Business Case: Aerofit - Descriptive Statistics & Probability

AeroFit is a leading brand in the field of fitness equipment. AeroFit provides a product range including machines such as treadmills, exercise bikes, gym equipment, and fitness accessories to cater to the needs of all categories of people.

- **Business Problem: -**

The market research team at AeroFit wants to identify the characteristics of the target audience for each type of treadmill offered by the company, to provide a better recommendation of the treadmills to the new customers. The team decides to investigate whether there are differences across the product with respect to customer characteristics.

1. Perform descriptive analytics **to create a customer profile** for each AeroFit treadmill product by developing appropriate tables and charts.
2. For each AeroFit treadmill product, construct **two-way contingency tables** and compute all **conditional and marginal probabilities** along with their insights/impact on the business.

- **Importing the libraries we need: -**

Import numpy as np
Import pandas as pd
Import matplotlib.pyplot as plt
Import seaborn as sns

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

- **Loading The Data Set**: -

So, using Pandas Library we will load the csv file. Named it as the > df for the data set.

```
import pandas as pd
df = pd.read_csv("/aerofit_treadmill.csv")
df
```

	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

- Checking the shape of the data frame: -

```
10] df.shape

    (180, 9)
```

So, we have found that the data consists of 180 rows along with 9 columns.

- About the Information: -

```
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
```

So, from the information about the data, we found that it is a pandas data frame & it started from 0 & ends at 179, Which have 9 columns & in the last it shows about the data types.

- Description of the Data in the DataFrame: --

```
df.describe(include="all")
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

So, to Calculate descriptive statistics for every column in the DataFrame, we can use include all argument which generate descriptive statistics for all the columns.

- Checking the Missing or Null values: -

```
print("Columns 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
```

So isnull() the isnull() functions in pandas is a convenient method to detect missing or null values with a DataFrame or Series.

- Observations: -

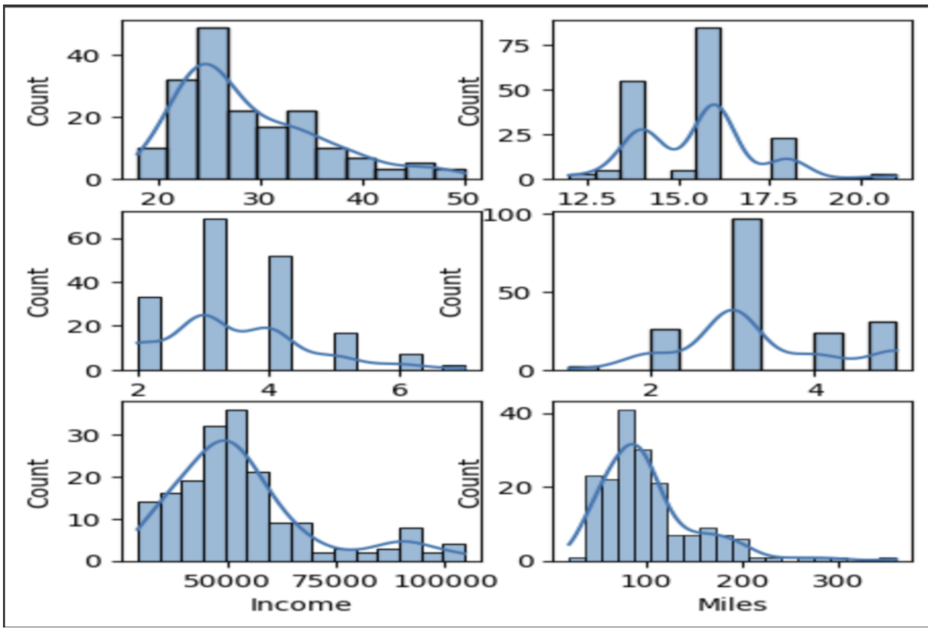
1. There are no missing values in the data.
2. There are 3 Unique products in the data set.
3. **KP281** is the most frequent product.
4. Minimum & Maximum age of the of the person is 18 & 50. Mean is 28.79 & 75% of persons have the age less than or equal to 33.
5. Most of the people are having 16 years of education i.e. 75% of persons are having the education <= 16 years.

6. Out of 180 Data Points, 104's gender is Male & rest are the Female.
7. Standard Deviation for Income & Miles are very high. This variables might have the outliers in it.

- **Univariate Analysis: --**

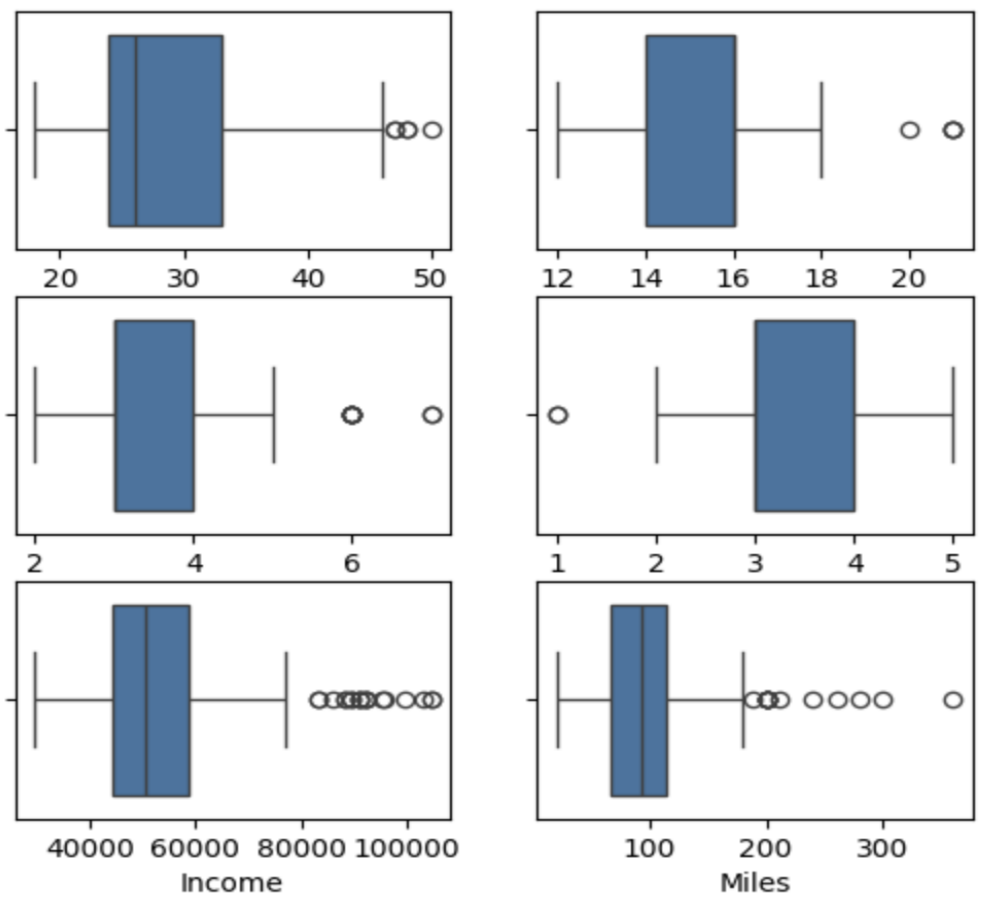
    1. Age
    2. Education
    3. Usage
    4. Fitness
    5. Income
    6. Miles

```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(5, 4))
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()
```



- **Now Outliers detection using the Box plot: -**

```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(6, 5))
fig.subplots_adjust(top=1.0)
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()
```



<u>Observations: -</u>

So even from the Box plots it is quite clear that: -

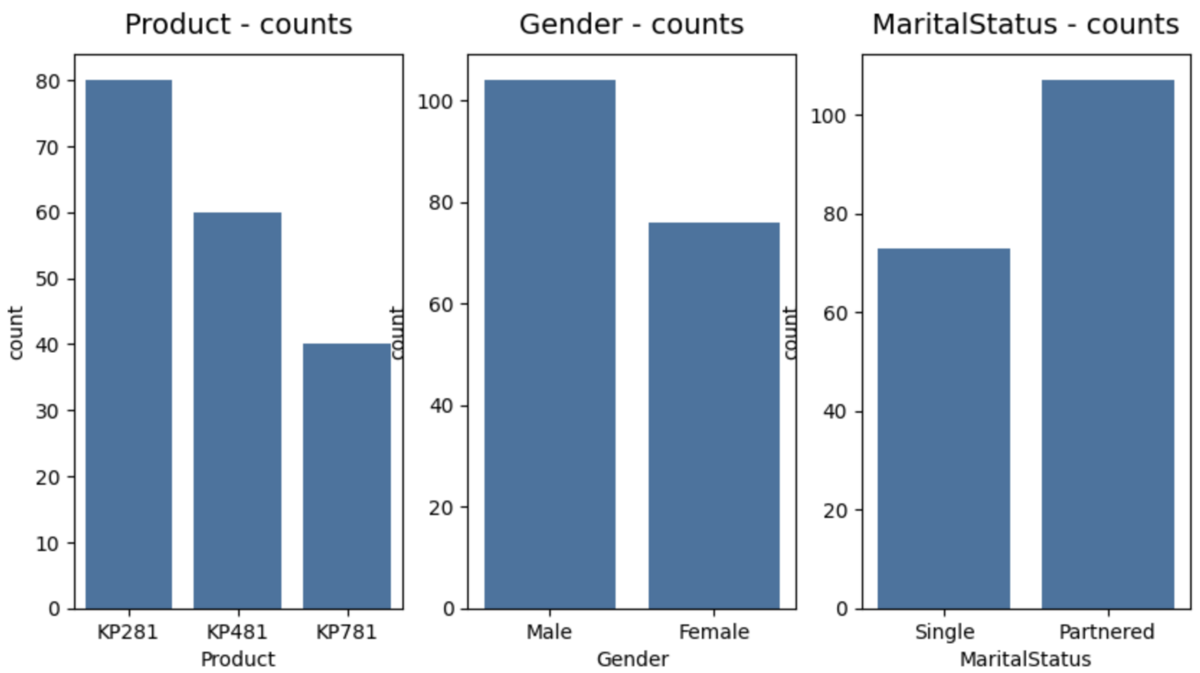
- Age, Educations & the Usage are having very few outliers.
  - While Income & Miles are having more outliers.

- **Understanding The Distribution of The Data for Qualitative Attributes: ---**

1) Product
2) Gender
3) Marital Status

```
fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(10,5))
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: --

- KP281 is the most frequent product.
- There are more males in the data then females.
- More Partnered persons are there in the data.

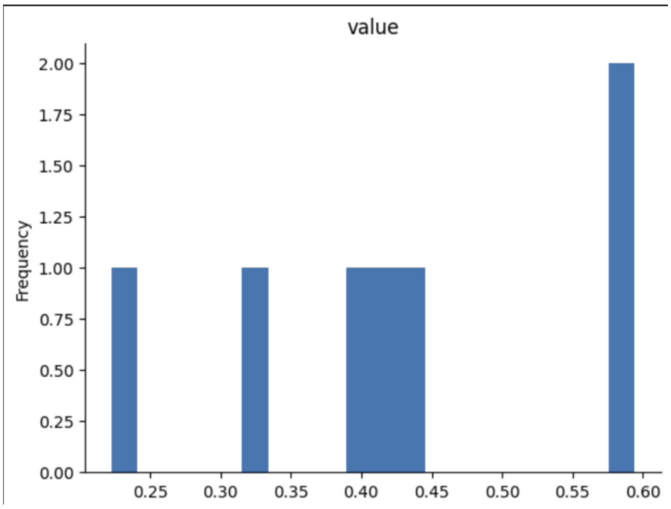
Precisely – Normalized count for each variable is shown below: --

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

		value
variable	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

Graphical Presentation: --

```
from matplotlib import pyplot as plt
_df_0['value'].plot(kind='hist', bins=20, title='value')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



Observations: --

1) **Product**
   - 44.44 % of the customers have purchased - KP2821
   - 33.33 % of the customers have purchased - KP481
   - 22.22 % of the customers have purchased – KP781

2) **Gender**

- 57.78 % of the customers are the Males.

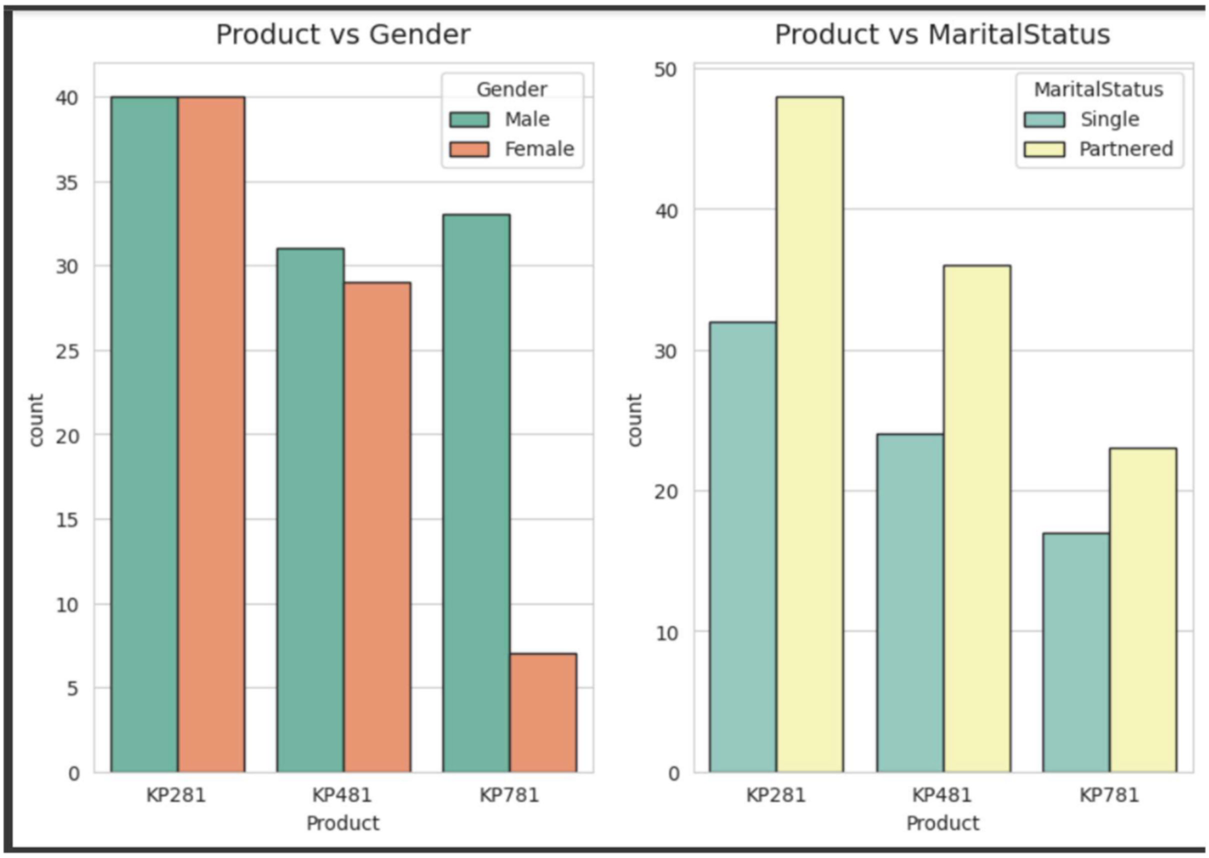
3) **Marital Status**

- 59.44 % of the customers are partnered.

Bivariate Analysis: --

Checking if the features – Gender of Marital Status have any effect on the Product Purchased.

```
sns.set_style(style='whitegrid')
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(10, 7))
sns.countplot(data=df, x='Product', hue='Gender', edgecolor="0.15",
palette='Set2', ax=axs[0])
sns.countplot(data=df, x='Product', hue='MaritalStatus',
edgecolor="0.15", palette='Set3', ax=axs[1])
axs[0].set_title("Product vs Gender", pad=10, fontsize=14)
axs[1].set_title("Product vs MaritalStatus", pad=10, fontsize=14)
plt.show()
```



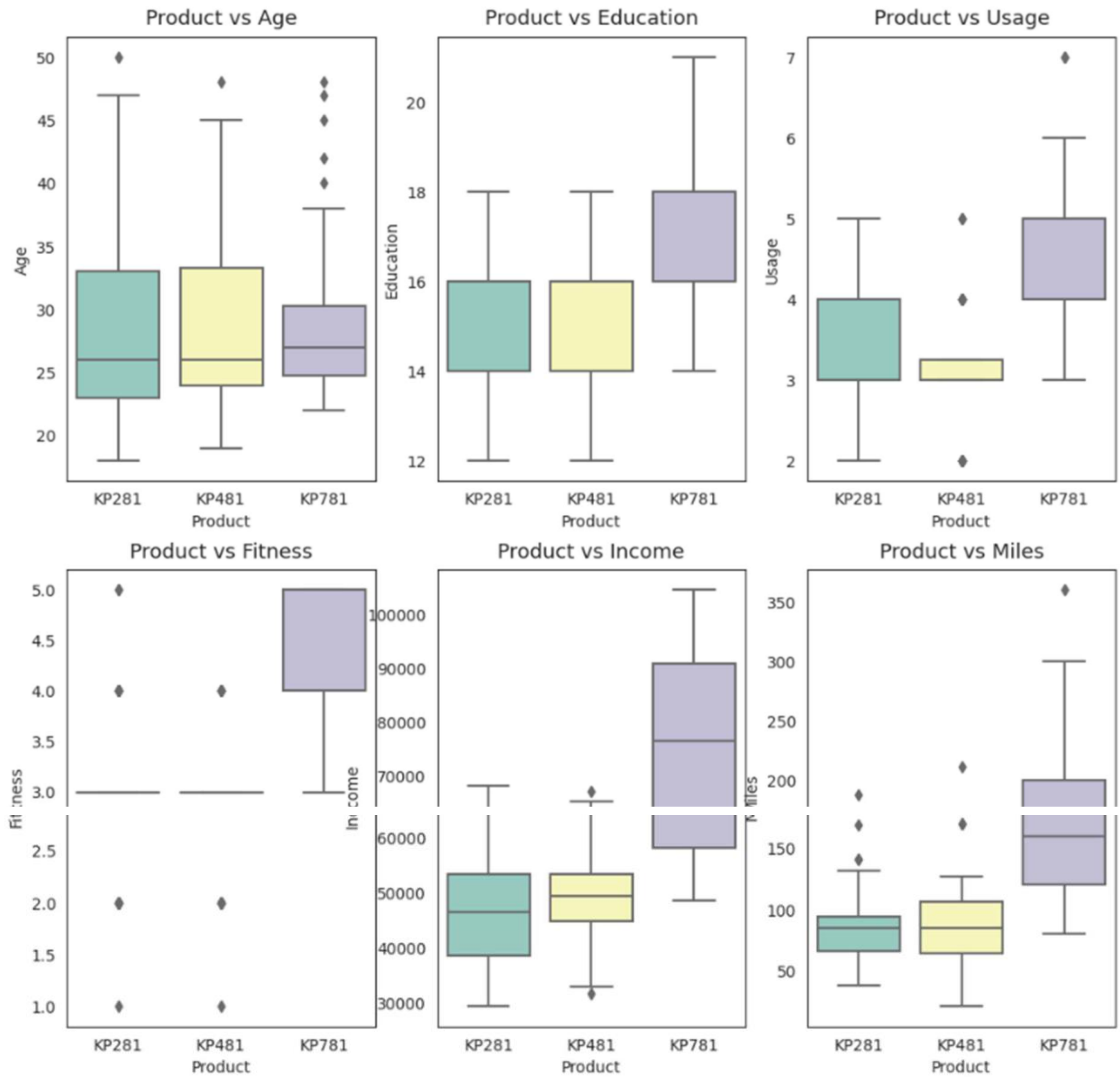
Observations: ---

- **Product VS Gender**
  I. Equal number of males & females have purchased KP281 & almost same for the product KP481.
  II. Most of the male customers have purchased the KP781 product.
- **Product VS Marital Status**
  I. Customers who are Partnered, are more likely to purchase the product.

Checking If the Following Features having any effect While Purchasing the Product: ----

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

```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income','Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(12, 8))
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=8, fontsize=13)
count += 1
```



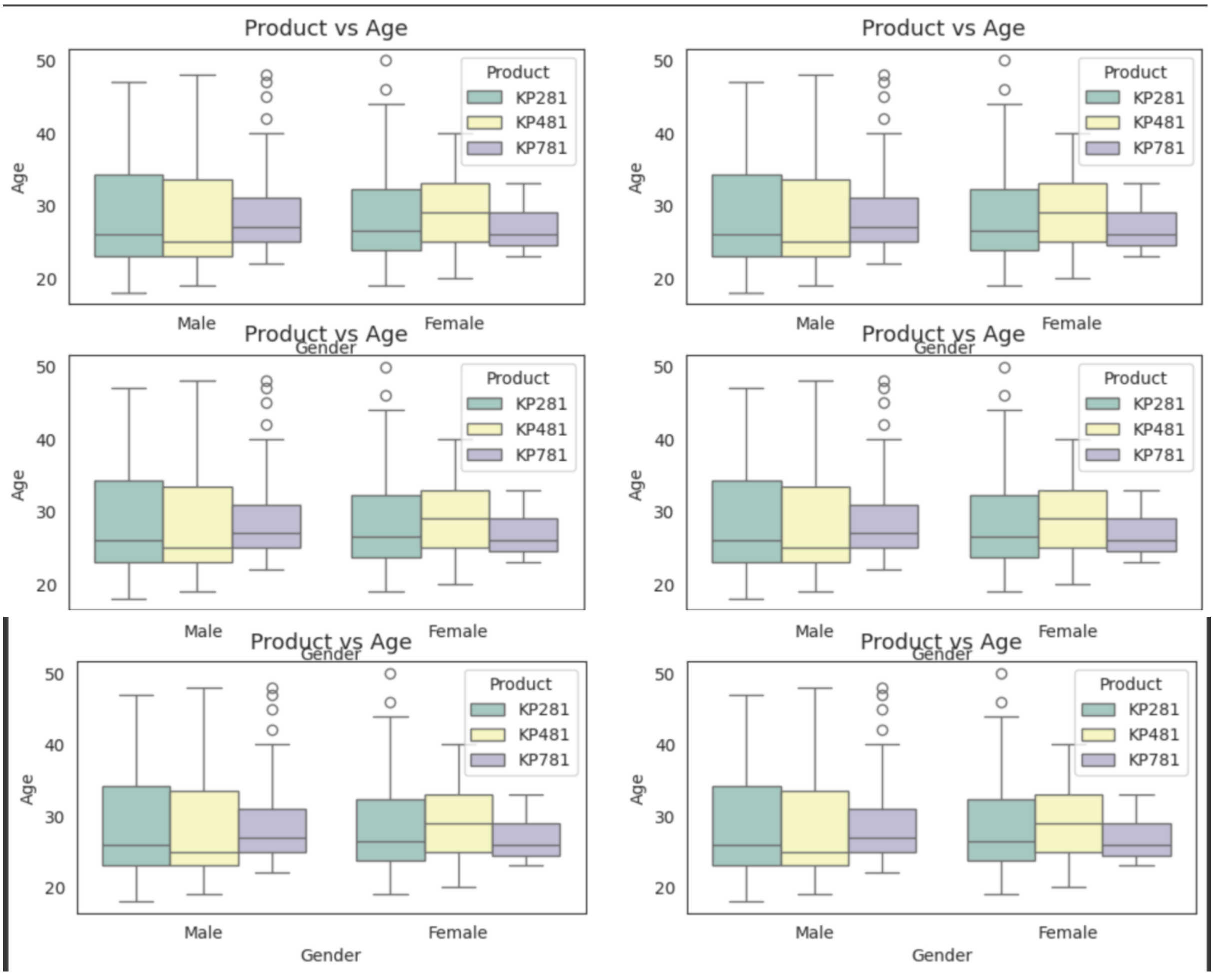
Observations: ---

1. **Product VS Age**
   - Customers who are purchasing Products KP281 & KP481 are having same age median values.
   - Customers whose age lies between 25 – 30, are more likely to buy KP781 product.
2. **Product VS Education**
   - **Customers whose education is greater than 16 have more chances to purchase the KP781 Product.**
   - **While the customers with the education less than 16 have equal chances of purchasing the products of KP281 or KP481.**

3. **Product VS Usages: ---**
    - Customers who are planning to use the Treadmill greater than 4 times a week are more likely to purchase the KP781 Product.
    - While the other customers are likely to purchase KP281 & KP481.
4. **Product VS Fitness: ---**
    - The more the customers are fit (fitness >= 3), higher the chances of the customers to purchases the KP781 Product.
5. **Product VS Income: ----**
    - Higher the Income of the customer (Income>=60000), higher the chances of the customer to purchase the KP781 Product.
6. **Product VS Miles: ---**
    - If the customer expects to walk / run greater than 120 miles per week, It is more likely that the customer will buy KP781 Product.

Multi – Variate Analysis: ----

```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=3, ncols=2, figsize=(12, 8))
fig.subplots_adjust(top=1)
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], palette='Set3')
    axs[i,j].set_title(f"Product vs {attrs[count]}", pad=8,fontsize=13)
count += 1
```



Conditional Probability: ---

Probability of Each Product given Gender

```
def p_prod_given_gender(gender, print_marginal=False):
  if gender is not "Female" and gender is not "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')
```

```
if gender is not 'Female' and gender is not 'Male':
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
```

Business Insights & Recommendations: ----

- As AeroFit is the leading company in the field of fitness equipment’s, it has been observed that KP2821 is the most frequent product, after that KP481 & KP781 respectively.
- Customers who are associated with us It has been observed that (57.78, 59.44) % of them are males & partnered respectively.
- We also found that equal number of males & females has purchased the KP281 product & almost same for the product KP481 but males preferred KP781.
- We also observed some features like – Age, Education, Usages, Fitness, Income, Miles have some impact which purchasing the product.
- We have found While checking gender wise Probability for each product that P (0.58) are the males while remaining P (0.42) are the females, as far as our product are concerned.
- Finally, we found while checking the gender wise conditional probability that KP281 is the product which stands highest for both males & Females.
- As far as the higher income of the Customers are concerned, we found that KP781 is the product which has the positive correlations with the Income of the customers.
- Following this information’s AeroFit should look on their growth perspectives