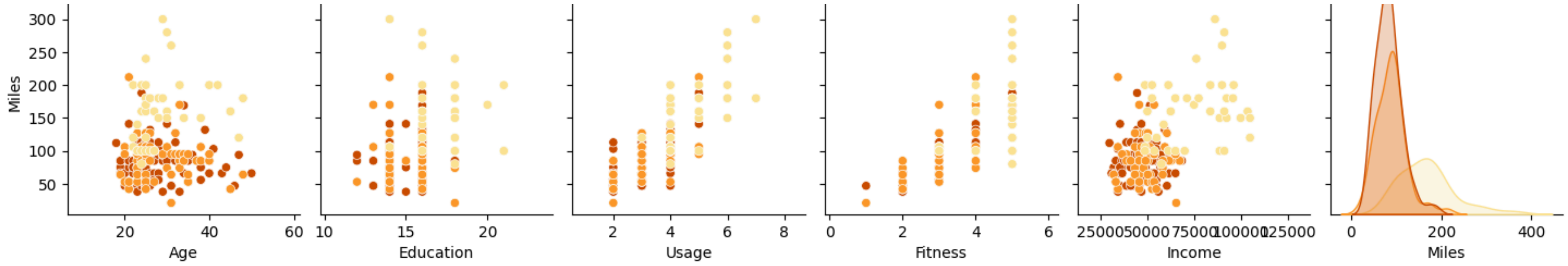
5. Checking the correlation among different factors

5.1: Pairplot

In [ ]:

```
sns.pairplot(df, hue='Product', palette='YlOrBr_r')
plt.show()
```


Age
Education
Usage
Fitness
Income
Product
KP281
KP481
KP781



Insights:

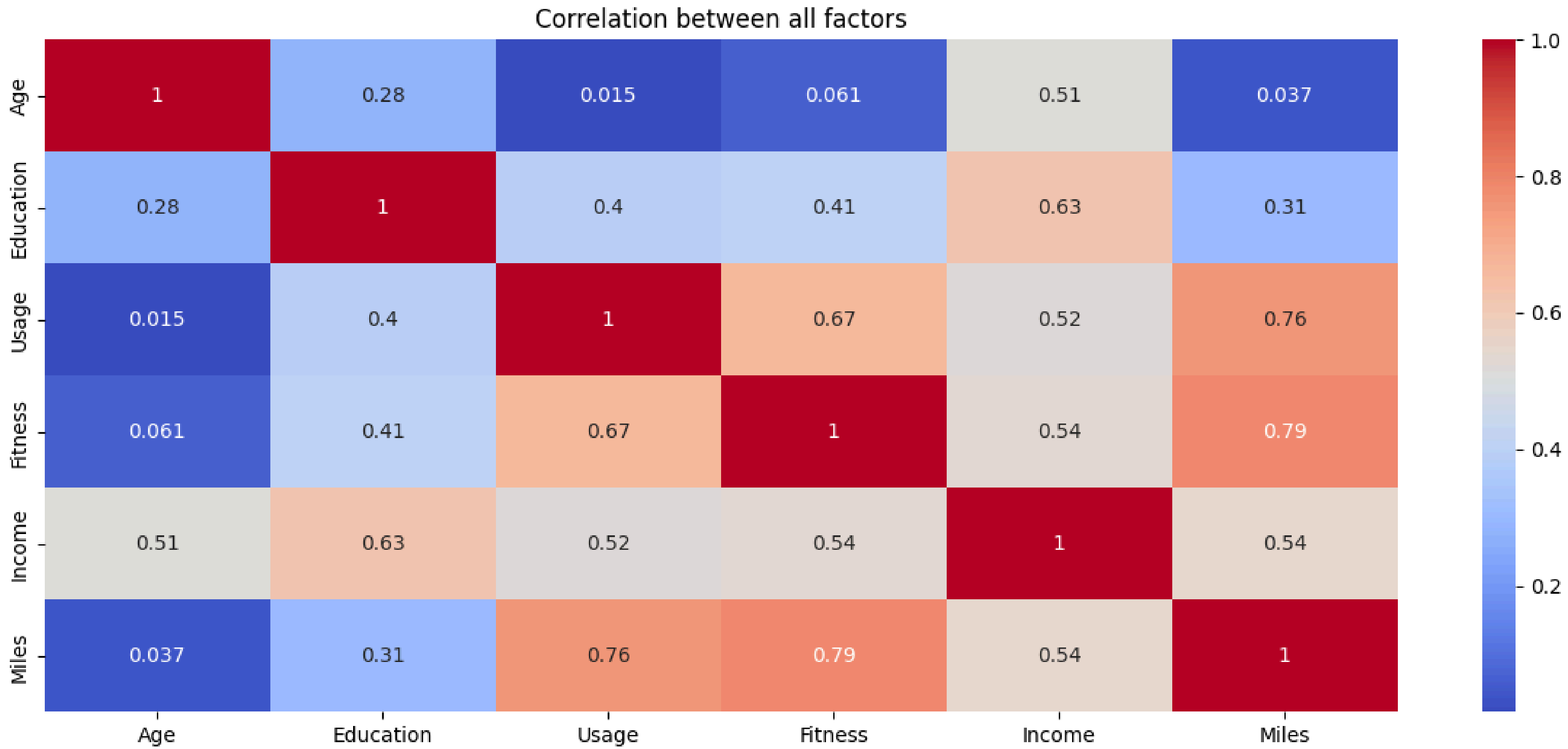
Upon examining the pair plot, we observe a positive correlation between Age and Income, consistent with the heatmap's depiction of a strong correlation between these variables.

Additionally, there's a noticeable correlation between Education and Income, as anticipated. Education also exhibits a significant correlation with both Fitness rating and Usage of the treadmill.

Moreover, Usage displays a strong correlation with Fitness and Miles, indicating that higher treadmill usage aligns with increased fitness levels and mileage covered.

5.2: Heatmap

In [ ]:
```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
import pandas as pd

plt.figure(figsize=(15, 6))
sns.heatmap(df.corr(), cmap='coolwarm', annot=True)
plt.title('Correlation between all factors')
plt.show()
```



Insights:

1. **Usage and Fitness Connection**: There is a strong positive correlation between usage and fitness level (0.76 and 0.67, respectively). This indicates that individuals who use fitness equipment more frequently tend to have higher fitness levels. Consistent exercise is crucial for maintaining fitness levels.

2. **Income Influence**: Income is significantly associated with education (0.63) and miles covered (0.54). Higher-income customers may have pursued more education and might prefer treadmills with longer mileage. This suggests that socioeconomic factors can impact consumer preferences and behaviors.

3. **Limited Influence of Age**: Age shows relatively weak correlations with other variables, suggesting that age alone may not strongly influence factors like income, fitness, or usage patterns. While age can play a role in determining fitness and usage, its influence appears less significant compared to other variables.

4. **Education's Role**: Education correlates positively with income (0.63) and, to a lesser extent, with fitness and usage (0.41 and 0.4, respectively). This implies that individuals with higher education levels may earn more and engage more in fitness activities. The positive correlation underscores the potential impact of education on socioeconomic status and health-related behaviors.

These insights provide valuable understanding of the relationships between income, education, age, fitness levels, and treadmill usage patterns, shedding light on the complex dynamics influencing consumer behavior in the fitness equipment market.

6. Customer Profiling:

6.1.1 Overview:

- Probability of purchasing KP281: 44%
- Probability of purchasing KP481: 33%
- Probability of purchasing KP781: 22%

6.1.2 Customer Profile for KP281 Treadmill:

- Age: Predominantly 18 to 35 years, with some aged 35 to 50
- Education: 13 years and above
- Income: Below USD 60,000 annually
- Usage: 2 to 4 times weekly
- Fitness: Scale of 2 to 4
- Miles: 50 to 100 miles per week

6.1.3 Customer Profile for KP481 Treadmill:

- Age: Mainly 18 to 35 years, with some aged 35 to 50
- Education: 13 years and above
- Income: Between USD 40,000 to USD 80,000 annually
- Usage: 2 to 4 times weekly

- Fitness: Scale of 2 to 4
- Miles: 50 to 200 miles per week

6.1.4 Customer Profile for KP781 Treadmill:

- Gender: Male
- Age: Primarily 18 to 35 years
- Education: 15 years and above
- Income: USD 80,000 and above annually
- Usage: 4 to 7 times weekly
- Fitness: Scale of 3 to 5
- Miles: 100 miles and above per week

6.2. Recommendations

6.2.1 Targeted Marketing:

Utilize demographic insights to tailor marketing strategies effectively:

- **KP281:** Target females and lower-income customers with campaigns emphasizing affordability and moderate exercise suitability.
- **KP781:** Highlight advanced features for higher-income and male customers through premium advertising channels.

6.2.2 Product Development:

Leverage insights to enhance product features:

- **KP281:** Consider enhancing features for wider appeal.
- **KP781:** Explore customization options to cater to higher-income customers' needs.

6.2.3 Pricing Strategies:

Optimize pricing to align with customer income levels:

- **Tiered Pricing:** Introduce entry-level pricing for KP281, mid-range pricing for KP481, and premium pricing for KP781.
- **Bundle Deals:** Offer package deals to add value and justify higher price points.

6.2.4 Education and Engagement:

Engage customers through educational content:

- **Webinars:** Host online sessions focusing on fitness topics tailored to different education levels.
- **Product Demonstrations:** Showcase how treadmill models support various fitness goals.

6.2.5 Inventory Management:

Optimize inventory based on product popularity:

- **Demand Analysis:** Ensure adequate stock levels for each product based on demographic preferences and sales data.

By implementing these recommendations, Aerofit can effectively target diverse customer segments, enhance product appeal, optimize pricing strategies, engage customers through educational content, and manage inventory efficiently to drive sales and brand loyalty.

In [63]:
```
! jupyter nbconvert --to html /content/Aerofit_Case_study.ipynb
```

```
[NbConvertApp] Converting notebook /content/Aerofit_Case_study.ipynb to html
[NbConvertApp] Writing 1917488 bytes to /content/Aerofit_Case_study.html
```