Aerofit Business Case Study by **B Dinesh Prabhu DSML DEC 2022**

Importing the Required Libraries

```
#Import the Required Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

```
! gdown 1m9wpycyucsO8z1_1ga6rcwd2rSZp8Nln
```

```
Downloading...
From: https://drive.google.com/uc?id=1m9wpycyucsO8z1_1ga6rcwd2rSZp8Nln
To: /content/Aerofit_data.csv.txt
100% 7.28k/7.28k [00:00<00:00, 26.3MB/s]
```

```
#Creating Aerofit data frame
ARF_df=pd.read_csv("Aerofit_data.csv.txt")
ARF_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

1.Defining Problem Statement and Analysing basic metrics

Perform Exploratory Data Analysis (EDA) on Aerfit Data and Extract meaningful insights from it to improve the Business

> 1.1 Columns in the data

```
#Columns
ARF_df.columns
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

> 1.2 Information About the data

```
# Checking the Structure of the data
ARF_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
```

> 1.3Shape of the data

```
ARF_df.shape
```

```
(180, 9)
```

> 1.4 Data Types the data

```
ARF_df.dtypes
```

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

> 1.5 Conversion of categorical objects into category

```
ARF_df['Product']=ARF_df['Product'].astype('category')
ARF_df['Gender']=ARF_df['Gender'].astype('category')
ARF_df['MaritalStatus']=ARF_df['MaritalStatus'].astype('category')
ARF_df.dtypes
```

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

2. Non-Graphical Analysis: Value counts and unique attributes

> 2.1 **Value Counts**

> **OBSERVATION** :: Most number of sales were happened for KP281 model

```
#Procuts counts
ARF_df['Product'].value_counts()
```

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

> **Observation** Most of the customers Marital status is "**Partnered**"

```
#Marital Status
ARF_df['MaritalStatus'].value_counts()
```

```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

> **Observation** Most of the Customers are **Males**

```
#Gender Count
ARF_df['Gender'].value_counts()
```

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

> **Observation**: Most of the customers lies in the age group between **22 to 30**

```
#Age count
ARF_df['Age'].value_counts()
```

```
25    25
23    18
24    12
26    12
28     9
35     8
33     8
30     7
38     7
21     7
22     7
27     7
31     6
34     6
29     6
20     5
40     5
32     4
19     4
48     2
37     2
45     2
47     2
46     1
50     1
18     1
44     1
43     1
41     1
39     1
36     1
42     1
Name: Age, dtype: int64
```

> **Observation** On an average Most of the customers are using tread mills **3 time's a week**

```
#Average Usage countly weekly
ARF_df['Usage'].value_counts()
```

```
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

> **Observation**: More than 50% of the customers rated 3 as their fitness level and 17% of the customers rated their fitness level as 5

```
#Salf rated fitness level
ARF_df['Fitness'].value_counts()
```

```
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

> 2.2 **Unique Values**

Observation:: 3 different treadmills were released by Aerofit

```
print(ARF_df['Product'].unique())
```

```
['KP281', 'KP481', 'KP781']
Categories (3, object): ['KP281', 'KP481', 'KP781']
```

> Some Basic Matrics using non graphical analysis

> **Average** Miles Ran by Customers

```
ARF_df['Miles'].mean()
```

```
103.19444444444444
```

> ***Observation**: we can see that There is not even one male customer who bought KP281 product with Marital status as **SINGLE** , Patnered females are the Highest number of customer's using KP281

```
x1=ARF_df[(ARF_df.Product=='KP281') & (ARF_df.MaritalStatus=="Single") & (ARF_df.Gender=="Female")].shape[0]
print('Number of Female customers who bought KP281 With marital status as single ::',x1)
x2=ARF_df[(ARF_df.Product=='KP281') & (ARF_df.MaritalStatus=="Partnered") & (ARF_df.Gender=="Female")].shape[0]
print('Number of Female customers who bought KP281 With marital status as Partnered ::',x2)
x3=ARF_df[(ARF_df.Product=='KP281') & (ARF_df.MaritalStatus=="Singles") & (ARF_df.Gender=="Male")].shape[0]
print('Number of Male customers who bought KP281 With marital status as single ::',x3)
x4=ARF_df[(ARF_df.Product=='KP281') & (ARF_df.MaritalStatus=="Partnered") & (ARF_df.Gender=="Male")].shape[0]
print('Number of Male customers who bought KP281 With marital status as Partnered ::',x4)
```

```
Number of Female customers who bought KP281 With marital status as single :: 13
Number of Female customers who bought KP281 With marital status as Partnered :: 27
Number of Male customers who bought KP281 With marital status as single :: 0
Number of Male customers who bought KP281 With marital status as Partnered :: 21
```

> **Observation** Male Partnered customers are Mostly using **KP481**

```
y1=ARF_df[(ARF_df.Product=='KP481') & (ARF_df.MaritalStatus=="Single") & (ARF_df.Gender=="Female")].shape[0]
print('Number of Female customers who bought KP481 With marital status as single ::',y1)
y2=ARF_df[(ARF_df.Product=='KP481') & (ARF_df.MaritalStatus=="Partnered") & (ARF_df.Gender=="Female")].shape[0]
print('Number of Female customers who bought KP481 With marital status as Partnered ::',y2)
y3=ARF_df[(ARF_df.Product=='KP481') & (ARF_df.MaritalStatus=="Single") & (ARF_df.Gender=="Male")].shape[0]
print('Number of Male customers who bought KP481 With marital status as single ::',y3)
y4=ARF_df[(ARF_df.Product=='KP481') & (ARF_df.MaritalStatus=="Partnered") & (ARF_df.Gender=="Male")].shape[0]
print('Number of Male customers who bought KP481 With marital status as Partnered ::',y4)
```

```
Number of Female customers who bought KP481 With marital status as single :: 14
Number of Female customers who bought KP481 With marital status as Partnered :: 15
Number of Male customers who bought KP481 With marital status as single :: 10
Number of Male customers who bought KP481 With marital status as Partnered :: 21
```

> **Observation** just like KP481,in case of **KP781** most of the customers are **Male Partnered** Users

```
z1=ARF_df[(ARF_df.Product=='KP781') & (ARF_df.MaritalStatus=="Single") & (ARF_df.Gender=="Female")].shape[0]
print('Number of Female customers who bought KP781 With marital status as single ::',z1)
z2=ARF_df[(ARF_df.Product=='KP781') & (ARF_df.MaritalStatus=="Partnered") & (ARF_df.Gender=="Female")].shape[0]
print('Number of Female customers who bought KP781 With marital status as Partnered ::',z2)
z3=ARF_df[(ARF_df.Product=='KP781') & (ARF_df.MaritalStatus=="Single") & (ARF_df.Gender=="Male")].shape[0]
print('Number of Male customers who bought KP781 With marital status as single ::',z3)
z4=ARF_df[(ARF_df.Product=='KP781') & (ARF_df.MaritalStatus=="Partnered") & (ARF_df.Gender=="Male")].shape[0]
print('Number of Male customers who bought KP781 With marital status as Partnered ::',z4)
```

```
Number of Female customers who bought KP781 With marital status as single :: 3
Number of Female customers who bought KP781 With marital status as Partnered :: 4
Number of Male customers who bought KP781 With marital status as single :: 14
Number of Male customers who bought KP781 With marital status as Partnered :: 19
```

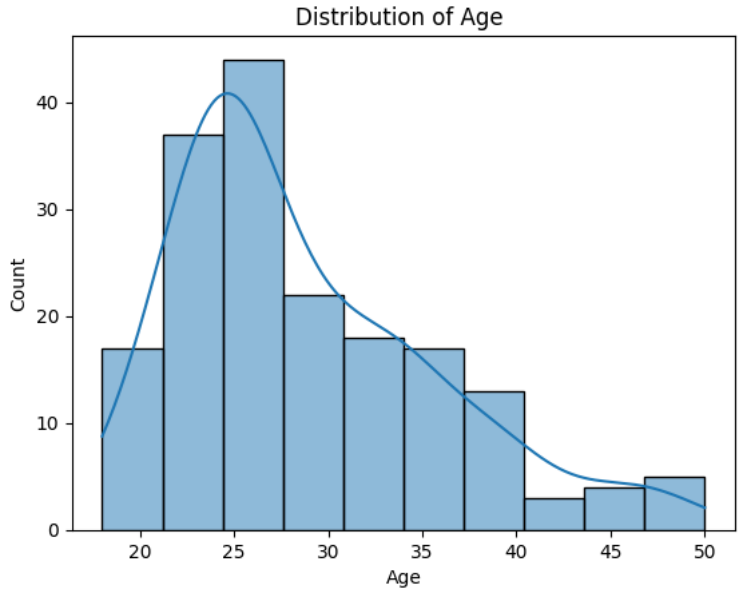
3. Visual Analysis - Univariate & Bivariate

> Distribution of Customers Age

> **Observation**: Distribution of Ages is skwed to the right

```
sns.histplot(data=ARF_df,x='Age',bins=10,kde=True)
plt.title('Distribution of Age ')
```

```
Text(0.5, 1.0, 'Distribution of Age ')
```

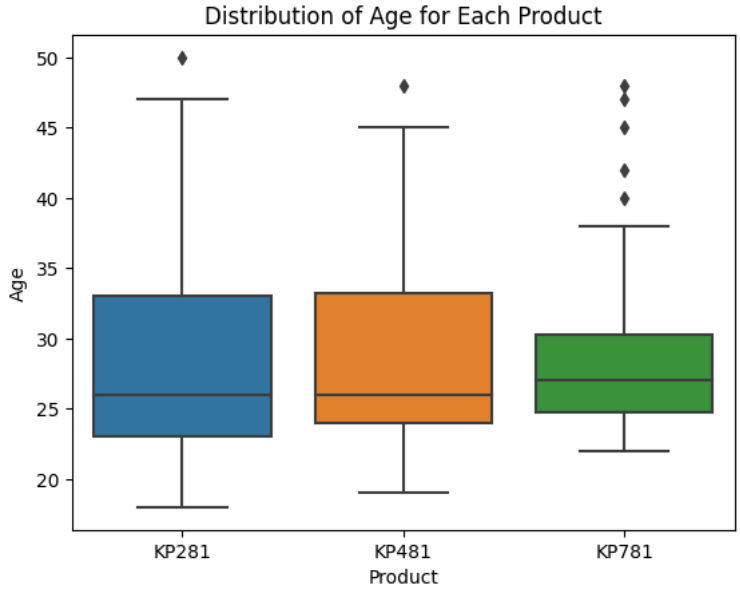


> **Box plot for each product**

> **Observation** we can see that there are more outliers in Ages of Customers using KP281 tread mill product

```
sns.boxplot(data=ARF_df,x='Product',y='Age')
plt.title('Distribution of Age for Each Product')
```

```
Text(0.5, 1.0, 'Distribution of Age for Each Product')
```

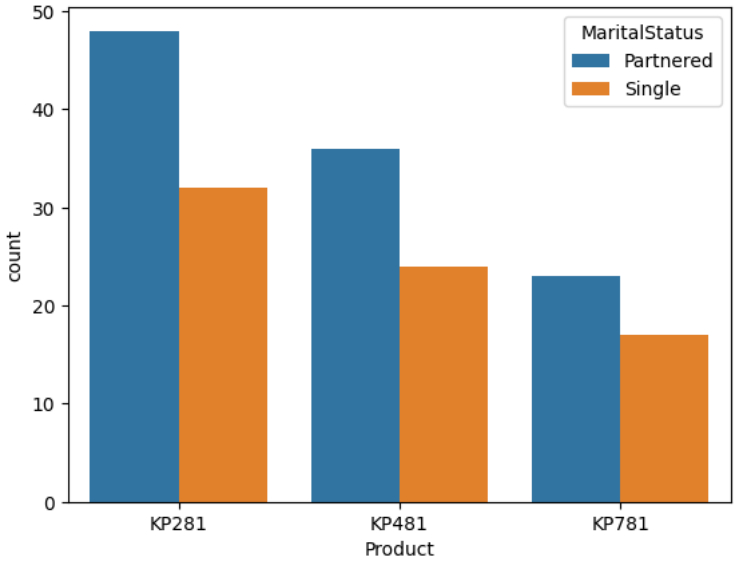


> **Count Plot**

> **OBSERVATION** we can Conclude that Among all three products that Aerofit Released most of them were used by people whose marital status is Partnered

```
sns.countplot(x='Product',hue='MaritalStatus',data=ARF_df)
```

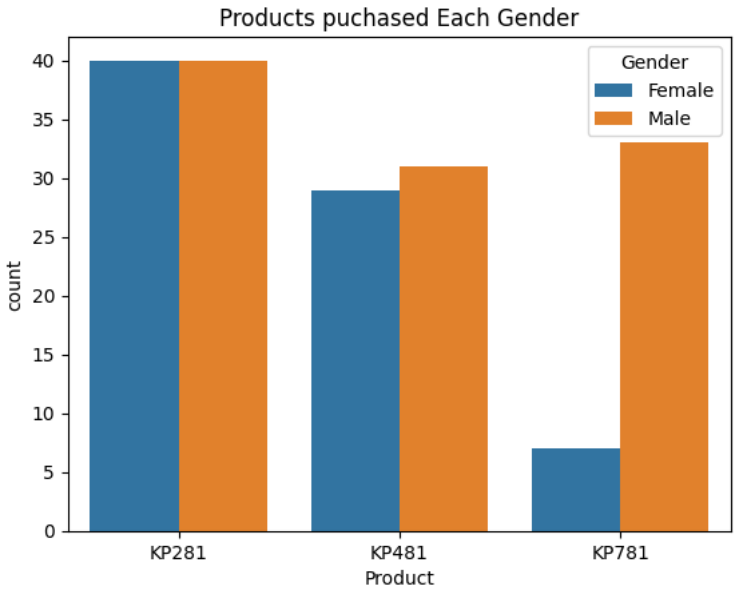
```
<Axes: xlabel='Product', ylabel='count'>
```



> **OBSERVATION** it is clear that The basic Model KP281 is used equally by both the Genders, looking at the KP481 (mid level tredmill) it has slightly more numbere of male users than female users, KP781(premium model tredmill) is mostly used by males(dominating) and there are less number of female users

```
sns.countplot(x='Product',hue='Gender',data=ARF_df)
plt.title("Products puchased Each Gender")
```

```
Text(0.5, 1.0, 'Products puchased Each Gender')
```



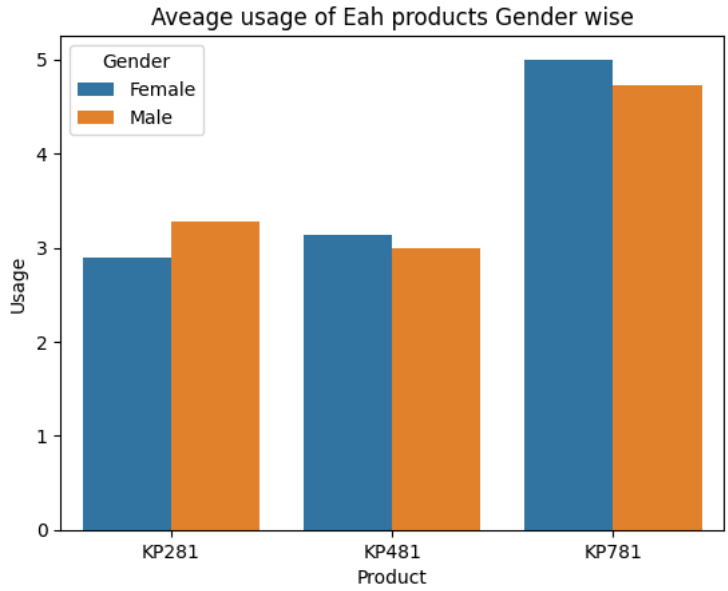
> 3.3 Bi variate Analysis

> Aveage usage of Eah products Gender wise

> **Observation** The Aveage usage Gender wise is changing from one Product to other on an Overall if we look at the bar plot we can conclude that Male customers tend to use the Tread mills more frequently

```
sns.barplot(data=ARF_df,x="Product",y="Usage",hue="Gender",errorbar=None)
plt.title('Aveage usage of Eah products Gender wise')
```

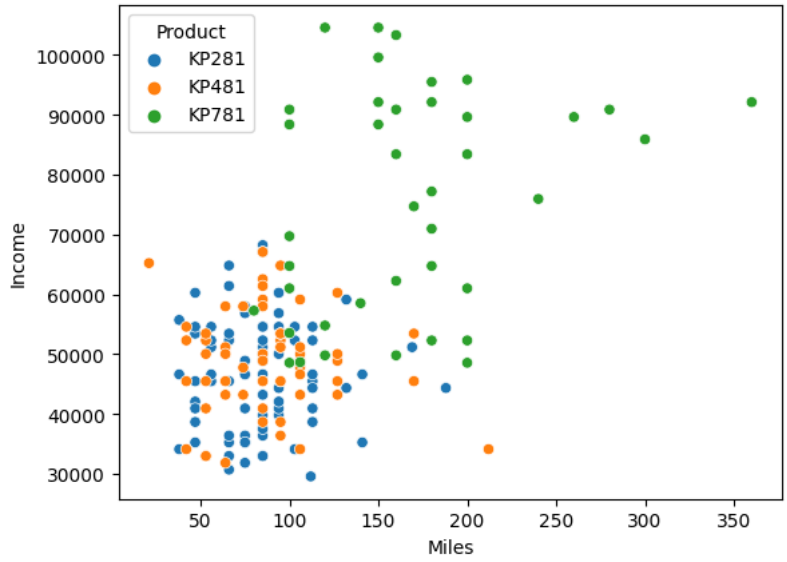
```
Text(0.5, 1.0, 'Aveage usage of Eah products Gender wise')
```



> **Observation**:: The Scatter plot is Direction is positive but non Linear in nature

```
sns.scatterplot(data=ARF_df,x='Miles',y='Income',hue='Product')
```

```
<Axes: xlabel='Miles', ylabel='Income'>
```



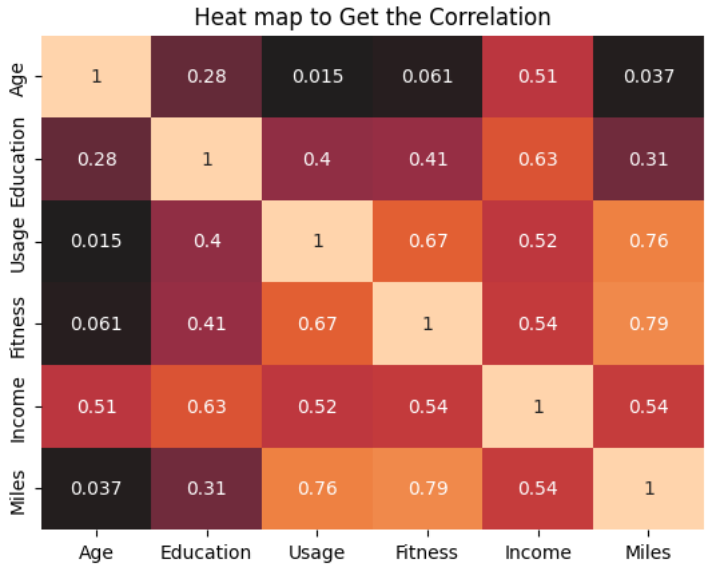
> **Heat map** for correlation index among different features in data

```
df1=ARF_df[['Age','Education','Usage','Fitness','Income','Miles']]
correlation=df1.corr()
correlation
```

	Age	Education	Usage	Fitness	Income	Miles
Age	1.000000	0.280496	0.015064	0.061105	0.513414	0.036618
Education	0.280496	1.000000	0.395155	0.410581	0.625827	0.307284
Usage	0.015064	0.395155	1.000000	0.668606	0.519537	0.759130
Fitness	0.061105	0.410581	0.668606	1.000000	0.535005	0.785702
Income	0.513414	0.625827	0.519537	0.535005	1.000000	0.543473
Miles	0.036618	0.307284	0.759130	0.785702	0.543473	1.000000

```
#Heat Map
sns.heatmap(correlation,cbar=False,annot=True,center=0)
plt.title(' Heat map to Get the Correlation ')
```

```
Text(0.5, 1.0, ' Heat map to Get the Correlation ')
```

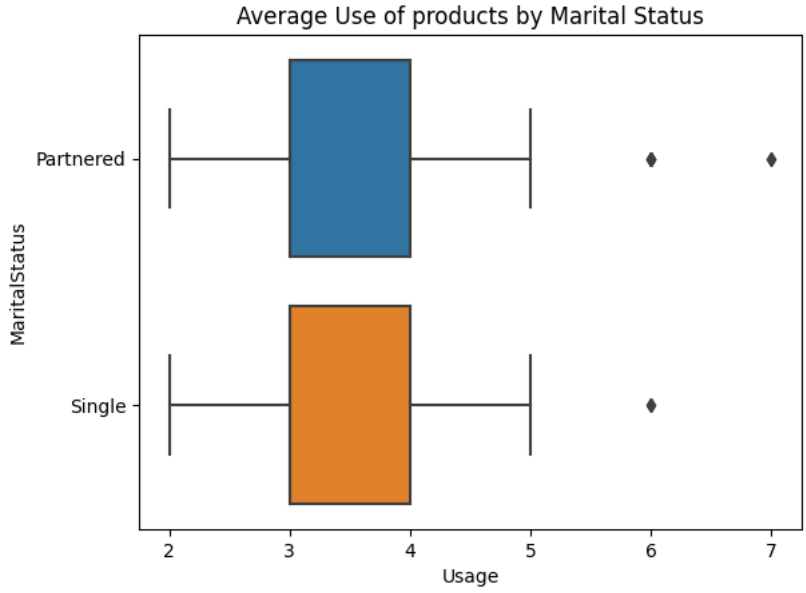


> **BOX plot**

> **Observation** we can Observe that the Average usage by both Patnered and Single Customers is almopst same with some outliers in partnered data

```
sns.boxplot(ARF_df,x='Usage',y='MaritalStatus',orient='h')
plt.title('Average Use of products by Marital Status')
```

```
Text(0.5, 1.0, 'Average Use of products by Marital Status')
```

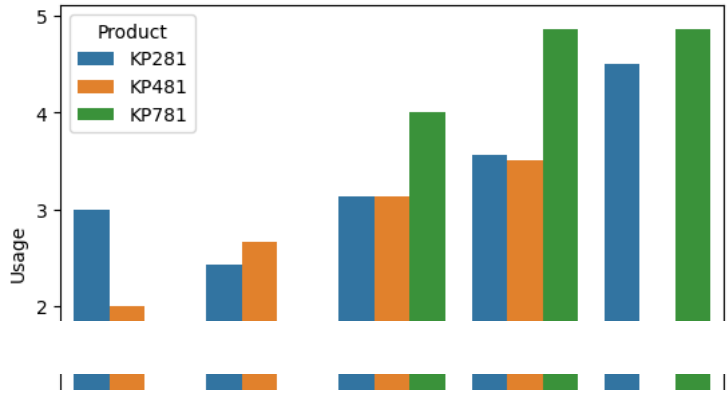


> **Fitness and Usage**: Customers with High Fitness rating are tend to use the Treadmill products more frequently

```
1.it is Observed that customers with 5fitness rating are not using KP481
2. customers with Fitness rating 1 and 2 are not at all using the premium model KP781
```

```
sns.barplot(data=ARF_df,x="Fitness",y="Usage",hue='Product',errorbar=None)
```

```
<Axes: xlabel='Fitness', ylabel='Usage'>
```




```
sns.pairplot(ARF_df,hue='Product')
plt.title('Scatter Plot ')
```

```
Text(0.5, 1.0, 'Scatter Plot ')
```

4.Missing Value & Outlier Detection

4.1 Checking If there are any Null Values in data

Observation : Looking at the information of the data we can conclude that the data contains ZERO Null values

```
ARF_df.isna().sum()
```

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

OBSERVATION: Looking at the information of the data we can conclude that the data contains ZERO Null values

4.2 **Statistical Summary Of The Data**

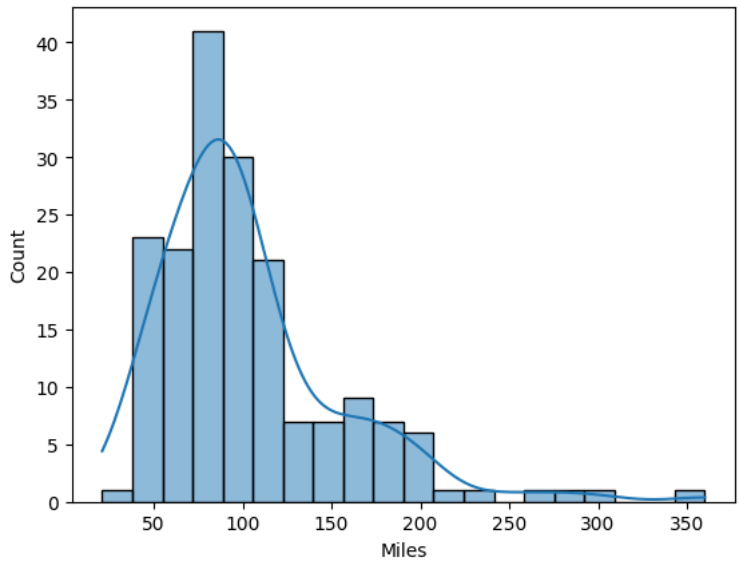
```
ARF_df.describe()[['Age','Education','Usage','Fitness','Income','Miles']]
```

	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

Outlier Check looking at the numeical columns of the data frame we can conclude that data have no outliers , how ever the column Miles need to be inspected, since mean is subjected outliers looking at the data, the mean value is 103.194 and max value is 360.0 which is more than the 3 sigma deviation(258.78).

```
#Distribution of Miles
sns.histplot(data=ARF_df['Miles'],kde=True)
```

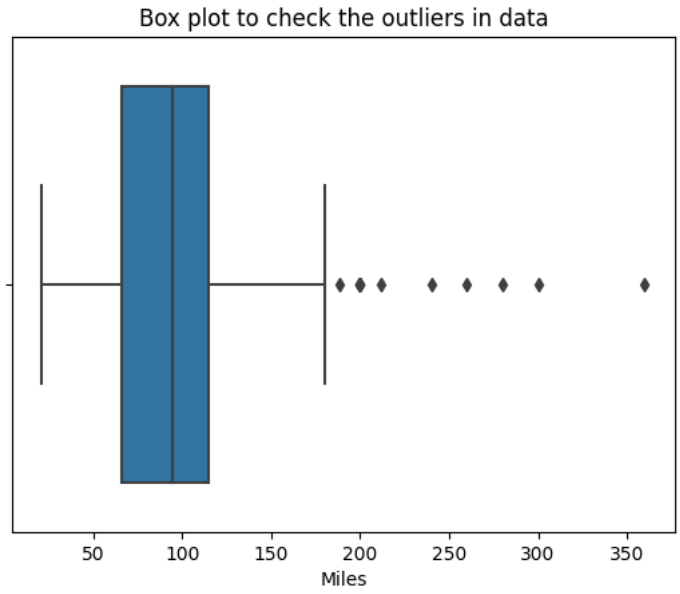
```
<Axes: xlabel='Miles', ylabel='Count'>
```



OBSERVATION we can see that the distribution of the Miles column is skewed to left we can use IRQ (inter quaantile range) method to determine the outliers

```
#Checking the outliers using box plot
sns.boxplot(data=ARF_df,x='Miles')
plt.title('Box plot to check the outliers in data')
```

```
Text(0.5, 1.0, 'Box plot to check the outliers in data')
```



CONTINGENCY TABLECREATION

```
cross_tab=pd.crosstab(ARF_df.Product,[ARF_df.MaritalStatus,ARF_df.Gender],margins='All',margins_name='SUM')
cross_tab
```

MaritalStatus	Partnered		Single		SUM
Gender	Female	Male	Female	Male	
Product					
KP281	27	21	13	19	80
KP481	15	21	14	10	60
KP781	4	19	3	14	40
SUM	46	61	30	43	180

> **Using above contingecy table we can find out the Conditional and Marginal probabilities**

```
#Normalsing values column wise to find the percentage of total contribution column wise
cross_tab_c=pd.crosstab(ARF_df.Product,[ARF_df.MaritalStatus,ARF_df.Gender],normalize='columns')*100
cross_tab_c
```

MaritalStatus	Partnered		Single	
Gender	Female	Male	Female	Male
Product				
KP281	58.695652	34.426230	43.333333	44.186047
KP481	32.608696	34.426230	46.666667	23.255814
KP781	8.695652	31.147541	10.000000	32.558140

```
#normalising row wise
cross_tab_r=pd.crosstab(ARF_df.Product,[ARF_df.MaritalStatus,ARF_df.Gender],normalize='index')*100
cross_tab_r
```

MaritalStatus	Partnered		Single	
Gender	Female	Male	Female	Male
Product				
KP281	33.75	26.25	16.250000	23.750000

```
#Taking Qualtiles
max_mile=ARF_df['Miles'].quantile(0.95)
max_mile
```

```
200.0
```

```
min_mile=ARF_df['Miles'].quantile(0.05)
ARF_df[ARF_df['Miles']<min_mile]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
19	KP281	23	Female	15	Partnered	2	2	34110	38
51	KP281	29	Female	14	Partnered	2	2	46617	38
59	KP281	33	Female	16	Single	2	2	55713	38
85	KP481	21	Male	16	Partnered	2	2	34110	42
99	KP481	25	Male	16	Partnered	2	2	52302	42
106	KP481	25	Female	14	Single	2	2	45480	42
117	KP481	31	Female	18	Single	2	1	65220	21
138	KP481	45	Male	16	Partnered	2	2	54576	42

> **Observation**: it is observed that there are some outlieres in Miles columns if we eleminate those outliers the shape got reduced to 151 rows,however since the outliers not that far from the Actual mean valuese it is decided to keep them in the data instead of removing them.

```
#to eleminate  the outliers
outlier_eleminated=ARF_df[(ARF_df['Miles']<max_mile) & (ARF_df['Miles']>min_mile)].shape
print('Shape of the data after eleminating outliers in Miles column',outlier_eleminated)
print('Original Shape',ARF_df.shape)
```

```
Shape of the data after eleminating outliers in Miles column (151, 9)
Original Shape (180, 9)
```

5. Business Insights based on Non-Graphical and Visual Analysis

> the Answer to this Question is addressed in 3rd (Non graphical analysis) and 4th(Graphical Analysis) questions, Relevant Comments were added to therespective Non_Graphical and Visual Analysis blocks

6. Recommendations- Actionable items for business.

Actionable Items

1. it is observed that Number of Male customers who bought KP281 With marital status as **Single** is Zero which clearly says that these set of of customers are not at all interested to use the KP281 product, keeping them as Targated customers Aerofit can offers some Discounts to Male single Customers which will encourage them to buy the product
2. Product wise Usage is 80 customers are using KP281,60 customers are using KP481, 40 customers are using KP781 which indicated that People are insterested to buy the basic model KP281 may be beacuase of Cost constraints to make them buy other two models Aerofit can Add addtional Features to KP481 which are not availabe in KP281 and also Aerofit can make an offer like who ever the people using KP281 They are allowed to get discount if they want to upgrade to higher model
3. it is Observed that customers with Fitness rating 1 and 2 are not at all using the premium model KP781 soAero fit can condut a capaingn stating "Use Our Premium model & Upgrade your fitness Level" by showing fitness level of customers who are using kp781(since i is observed that ost f the customers using kp781 are having 5 as their fitness level)
4. looking at the Data it is found that on an average a customeres ran 103 miles keeping this in mind Aerofit can engage their partnerships with fitness-related services or apps to enhance the customer experience and increase brand visibility.
5. Aeofit can offer complementory fitness products for the customers which encourages them to buy Aerofit models in turn it inceases the sales

✓ 0s completed at 8:29 AM