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.
- 1. 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.

In [1]:
```
# Importing necessary python libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Loading dataset
df = pd.read_csv('/Users/bose/Downloads/aerofit.csv')
```

In [377…
```
df.head()
```

Out[377]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	Good	29562	112
1	KP281	19	Male	15	Single	2	Average	31836	75
2	KP281	19	Female	14	Partnered	4	Average	30699	66
3	KP281	19	Male	12	Single	3	Average	32973	85
4	KP281	20	Male	13	Partnered	4	Poor	35247	47

Observation of basic metrics, shape of data, datatype of attributes -

In [ ]:
```
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
```

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

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

There are **180 rows** and **9 columns** in the dataset

In [250…
```
# Checking for null values
df.isna().sum().sum()
```

Out[250]:
```
0
```

There are **No Null values** in the dataset

In [12]:
```
# Datatype of attributes
df.dtypes
```

Out[12]:
```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

Statistical Summary -

In [19]:
```
df.describe()
```

Out[19]:

	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 [21]:

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

Out[21]:

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

Converting to Category -

In [383...

```
df2 = df
df2.head()
```

Out[383]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	Moderate	Good	29562	112
1	KP281	19	Male	15	Single	Low	Average	31836	75
2	KP281	19	Female	14	Partnered	Moderate	Average	30699	66
3	KP281	19	Male	12	Single	Moderate	Average	32973	85
4	KP281	20	Male	13	Partnered	Moderate	Poor	35247	47

In [384...

```
df2["Fitness"].replace({1:"Very Poor",
                        2:"Poor",
                        3:"Average",
                        4:"Good",
                        5:"Excellent"},inplace=True)
df2.head()
```

Out[384]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	Moderate	Good	29562	112
1	KP281	19	Male	15	Single	Low	Average	31836	75
2	KP281	19	Female	14	Partnered	Moderate	Average	30699	66
3	KP281	19	Male	12	Single	Moderate	Average	32973	85
4	KP281	20	Male	13	Partnered	Moderate	Poor	35247	47

In [386…

```
df2["Usage"].replace({1:"Low",
                      2:"Low",
                      3:"Normal",
                      4:"Normal",
                      5:"Normal",
                      6:"High",
                      7:"High"},inplace=True)
df2.head()
```

Out[386]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	Normal	Good	29562	112
1	KP281	19	Male	15	Single	Low	Average	31836	75
2	KP281	19	Female	14	Partnered	Normal	Average	30699	66
3	KP281	19	Male	12	Single	Normal	Average	32973	85
4	KP281	20	Male	13	Partnered	Normal	Poor	35247	47

Value Counts and Unique Values -

In [321…

```
# Number of Unique values in each column
for i in df.columns:
  print(i,':',df[i].nunique())
```

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

Product column -

In [ ]:

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

Out[ ]:

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

KP281 is the most poular product. Followed by KP481 and then KP781 in that order

In [345…

```
print("Unique attributes in Product column :",df['Product'].unique())
```

```
Unique attributes in Product column : ['KP281' 'KP481' 'KP781']
```

Gender Column -

In [355…

```
df['Gender'].value_counts()
```

Out[355]:

```
Male      104
Female     76
Name: Gender, dtype: int64
```

- There are more **Male** customers compared to **Female**
- **Male** - 104

- **Female** - 76

In [356…
```
print("Unique attributes in Gender column :",df['Gender'].unique())
```
```
Unique attributes in Gender column : ['Male' 'Female']
```

MaritalStatus Column -

In [358…
```
df['MaritalStatus'].value_counts()
```
Out[358]:
```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

- There are more **Partnered** customers compared to **Single** customers
- **Partnered** - 107
- **Single** - 73

In [359…
```
print("Unique attributes in MaritalStatus column :",df['MaritalStatus'].uniq
```
```
Unique attributes in MaritalStatus column : ['Single' 'Partnered']
```

Fitness Column -

In [387…
```
df2['Fitness'].value_counts()
```
Out[387]:
```
Average      97
Excellent    31
Poor         26
Good         24
Very Poor     2
Name: Fitness, dtype: int64
```

- Most customers **(97)** rate themselves **Average** in terms of fitness (fitness rating -3)
- **31** customers are in **Excellent** shape(fitness rating -5)

In [389…
```
print("Unique attributes in Fitness column :",df2['Fitness'].unique())
```
```
Unique attributes in Fitness column : ['Good' 'Average' 'Poor' 'Very Poor'
'Excellent']
```

Usage Column -

In [388…
```
df2['Usage'].value_counts()
```
Out[388]:
```
Normal    138
Low        33
High        9
Name: Usage, dtype: int64
```

- Most customers(138) have a **Normal usage** of their threadmill (3-5 days per week)
- 33 customers have **Low usage** (1-2 days a week)

In [390…
```
print("Unique attributes in Usage column :",df['Usage'].unique())
```
```
Unique attributes in Usage column : ['Normal' 'Low' 'High']
```

Visual Analysis

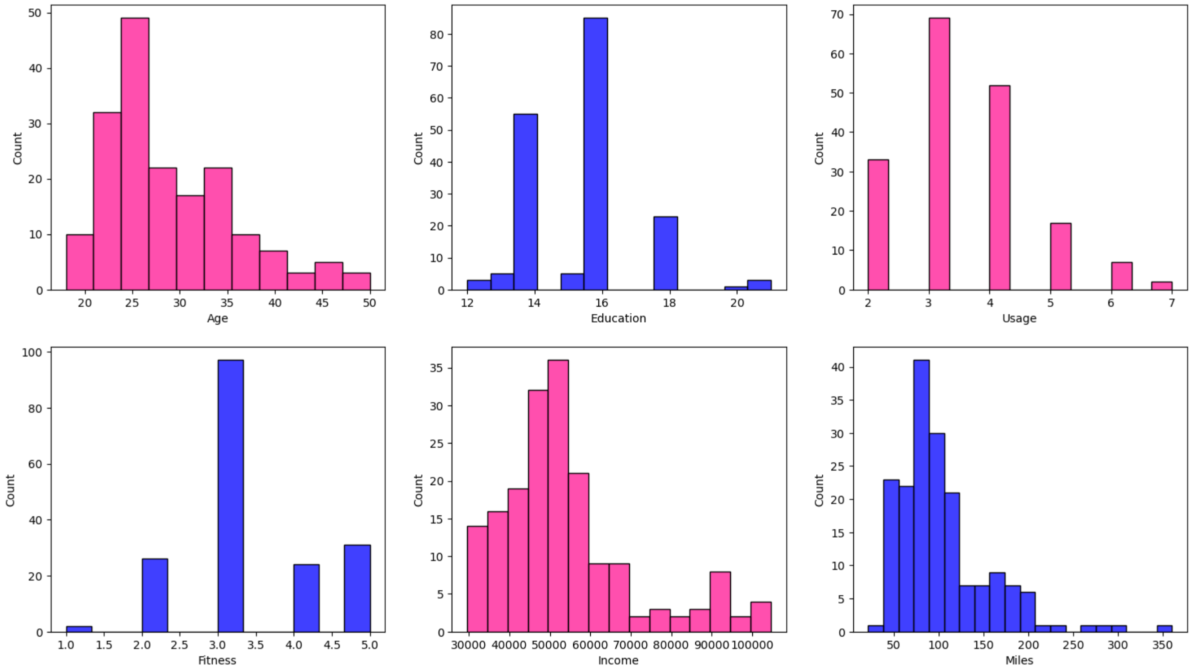
Distribution of data -

For Continuous Variables -

In [322…

```
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(18, 10))

sns.histplot(data=df, x="Age", ax=axis[0,0], color='deeppink')
sns.histplot(data=df, x="Education", ax=axis[0,1],color='blue')
sns.histplot(data=df, x="Usage", ax=axis[0,2],color='deeppink')
sns.histplot(data=df, x="Fitness", ax=axis[1,0],color='blue')
sns.histplot(data=df, x="Income", ax=axis[1,1],color='deeppink')
sns.histplot(data=df, x="Miles", ax=axis[1,2],color='blue')
plt.show()
```



Insights -

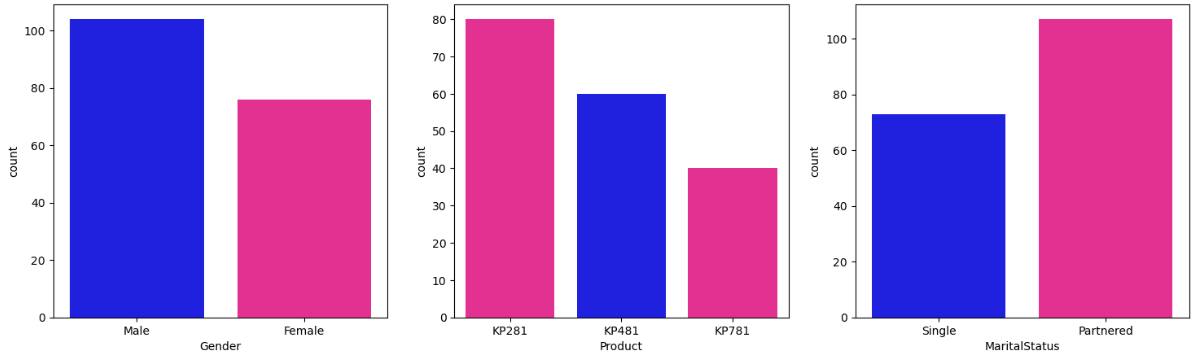
- Median Value for **Age** of customers is **26** years
- Median Value for **Education** is **16** years
- Median Value for **Usage** is **3** days a week
- Median Value for **Fitness** is **3**
- Median Value of **Income** comes around **54,000**
- Median Value for **Miles** covered is **90** miles

For Categorical Variables -

In [323…

```
fig, axis = plt.subplots(nrows=1, ncols=3, figsize=(18, 5))

sns.countplot(data=df, x="Gender", ax=axis[0], palette=['blue','deeppink'])
sns.countplot(data=df, x="MaritalStatus", ax=axis[2],palette=['blue','deeppi
sns.countplot(data=df, x="Product", ax=axis[1],palette=['deeppink','blue'])
plt.show()
```



Insights -

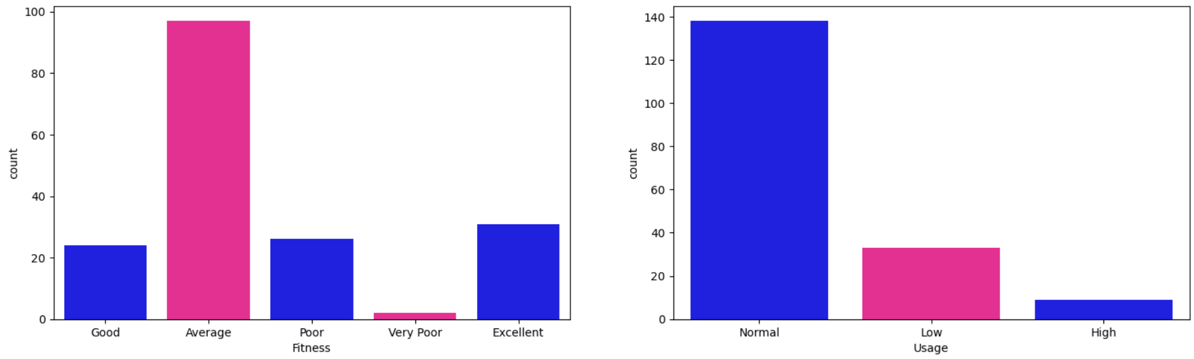
- **Male** customers are **more in number than Female** customers
- **KP281 is the most popular product, followed by KP481 and KP781** in that order
- **Partnered** customers tend to buy threadmill **more than Single** customers

For Newly created categories -

In [420…

```
fig, axis = plt.subplots(nrows=1, ncols=2, figsize=(18, 5))

sns.countplot(data=df2, x="Fitness", ax=axis[0], palette=['blue','deeppink']
sns.countplot(data=df2, x="Usage", ax=axis[1],palette=['blue','deeppink'])
plt.show()
```



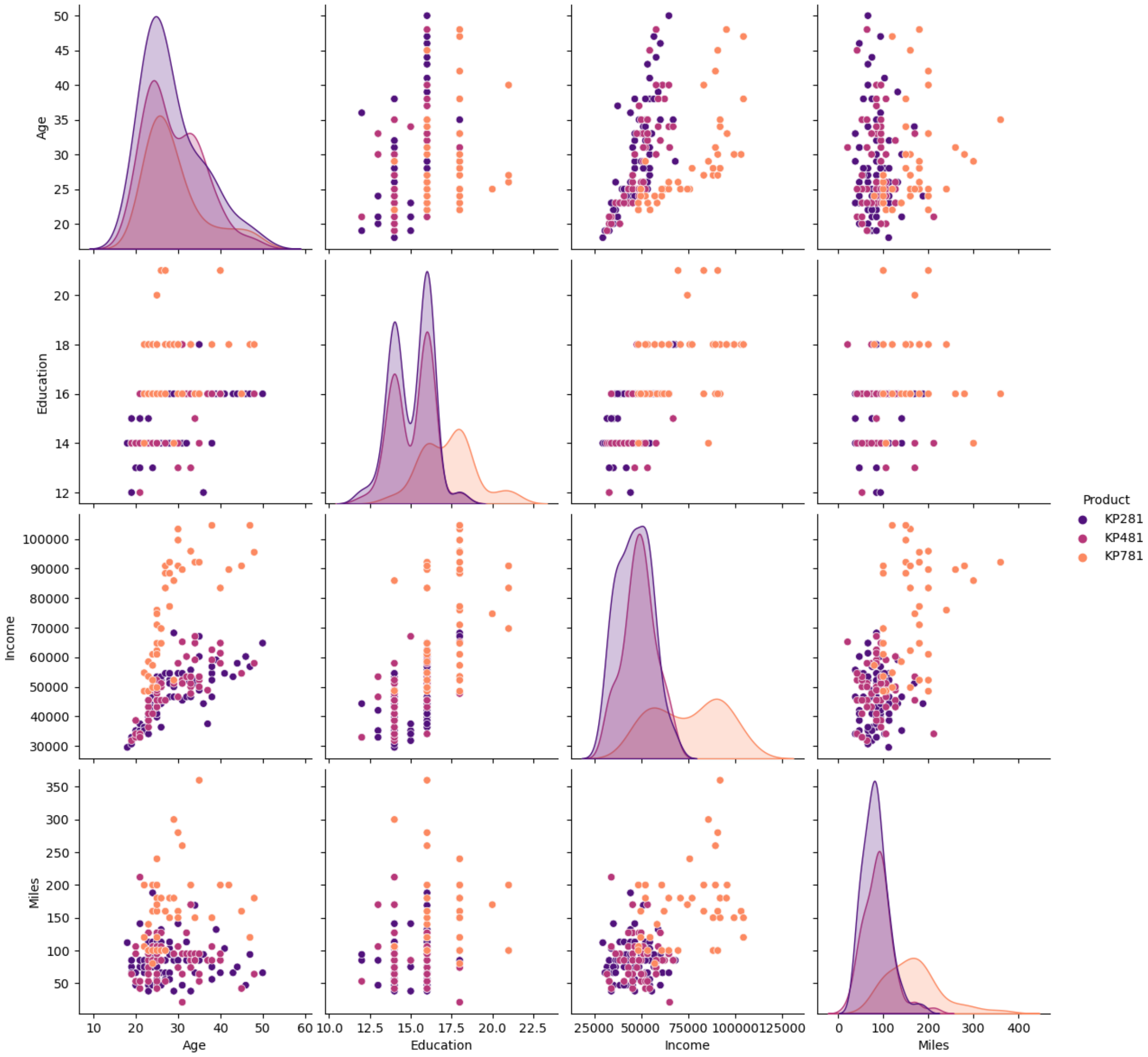
Insights -

- Most of the customers come in the **Average**(Fitness Rating = 3) category for **Fitness**
- Most of the customers have **Normal Usage** (3-5 days per week) of their Threadmill

Correlation among different attributes -

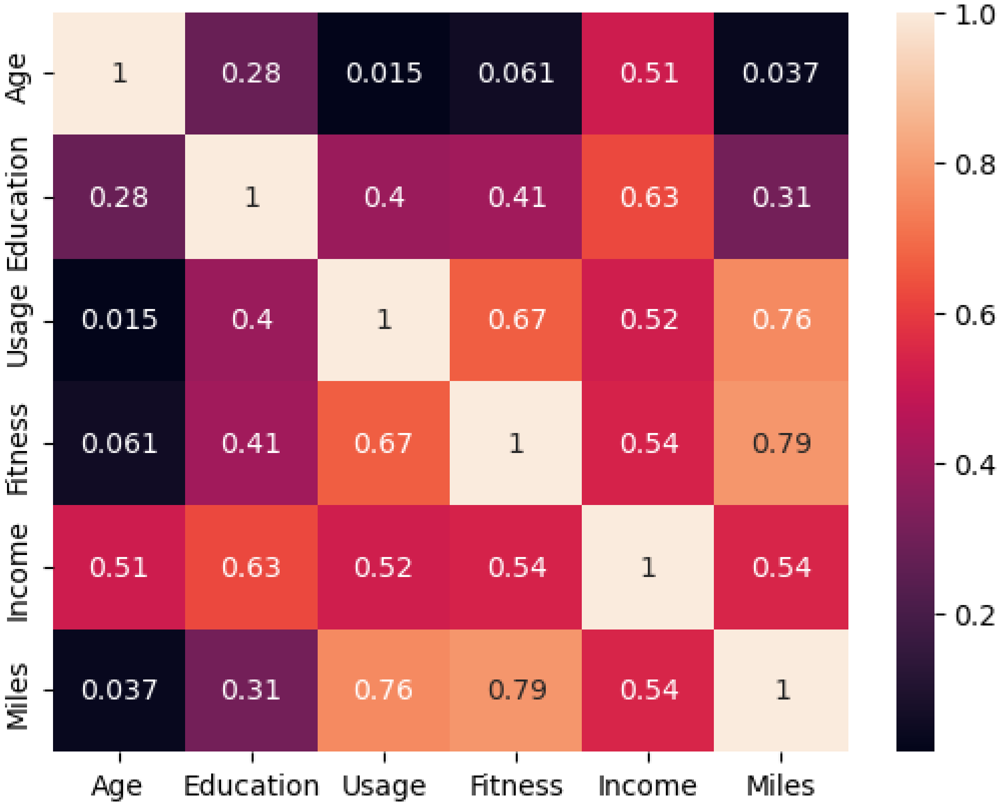
In [395…

```
sns.pairplot(data=df, hue='Product', palette= 'magma', height=3)
plt.show()
```



In [339…

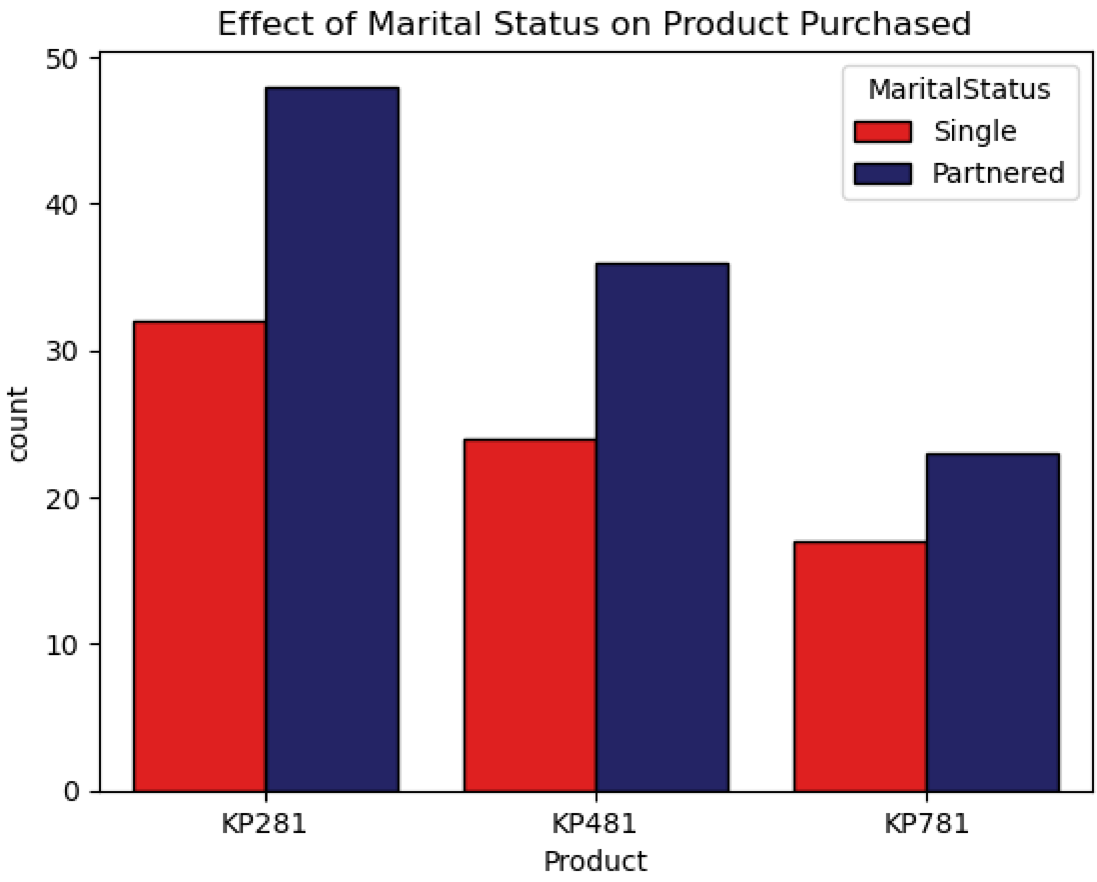
```
sns.heatmap(df.corr(), annot=True)
plt.show()
```



Inference -

- Fitness is strongly correlated to Miles (correlation factor of 0.79)
- Fitness is strongly correlated to Usage (correlation factor 0.67)
- Usage is strongly correlated to Miles (correlation factor 0.76)
- Age has no strong correlation with any factors except for Income (factor 0.51)
- Age is weakly correlated to Fitness and Usage(0.061 & 0.015)

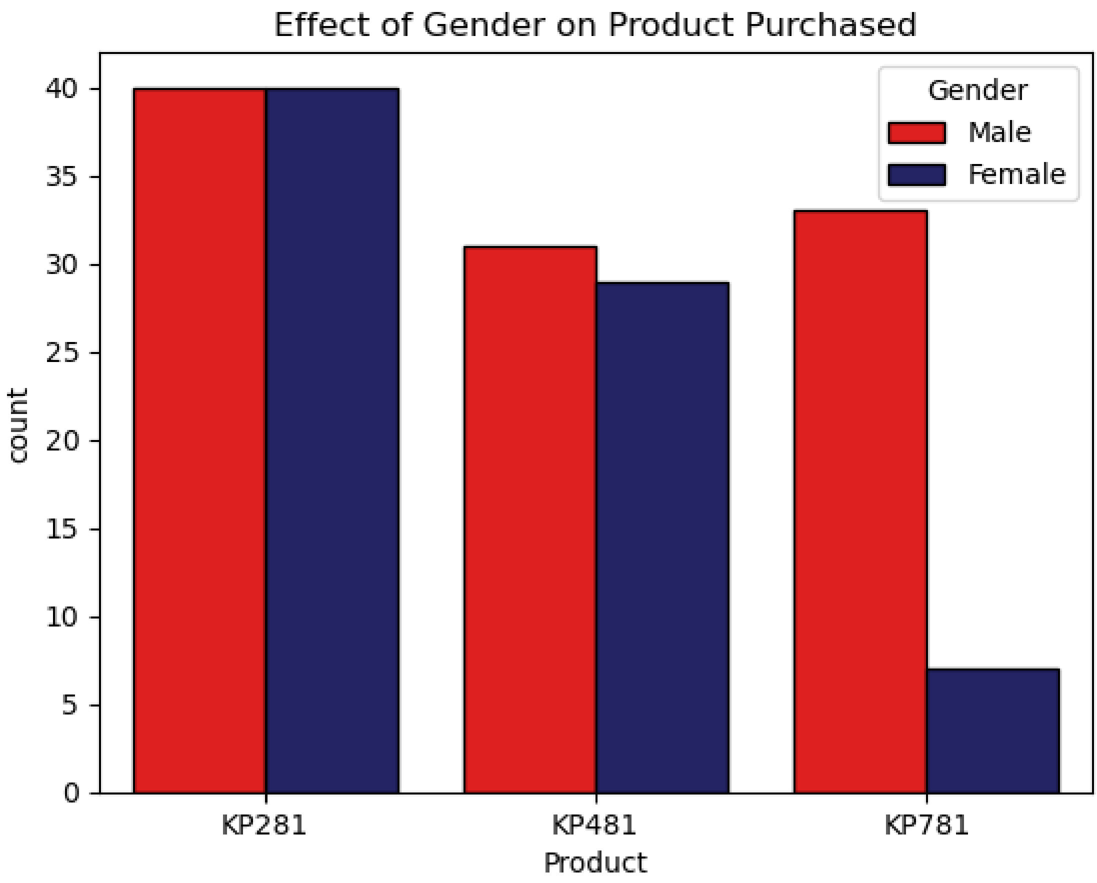
In [251…
```
sns.countplot(x='Product', hue='MaritalStatus', data=df, palette =['red','mi
plt.title('Effect of Marital Status on Product Purchased')
plt.show()
```



Inference -

- People having Marital Status of **Partnered purchase more Threadmills compared to people who are Single**
- It is true in the case of all 3 product categories(KP281, KP481, KP781)

In [396…
```
sns.countplot(x='Product', hue='Gender', data=df, palette =['red','midnightb
plt.title('Effect of Gender on Product Purchased')
plt.show()
```

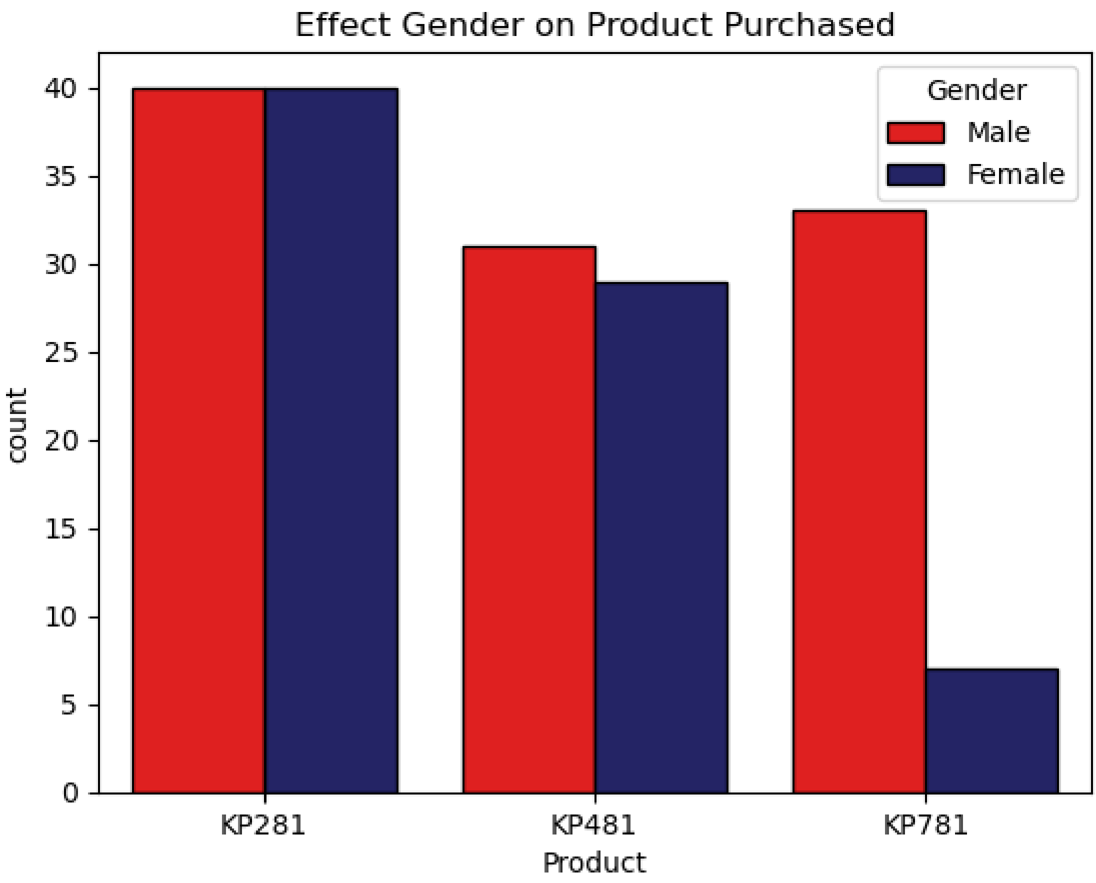


Inference -

- KP281 - Both **male and female customers equally prefer** KP281
- KP481 - There is **slightly more number of Male**customers compared to Female
- KP781 - **Most of the customers** using this product are **Male**

In [279…

```
sns.countplot(x='Product', hue='Gender', data=df, palette =['red','midnightb
plt.title('Effect Gender on Product Purchased')
plt.show()
```



Inference -

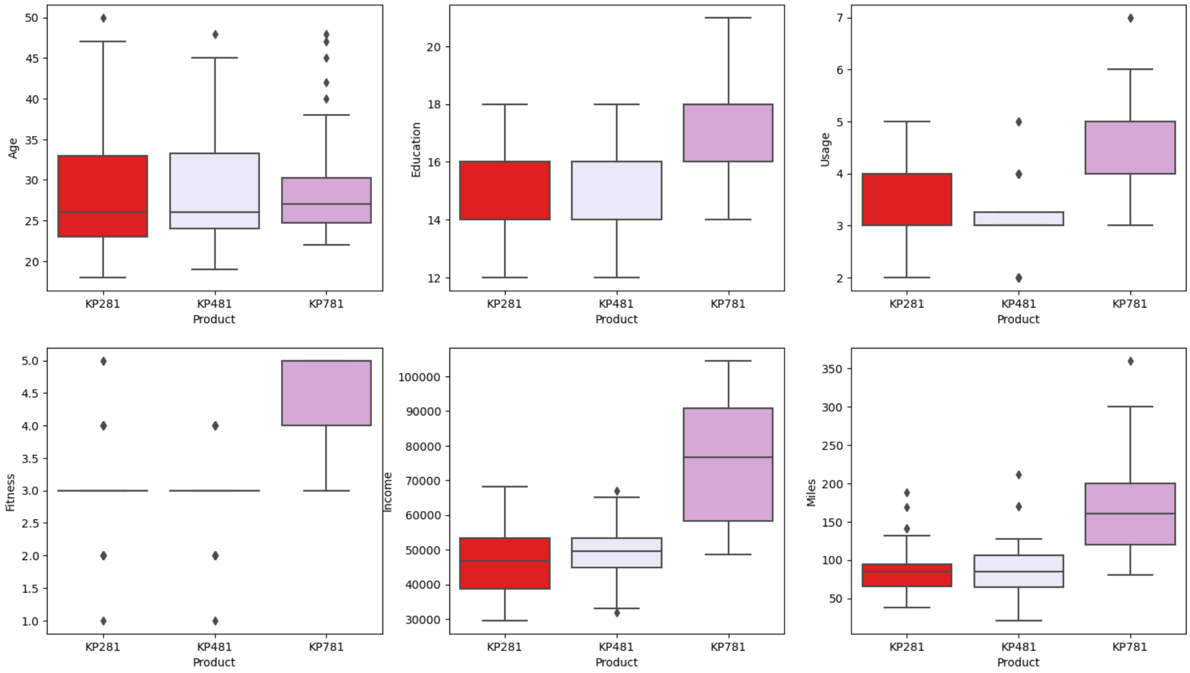
- **KP281 -** Equal number of Males and Females purachased this product
- **KP481 -** This category has a slightly more number of Male customers compared to Female
- **KP781 -** Very few number of Female customers compared to Male in this category

Effect of Quantitative attributes on Product Purchased -

In [178…

```
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(18, 10))

sns.boxplot(x="Product", y="Age", data=df, ax=axis[0,0], palette=['red','lav
sns.boxplot(data=df, x="Product" ,y="Education", ax=axis[0,1], palette=['rec
sns.boxplot(data=df, x="Product", y="Usage", ax=axis[0,2],palette=['red','la
sns.boxplot(data=df, x="Product", y="Fitness", ax=axis[1,0],palette=['red','
sns.boxplot(data=df, x="Product", y="Income", ax=axis[1,1],palette=['red','l
sns.boxplot(data=df, x="Product", y="Miles", ax=axis[1,2],palette=['red','la
plt.show()
```



Effect of Quantitative attributes on Product Purchased -

- *Age* :
  - Cutomers buying products KP281 and KP481 have the same median age of 27 and in the range (23 to 34) years
  - Customers buying KP781 have a **higher median age** of 28 years and fall in the range of (25 to 30) years
- *Education* :
  - Cutomers of **KP281 and KP481** have an education in the range of **14 to 16 years**
  - Customers of **KP781** have an higher education in the range of **16 to 18 years**
- *Usage* :
  - **KP281** customers use it for **(3 to 4) times a week**
  - **KP481** customers use it for **3 days per week**
  - **KP781** customers use their threadmill more times a week, around **4 to 5 times per week**
- *Fitness* :
  - More **fit customers** (rating of 4 to 5) tend to **purchase KP781**
- *Income* :
  - **Low income customers prefer KP281 and KP481** (KP281 - 40k to 54k) (KP481 - 45k to 54k)
  - People with **High Income** tend to **purchase KP781** more than the other models (KP781 - 60k to 90k)
- *Miles* :
  - **KP281 and KP481** customers cover around **60 to 110 miles a week**
  - **KP781** customers cover around **120 to 200 miles a week**

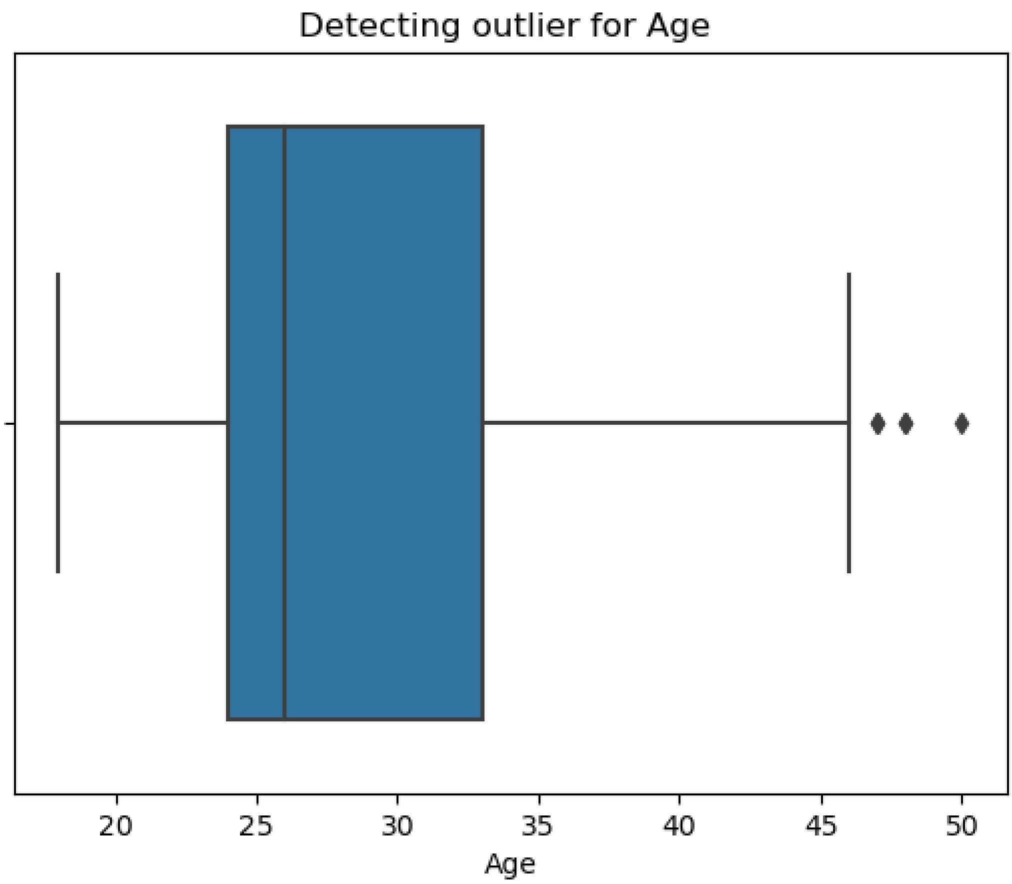
In [422…

```
# Check for missing or null values
df.isna().sum()
```

Out[422]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

As you can see there is **No Missing or Null values** in the dataset

In [266…
```
sns.boxplot(x='Age',data=df)
plt.title('Detecting outlier for Age ')
plt.show()
print('Median is ',df['Age'].median())
print('Mean is ',df['Age'].mean())
print('Difference between mean and median is 2.78')
```



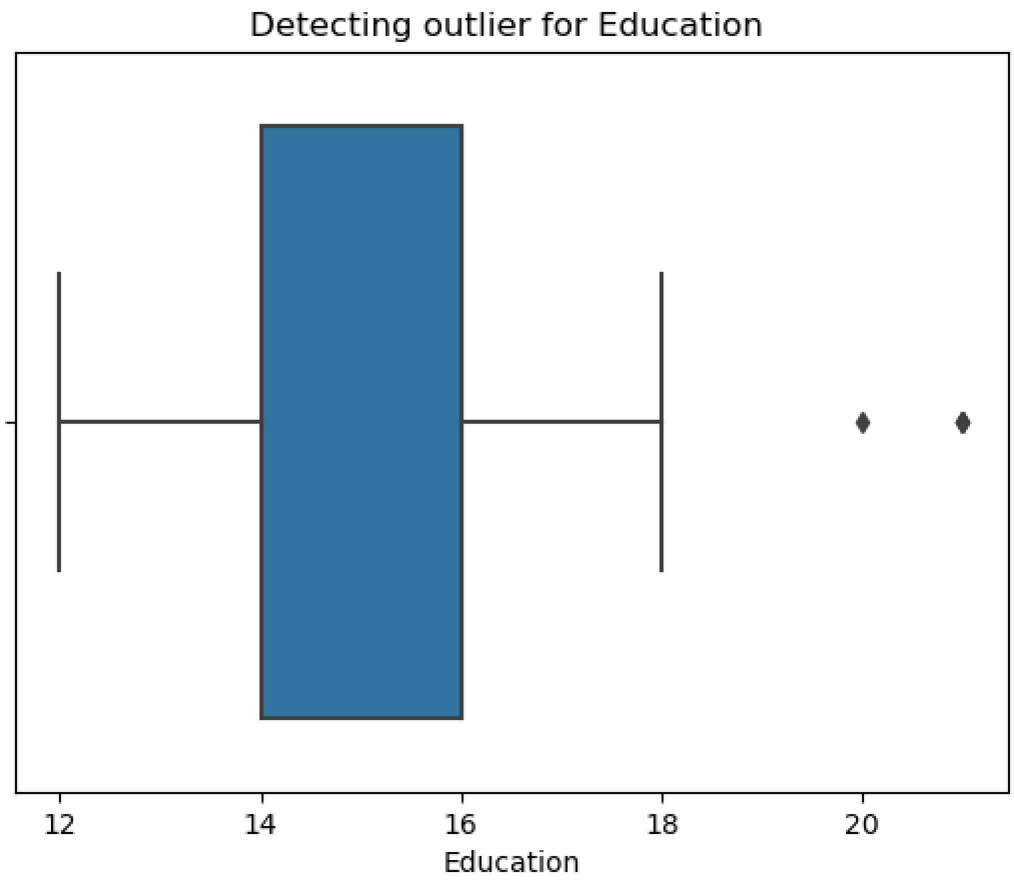

```
Median is  26.0
Mean is  28.788888888888888
Difference between mean and median is 2.78
```

Inference -

- Majority of the customer's **Age** fall in the range **24 to 33**
- There are **3 outliers** above the age of **45**
- Differnce between mean and median is **2.78**

In [268…
```
sns.boxplot(x='Education',data=df)
plt.title('Detecting outlier for Education ')
plt.show()
print('Median is ',df['Education'].median())
```

```
print('Mean is ',df['Education'].mean())
print('Difference between mean and median is 0.427')
```



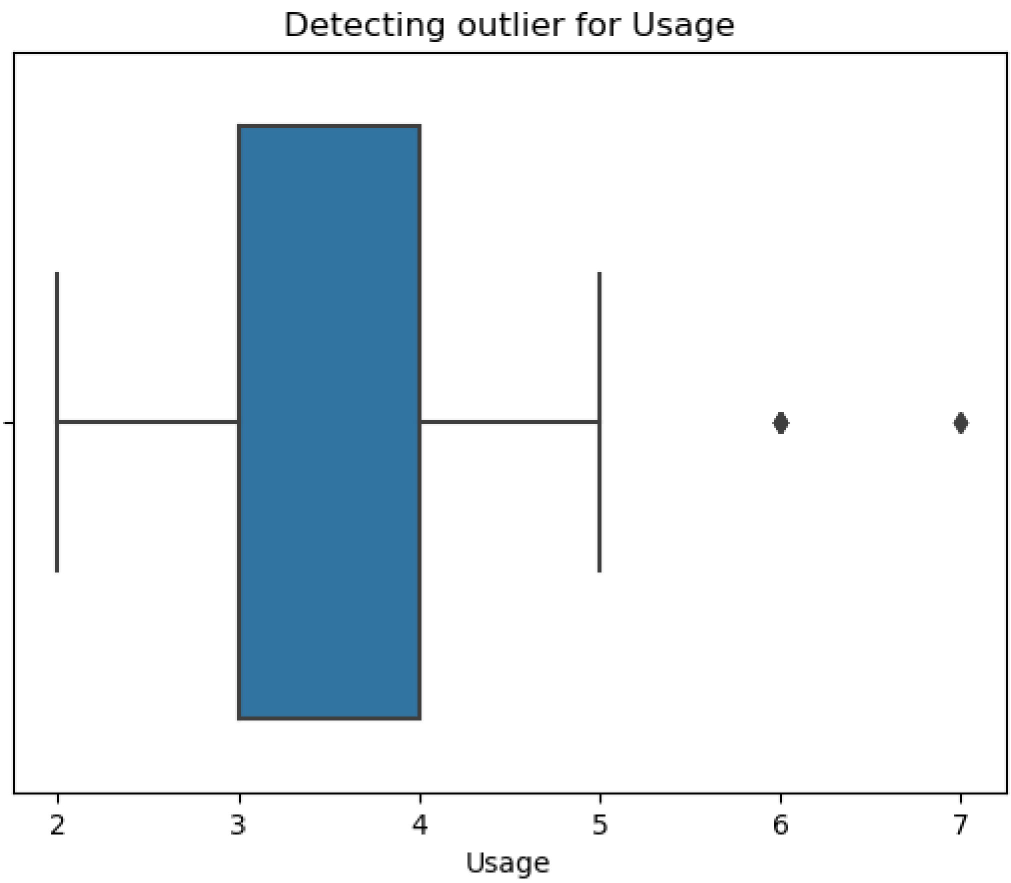

```
Median is  16.0
Mean is  15.572222222222223
Difference between mean and median is 0.427
```

Inference -

- Majority of the customer's **Education** fall in the range **14 to 16 years**
- There are **2 outliers** above the value of **18 years**
- Differnce between mean and median is **0.427**

In [271…

```
sns.boxplot(x='Usage',data=df)
plt.title('Detecting outlier for Usage')
plt.show()
print('Median is ',df['Usage'].median())
print('Mean is ',df['Usage'].mean())
print('Difference between mean and median is 0.455')
```



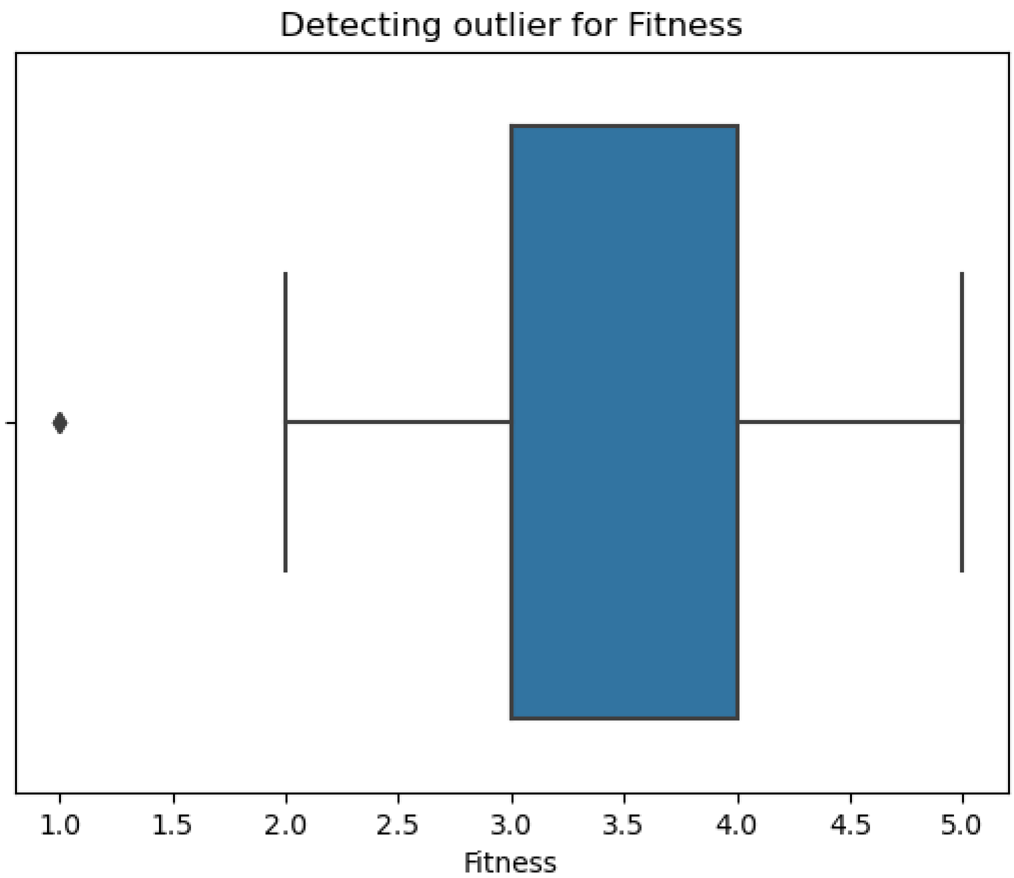

```
Median is  3.0
Mean is  3.4555555555555557
Difference between mean and median is 0.455
```

Inference -

- Majority of the customer's **Usage** fall in the range **3 to 4 times a week**
- There are **2 outliers** above the value of **5 times a week**
- Differnce between mean and median is **0.455**

In [320…

```
sns.boxplot(x='Fitness',data=df)
plt.title('Detecting outlier for Fitness')
plt.show()
print('Median is ',df['Fitness'].median())
print('Mean is ',df['Fitness'].mean())
print('Difference between mean and median is 0.311')
```



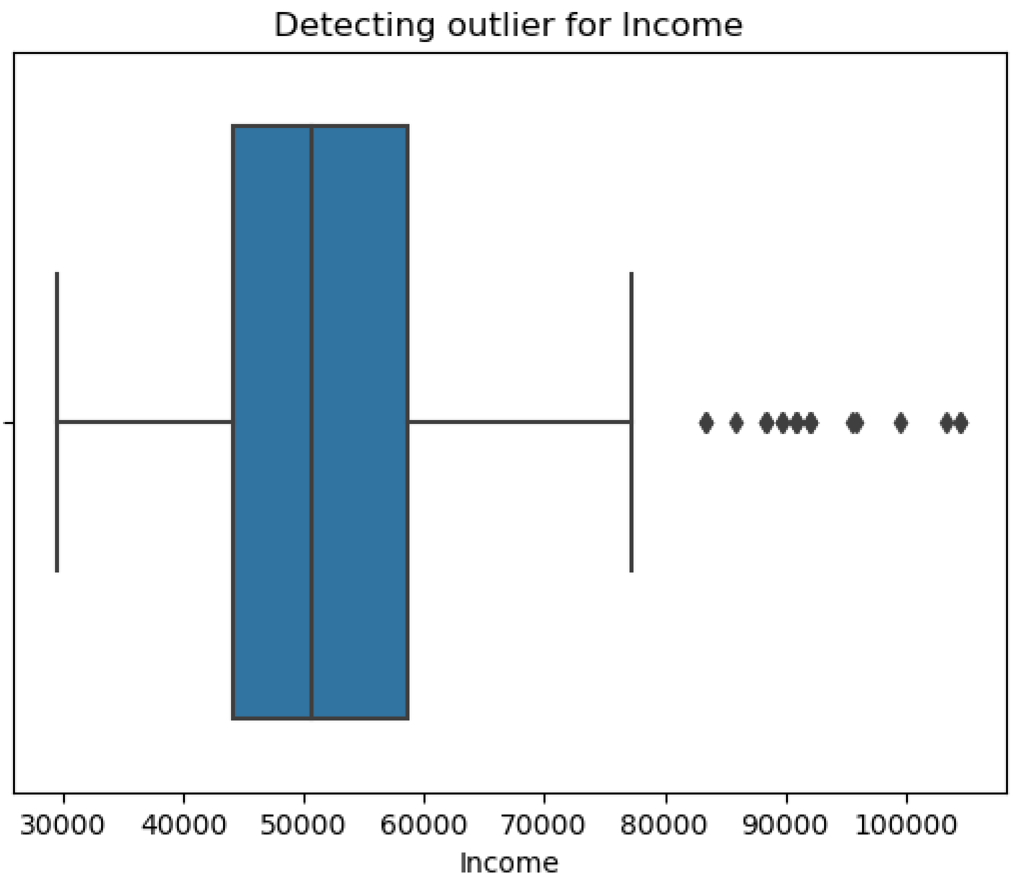

```
Median is  3.0
Mean is  3.311111111111111
Difference between mean and median is 0.311
```

Inference -

- Majority of the customer's gave themselves a **Fitness Rating** in the range **3 to 4**
- There is **1 outlier** below the value of **2**
- Differnce between mean and median is **0.311**

In [276…

```
sns.boxplot(x='Income',data=df)
plt.title('Detecting outlier for Income')
plt.show()
print('Median is ',df['Income'].median())
print('Mean is ',df['Income'].mean())
print('Difference between mean and median is 3123.077')
```



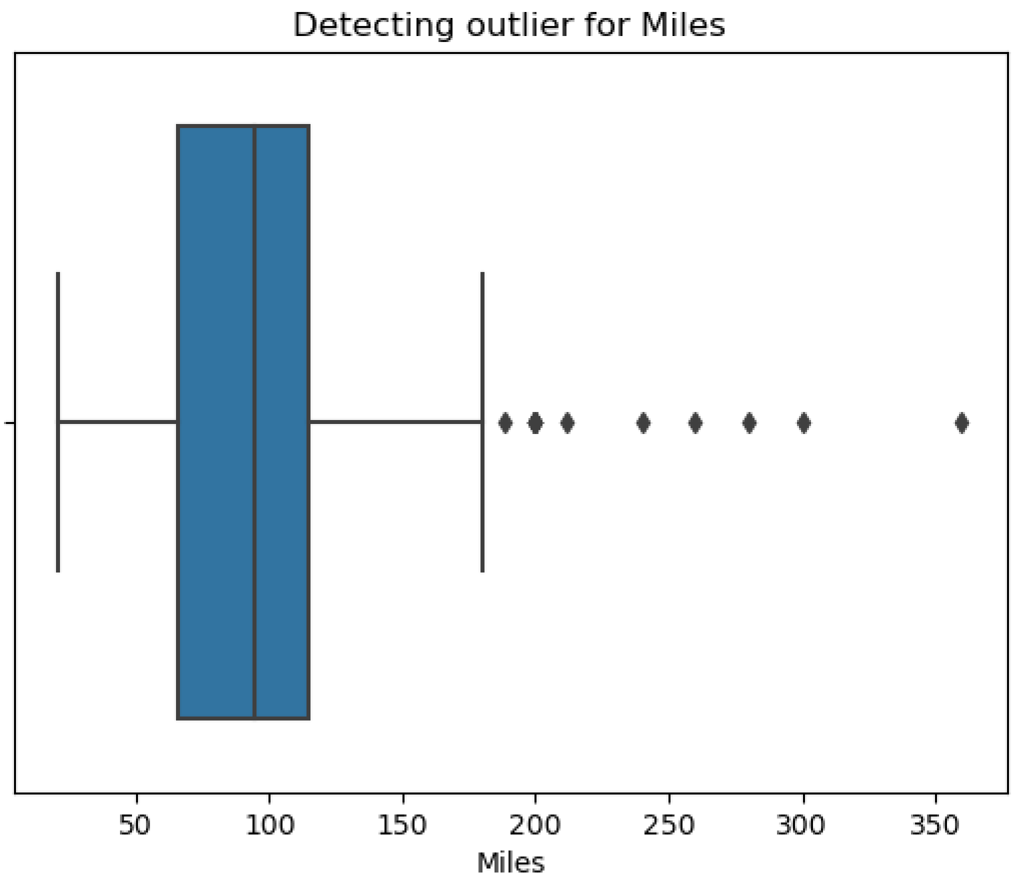

```
Median is  50596.5
Mean is  53719.57777777778
Difference between mean and median is 3123.077
```

Inference -

- Majority of the customer's have **Income** in the range **45,000 to 60,000 dollars**
- There are **11 outlier** above the value of **78,000 dollars**
- Differnce between mean and median is **3123.077 dollars**

In [278…

```
sns.boxplot(x='Miles',data=df)
plt.title('Detecting outlier for Miles')
plt.show()
print('Median is ',df['Miles'].median())
print('Mean is ',df['Miles'].mean())
print('Difference between mean and median is 9.194')
```




```
Median is  94.0
Mean is  103.19444444444444
Difference between mean and median is 9.194
```

Inference -

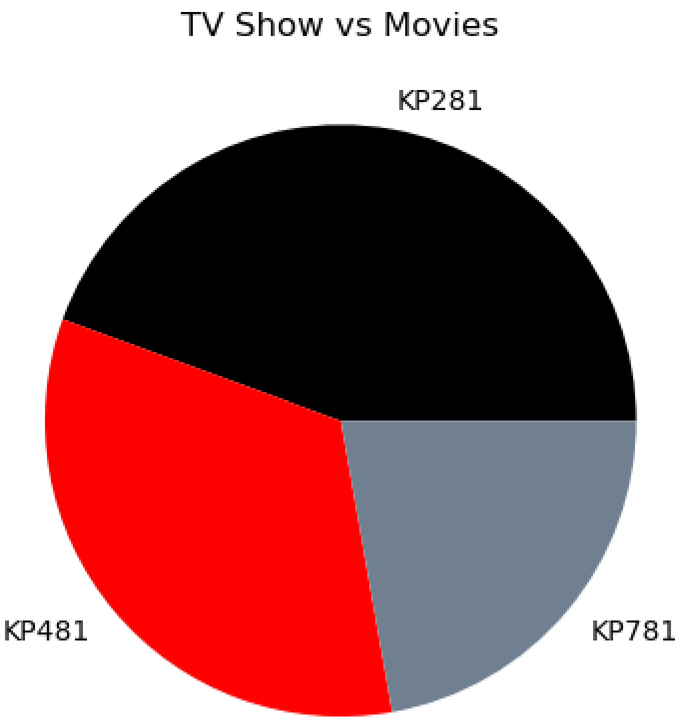
- Average Number of **Miles** the customer expects to walk/run each week falls in the range **60 to 110 miles**
- There are **8 outlier** above the value of **180 miles**
- Differnce between mean and median is **9.194 miles**

Marginal Probability and Conditional Probability -

In [291…
```
# Marginal Probability
round(df['Product'].value_counts(normalize=True)*100,2)
```

Out[291]:
```
KP281    44.44
KP481    33.33
KP781    22.22
Name: Product, dtype: float64
```

In [292…
```
d = df['Product'].value_counts()
colors = ['black','red','slategray']
plt.pie(d.values, labels=d.index, colors=colors)
plt.title('Products')
plt.show()
```



Inference -

- **KP281** type treadmill was purchased by **44.4%** of the customers.
- **KP481** type was purchased by **33.3%** of customers.
- Wheres type **KP781** was purchased by **22.2%** customers

In [ ]:
```
# Gender
round(df['Gender'].value_counts(normalize=True)*100,2)
```

Out[ ]:
```
Male      57.78
Female    42.22
Name: Gender, dtype: float64
```

- **57.78%** customers are **Males**
- **Whereas 42.22%** customers are **Females**

In [ ]:
```
# Marital Status
round(df['MaritalStatus'].value_counts(normalize=True)*100,2)
```

Out[ ]:
```
Partnered    59.44
Single       40.56
Name: MaritalStatus, dtype: float64
```

- **59.44%** customers are **Partneres=d**
- **Whereas 40.56%** customers are **Single**

In [ ]:
```
# Usage
round(df['Usage'].value_counts(normalize=True)*100,2).reset_index()
```

Out[ ]:

	index	Usage
0	3	38.33
1	4	28.89
2	2	18.33
3	5	9.44
4	6	3.89
5	7	1.11

- Majority of customers (**38.33%**) customers use their threadmill for **3 days** a week
- **Around 1.11%** customers use it for **7 days** a week

In [ ]:
```
# Usage
round(df['Fitness'].value_counts(normalize=True)*100,2).reset_index()
```

Out[ ]:

	index	Fitness
0	3	53.89
1	5	17.22
2	2	14.44
3	4	13.33
4	1	1.11

- Majority of customers (**53.89%**) customers have given themselves a rating of **3** out of 5
- **Around 17.22%** customers have **5/5** ratings

Conditional Probability -

Probability for each Product given Gender -

In [414…
```
#Joint Probability Table
pd.crosstab(df['Gender'],df['Product'],margins=True,normalize=True).round(2)
```

Out[414]:

Product	KP281	KP481	KP781	All
Gender				
Female	22.0	16.0	4.0	42.0
Male	22.0	17.0	18.0	58.0
All	44.0	33.0	22.0	100.0

In [415…
```
#Conditional Probability Table
pd.crosstab(index = df["Gender"], columns = df["Product"], margins = True, n
```

Out[415]:

Product	KP281	KP481	KP781
Gender			
Female	53.0	38.0	9.0
Male	38.0	30.0	32.0
All	44.0	33.0	22.0

Inference -

```
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
```

Insights -

Probability of customer being **Male** is **58%**
Probability of customer being **Female** is **42%**

Probability of **Male customer buying KP781** is **32%**
Probability of **Male customer buying KP481** is **30%**
Probability of **Male customer buying KP281** is **38%**

Probability of **Female customer buying KP781** is **9%**
Probability of **Female customer buying KP481** is **38%**
Probability of **Female customer buying KP281** is **53%**

Probability for each Product given MaritalStatus -

In [416…

```
#Joint Probability Table
pd.crosstab(df['MaritalStatus'],df['Product'],margins=True,normalize=True).r
```

Out[416]:

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	27.0	20.0	13.0	59.0
Single	18.0	13.0	9.0	41.0
All	44.0	33.0	22.0	100.0

In [417…

```
#Conditional Probability Table
pd.crosstab(index = df["MaritalStatus"], columns = df["Product"], margins =
```

Out[417]:

Product	KP281	KP481	KP781
MaritalStatus			
Partnered	45.0	34.0	21.0
Single	44.0	33.0	23.0
All	44.0	33.0	22.0

Inference -

```
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
```

Insights -

Probability of customer being **Single** is **41%**
Probability of customer being **Partnered** is **59%**

Probability of **Single customer buying KP781** is **23%**
Probability of **Single customer buying KP481** is **33%**
Probability of **Single customer buying KP281** is **44%**

Probability of **Partnered customer buying KP781** is **21%**
Probability of **Partnered customer buying KP481** is **34%**
Probability of **Partnered customer buying KP281** is **45%**

Business Insights -

KP281 -

- KP281 is the most **economical** product as well as the product with **most number of sales**
- **Age** group of customers fall in the range of **23 to 34 years**. Median Age of customers is 27
- Customers have **education** of around **14 to 16 years**
- Users of KP281 **use** it for **3 to 4 times a week**
- Users of KP281 have given themselves **low rating in terms of Fitness**
- Mostly **Low Income** groups prefer KP281 (ie Income in the range **40k to 54k**)
- Users using KP281 **walk/run** around **60 to 100 miles a week**

KP481 -

- Age group of customers fall in the range of **24 to 34 years**. Median Age of customers is 27

- Customers have **education** of around **14 to 16 years**

- Users of KP481 **use** it for **3 times a week**
- Users of KP481 have given themselves **low rating in terms of Fitness**
- Mostly **Low Income** groups prefer KP481 (ie Income in the range **45k to 54k**)
- Users using KP481 **walk/run** around **60 to 110 miles a week**

```
In [ ]:
```

KP781 -

- This is the most **premium product** from the brand

- **Age** group of customers fall in the range of **25 to 30 years**. Median Age of customers is 28

- Customers have **education** of around **16 to 18 years**

- Users of KP781 **use** it for **4 to 5 times a week**
- Users of KP781 have given themselves **High rating (4 to 5) in terms of Fitness**
- Mostly **High Income** groups prefer KP781 (ie Income in the range **60k to 90k**)
- Users using KP781 **walk/run** around **120 to 200 miles a week**
- Customers of KP781 tend to take their **Fitness very seriously**. They are the most **dedicated** set of customers.

Recommendation -

- Participation of **Single customers are less compared to Partnered customers**. So aerofit should do **Ad campaigns for Single customers** to make them more intersted in their fitness and hence their products.

- **Female customers are very low in the KP781 product category**. KP781 are mostly only bought by High Income group customers. From this group Female participation is very less. Aerofit should do some **social media marketing and and enagage with more Female audiences.**

- Most of the customers fall in the age range of 24 to 34 years. Aerofit should do more direct **advertising** through retail stores and Supermarkets for the people above the **age of 35**. These advertising should **highlight the risk of diseases in Old Age from the lack of exercise**. This can be made more convincing by including recommendations from Doctors.

- **Create a Mobile App to interact and motivate the customers.** Things like Daily Tasks and Rewards can be included to motivate the customers

- Usage of KP281 and KP 481 are low. We shouuld motivate the customers to workout more often. These can be done through **alerts from the mobile app.**

In [ ]: