Aerofit Business Case Study on Treadmills

In [146…

```
# Importing the necessary libraries

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

In [147…

```
# Loading the dataset

aerofit = pd.read_csv("aerofit_treadmill.csv")
```

In [148…

```
# Top 5 rows of the dataset

aerofit.head()
```

Out[148]:

	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 [149…

```
# Last 5 rows of the dataset

aerofit.tail()
```

Out[149]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

In [150…

```
# No of rows & columns

aerofit.shape
```

Out[150]:

```
(180, 9)
```

In [151…

```
# Check for Dimension

aerofit.ndim
```

Out[151]:

```
2
```

In [152…

```
aerofit.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
```

In [153…

```
# Checking of null values

aerofit.isna().sum()
```

Out[153]:

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

In [154…

```
# Duplicate values check

aerofit.duplicated().sum()
```

Out[154]:

```
0
```

In [155…

```
# Uniques values of each columns

aerofit.nunique()
```

Out[155]:

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

In [156…

```
# making an copy of the dataset

df = aerofit.copy()
```

Descriptive Analysis

In [157…

```
df.describe()
```

Out[157]:

	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

In [158…

```
# Object columns

df.describe(include = "object").T
```

Out[158]:

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

In [159…

```
# Product names

df['Product'].unique()
```

Out[159]:

```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

In [160…

```
# Product prices in dollars

data = {
    'Treadmill Model': ['KP281', 'KP481', 'KP781'],
    'User Level': ['Entry-Level', 'Mid-Level', 'Advanced'],
    'Price': [1500, 1750, 2500]
}
treadmill_df = pd.DataFrame(data)
treadmill_df
```

Out[160]:

	Treadmill Model	User Level	Price
0	KP281	Entry-Level	1500
1	KP481	Mid-Level	1750
2	KP781	Advanced	2500

In [161…

```
# Number of customers for each product types

product_counts = df['Product'].value_counts().sort_index()
product_counts
```

Out[161]:

```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

Outliers & count of outliers in each columns

In [162…

```
def outliers(df, column_name):

    # Calculate quartiles and IQR for the specified column
    Q1 = np.percentile(df[column_name], 25)
    Q3 = np.percentile(df[column_name], 75)
    IQR = Q3 - Q1

    # upper bounds for outliers
    upper_band = Q3 + 1.5 * IQR

    # outliers in the specified column
    outliers_df = df[df[column_name] > upper_band]

    return outliers_df
```

In [163…

```
# outliers in Age column

outliers(df,'Age')
```

Out[163]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
78	KP281	47	Male	16	Partnered	4	3	56850	94
79	KP281	50	Female	16	Partnered	3	3	64809	66
139	KP481	48	Male	16	Partnered	2	3	57987	64
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

In [164…

```
# Count of outliers in Age column

len(outliers(df,'Age'))
```

Out[164]:

```
5
```

In [165…

```
# outliers in Education column

outliers(df,'Education')
```

Out[165]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
156	KP781	25	Male	20	Partnered	4	5	74701	170
157	KP781	26	Female	21	Single	4	3	69721	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
175	KP781	40	Male	21	Single	6	5	83416	200

In [166…

```
# Count of outliers in Education column

len(outliers(df,'Education'))
```

Out[166]:

```
4
```

In [167…

```
# outliers in Usage column

outliers(df,'Usage')
```

Out[167]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
154	KP781	25	Male	18	Partnered	6	4	70966	180
155	KP781	25	Male	18	Partnered	6	5	75946	240
162	KP781	28	Female	18	Partnered	6	5	92131	180
163	KP781	28	Male	18	Partnered	7	5	77191	180
164	KP781	28	Male	18	Single	6	5	88396	150
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
170	KP781	31	Male	16	Partnered	6	5	89641	260
175	KP781	40	Male	21	Single	6	5	83416	200

In [168…

```
# Count of outliers in Usage column

len(outliers(df,'Usage'))
```

Out[168]: 9

In [169…

```
# outliers in Income column

outliers(df,'Income')
```

Out[169]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
159	KP781	27	Male	16	Partnered	4	5	83416	160
160	KP781	27	Male	18	Single	4	3	88396	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
162	KP781	28	Female	18	Partnered	6	5	92131	180
164	KP781	28	Male	18	Single	6	5	88396	150
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
168	KP781	30	Male	18	Partnered	5	4	103336	160
169	KP781	30	Male	18	Partnered	5	5	99601	150
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
172	KP781	34	Male	16	Single	5	5	92131	150
173	KP781	35	Male	16	Partnered	4	5	92131	360
174	KP781	38	Male	18	Partnered	5	5	104581	150
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

In [170…

```
# Count of outliers in Income column

len(outliers(df,'Income'))
```

Out[170]: 19

In [171…

```
# outliers in Miles column

outliers(df,'Miles')
```

Out[171]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
23	KP281	24	Female	16	Partnered	5	5	44343	188
84	KP481	21	Female	14	Partnered	5	4	34110	212
142	KP781	22	Male	18	Single	4	5	48556	200
148	KP781	24	Female	16	Single	5	5	52291	200
152	KP781	25	Female	18	Partnered	5	5	61006	200
155	KP781	25	Male	18	Partnered	6	5	75946	240
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
173	KP781	35	Male	16	Partnered	4	5	92131	360
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200

In [172…

```
# Count of outliers in Miles column

len(outliers(df,'Miles'))
```

Out[172]:

```
13
```

In [173…

```
# outliers in Fitness column

Q1_Fitness = np.percentile(df['Fitness'],25)
Q3_Fitness = np.percentile(df['Fitness'],75)
IQR = Q3_Fitness - Q1_Fitness
lower_band = Q1_Fitness - 1.5*(IQR)

Fitness_outliers = df[df['Fitness'] < lower_band]

Fitness_outliers
```

Out[173]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
14	KP281	23	Male	16	Partnered	3	1	38658	47
117	KP481	31	Female	18	Single	2	1	65220	21

In [174…

```
# Count of outliers in Fitness column

len(Fitness_outliers)
```

Out[174]:

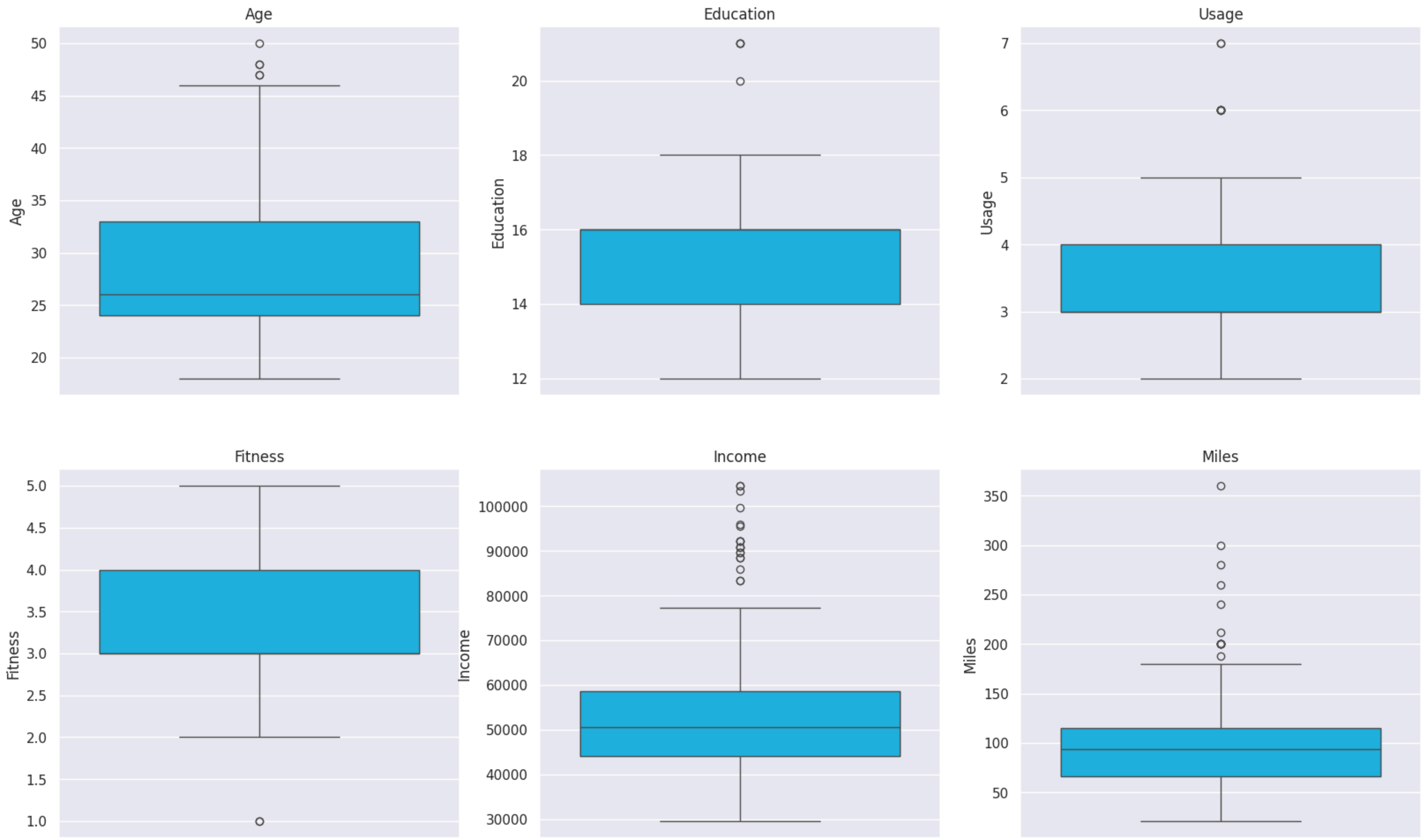
```
2
```

In [175…

```
# combining all the int columns into a list

columns = ["Age", "Education", "Usage", "Fitness", "Income", "Miles"]
```

In [176…

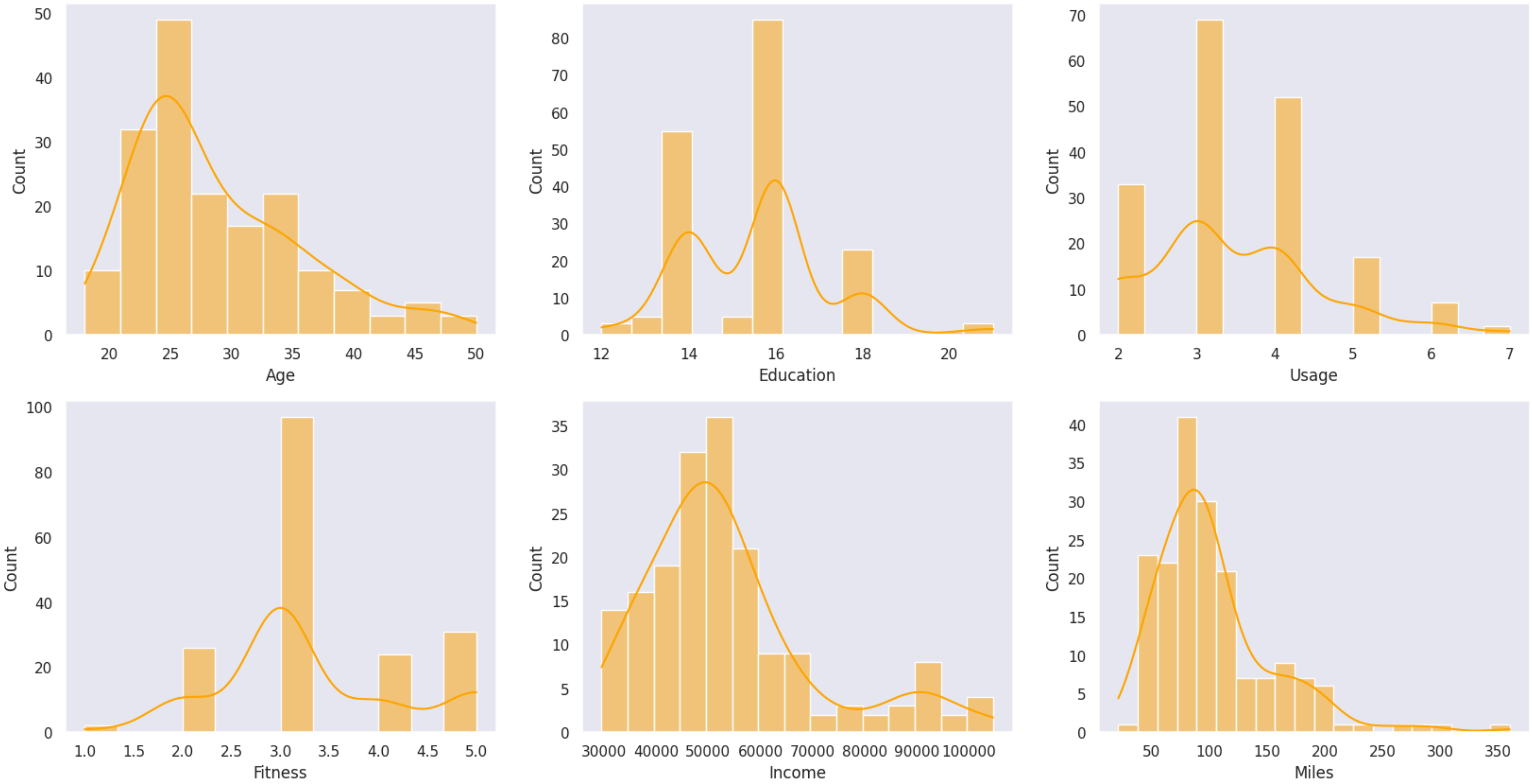
```
# Plotting the outliers using boxplots

fig, axes = plt.subplots(2, 3, figsize=(20, 12))
sns.set(style="dark")

for i in range(2):
    for j in range(3):
        variable = columns[i * 3 + j]
        sns.boxplot(ax=axes[i, j], data=df, y=variable, color="deepskyblue")
        axes[i, j].set_title(variable)

plt.show();
```



Univariate Analysis

In [177…

```
# Hisplot for the above mentioned columns

fig, axes = plt.subplots(2, 3, figsize=(20, 10))
sns.set(style="darkgrid")

# Iterate through the rows and columns of the subplot grid
for i in range(2):
    for j in range(3):
        variable = columns[i * 3 + j] # Variable Assignment for Subplot Analysis
        sns.histplot(ax=axes[i, j], data=df, x=variable, kde=True, color="Orange")
        axes[i, j]

plt.show();
```



Insights :

Age: The majority of individuals fall between age range of 24 to 33, indicating a concentration of users in the young to early-mid adulthood demographic.

Education: Most individuals have an education level between 14 and 16 years, with outliers having exceptionally high education levels of 20 and 21 years.

Usage: The average treadmill usage is 3 to 4 times per week, suggesting a moderate and consistent engagement in treadmill activities among users.

Fitness: The majority of users rate their fitness between 3 and 4, reflecting a moderate to good fitness level among the sample.

Income: The median income falls between 50,596.50, indicating a middle-income range among the sample.

Miles: Most individuals aim to walk between 66 and 114.75 miles weekly, with some exceeding 175 miles.

Bivariate Analysis

In [178…

```
# cross tab function

cross_tab = lambda x, y, df: pd.crosstab(df[x], df[y])
```

In [179…

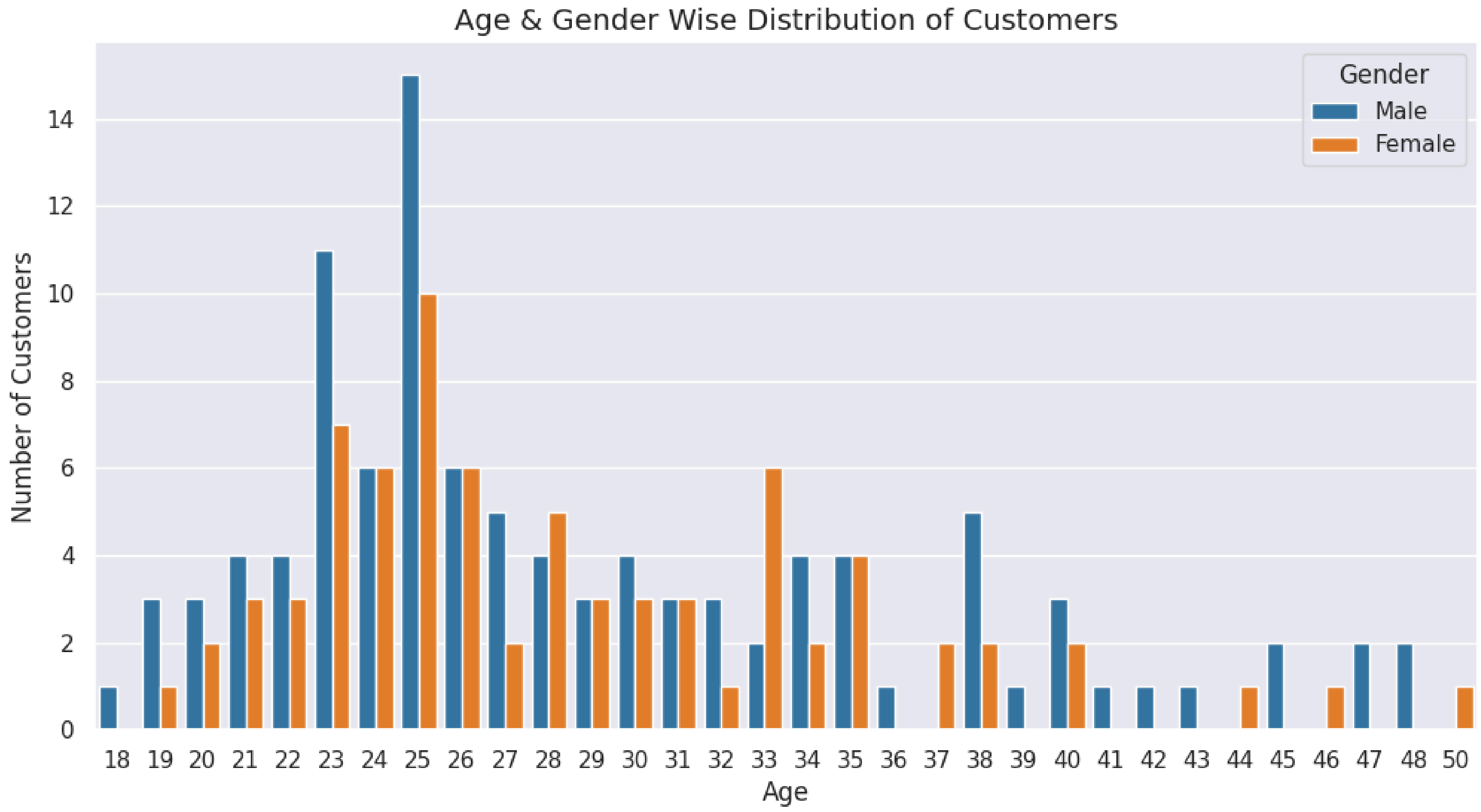
```
# Age & gender wise distribution of customers

cross_tab('Age', 'Gender', df)
```

Out[179]:

Gender	Female	Male
Age		
18	0	1
19	1	3
20	2	3
21	3	4
22	3	4
23	7	11
24	6	6
25	10	15
26	6	6
27	2	5
28	5	4
29	3	3
30	3	4
31	3	3
32	1	3
33	6	2
34	2	4
35	4	4
36	0	1
37	2	0
38	2	5
39	0	1
40	2	3
41	0	1
42	0	1
43	0	1
44	1	0
45	0	2
46	1	0
47	0	2
48	0	2
50	1	0

In [180…

```
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Age', hue='Gender', palette='tab10')
plt.title("Age & Gender Wise Distribution of Customers", fontsize=14)
plt.xlabel('Age')
plt.ylabel('Number of Customers')
plt.show()
```



Insights:

The distribution of age among males and females indicates a diverse representation, with individuals ranging from 18 to 50 years old.

In the age group between 23 and 25, there is a noticeable increase in the number of females compared to males.

In [181…

```
(pd.crosstab(df['Product'],df['Gender'],margins = True, normalize = True)*100).T.round(2)
```

Out[181]:

Product	KP281	KP481	KP781	All
Gender				
Female	22.22	16.11	3.89	42.22
Male	22.22	17.22	18.33	57.78
All	44.44	33.33	22.22	100.00

In [182…

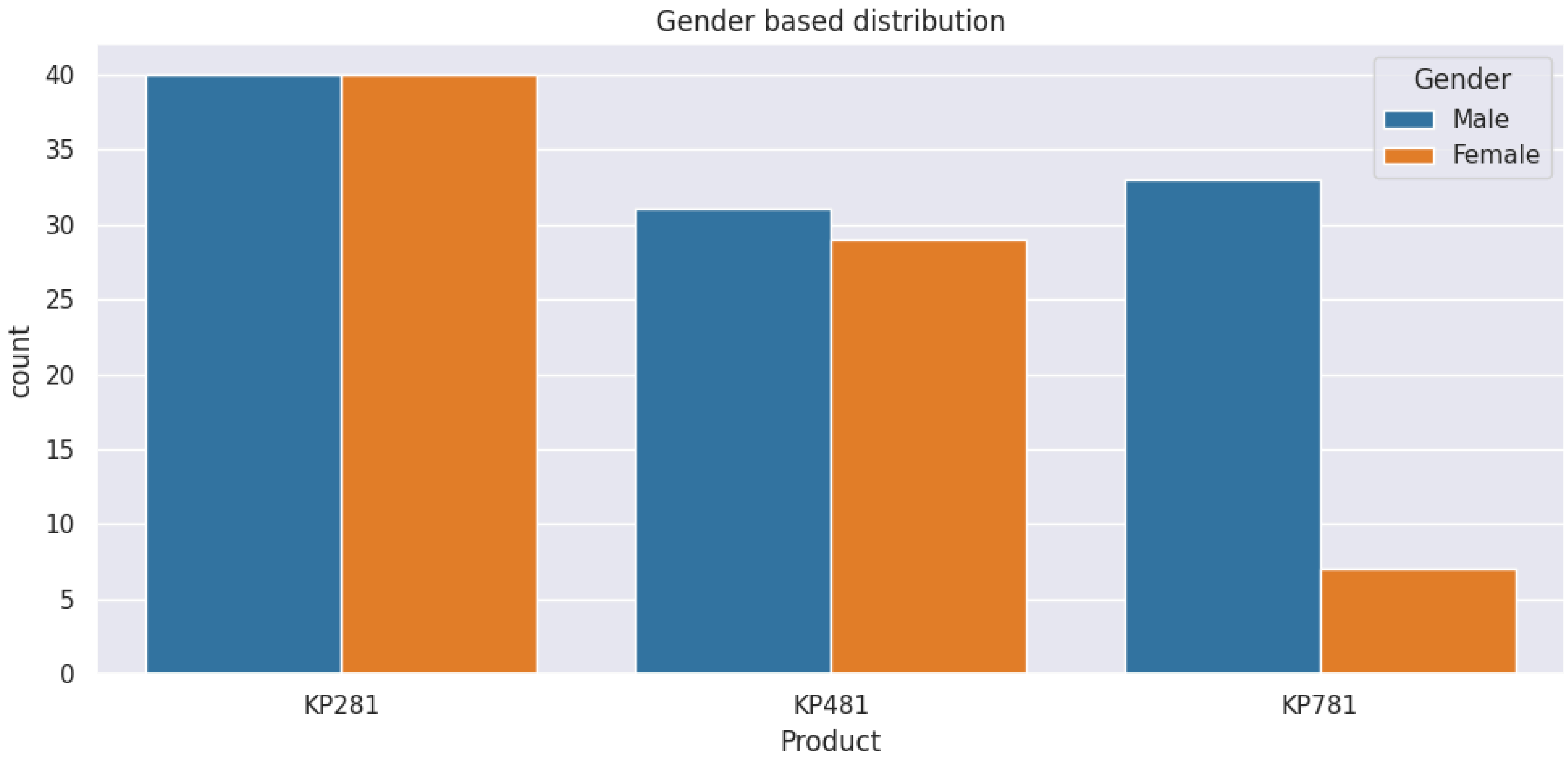
```
# gender wise distribution of each products

cross_tab('Product','Gender',df).T
```

Out[182]:

Product	KP281	KP481	KP781
Gender			
Female	40	29	7
Male	40	31	33

In [183…

```
# Gender Plot of each products
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Product', hue='Gender', palette='tab10')
plt.title("Gender based distribution")
plt.tight_layout()
```



In [184…

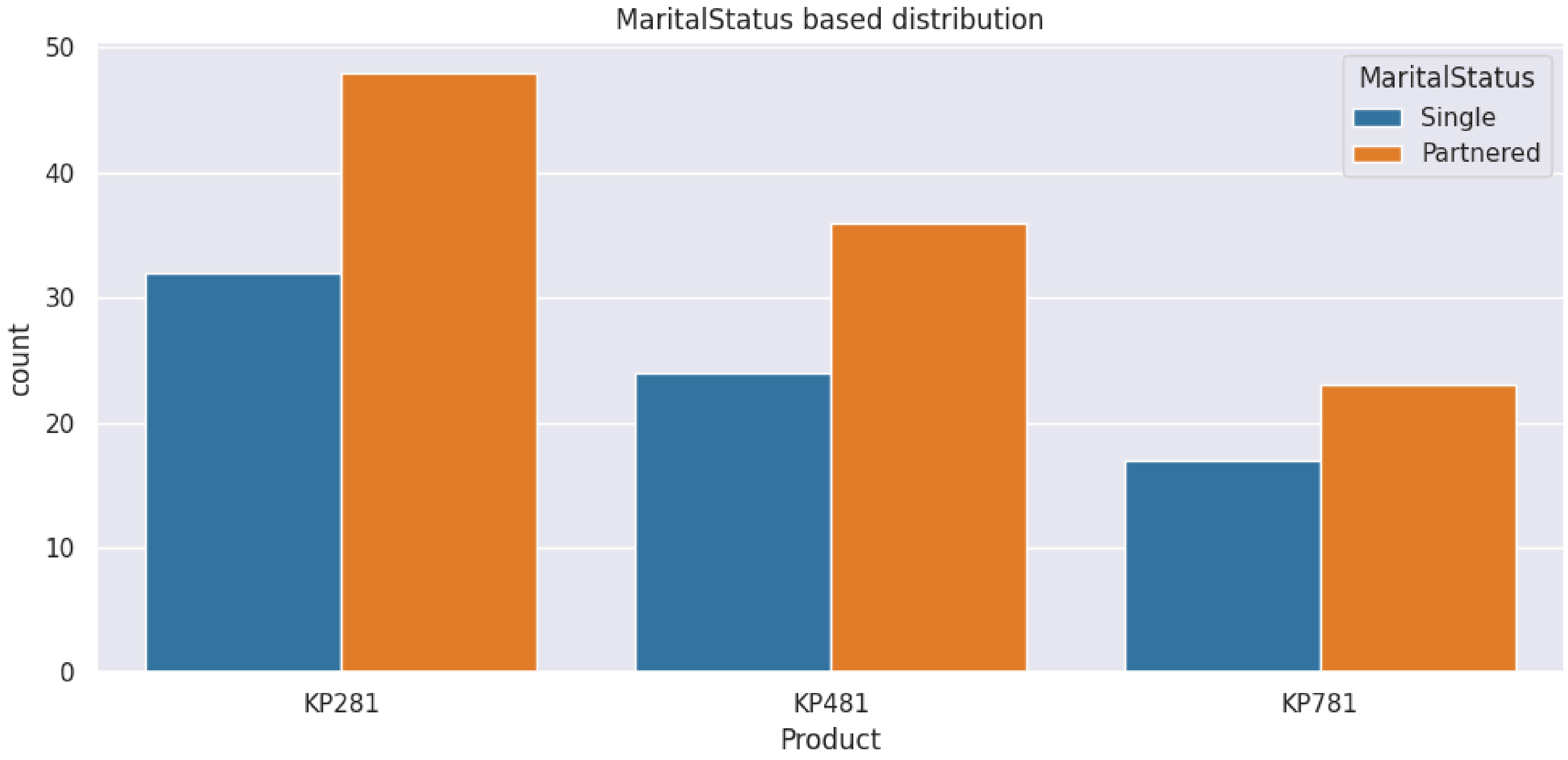
```
# MartialStatus wise distribution of each products

cross_tab('Product','MaritalStatus',df).T
```

Out[184]:

Product MaritalStatus	KP281	KP481	KP781
Partnered	48	36	23
Single	32	24	17

In [185…

```
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Product', hue='MaritalStatus', palette='tab10')
plt.title("MaritalStatus based distribution")
plt.tight_layout()
```



Insights :

- The distribution of product preferences among genders reveals that both females and males show a preference for KP281 and KP481.
- Notably, the KP781 treadmill is more popular among males, with a substantial count, while females show a lower preference for this particular product.
- For both partnered and single individuals, KP281 is the most preferred treadmill model, followed by KP481 and KP781.
- Partnered individuals show a higher overall preference for all three models compared to singles.

In [186…

```
# creating bins for age

bins = [14,20,30,40,60]
labels =["Teens","Young Adults","Adults","Over 40s"]
df['AgeCategory'] = pd.cut(df['Age'], bins,labels=labels)
```

In [187…

```
# creating bins for Income

bins_income = [29000, 35000, 60000, 85000, 105000]
labels_income = ['Low Income','Middle-class','Upper-Middle class', 'Wealthy']
df['IncomeSlab'] = pd.cut(df['Income'],bins_income,labels = labels_income)
```

In [188…

```
df.head(3)
```

Out[188]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeCategory	IncomeSlab
0	KP281	18	Male	14	Single	3	4	29562	112	Teens	Low Income
1	KP281	19	Male	15	Single	2	3	31836	75	Teens	Low Income
2	KP281	19	Female	14	Partnered	4	3	30699	66	Teens	Low Income

In [189…

```
# AgeCategory wise distribution of each products

cross_tab('Product','AgeCategory',df).T
```

Out[189]:

Product	KP281	KP481	KP781
AgeCategory			
Teens	6	4	0
Young Adults	49	31	30
Adults	19	23	6
Over 40s	6	2	4

In [190…

```
# IncomeSlab wise distribution of each products

cross_tab('Product','IncomeSlab',df).T
```

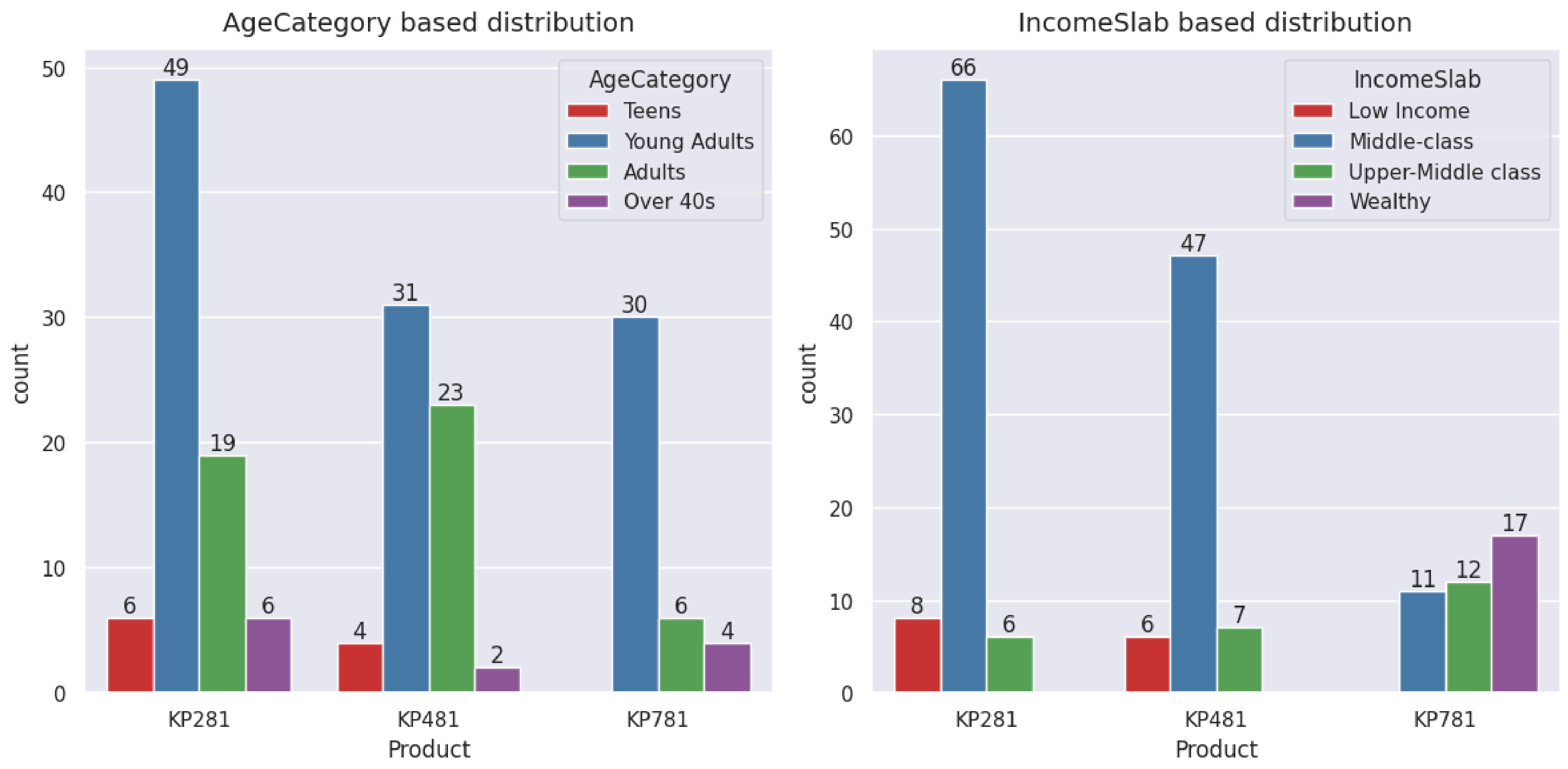
Out[190]:

Product	KP281	KP481	KP781
IncomeSlab			
Low Income	8	6	0
Middle-class	66	47	11
Upper-Middle class	6	7	12
Wealthy	0	0	17

In [190…

In [191…

```
fig, axs = plt.subplots(ncols=2, figsize=(12,6))

# AgeCategory Plot
label1 = sns.countplot(data=df, x='Product', hue='AgeCategory', palette='Set1', ax=axs[0])
for i in label1.containers:
    label1.bar_label(i)
# IncomeSlab Plot
label2 = sns.countplot(data=df, x='Product', hue='IncomeSlab', palette='Set1', ax=axs[1])
for i in label2.containers:
    label2.bar_label(i)

# titles for subplots
axs[0].set_title("AgeCategory based distribution", pad=10, fontsize=14)
axs[1].set_title("IncomeSlab based distribution", pad=10, fontsize=14)
plt.tight_layout()
plt.show()
```



Insights:

- Young Adults (between the ages of 20 and 40) preference for all three treadmill models, with KP281 being the most popular.
- Teens show minimal interest in these treadmill models, while Adults and individuals Over 40 also demonstrate interest, albeit with lower counts.
- Middle-class individuals overwhelmingly prefer KP281, followed by KP481 and KP781.
- The Upper-Middle class and Wealthy categories show a distinct preference for KP781, suggesting its appeal to individuals with higher disposable income.

In [192…

```
# Week usage wise distribution of each products

cross_tab('Product','Usage',df).T
```

Out[192]:

Product	KP281	KP481	KP781
Usage			
2	19	14	0
3	37	31	1
4	22	12	18
5	2	3	12
6	0	0	7
7	0	0	2

In [193…

```
# Miles wise distribution of each products

cross_tab('Product','Fitness',df).T
```

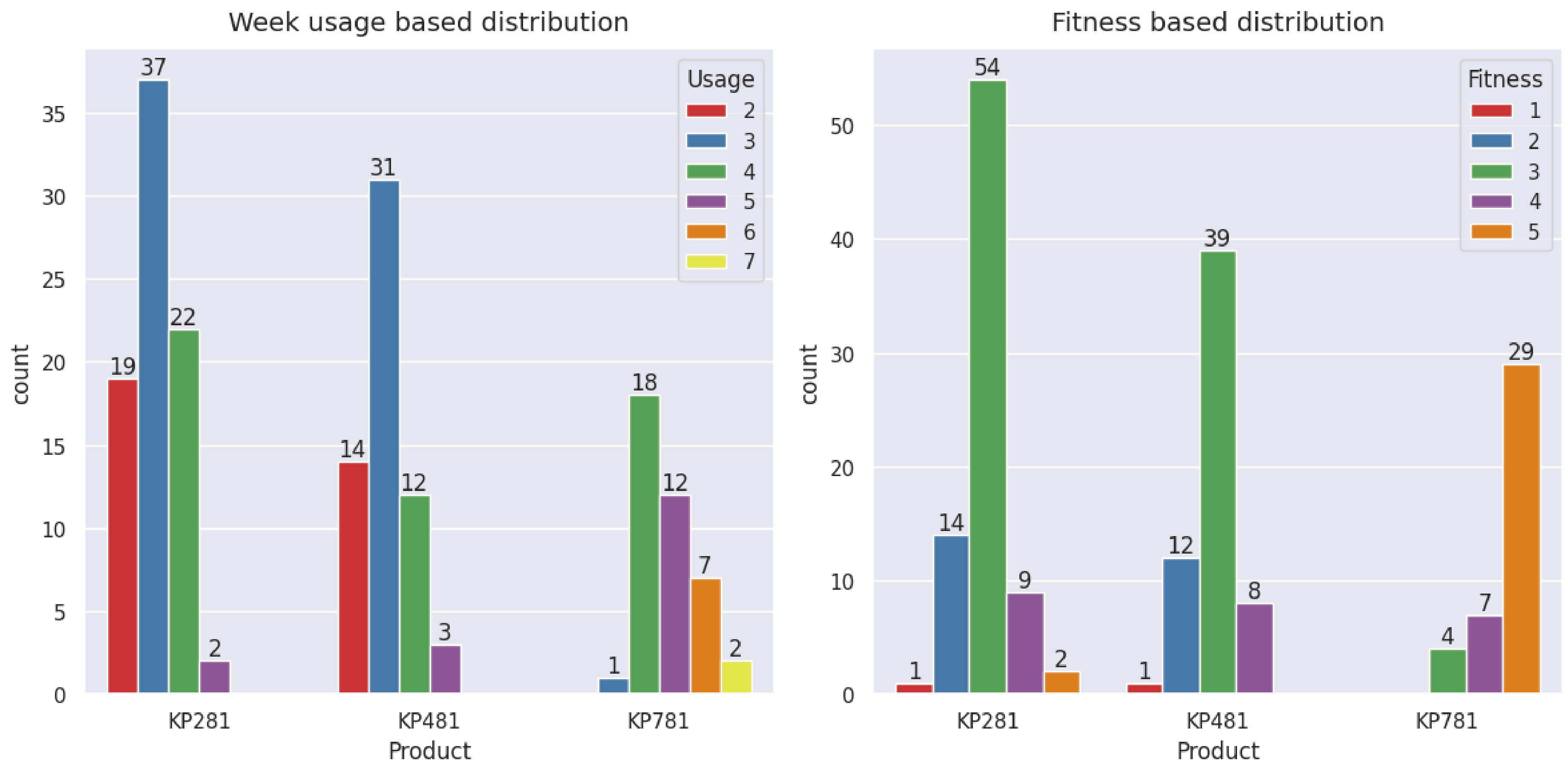
Out[193]:

Product	KP281	KP481	KP781
Fitness			
1	1	1	0
2	14	12	0
3	54	39	4
4	9	8	7
5	2	0	29

In [194…

```
# size of the figure and subplots
fig, axs = plt.subplots(ncols=2, figsize=(12,6))

# AgeCategory Plot
AgeCategory = sns.countplot(data=df, x='Product', hue='Usage', palette='Set1', ax=axs[0])
for i in AgeCategory.containers:
    AgeCategory.bar_label(i)

# IncomeSlab Plot
IncomeSlab = sns.countplot(data=df, x='Product', hue='Fitness', palette='Set1', ax=axs[1])
for i in IncomeSlab.containers:
    IncomeSlab.bar_label(i)

# titles for subplots
```

```
axs[0].set_title("Week usage based distribution", pad=10, fontsize=14)
axs[1].set_title("Fitness based distribution", pad=10, fontsize=14)
plt.tight_layout()
plt.show()
```



Insights:

- The majority of users across all three treadmill models have a usage frequency of 3 times per week, with KP281 having the highest count.
- KP781 has usage frequency of 4, 5, 6 & 7 times per week, shows that this is preferred by customers with a higher frequency of treadmill usage.
- Fitness ratings reveal that the majority of users across all treadmill models fall within the range of 2 to 4, with a higher count in the Fitness 3 category.
- KP781 attracts users with the highest fitness ratings (5), indicating its popularity among individuals who prioritize and maintain a high level of fitness.

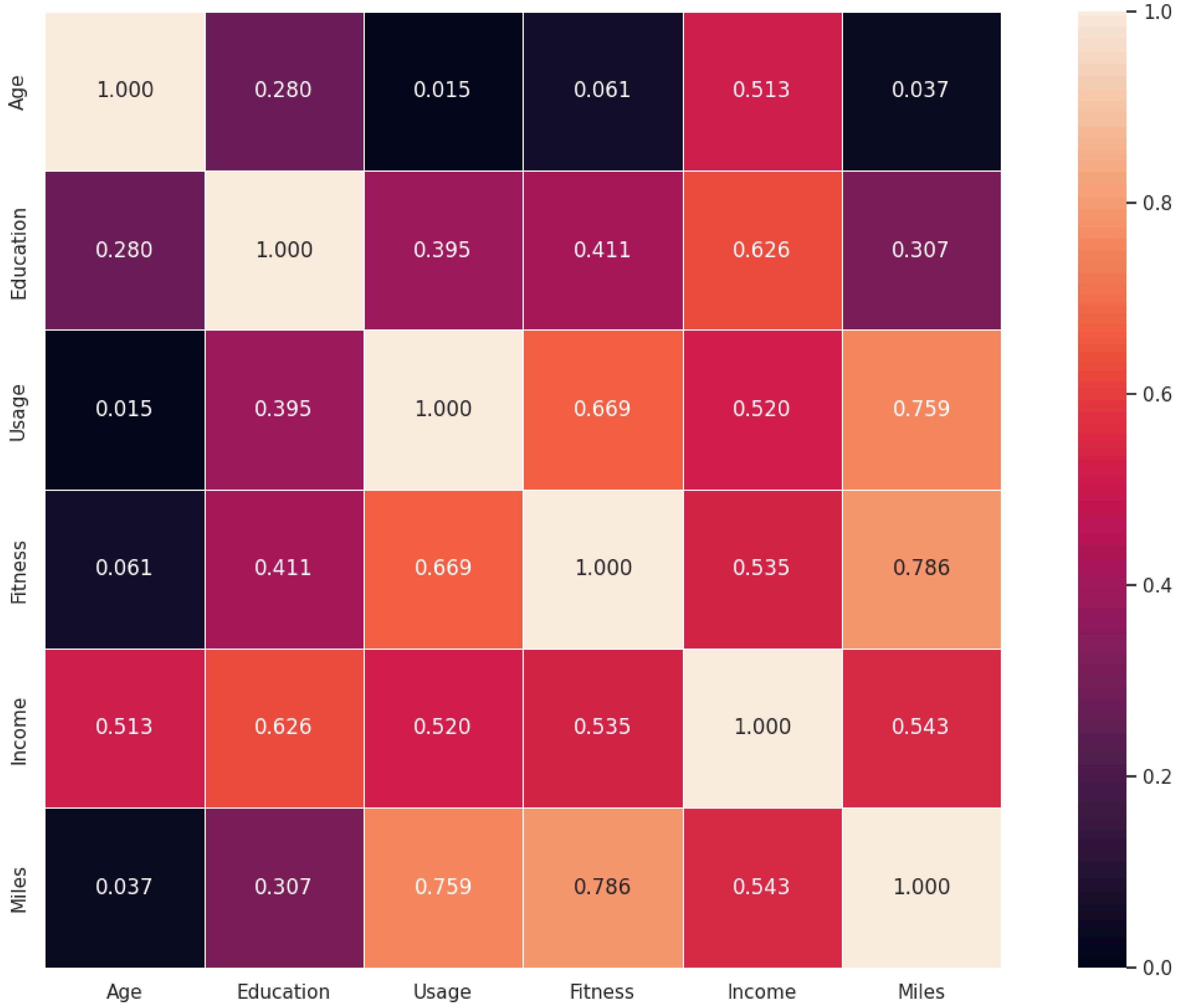
Correlation analysis among different factors

In [195…

```
# Exclude non-numeric columns

numeric_df = df.select_dtypes(include=['int64'])
```

In [196…

```
# Setting up the matplotlib figure
plt.figure(figsize=(16, 10))

ax = sns.heatmap(
    numeric_df.corr(),
    annot=True,
    fmt='.3f',
    linewidths=0.5,
    cmap='rocket',
    vmin=-0, vmax=1,
    square=True
)
```



Insights:

Age:

Age is positively correlated with Education and Income, indicating that older individuals tend to have higher levels of education and income.

Education:

Education is positively correlated with Income, Usage, and Fitness, suggesting that individuals with higher education levels tend to have higher incomes, use the treadmill more frequently, and maintain higher fitness levels.

Usage:

Usage has strong positive correlations with Fitness and Miles walked per week, indicating that more frequent treadmill usage is associated with higher fitness levels and covering a greater distance.

Fitness:

Fitness has a strong positive correlation with Miles walked per week, highlighting that individuals with higher fitness levels tend to walk more miles per week.

Income:

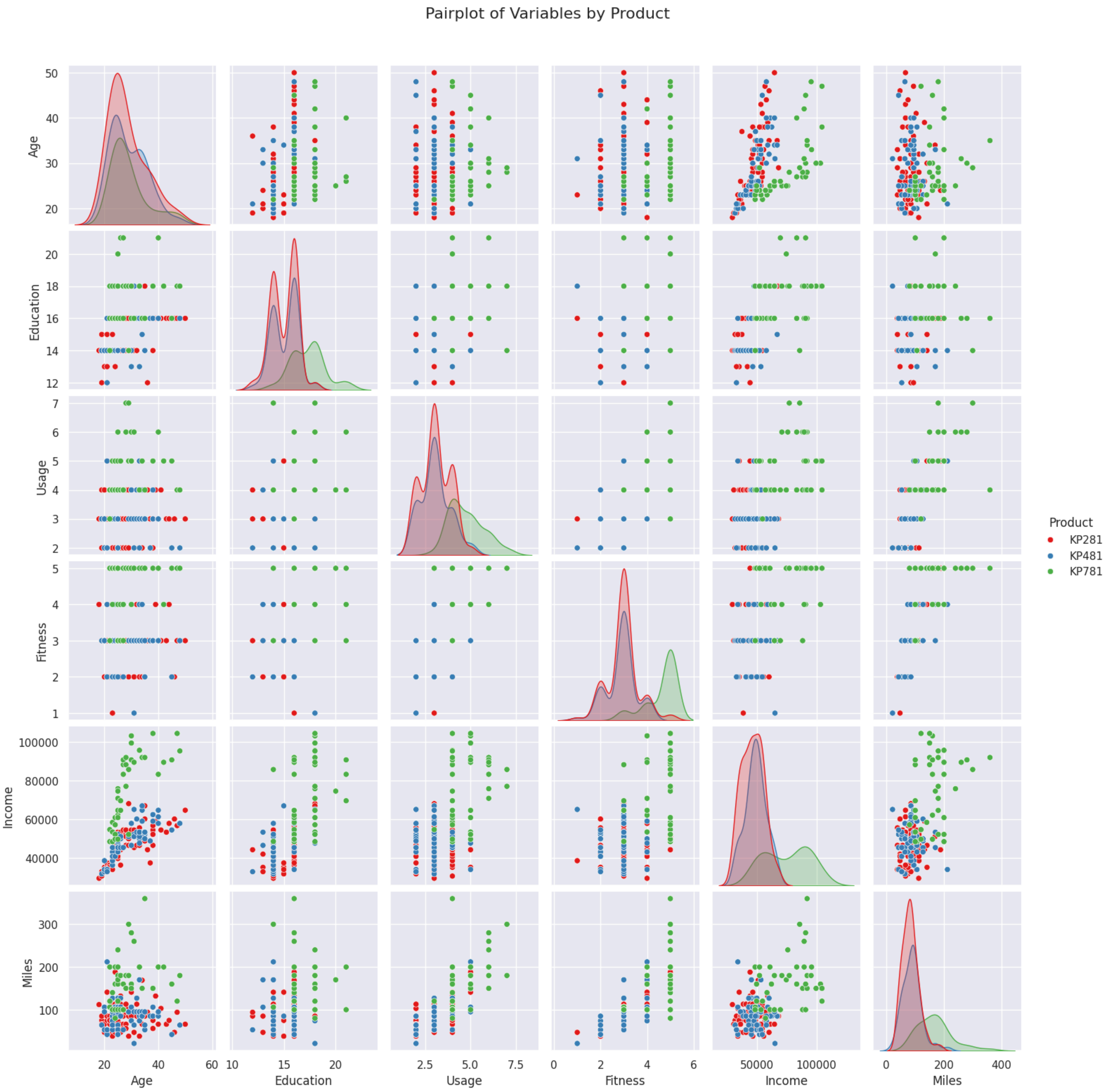
Income is positively correlated with Age, Education, Fitness, Usage, and Miles walked per week, indicating that individuals with higher incomes tend to be older, more educated, have higher fitness levels, use the treadmill more frequently, and walk more miles per week.

Miles walked per week:

Miles walked per week has the strongest positive correlation with Fitness and a substantial positive correlation with Usage, suggesting that individuals who walk more miles tend to have higher fitness levels and use the treadmill more frequently.

In [197…

```
# Using a Pairplot to visualize relationships between above variables, especially in terms of distribution and correlation

sns.pairplot(df, hue="Product",
             palette= 'Set1')

plt.suptitle("Pairplot of Variables by Product",
             y= 1.04,
             fontsize=16)
plt.show()
```



Descriptive Statistics Grouped by Product.

In [198…:

```
df.groupby("Product")['Age'].describe().reset_index().T
```

Out[198]:

	0	1	2
Product	KP281	KP481	KP781
count	80.0	60.0	40.0
mean	28.55	28.9	29.1
std	7.221452	6.645248	6.971738
min	18.0	19.0	22.0
25%	23.0	24.0	24.75
50%	26.0	26.0	27.0
75%	33.0	33.25	30.25
max	50.0	48.0	48.0

Age

- The average age across all products is similar, ranging from 28.55 to 29.1.
- Product KP781 has a slightly higher average age compared to the others.

In [199…

```
df.groupby("Product")['Education'].describe().reset_index().T
```

Out[199]:

	0	1	2
Product	KP281	KP481	KP781
count	80.0	60.0	40.0
mean	15.0375	15.116667	17.325
std	1.216383	1.222552	1.639066
min	12.0	12.0	14.0
25%	14.0	14.0	16.0
50%	16.0	16.0	18.0
75%	16.0	16.0	18.0
max	18.0	18.0	21.0

Education

- The education level is relatively consistent across products.
- Product KP781 has a higher average education level (17.325) compared to the others.

In [200…

```
df.groupby("Product")['Usage'].describe().reset_index().T
```

Out[200]:

	0	1	2
Product	KP281	KP481	KP781
count	80.0	60.0	40.0
mean	3.0875	3.066667	4.775
std	0.782624	0.799717	0.946993
min	2.0	2.0	3.0
25%	3.0	3.0	4.0
50%	3.0	3.0	5.0
75%	4.0	3.25	5.0
max	5.0	5.0	7.0

Usage

- Users of Product KP781 tend to use it more frequently, with an average usage of 4.775 times.
- Products KP281 and KP481 have lower average usage at 3.0875 and 3.066667, respectively.

In [201…

```
df.groupby("Product")['Fitness'].describe().reset_index().T
```

Out[201]:

	0	1	2
Product	KP281	KP481	KP781
count	80.0	60.0	40.0
mean	2.9625	2.9	4.625
std	0.66454	0.62977	0.667467
min	1.0	1.0	3.0
25%	3.0	3.0	4.0
50%	3.0	3.0	5.0
75%	3.0	3.0	5.0
max	5.0	4.0	5.0

Fitness

- Users of Product KP781 have a higher average fitness level (4.625) compared to the other products.
- Fitness levels are relatively consistent for products KP281 and KP481.

In [202…

```
df.groupby("Product")['Income'].describe().reset_index().T
```

Out[202]:

	0	1	2
Product	KP281	KP481	KP781
count	80.0	60.0	40.0
mean	46418.025	48973.65	75441.575
std	9075.78319	8653.989388	18505.83672
min	29562.0	31836.0	48556.0
25%	38658.0	44911.5	58204.75
50%	46617.0	49459.5	76568.5
75%	53439.0	53439.0	90886.0
max	68220.0	67083.0	104581.0

Income

- Product KP781 is associated with higher average income (75441.575) compared to the other products.
- Product KP281 has the lowest average income at 46418.025.

In [203…

```
df.groupby("Product")['Miles'].describe().reset_index().T
```

Out[203]:

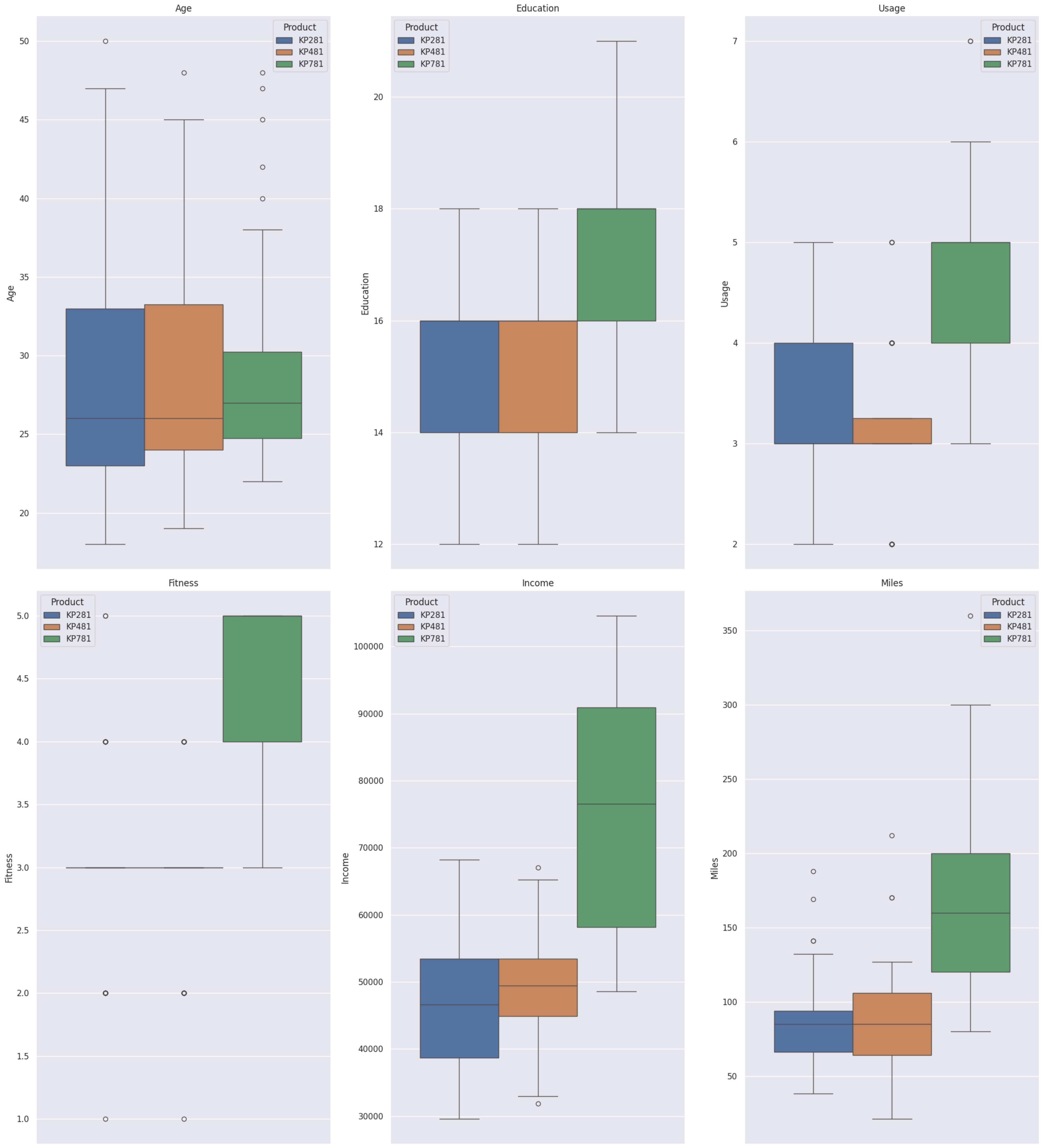
	0	1	2
Product	KP281	KP481	KP781
count	80.0	60.0	40.0
mean	82.7875	87.933333	166.9
std	28.874102	33.263135	60.066544
min	38.0	21.0	80.0
25%	66.0	64.0	120.0
50%	85.0	85.0	160.0
75%	94.0	106.0	200.0
max	188.0	212.0	360.0

Miles

- Users of Product KP781 tend to cover more miles on average (166.9) compared to the other products.
- Product KP281 has the lowest average miles covered at 82.7875.

In [204…

```
# Boxplots of Product Distribution for Various Variables.

fig, axes = plt.subplots(2, 3, figsize=(20, 22))

for i in range(2):
    for j in range(3):
        variable = columns[i * 3 + j]
        sns.boxplot(ax=axes[i, j], data=df, y=variable, hue="Product")
        axes[i, j].set_title(variable)
plt.tight_layout()
plt.show();
```

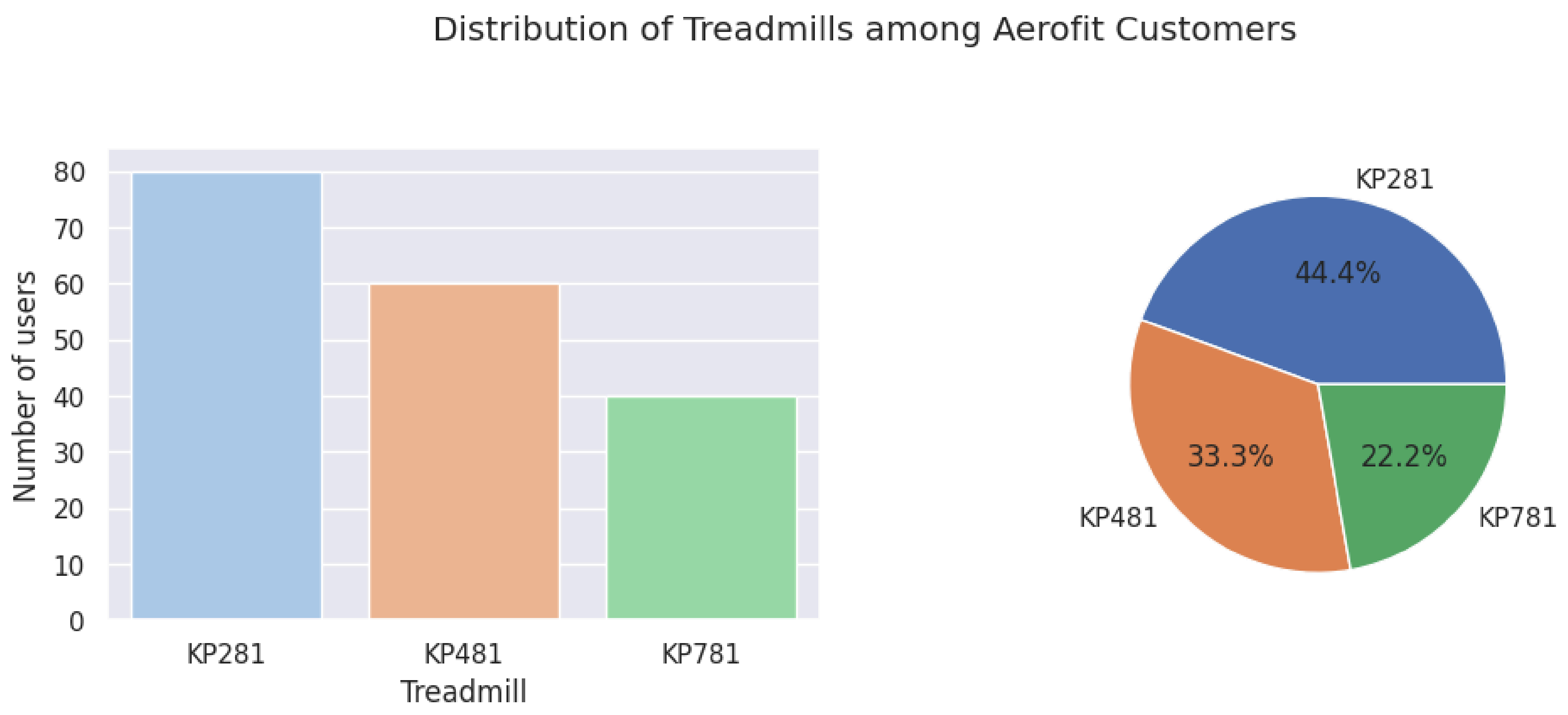


Marginal Probability & Insights of customer buying

```
In [205…
product = (df["Product"].value_counts(normalize=True)*100).T.round(2)
product
```

```
Out[205]:
KP281    44.44
KP481    33.33
KP781    22.22
Name: Product, dtype: float64
```

```
In [206…
plt.figure(figsize=(12,8))
plt.subplot(2,2,1)
sns.countplot(data=df,x='Product', palette='pastel',hue = 'Product',legend=False)
plt.xlabel('Treadmill')
plt.ylabel('Number of users')
plt.subplot(2,2,2)
plt.pie(df['Product'].value_counts(), labels=df['Product'].unique(), autopct='%1.1f%%')
plt.suptitle('Distribution of Treadmills among Aerofit Customers')
plt.show()
```



Insights :

1. Among the users, 44.44% prefer using the KP281 treadmill, while 33.33% opt for the KP481 treadmill, and only 22.22% of users favor the KP781 treadmill.
2. KP281, being an entry-level and more affordable treadmill compared to the others, is the preferred choice among the majority of customers. 3.33.3% of customers favor the KP481 treadmill, drawn by its ideal fit for mid-level runners and its excellent value-for-money offering.
3. KP781 treadmill, being more advanced and costlier than the other two options, is chosen by only 22.2% of customers. **Recommendations:**
4. Emphasize the budget-friendly nature of the KP281 treadmill to attract more customers.
5. Highlight the key features of the KP281 that make it a great entry-level option for fitness enthusiasts.
6. Provide special offers or discounts to further entice customers looking for a cost-effective option.
7. Engage with fitness communities online to showcase the KP281's appeal to beginners. 5.Focus marketing efforts on reaching out to mid-level runners, emphasizing how the KP481 is tailored to meet their specific fitness needs and goals.
8. Showcase the competitive pricing and the outstanding features of the KP481 that make it a cost-effective choice for customers.
9. Launch targeted marketing campaigns to increase awareness and interest in the KP781 among potential customers who may value its advanced capabilities. Utilize various channels such as social media, fitness forums, and influencer collaborations.

In [207…

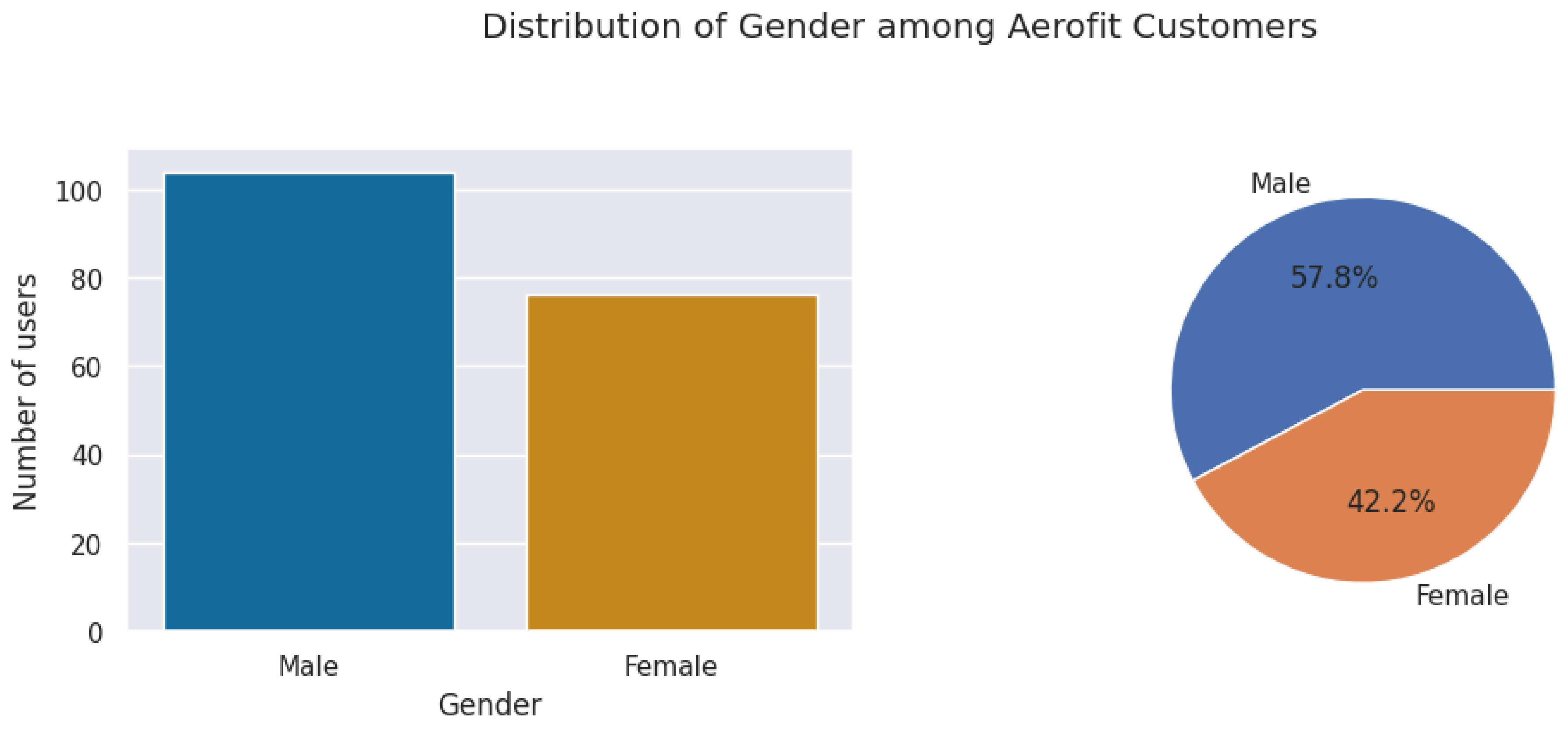
```
mp_of_gender = (pd.crosstab(df['Product'],df['Gender'],margins = True, normalize = True)*100).T.round(2)
mp_of_gender
```

Out[207]:

Product	KP281	KP481	KP781	All
Gender				
Female	22.22	16.11	3.89	42.22
Male	22.22	17.22	18.33	57.78
All	44.44	33.33	22.22	100.00

In [208…

```
plt.figure(figsize=(12,8))
plt.subplot(2,2,1)
sns.countplot(data=df,x='Gender', palette='colorblind',hue ='Gender',legend=False)
plt.xlabel('Gender')
plt.ylabel('Number of users')
plt.subplot(2,2,2)
plt.pie(df['Gender'].value_counts(), labels=df['Gender'].unique(), autopct='%1.1f%%')
plt.suptitle('Distribution of Gender among Aerofit Customers')
plt.show()
```



Insight: Aerofit has 57.78% male customers and 42.22% female customers.

Recommendations:

1. Create targeted advertisements and promotions that appeal to women, showcasing how fitness can positively impact their lives.
2. Showcase the female-friendly features and benefits of Aerofit treadmills to attract more female customers.

In [209…

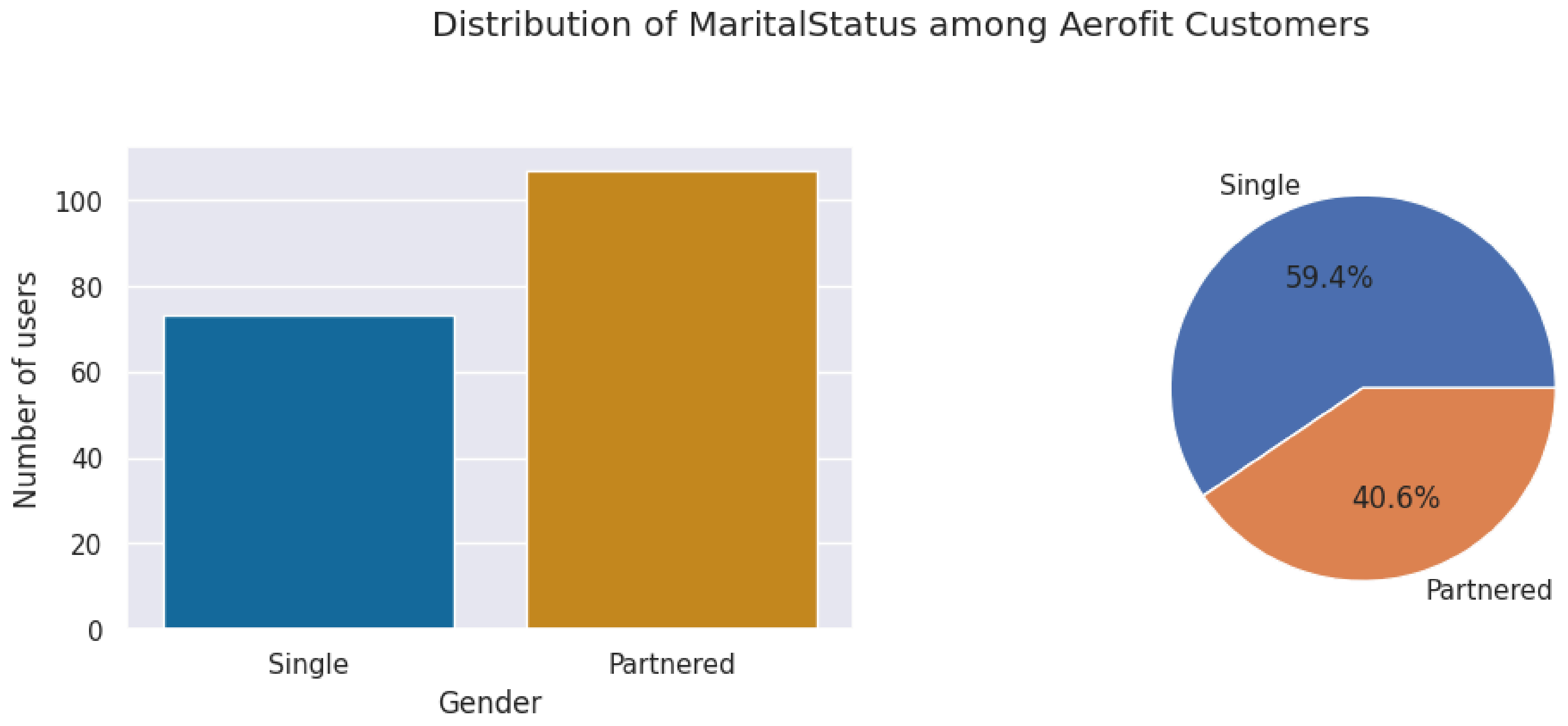
```
mp_of_marital = (pd.crosstab(df['Product'],df['MaritalStatus'],margins = True, normalize = True)*100).T.round(2)
mp_of_marital
```

Out[209]:

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	26.67	20.00	12.78	59.44
Single	17.78	13.33	9.44	40.56
All	44.44	33.33	22.22	100.00

In [210…

```
plt.figure(figsize=(12,8))
plt.subplot(2,2,1)
sns.countplot(data=df,x='MaritalStatus', palette='colorblind',hue ='MaritalStatus',legend=False)
plt.xlabel('Gender')
plt.ylabel('Number of users')
plt.subplot(2,2,2)
plt.pie(df['MaritalStatus'].value_counts(), labels=df['MaritalStatus'].unique(), autopct='%1.1f%%')
plt.suptitle('Distribution of MaritalStatus among Aerofit Customers')
plt.show()
```



In [211…

```
mp_of_Age = (pd.crosstab(df['Product'],df['AgeCategory'],margins = True, normalize = True)*100).T.round(2)

mp_of_Age
```

Out[211]:

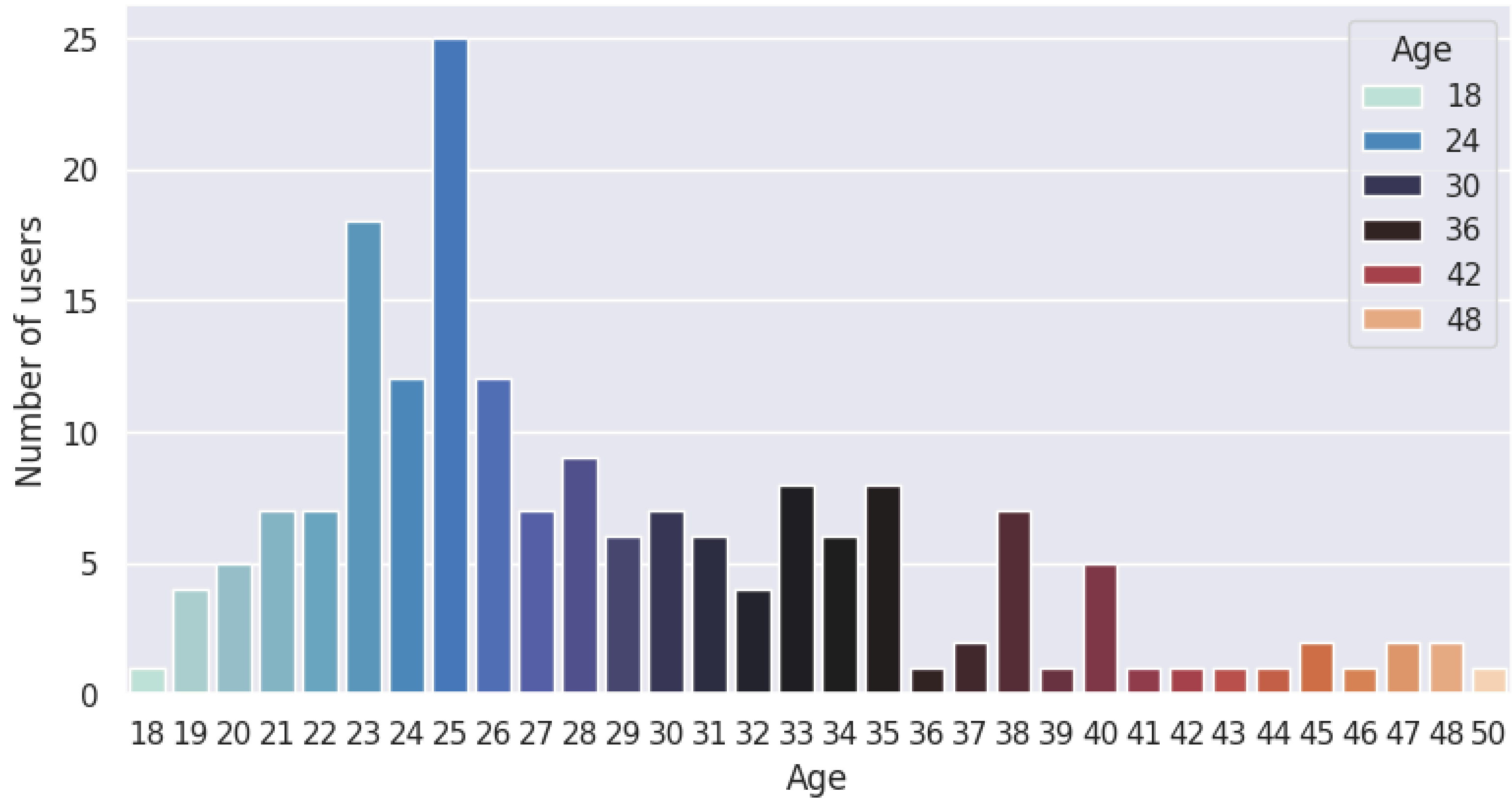
Product	KP281	KP481	KP781	All
AgeCategory				
Teens	3.33	2.22	0.00	5.56
Young Adults	27.22	17.22	16.67	61.11
Adults	10.56	12.78	3.33	26.67
Over 40s	3.33	1.11	2.22	6.67
All	44.44	33.33	22.22	100.00

In [212…

```
plt.figure(figsize=(20,10))
plt.subplot(2,2,1)
sns.countplot(data=df,x='Age',palette='icefire',hue='Age')
plt.xlabel('Age')
plt.ylabel('Number of users')
```

Out[212]:

```
Text(0, 0.5, 'Number of users')
```



Insight

1. Most of the Aerofit customer falls under young age-group (18-29). 2.27.78% of middle-aged(30-39) users prefer to use the Aerofit Treadmills. 3.9.4% of users in the old (40-50) age group prefer purchasing Aerofit treadmills. **Recommendations:**

Offer personalized assistance to help customers aged 40-50 select the ideal treadmill model, providing them with the tools to maintain an active and healthy lifestyle. With Aerofit's expert guidance, customers can feel confident and motivated to make the most of their treadmills effectively.

In [213…

```
mp_of_income = (pd.crosstab(df['Product'],df['IncomeSlab'],margins = True, normalize = True)*100).T.round(2)

mp_of_income
```

Out[213]:

Product	KP281	KP481	KP781	All
IncomeSlab				
Low Income	4.44	3.33	0.00	7.78
Middle-class	36.67	26.11	6.11	68.89
Upper-Middle class	3.33	3.89	6.67	13.89
Wealthy	0.00	0.00	9.44	9.44
All	44.44	33.33	22.22	100.00

Conditional probability

In [214…

```
cp_of_gender = (pd.crosstab(df['Product'],df['Gender'],margins=True,normalize="columns")*100).T.round(2)
cp_of_gender
```

Out[214]:

Product	KP281	KP481	KP781
Gender			
Female	52.63	38.16	9.21
Male	38.46	29.81	31.73
All	44.44	33.33	22.22

Insights:

- Probability of Female customer buying KP281(52.63%) is more than male(38.46%).
- KP281 is more recommended for female customers.
- Probability of Male customer buying Product KP781(31.73%) is way more than female(9.21%).
- KP481 product is specifically recommended for Female customers who are intermediate user.

In [215…

```
cp_of_marital = (pd.crosstab(df['Product'],df['MaritalStatus'],margins=True,normalize='columns')*100).T.round(2)
cp_of_marital
```

Out[215]:

Product	KP281	KP481	KP781
MaritalStatus			
Partnered	44.86	33.64	21.50
Single	43.84	32.88	23.29
All	44.44	33.33	22.22

Insights:

- KP281 is slightly more favored by partnered customers (44.86%) compared to single customers (43.84%).
- KP481 enjoys consistent popularity across both partnered (33.64%) and single (32.88%) customers.
- KP781, an advanced product, has a lower probability of selection.
- But shows a slightly higher preference among single customers (23.29%) compared to partnered customers (21.50%).

In [216…

```
cp_of_Age = (pd.crosstab(df['Product'],df['AgeCategory'],margins = True, normalize = 'columns')*100).T.round(2)
cp_of_Age
```

Out[216]:

Product	KP281	KP481	KP781
AgeCategory			
Teens	60.00	40.00	0.00
Young Adults	44.55	28.18	27.27
Adults	39.58	47.92	12.50
Over 40s	50.00	16.67	33.33
All	44.44	33.33	22.22

Insights:

- Among teens, KP281 is the predominant choice (60%), while KP481 is chosen by 40%. KP781 does not seem to appeal to this age group.
- In the young adults category, KP281 has a slightly higher preference (44.55%) compared to KP481 (28.18%) and KP781 (27.27%).
- Among adults, there is a balanced distribution. KP481 is the most favored (47.92%), followed by KP281 (39.58%), KP781 with a low preference (12.50%).
- Customers over 40 show a clear preference for KP281 (50%), followed by KP781 (33.33%), while KP481 has a lower preference (16.67%).

In [217…

```
cp_of_Income = (pd.crosstab(df['Product'],df['IncomeSlab'],margins = True, normalize = 'columns')*100).T.round(2)

cp_of_Income
```

Out[217]:

Product	KP281	KP481	KP781
IncomeSlab			
Low Income	57.14	42.86	0.00
Middle-class	53.23	37.90	8.87
Upper-Middle class	24.00	28.00	48.00
Wealthy	0.00	0.00	100.00
All	44.44	33.33	22.22

Insights:

- In the Low-Income category, KP281 is dominant (57.14%), while KP481 is chosen by 42.86%. KP781 doesn't seem to be preferred in this income bracket.

- Among the Middle-Class, KP281 remains popular (53.23%), followed by KP481 (37.90%), and KP781 has a lower preference (8.87%).
- In the Upper-Middle class,KP781 being the most favored (48.00%), followed by KP481 (28.00%), and KP281 has a lower preference (24.00%).
- Among the Wealthy, KP781 is the exclusive choice (100.00%), with KP281 and KP481 having no preference.

Customer Profiling

KP281

- Affordable Entry-Level Choice: KP281 is a budget-friendly option and the best-selling product.
- Popularity Among Beginners: It's favored by entry-level customers, both male and female.
- Usage Patterns: Typically used 3 to 4 times a week, covering 70 to 90 miles.
- Fitness Rating: Most customers rate their fitness as average.
- Demographic Preferences: Attracts younger to elder beginners, especially preferred by single females and partnered males.
- Income Range: Preferred by customers with incomes between 39K to 53K.

KP481

- Intermediate Level Product: KP481 is an intermediate-level treadmill.
- Usage Characteristics: Customers cover 70 to 130 miles per week, with usage averaging 3 days.
- Fitness and Mileage Focus: Fitness level varies from bad to average, and the focus is on covering more miles.
- Demographic Preferences: More popular among females, recommended for intermediate female users.
- Age Groups: Attracts customers across different age groups - teens, adults, and middle-aged.
- Income and Relationship Preferences: Average income around 49K, more preferred by partnered customers.

KP781

- Advanced and High-Priced: KP781 is an advanced and higher-priced product.
- Distance and Usage: Customers cover 120 to 200+ miles per week, using it 4 to 5 times.
- Fitness Rating: Users rate their fitness as excellent.
- Demographic Insights: Preferred by middle to higher age groups, especially single individuals with higher exercise levels.
- Gender Preference: More favored by males, especially those with extensive exercise routines.
- Income Influence: Preferred by higher-income individuals, correlated with higher education.
- Experience Factor: Attracts customers familiar with previous aerofit products.
- Partnered Preference: Partnered females show a preference for KP781.

Recommendations

Attracting Female Customers:

To increase female engagement with exercise equipment, consider launching a targeted marketing campaign specifically aimed at encouraging women to participate in fitness activities.

Strategic Pricing for Budget-Conscious Customers: Position the KP281 and KP481 treadmills as budget-friendly options. These models are ideal for customers with an annual income in the range of 39K to 53K dollars.

Highlighting Premium Features: Promote the KP781 treadmill as a premium product. Leverage its advanced features to attract professionals, athletes, and fitness enthusiasts. Collaborate with influencers and international athletes to enhance visibility.

Exploring Market Expansion: Investigate the possibility of expanding the market beyond the age of 50. Conduct thorough research to understand potential health benefits and challenges associated with this demographic.

Enhancing Customer Support and Upgrades: Strengthen your customer support system. Encourage users to consider upgrading from basic treadmill models to higher-level ones based on their consistent usage patterns.

Tailored Promotion for Female Users: Specifically target female customers who engage in regular exercise routines. Provide clear guidance on the benefits of the KP781 treadmill to make it more appealing.

Age-Specific Marketing: Customize marketing efforts for individuals aged 40 and above. Highlight health benefits and user-friendly features of the KP781 model to attract this demographic

In [217…