Business Case: Aerofit - Descriptive Statistics & Probability

Aerofit stands as a prominent brand in the fitness equipment industry, offering a diverse product line that encompasses treadmills, exercise bikes, gym equipment, and fitness accessories. This comprehensive range is designed to meet the varied needs of individuals across all categories.

Business Problem

The AeroFit market research team aims to discern the distinct characteristics of the audience associated with each treadmill variant offered by the company. This initiative seeks to enhance recommendations for new customers. The team will analyze potential differences across products concerning customer attributes.

Conduct descriptive analytics to formulate a customer profile for each AeroFit treadmill product, utilizing relevant tables and charts. Generate two-way contingency tables for each AeroFit treadmill product, calculating both conditional and marginal probabilities. Provide insights into their impact on the business.

Defining Problem Statement and Analysing basic metrics

Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), statistical summary

Importing libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from scipy.stats import norm
import warnings
warnings.filterwarnings('ignore')
```

Loading and Reading Dataset

In [2]:
```
url = "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original
```

In [3]:
```
df = pd.read_csv(url)
```

Shape of the data

In [4]:
```
df.shape
```

Out[4]:
```
(180, 9)
```

In [5]:
```
df.head
```

Out[5]:
```
<bound method NDFrame.head of     Product  Age  Gender  Education MaritalStatus  Usag
e  Fitness  Income  \
0     KP281   18    Male         14        Single      3        4   29562
1     KP281   19    Male         15        Single      2        3   31836
2     KP281   19  Female         14     Partnered      4        3   30699
3     KP281   19    Male         12        Single      3        3   32973
4     KP281   20    Male         13     Partnered      4        2   35247
..      ...  ...     ...        ...           ...    ...      ...     ...
175   KP781   40    Male         21        Single      6        5   83416
176   KP781   42    Male         18        Single      5        4   89641
177   KP781   45    Male         16        Single      5        5   90886
178   KP781   47    Male         18     Partnered      4        5  104581
179   KP781   48    Male         18     Partnered      4        5   95508

     Miles
0      112
1       75
2       66
3       85
4       47
..     ...
175    200
176    200
177    160
178    120
179    180

[180 rows x 9 columns]>
```

In [6]:
```
df.info
```

Out[6]:
```
<bound method DataFrame.info of     Product  Age  Gender  Education MaritalStatus  Us
age  Fitness  Income  \
0     KP281   18    Male         14        Single      3        4   29562
1     KP281   19    Male         15        Single      2        3   31836
2     KP281   19  Female         14     Partnered      4        3   30699
3     KP281   19    Male         12        Single      3        3   32973
4     KP281   20    Male         13     Partnered      4        2   35247
..      ...  ...     ...        ...           ...    ...      ...     ...
175   KP781   40    Male         21        Single      6        5   83416
176   KP781   42    Male         18        Single      5        4   89641
177   KP781   45    Male         16        Single      5        5   90886
178   KP781   47    Male         18     Partnered      4        5  104581
179   KP781   48    Male         18     Partnered      4        5   95508

     Miles
0      112
1       75
2       66
3       85
4       47
..     ...
175    200
176    200
177    160
178    120
179    180

[180 rows x 9 columns]>
```

In [7]:
```
df.describe(include="all")
```

Out[7]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Incom
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.00000
unique	3	NaN	2	NaN	2	NaN	NaN	Nal
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	Nal
freq	80	NaN	104	NaN	107	NaN	NaN	Nal
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.57777
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.68422
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.00000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.75000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.50000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.00000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.00000

In [8]:
```
print('\nColumns with missing value:')
print(df.isnull().any())
```

```
Columns with missing value:
Product          False
Age              False
Gender           False
Education        False
MaritalStatus    False
Usage            False
Fitness          False
Income           False
Miles            False
dtype: bool
```

Observations:

1. The dataset is devoid of any missing values.
2. There are three distinct products featured in the dataset, with KP281 emerging as the most frequently purchased.
3. The age of individuals spans from 18 to 50, with a mean age of 28.79. Moreover, 75% of individuals are aged 33 or below.
4. A significant proportion of individuals possess 16 years of education, as 75% of the sample has an education level of 16 years or less.

Out of the 180 data points, 104 correspond to males, while the remaining entries pertain to females.

Both Income and Miles exhibit a notably high standard deviation, suggesting the potential presence of outliers in these variables.

Non-Graphical Analysis: Value counts and unique attributes

In [9]:
```
df['Product'].unique()
```

Out[9]:
```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

There are 3 unique products available in the dataset.

Visual Analysis - Univariate & Bivariate

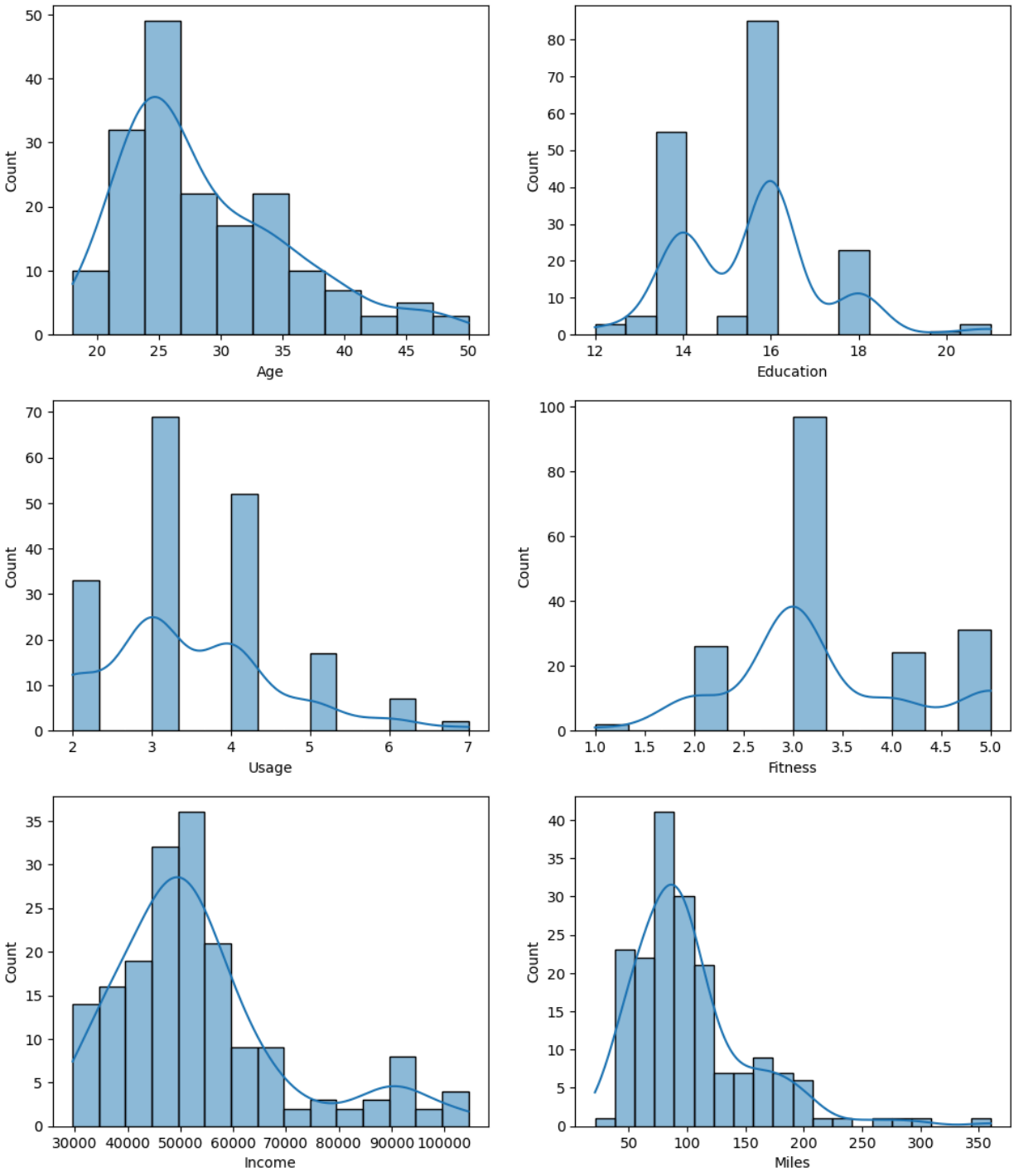
Univariate Analysis

Understanding the distribution of the data for the quantitative attributes:

1. Age
2. Education

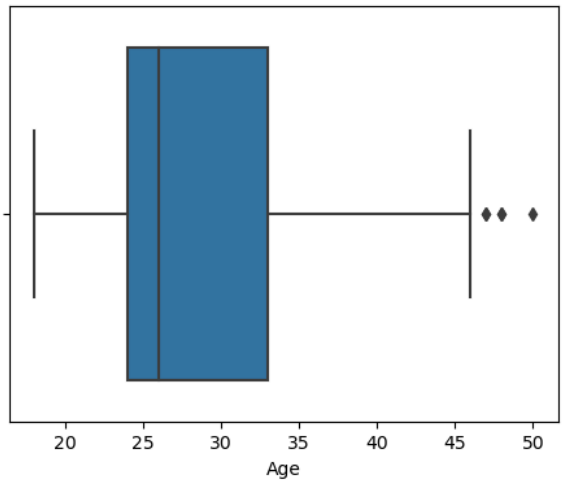
3. Usage
4. Fitness
5. Income
6. Miles

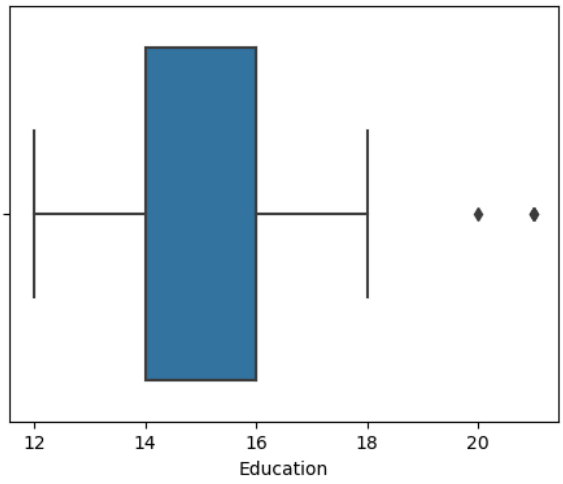
In [10]:
```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))
fig.subplots_adjust(top=1.2)
sns.histplot(data=df, x="Age", kde=True, ax=axis[0,0])
sns.histplot(data=df, x="Education", kde=True, ax=axis[0,1])
sns.histplot(data=df, x="Usage", kde=True, ax=axis[1,0])
sns.histplot(data=df, x="Fitness", kde=True, ax=axis[1,1])
sns.histplot(data=df, x="Income", kde=True, ax=axis[2,0])
sns.histplot(data=df, x="Miles", kde=True, ax=axis[2,1])
plt.show()
```

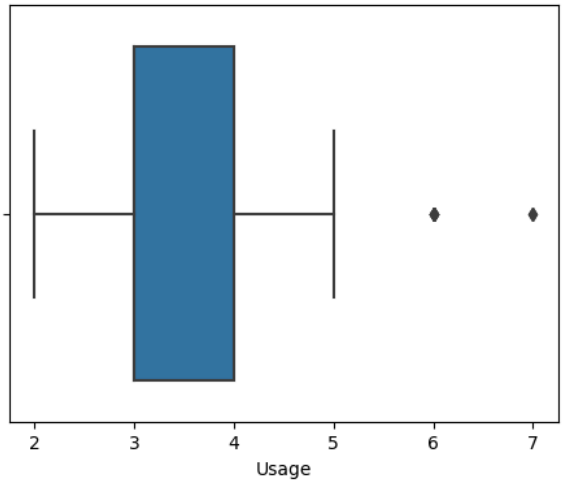


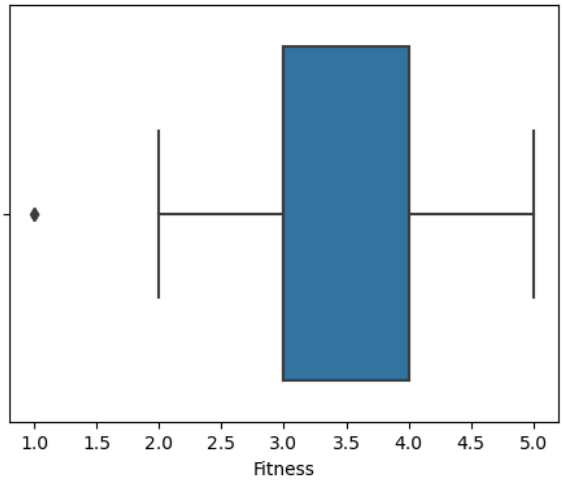
Outliers detection using BoxPlots

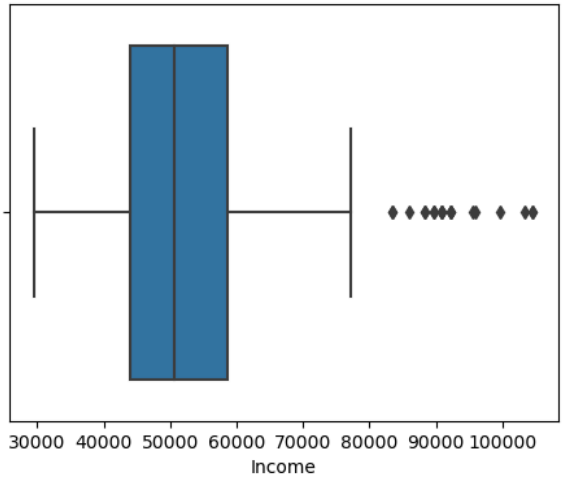
In [11]:
```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))
fig.subplots_adjust(top=1.2)
sns.boxplot(data=df, x="Age", orient='h', ax=axis[0,0])
sns.boxplot(data=df, x="Education", orient='h', ax=axis[0,1])
sns.boxplot(data=df, x="Usage", orient='h', ax=axis[1,0])
sns.boxplot(data=df, x="Fitness", orient='h', ax=axis[1,1])
sns.boxplot(data=df, x="Income", orient='h', ax=axis[2,0])
sns.boxplot(data=df, x="Miles", orient='h', ax=axis[2,1])
plt.show()
```

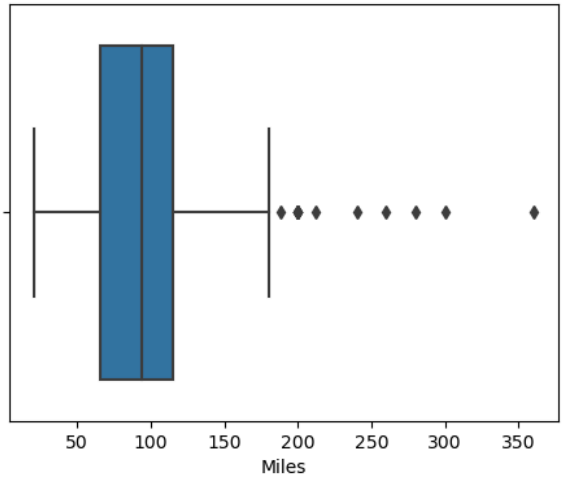










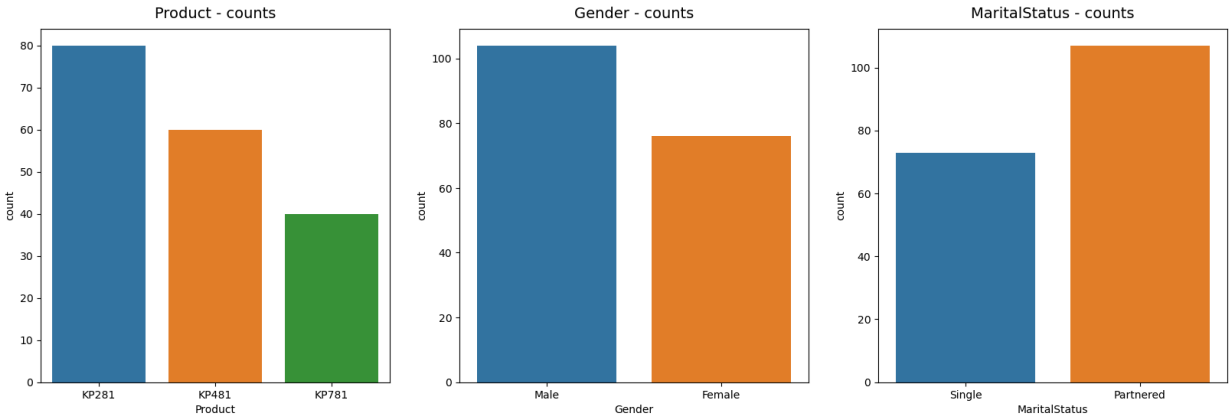


Observation: **Upon examining the boxplots, several insights emerge:

- Age, Education, and Usage exhibit minimal outliers.
- Conversely, Income and Miles display a higher prevalence of outliers.
- Understanding the distribution of data for qualitative attributes:

1. Product
2. Gender
3. MaritalStatus

In [12]:
```
fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(20, 6))
sns.countplot(data=df, x='Product', ax=axs[0])
sns.countplot(data=df, x='Gender', ax=axs[1])
sns.countplot(data=df, x='MaritalStatus', ax=axs[2])

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



Observations:

1. KP281 is the most frequently purchased product.
2. There are more males than females in the dataset.
3. The data comprises a higher number of partnered individuals.
4. To be precise, the normalized count for each variable is provided below.

In [13]:
```
df1 = df[['Product', 'Gender', 'MaritalStatus']].melt()
df1.groupby(['variable', 'value'])[['value']].count() / len(df)
```

Out[13]:

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

Observations:

**Product:

1. 44.44% of customers have purchased the KP281 product.
2. 33.33% of customers have purchased the KP481 product.

3. 22.22% of customers have purchased the KP781 product.

**Gender:

1. 57.78% of customers are male.

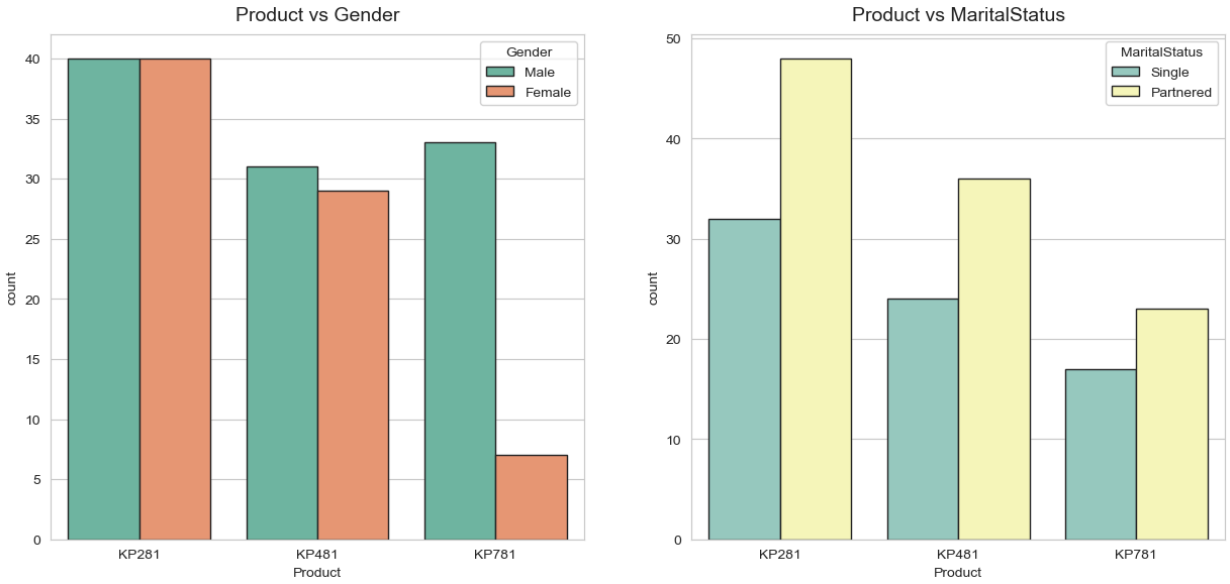
**MaritalStatus:

1. 59.44% of customers are partnered.

Bivariate Analysis:

Investigating whether features such as Gender or MaritalStatus have any impact on the purchased product.

In [14]:
```
sns.set_style(style='whitegrid')
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15, 6.5))
sns.countplot(data=df, x='Product', hue='Gender', edgecolor="0.15", palette='Set2', ax
sns.countplot(data=df, x='Product', hue='MaritalStatus', edgecolor="0.15", palette='Se
axs[0].set_title("Product vs Gender", pad=10, fontsize=14)
axs[1].set_title("Product vs MaritalStatus", pad=10, fontsize=14)
plt.show()
```



Observations:

**Product vs Gender:

- An equal number of males and females have purchased the KP281 product.
- A similar trend is observed for the KP481 product.
- The majority of male customers have opted for the KP781 product.

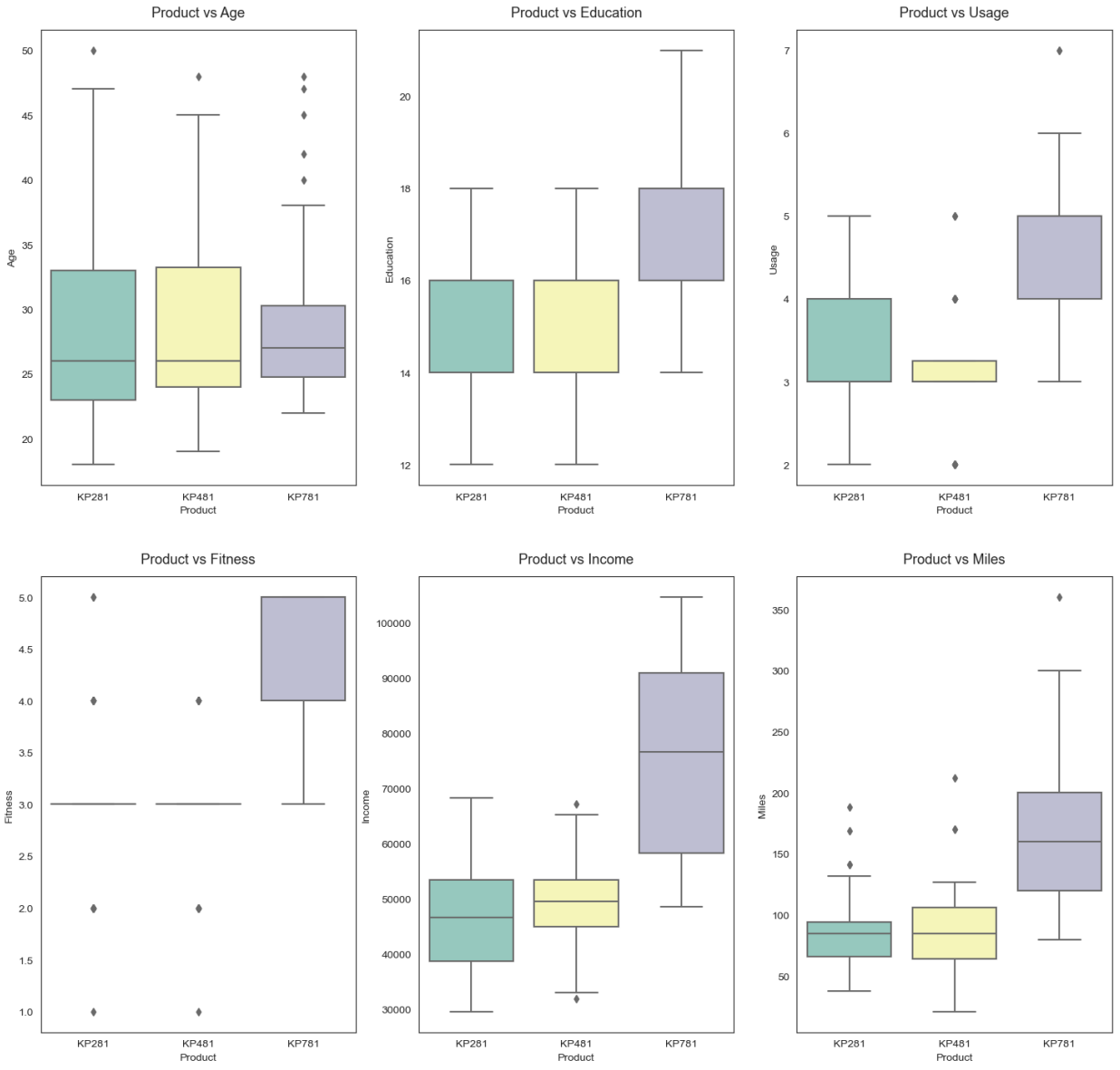
**Product vs MaritalStatus:

- Customers who are partnered are more inclined to purchase the product.

- Checking if the following features have any effect on the product purchased:

1. Age
2. Education
3. Usage
4. Fitness
5. Income
6. Miles

In [15]:
```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(18, 12))
fig.subplots_adjust(top=1.2)
count = 0
for i in range(2):
    for j in range(3):
        sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i,j], palette='Set3')
        axs[i,j].set_title(f"Product vs {attrs[count]}", pad=12, fontsize=13)
        count += 1
```



Observations:

**Product vs Age:

1. Customers purchasing products KP281 and KP481 exhibit the same median age value.
2. Customers aged between 25-30 are more likely to buy the KP781 product.

**Product vs Education:

1. Customers with an education level greater than 16 are more inclined to purchase the KP781 product.
2. Customers with an education level less than 16 have an equal likelihood of purchasing KP281 or KP481.

**Product vs Usage:

1. Customers planning to use the treadmill more than 4 times a week are more likely to purchase the KP781 product.
2. Other customers are more inclined to purchase KP281 or KP481.

**Product vs Fitness:

1. The higher the customer's fitness level (fitness >= 3), the more likely they are to purchase the KP781 product.

**Product vs Income:

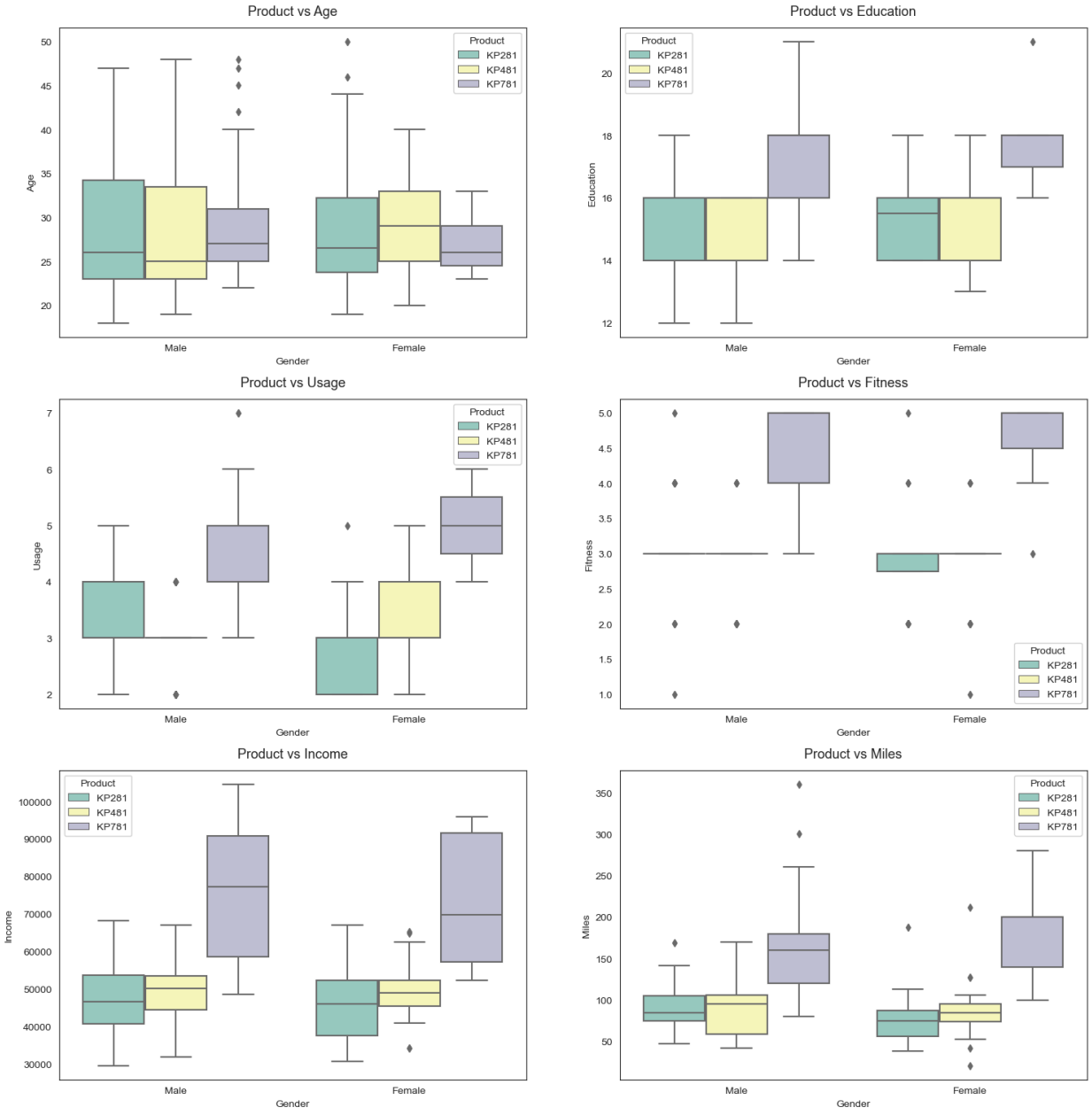
1. Higher customer income (Income >= 60000) corresponds to a higher likelihood of purchasing the KP781 product.

**Product vs Miles:

1. If the customer expects to walk/run more than 120 miles per week, there is a higher likelihood of purchasing the KP781 product.

Multivariate Analysis

In [16]:
```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(18, 12))
fig.subplots_adjust(top=1.3)
count = 0
for i in range(3):
    for j in range(2):
        sns.boxplot(data=df, x='Gender', y=attrs[count], hue='Product', ax=axs[i,j], p
        axs[i,j].set_title(f"Product vs {attrs[count]}", pad=12, fontsize=13)
        count += 1
```



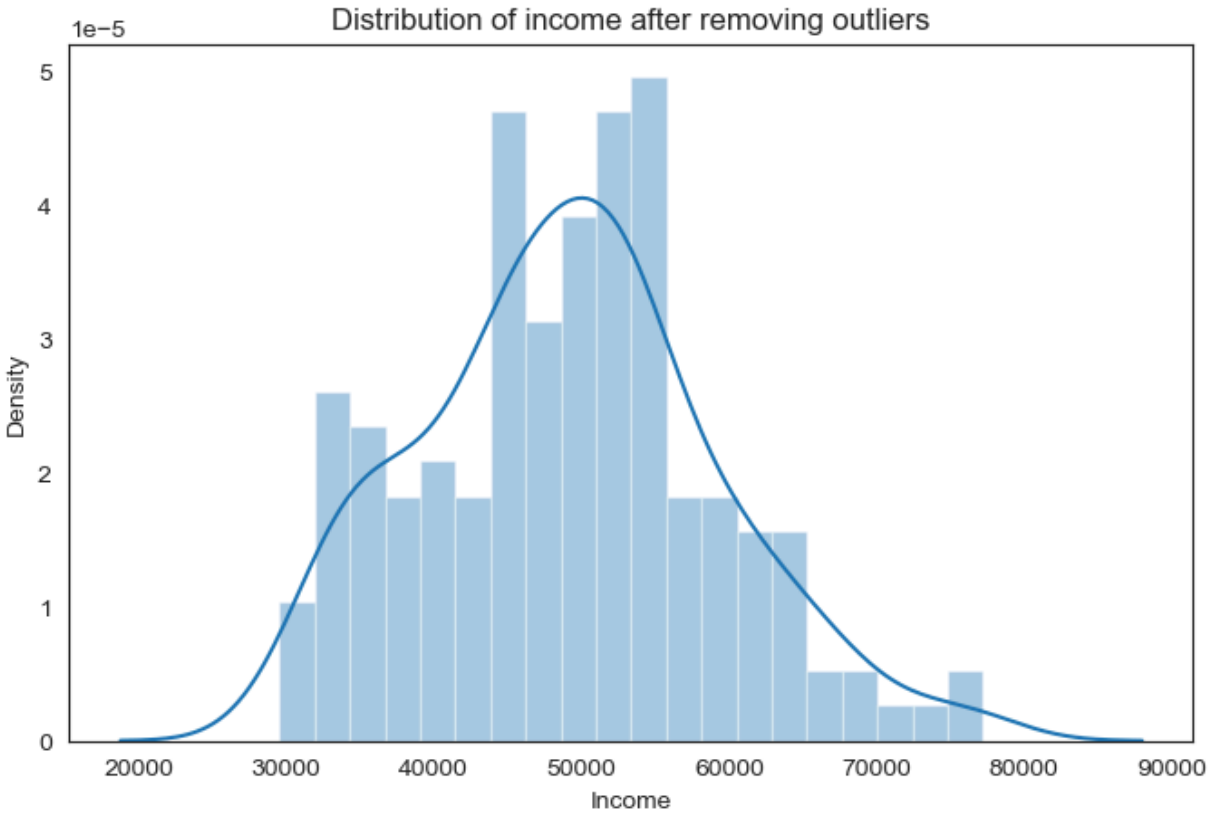
Observations: **Females who plan to use the treadmill 3-4 times a week are more inclined to purchase the KP481 product.

In [17]:
```
def remove_outliers(df, label, iqr_factor = 1.5) :
    """
    Outlier Treatment :
    In income columns we can see some outliers we can remove them.

    """

    q1 = df[label].quantile(0.25)
    q3 = df[label].quantile(0.75)
    IQR = q3-q1
    print(f'IQR for the label column is : {IQR}')
    print(df.shape)
    df = df[ (df[label] > (q1 - (iqr_factor * IQR))) & (df[label] < (q3 + (iqr_factor
    print(df.shape)
    plt.figure(figsize= (8,5))
    sns.distplot(df['Income'], bins =20)
```

```
    plt.title('Distribution of income after removing outliers')
    plt.show()
    return df
```

In [19]:
```
df_remove_outliers = remove_outliers(df,'Income',iqr_factor=1.5)
```

```
IQR for the label column is : 14609.25
(180, 9)
(161, 9)
```

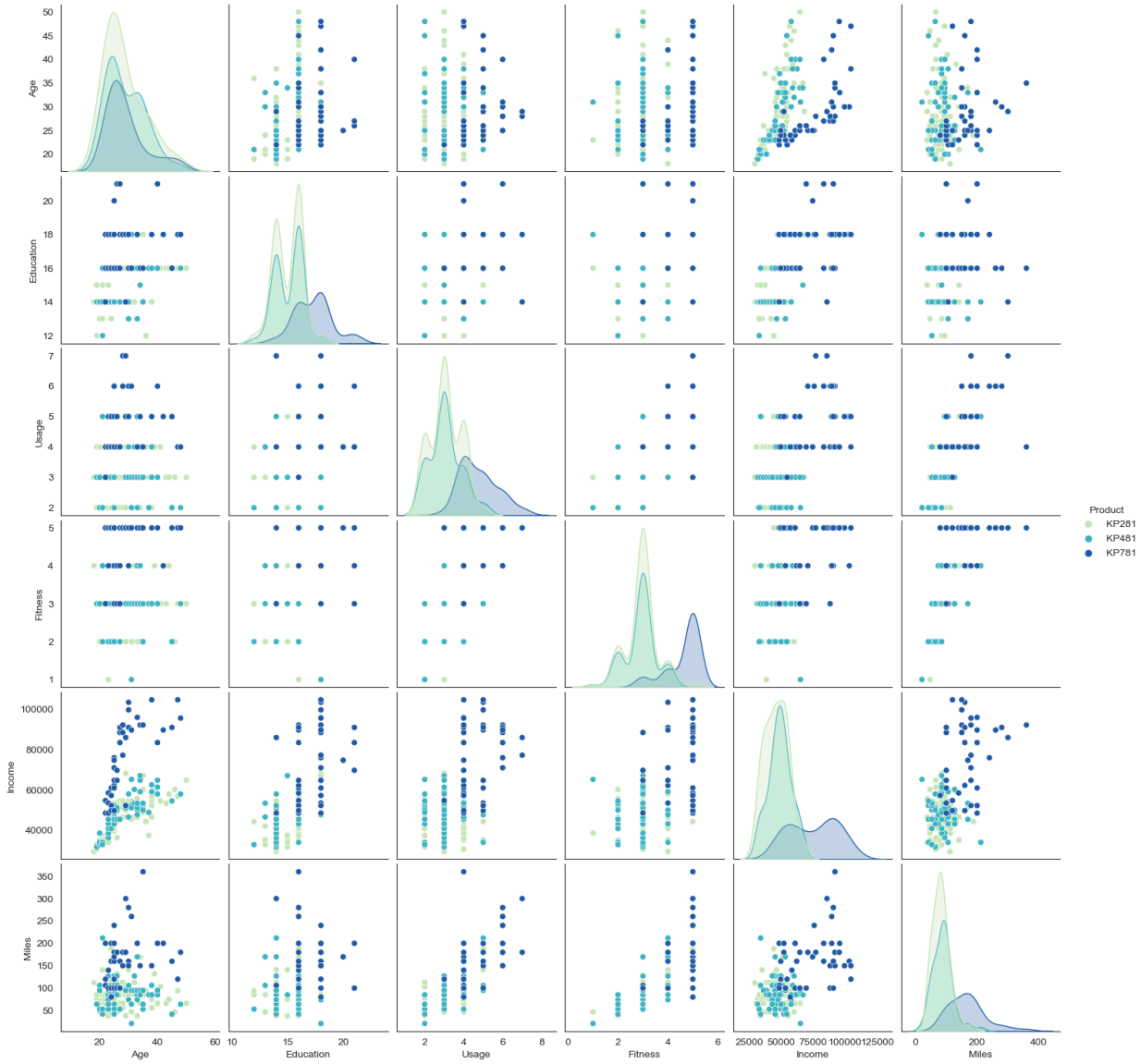


1. To increase the sales of the KP781 treadmill, the company should focus on individuals with higher salaries (>70k), particularly targeting males who show a higher inclination to purchase this product.
2. The ideal target demographic for the KP781 treadmill comprises individuals with ages ranging from 22 to 35.
3. Single individuals with a higher income are predominantly inclined towards the KP781 treadmill. Hence, this demographic becomes a crucial target audience for Aerofit.

Correlation between Variables

Pairplot

In [18]:
```
sns.pairplot(df, hue ='Product', palette= 'YlGnBu')
plt.show()
```



Heatmap

In [20]:
```
# First we need to convert object into int datatype for usage and fitness columns
df['Usage'] = df['Usage'].astype('int')
df['Fitness'] = df['Fitness'].astype('int')
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    int32 
 6   Fitness        180 non-null    int32 
 7   Income         180 non-null    int64 
 8   Miles          180 non-null    int64 
dtypes: int32(2), int64(4), object(3)
memory usage: 11.4+ KB
```

In [21]:
```
corr_mat = df.corr()
plt.figure(figsize=(15,6))
sns.heatmap(corr_mat,annot = True, cmap="YlGnBu")
plt.show()
```

**Observations:

- The pair plot indicates a positive correlation between Age and Income, which is further supported by the heatmap showing a strong relationship between them.
- Education and Income are evidently highly correlated, which is expected. Additionally, Education shows a significant correlation with Fitness rating and Treadmill usage.
- Treadmill usage exhibits a strong correlation with Fitness and Miles, implying that higher usage is associated with greater fitness levels and mileage.

Computing Marginal & Conditional Probabilities

In [22]:
```
df['Product'].value_counts(normalize=True)
```

Out[22]:
```
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: Product, dtype: float64
```

In [23]:
```
pd.crosstab(index =df['Product'],columns = df['Gender'],margins = True,normalize = Tru
```

Out[23]:

Product	Female	Male	All
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

Conditional Probabilities: The Probability of a treadmill being purchased by a female is 42%.

The conditional probability of purchasing the treadmill model given that the customer is female is

For Treadmill model KP281 - 22%

For Treadmill model KP481 - 16%

For Treadmill model KP781 - 4%

The Probability of a treadmill being purchased by a male is 58%.

The conditional probability of purchasing the treadmill model given that the customer is male is -

For Treadmill model KP281 - 22%

For Treadmill model KP481 - 17%

For Treadmill model KP781 - 18%

In [24]:
```
def p_prod_given_gender(gender, print_marginal=False):
    if gender != "Female" and gender != "Male":
        return "Invalid gender value."

    df1 = pd.crosstab(index=df['Gender'], columns=[df['Product']])
    p_781 = df1['KP781'][gender] / df1.loc[gender].sum()
    p_481 = df1['KP481'][gender] / df1.loc[gender].sum()
    p_281 = df1['KP281'][gender] / df1.loc[gender].sum()

    if print_marginal:
        print(f"P(Male): {df1.loc['Male'].sum()/len(df):.2f}")
        print(f"P(Female): {df1.loc['Female'].sum()/len(df):.2f}\n")

    print(f"P(KP781/{gender}): {p_781:.2f}")
    print(f"P(KP481/{gender}): {p_481:.2f}")
    print(f"P(KP281/{gender}): {p_281:.2f}\n")
```

```
p_prod_given_gender('Male', True)
p_prod_given_gender('Female')
```

```
P(Male): 0.58
P(Female): 0.42

P(KP781/Male): 0.32
P(KP481/Male): 0.30
P(KP281/Male): 0.38

P(KP781/Female): 0.09
P(KP481/Female): 0.38
P(KP281/Female): 0.53
```

Probability of Each Product Given Marital Status:

- In this context, we aim to determine the likelihood of purchasing each product based on the customer's marital status

In [25]:
```
def p_prod_given_mstatus(status, print_marginal=False):
    if status is not "Single" and status is not "Partnered":
        return "Invalid marital status value."

    df1 = pd.crosstab(index=df['MaritalStatus'], columns=[df['Product']])
    p_781 = df1['KP781'][status] / df1.loc[status].sum()
    p_481 = df1['KP481'][status] / df1.loc[status].sum()
    p_281 = df1['KP281'][status] / df1.loc[status].sum()

    if print_marginal:
        print(f"P(Single): {df1.loc['Single'].sum()/len(df):.2f}")
        print(f"P(Partnered): {df1.loc['Partnered'].sum()/len(df):.2f}\n")

    print(f"P(KP781/{status}): {p_781:.2f}")
    print(f"P(KP481/{status}): {p_481:.2f}")
    print(f"P(KP281/{status}): {p_281:.2f}\n")

p_prod_given_mstatus('Single', True)
p_prod_given_mstatus('Partnered')
```

```
P(Single): 0.41
P(Partnered): 0.59

P(KP781/Single): 0.23
P(KP481/Single): 0.33
P(KP281/Single): 0.44

P(KP781/Partnered): 0.21
P(KP481/Partnered): 0.34
P(KP281/Partnered): 0.45
```

In [25]:
```
df
```

Out[25]:

	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
...	...	...	...	...	...	...	...	...	...
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

180 rows × 9 columns

Customer Profiles:

**For KP281 Treadmill:

- Most customers are between 18 to 35 years old, with a few between 35 to 50 years old.
- Education level tends to be 13 years or higher. -Annual income is typically below $60,000.
- Weekly usage ranges from 2 to 4 times.
- Fitness scale is usually between 2 to 4.
- Weekly running mileage falls between 50 to 100 miles.

**For KP481 Treadmill:

- Most customers are between 18 to 35 years old, with a few between 35 to 50 years old.
- Education level tends to be 13 years or higher.
- Annual income falls between USD 40000 to USD 80000.
- Weekly usage ranges from 2 to 4 times.
- Fitness scale is typically between 2 to 4.
- Weekly running mileage ranges from 50 to 200 miles.

**For KP781 Treadmill:

- Customers are mainly male.
- Age of customers typically falls between 18 to 35 years.
- Education level tends to be 15 years or higher.
- Annual income is $80,000 and above.
- Weekly usage ranges from 4 to 7 times.

- Fitness scale typically falls between 3 to 5.
- Weekly running mileage is usually 100 miles and above.

Recommendations:

**Targeted Marketing for KP781:

- Focus on advertising to women to balance out the sales gap.
- Offer special deals tailored to what women like to encourage their interest.

**Affordable Pricing and Payment Plans:

- Make sure the prices for KP281 and KP481 Treadmills are reasonable for the kind of people who would buy them.
- Allow payment in installments to make it easier for people with different budgets to buy.

**User-Friendly App Integration:

- Create a phone app that works smoothly with the treadmills.
- The app should track how much you run each week, give you feedback, and suggest workouts that fit your fitness level and goals.
- Include fun features like challenges and rewards to keep you motivated.

**Community Building and Support:

- Set up online groups where people who use the treadmills can chat, share tips, and help each other reach their fitness goals.
- Provide helpful info on workouts, nutrition, and how to use the treadmill effectively.

**Regular Maintenance and Help:

- Offer to fix the treadmills if something goes wrong and answer any questions people have about them.
- Give clear instructions on setting up, using, and fixing any issues with the treadmill.

**Improving the Product:

- Listen to what customers say about the KP781 model through surveys and reviews.
- Use this feedback to make the treadmill even better by adding new features and technologies that people want.

**Partnerships and Collaborations:

- Work with fitness experts and gyms to get the word out about the treadmill.
- Get endorsements from doctors and trainers to show that the KP781 Treadmill is good for reaching fitness goals.

**Being Eco-Friendly:

- Make the treadmill in a way that's kind to the environment by using recycled materials and energy-saving parts.
- Tell people about these efforts in ads and on the packaging to show that you care about the planet.