Business Case: Aerofit - Descriptive Statistics & Probability

Problem Statement

Analyse the data and help aerofit in deciding the target audience based on gender, age, income and other factors. Deciding the buyer persona is critical in providing the best recommendations. This will be done based the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months.

Importing Python Libraries necessary to carry out data exploration & visualisation

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

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

Dataset Description

Dataset Consists of:

- Product: Product Purchased KP281, KP481, or KP781
- Age: In years
- Gender: Male/Female
- Education: in years
- MaritalStatus: single or partnered
- Usage: average number of times the customer plans to use the treadmill each week
- Income: annual income (in $)
- Fitness: self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 is the excellent shape.
- Miles: average number of miles the customer expects to walk/run each week

Inspecting Dataset & Analyzing Different Metrics

```
df.head()
```

	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

```
df.tail()
```

	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

Observations on

- 1) shape of data 2) data types 3) Statistical summary

```
df.shape
```

```
(180, 9)
```

```
df.columns
```

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

```
df.size
```

```
1620
```

```
df.dtypes
```

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

```
df.info
```

```
<bound method DataFrame.info of     Product  Age  Gender  Education MaritalStatus  Usage  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]>
```

```
df.describe()
```

	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

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

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

Data Cleaning : Optional Treatment

```
df.isnull().sum().sort_values(ascending =False)
```

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

The dataset doesn't have any null values so data cleaning is not required

Mean and Median

```
df.mean()
```

```
<ipython-input-51-c61f0c8f89b5>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future ve
  df.mean()
Age             28.788889
Education       15.572222
Usage            3.455556
Fitness          3.311111
Income       53719.577778
Miles          103.194444
dtype: float64
```

```
df.median()
```

```
<ipython-input-53-6d467abf240d>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future
  df.median()
Age             26.0
Education       16.0
Usage            3.0
Fitness          3.0
Income       50596.5
Miles           94.0
dtype: float64
```

Inference: Difference between Mean and Median is not significant

Check the characteristics of the data

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

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

Out of 180 samples, there are 80 KP281, 60 KP481 and 40 KP781

```
round(df['Product'].value_counts(normalize=True)*100,2)
```

```
KP281    44.44
KP481    33.33
KP781    22.22
Name: Product, dtype: float64
```

which means 44.44% are KP281, 33.33% are KP481 and 22.22% are KP781

Age wise unique count & value count

```
df["Age"].unique()
```

```
array([18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 43, 44, 46, 47, 50, 45, 48, 42])
```

```
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
```

Inference: 1) we have customers from age range of 18-42. 2) Most of the customers lies in the range of 25-30.

Gender wise unique count & value count

```
df["Gender"].unique()
```

```
array(['Male', 'Female'], dtype=object)
```

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

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

Inference: 1) we have male customers more than female.

Marital Status unique count & value count

```
df["MaritalStatus"].unique()
```

```
array(['Single', 'Partnered'], dtype=object)
```

```
df["MaritalStatus"].value_counts()
```

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

```
df["MaritalStatus"].value_counts(normalize=True).round(2)*100
```

```
Partnered    59.0
Single       41.0
Name: MaritalStatus, dtype: float64
```

Inference: The data contains 59% partnered customers and 41% single people. Not really useful as sample size is pretty small.

Usage unique count & value count

```
df["Usage"].unique()
```

```
array([3, 2, 4, 5, 6, 7])
```

```
df["Usage"].value_counts(normalize=True).round(2)*100
```

```
3    38.0
4    29.0
2    18.0
5     9.0
6     4.0
7     1.0
Name: Usage, dtype: float64
```

Inference: Mostly users use the threadmill 2-4 days a week. Only 5% of the customers use it 6-7 days a week.

Checking data fitness wise

```
# Fitness wise unique value & count -

df["Fitness"].unique()
```

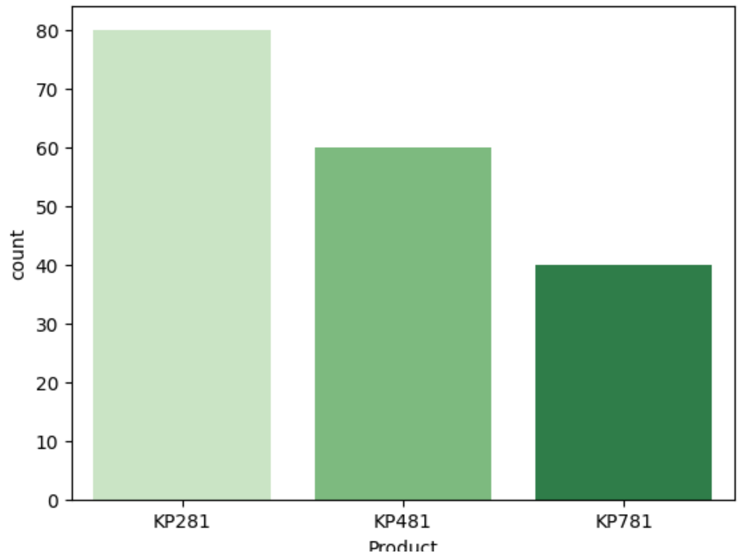
```
array([4, 3, 2, 1, 5])
```

```
df["Fitness"].value_counts(normalize=True).round(2)*100
```

```
3    54.0
5    17.0
2    14.0
4    13.0
1     1.0
Name: Fitness, dtype: float64
```

Checking the preference of the products among the users

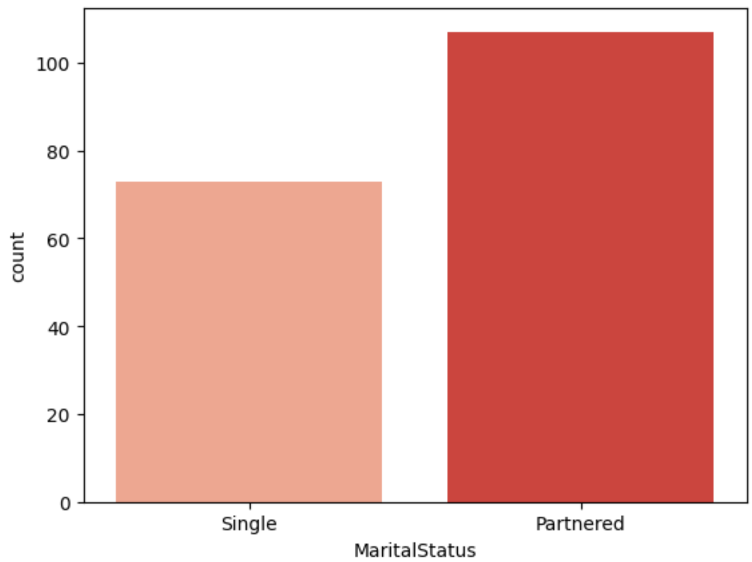
```
# Product countplot
sns.countplot(x = "Product", data = df, palette = "Greens")
plt.show()
```



Inference: KP281 (the cheapest product) is also the most popular product.

Checking the gender wise data

```
sns.countplot(x = "MaritalStatus", data = df, palette = "Reds")
plt.show()
```



Inference: Married people are higher in number in this dataset compared to single people.

Checking data age wise

```
plt.figure(figsize = (8,5))
sns.countplot(x = "Age", data = df, palette = "twilight")
plt.xticks(rotation = 90)
plt.show()
```

Inference : From above graph, people in age group of 23-28 are more in numbers than rest of the age group.

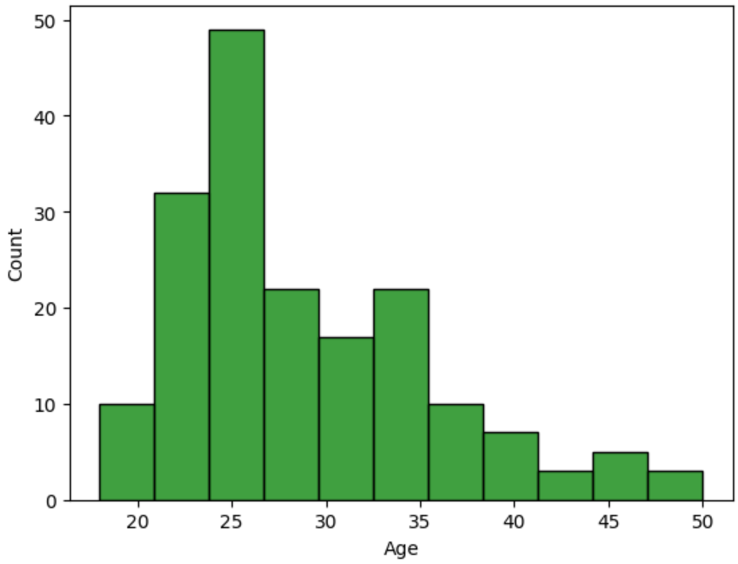
Inference: 1) Fitness value indicates the fitness of person and from the above values it can be seen that there are a good amount of people who have maintained their fitness. 2) The count of people who are having poor fitness are low in numbers.

```
df.head()
```

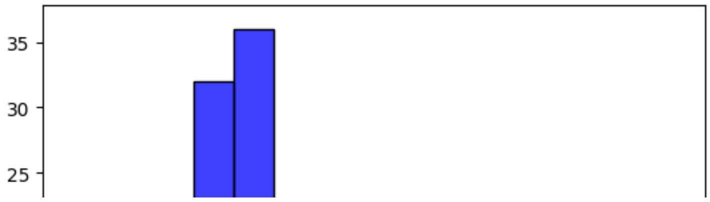
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

Univariate Analysis - Check different columns using histograms

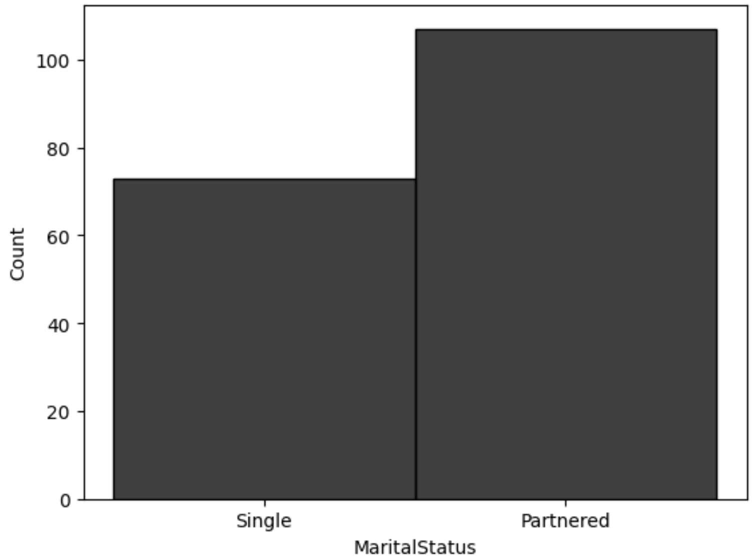
```
sns.histplot(df["Age"], color = "g")
plt.show()
```



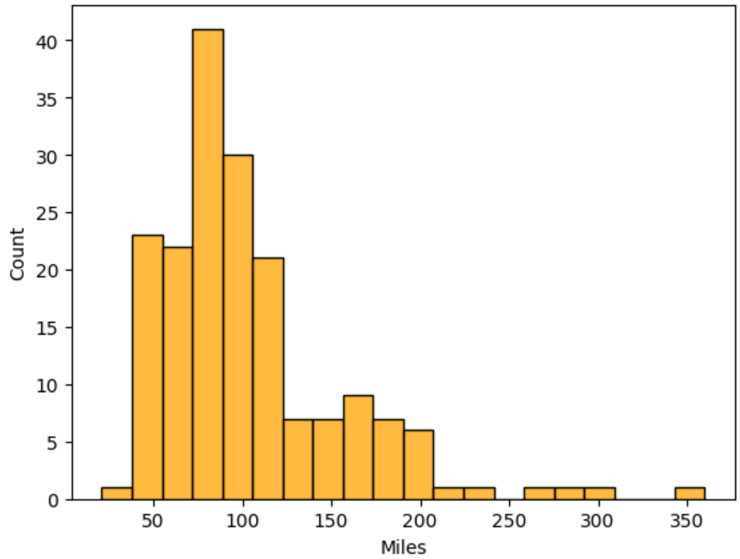

```
sns.histplot(df["Income"], color = "b")
plt.show()
```



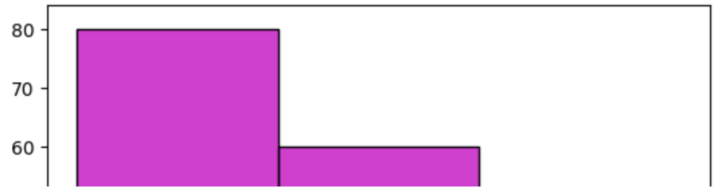

```
sns.histplot(df["MaritalStatus"], color = "k")
plt.show()
```




```
sns.histplot(df["Miles"], color = "orange")
plt.show()
```

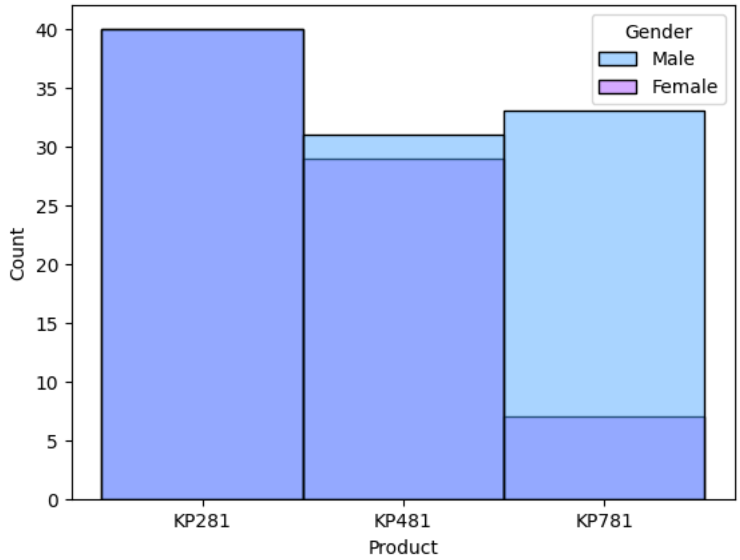



```
sns.histplot(df["Product"], color = "m")
plt.show()
```



Bivariate Analysis

```
# Product vs Gender -
sns.histplot(x = "Product", hue = "Gender", data = df, palette = "cool")
plt.show()
```

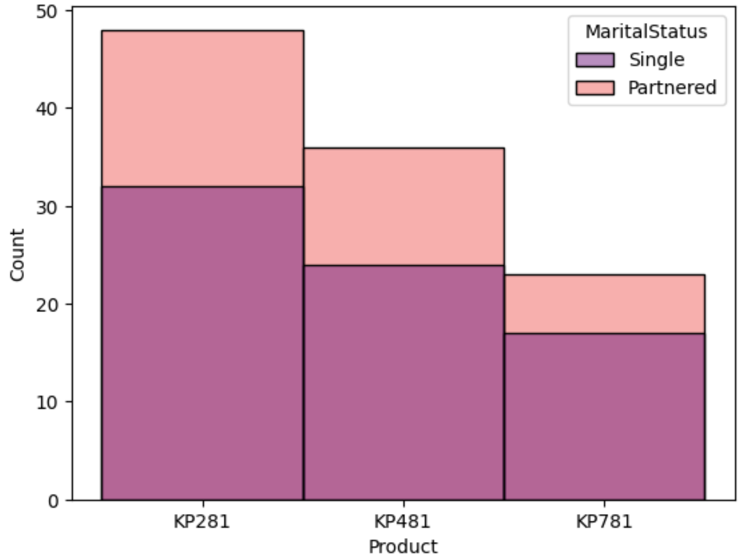


Inference:

- Maximum buyers for KP281 are female.
- More women buy KP481 as compared to male.
- Majority of men buy KP781.

```
# Product vs Marital Status-
sns.histplot(x = "Product", hue = "MaritalStatus", data = df, palette = "magma")

plt.show()
```



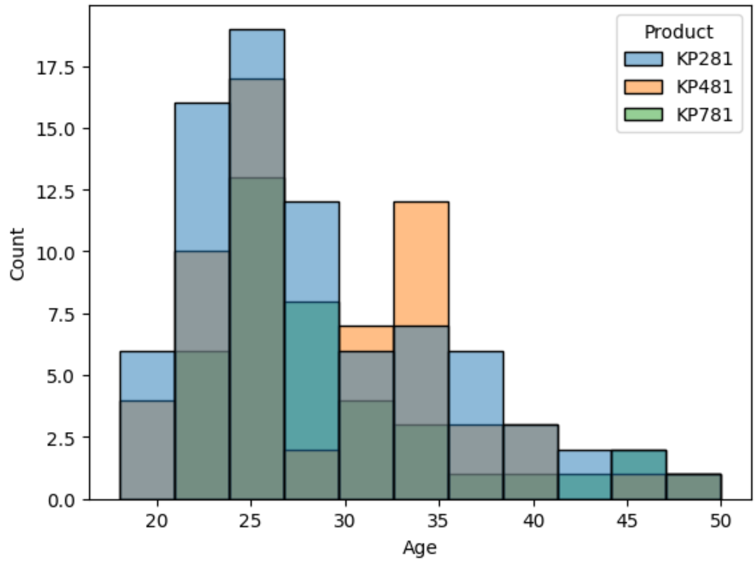
Inference:

- More KP281 is bought by single people than partnered.

- The same pattern is visible for the other two variants as well.

Double-click (or enter) to edit

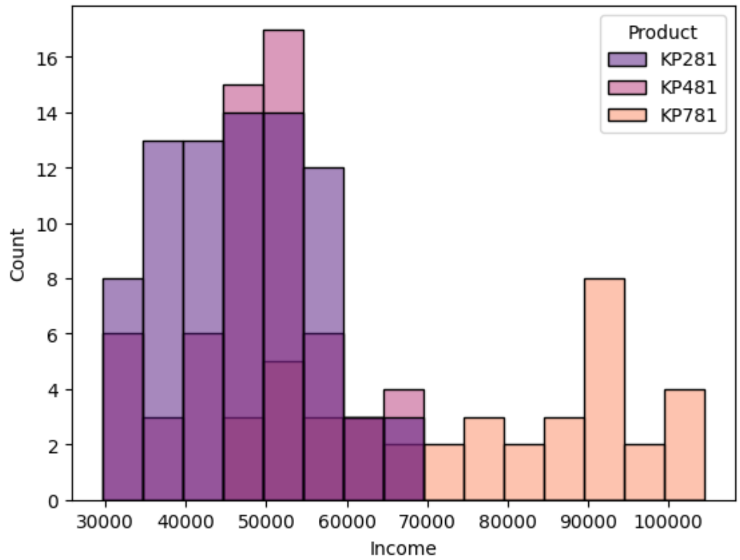
```
# Product vs Age -
sns.histplot(x = "Age", hue = "Product", data = df)
plt.show()
```



Inference:

- KP281 is widely popular under 30 age groups
- KP781 has medium popularity
- KP481 is most popular in the age group between 30-35

```
# Product vs Income -
sns.histplot(x = "Income", hue = "Product", data = df, palette = "magma")
plt.show()
```

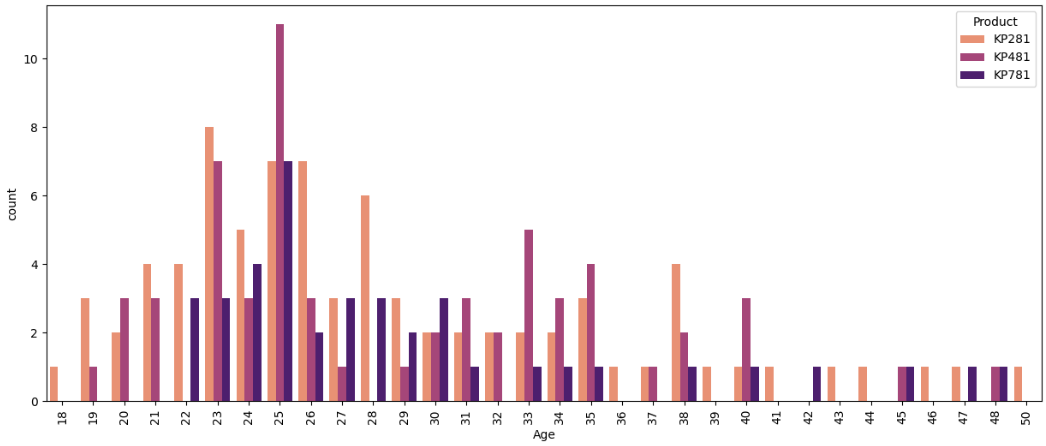


Inference:

- KP281 is only bought in income groups between 30k to 70k, which makes sense because it is the cheapest variant
- KP481 is most bought in income groups between 45k to 55k
- KP781 is only bought in income groups between 60k to 100k, which makes sense because it is the most expensive variant

Countplots

```
plt.figure(figsize = (15,6))
sns.countplot(x = "Age", hue = "Product", data = df, palette = "magma_r" )
plt.xticks( rotation = 90)
plt.show()
```



Inference:

- KP281 -People of almost every age use this product.
- KP481 - Users mainly are between 19-48. In the age group between 23-26, people prefer using this medium range product.
- KP781 - Used mainly by users ages in between 23-30, reflective of the fact that high earners use this product. This may also reflect the fact that young athletes may prefer this product to other variants. The number of count decreases for the people having age greater than 30.

Checking if income has any effect on the choice of products

```
df.groupby('Product')["Income"].describe()
```

Product	count	mean	std	min	25%	50%	75%	max
KP281	80.0	46418.025	9075.783190	29562.0	38658.00	46617.0	53439.0	68220.0
KP481	60.0	48973.650	8653.989388	31836.0	44911.50	49459.5	53439.0	67083.0
KP781	40.0	75441.575	18505.836720	48556.0	58204.75	76568.5	90886.0	104581.0

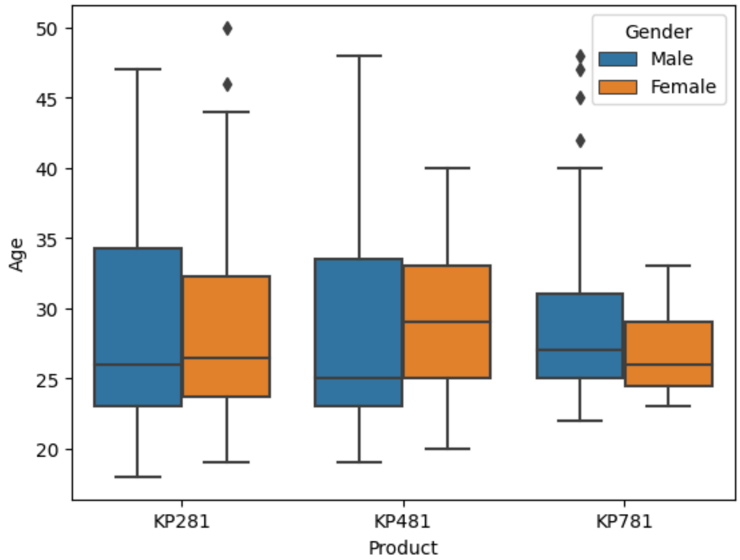
Lets visualise this with the help of a boxplot now

```
import seaborn as sns
sns.boxplot(x="Product", y="Income", hue="Gender", data=df)
plt.show()
```

Inference: From the above table it seems like income is directly dependent on the price of the product

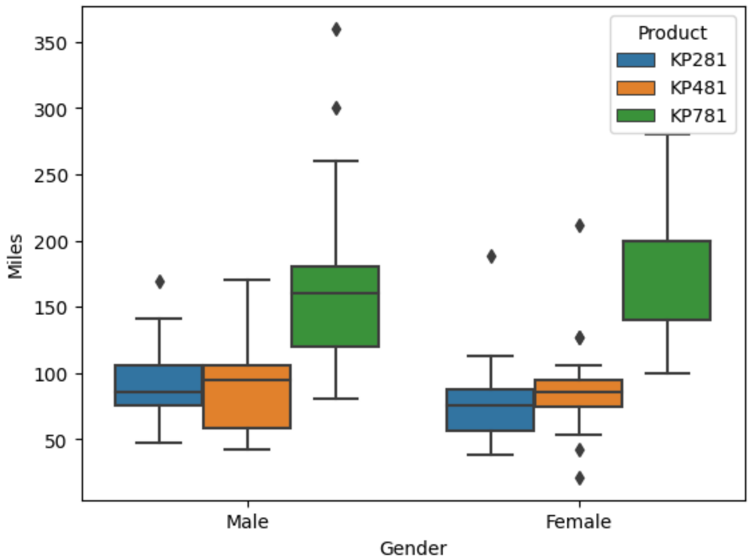
Checking if age and gender has any effect on the choice of products

```
sns.boxplot(x="Product", y="Age", hue="Gender", data=df)
plt.show()
```



Checking the relationship of Gender and Miles with the product

```
sns.boxplot(x="Gender", y="Miles", hue="Product", data=df)
plt.show()
```

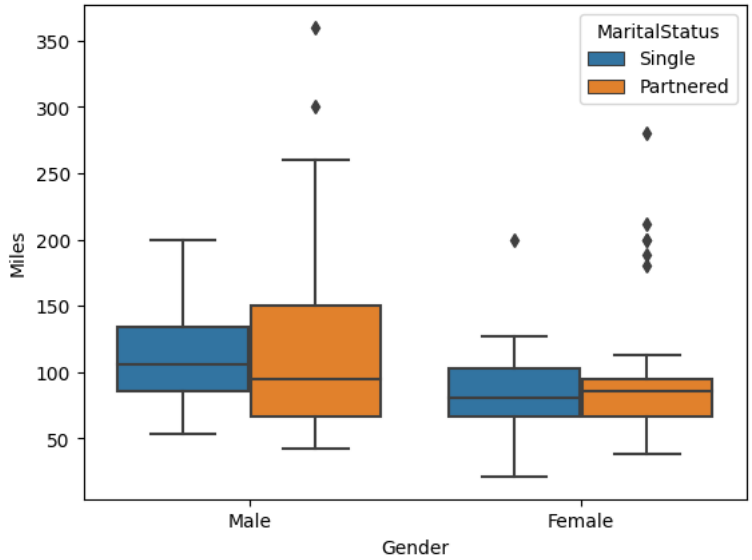


From the above plot it is evident that the expensive variant (KP781) is preferred by men and women who run more miles.

Checking Whether Gender and Marital Status has any impact on miles run

```
sns.boxplot(x="Gender", y="Miles", hue = "MaritalStatus", data=df)
```

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

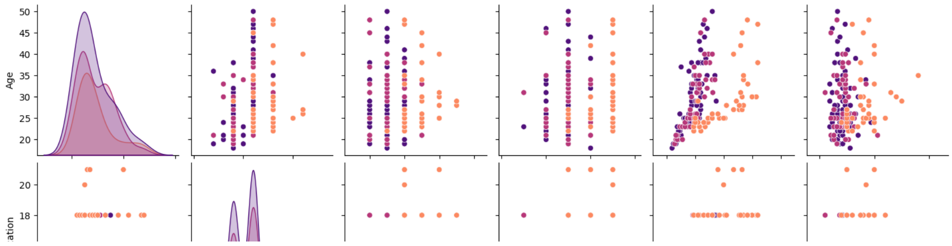


From the above graph it is obvious that there is no visible difference in miles run based on marital status. Males in general run more than females irrespective of marital status.

Multivariate Analysis

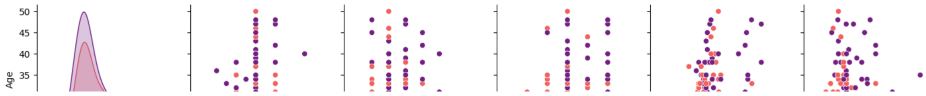
```
# Product analysis
plt.figure(figsize = (12, 10))
sns.pairplot(df, hue = "Product", palette = "magma")
plt.show()
```

```
<Figure size 1200x1000 with 0 Axes>
```



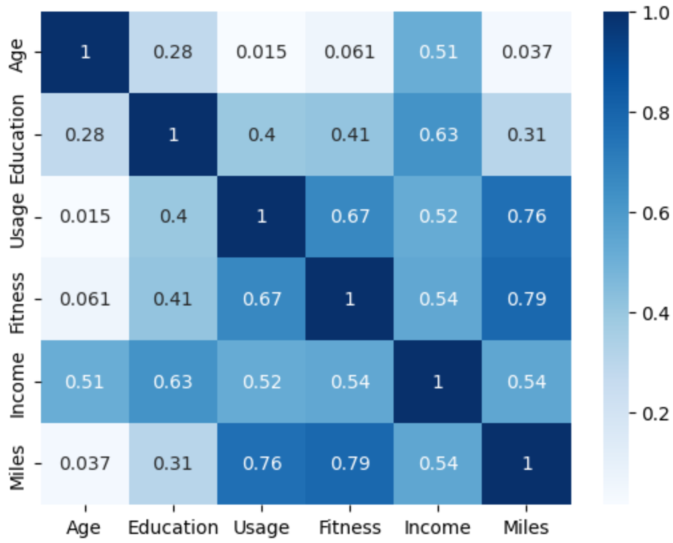

```
# Gender analysis
plt.figure(figsize = (12, 10))
sns.pairplot(df, hue = "Gender", palette = "magma")
plt.show()
```

```
<Figure size 1200x1000 with 0 Axes>
```



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

```
<ipython-input-33-10a01113c88c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr i
  sns.heatmap(df.corr(), annot=True,cmap="Blues")
```



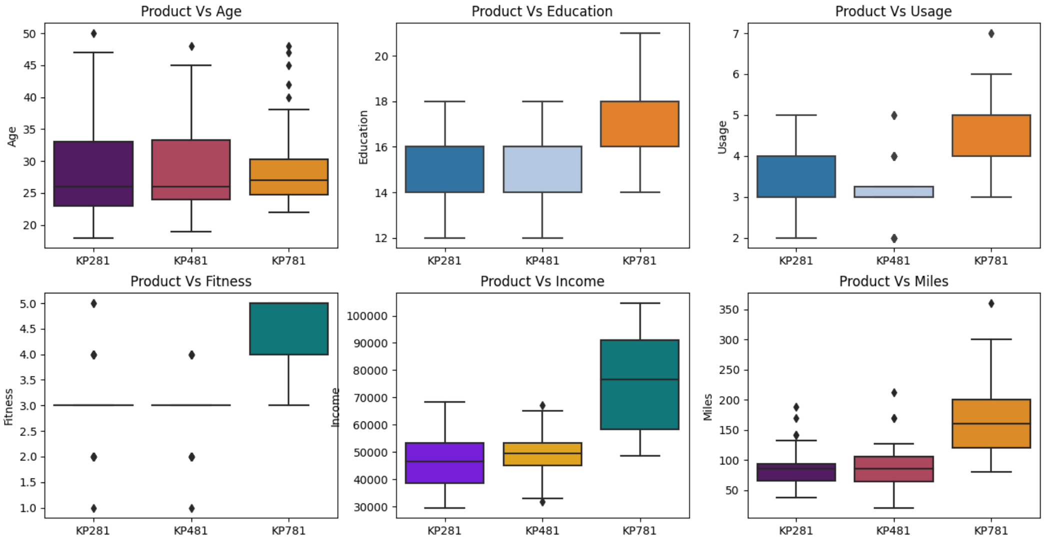
Inference:

- Usage, Fitness are strongly correlated to miles (correlation factor is 0.76 & 0.79 respectively).
- Education and income are strongly correlated to each other (correlation factor is 0.63).
- Age & fitness is weakly correlated (correlation factor is 0.015).

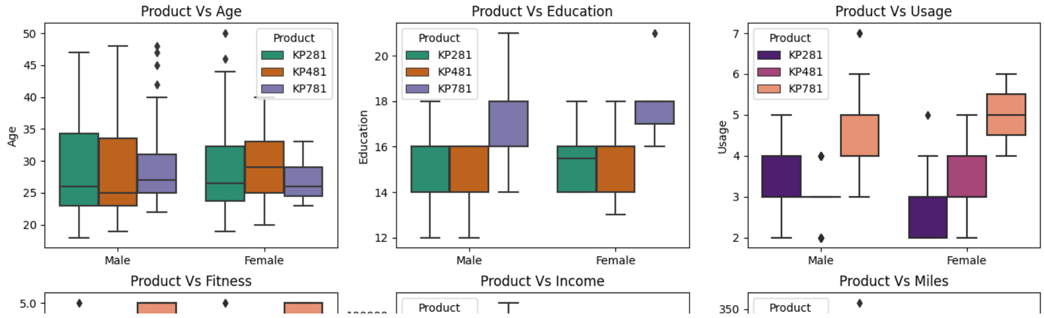
Visualisations using box plots

```
# Product vs Various Factors (Age, Education, Usage, Fitness, Income, Miles)

plt.figure(figsize = (16, 8))
plt.subplot(2, 3, 1)
sns.boxplot(data = df, x = "Product", y = "Age", palette = "inferno")
plt.title("Product Vs Age", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 2)
sns.boxplot(data = df, x = "Product", y = "Education", palette = "tab20")
plt.title("Product Vs Education", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 3)
sns.boxplot(data = df, x = "Product", y = "Usage", palette = "tab20")
plt.title("Product Vs Usage", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 4)
sns.boxplot(data = df, x = "Product", y = "Fitness", palette = "prism_r")
plt.title("Product Vs Fitness", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 5)
sns.boxplot(data = df, x = "Product", y = "Income", palette = "prism_r")
plt.title("Product Vs Income", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 6)
sns.boxplot(data = df, x = "Product", y = "Miles", palette = "inferno")
plt.title("Product Vs Miles", fontsize = 12)
plt.xlabel("")
plt.show()
```

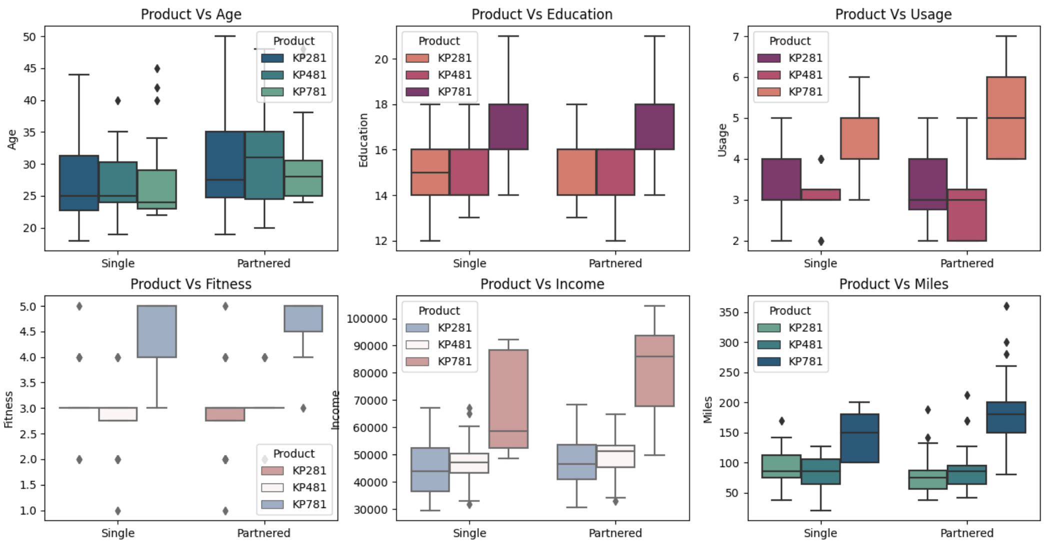



```
# Gender vs Variuos factor (Age, Education, Usage, Fitness, Income, Miles)
plt.figure(figsize = (16, 8))
plt.subplot(2, 3, 1)
sns.boxplot(data = df, x = "Gender", y = "Age", hue = "Product", palette = "Dark2")
plt.title("Product Vs Age", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 2)
sns.boxplot(data = df, x = "Gender", y = "Education", hue = "Product", palette = "Dark2")
plt.title("Product Vs Education", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 3)
sns.boxplot(data = df, x = "Gender", y = "Usage", hue = "Product", palette = "magma")
plt.title("Product Vs Usage", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 4)
sns.boxplot(data = df, x = "Gender", y = "Fitness", hue = "Product", palette = "magma")
plt.title("Product Vs Fitness", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 5)
sns.boxplot(data = df, x = "Gender", y = "Income", hue = "Product", palette = "crest")
plt.title("Product Vs Income", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 6)
sns.boxplot(data = df, x = "Gender", y = "Miles", hue = "Product", palette = "crest")
plt.title("Product Vs Miles", fontsize = 12)
plt.xlabel("")
plt.show()
```




```
# Marital Status vs Variuos factor (Age, Education, Usage, Fitness, Income, Miles)

plt.figure(figsize = (16, 8))
plt.subplot(2, 3, 1)
sns.boxplot(data = df, x = "MaritalStatus", y = "Age", hue = "Product", palette = "crest_r")
plt.title("Product Vs Age", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 2)
sns.boxplot(data = df, x = "MaritalStatus", y = "Education", hue = "Product", palette = "flare")
plt.title("Product Vs Education", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 3)
sns.boxplot(data = df, x = "MaritalStatus", y = "Usage", hue = "Product", palette = "flare_r")
plt.title("Product Vs Usage", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 4)
sns.boxplot(data = df, x = "MaritalStatus", y = "Fitness", hue = "Product", palette = "vlag_r")
plt.title("Product Vs Fitness", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 5)
sns.boxplot(data = df, x = "MaritalStatus", y = "Income", hue = "Product", palette = "vlag")
plt.title("Product Vs Income", fontsize = 12)
plt.xlabel("")
plt.subplot(2, 3, 6)
sns.boxplot(data = df, x = "MaritalStatus", y = "Miles", hue = "Product", palette = "crest")
plt.title("Product Vs Miles", fontsize = 12)
plt.xlabel("")
plt.show()
```



Analysis using Marginal, Joint & Conditional Probability -

Product VS Marital Status

```
# Product vs Marital Status -
```

```
pd.crosstab(df["MaritalStatus"], df["Product"], margins = True)
```

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	48	36	23	107
Single	32	24	17	73
All	80	60	40	180

```
# Joint Probability Table -
pd.crosstab(index = df["MaritalStatus"], columns = df["Product"], margins = True, normalize = True).round(2) * 100
```

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

```
# Conditional Probability Table -
pd.crosstab(df["MaritalStatus"], df["Product"], margins = True, normalize = "index").round(2) * 100
```

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: Interpretation of above table output in terms of probabilities :-

- 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:

> > Out of all the customers 41% are single and 59% are partnered. Out of all the customers who are single, 23% bought KP781, 33% bought KP481, 44% bought KP281. Out of all the customers who are partnered, 21% bought KP781, 34% bought KP481, 45% bought KP281

Product VS Income

For this we will create different income groups associated with different income.

```
df["Income_slab"] = df["Income"]
df.head()
```

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

```
# 0-21 -> Teen
# 22-35 -> Adult
```

```
# 36-45 -> Middle Age
# 46-60 -> Elder Age
df["Income_slab"] = pd.cut(df["Income_slab"], bins = [25000,50000,75000,105000],

labels = ["Low", "Average", "High"])
```

```
df.head()
```

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

```
df["Income_slab"].unique()
```

```
['Low', 'Average', 'High']
Categories (3, object): ['Low' < 'Average' < 'High']
```

```
df["Income_slab"].value_counts()
```

```
Low        83
Average    76
High       21
Name: Income_slab, dtype: int64
```

```
pd.crosstab(df["Income_slab"], df["Product"], margins = True)
```

Product	KP281	KP481	KP781	All
Income_slab				
Low	48	30	5	83
Average	32	30	14	76
High	0	0	21	21
All	80	60	40	180

```
# Joint Probability Table -
pd.crosstab(index = df["Income_slab"], columns = df["Product"], margins = True, normalize = True).round(2) * 100
```

Product	KP281	KP481	KP781	All
Income_slab				
Low	27.0	17.0	3.0	46.0
Average	18.0	17.0	8.0	42.0
High	0.0	0.0	12.0	12.0
All	44.0	33.0	22.0	100.0

Inference: Interpretation of above table output in terms of probabilities :-

- P(Low): 0.46
- P(Average): 0.42
- P(High): 0.12
- P(KP781 | Low): 0.06
- P(KP481 | Low): 0.36
- P(KP281 | Low): 0.58
- P(KP781 | Average): 0.18
- P(KP481 | Average): 0.39
- P(KP281 | Average): 0.42
- P(KP781 | High): 0.100
- P(KP481 | High): 0.0
- P(KP281 | High): 0.0

Insights:

> > Out of all the customers 46% are Low income and 42% is Average income and 12% are High income. Out of all the customers who are Low income, 6% bought KP781, 36% bought KP481, 58% bought KP281. Out of all the customers who are Average income, 18% bought KP781, 39% bought KP481, 42% bought KP281 Out of all the customers who are High income, 100% bought KP781.

Business Insights -

> > KP281
> >
> > - This product is an affordable model
> > - Income range between 38K to 52K have preferred this product.
> > - This Product is used 3 to 4 times a week mostly
> > - Beginner level customers of all age groups prefer this product.
> > - Average distance covered on this model is around 70 to 90 miles.

> > KP481
> >
> > - KP481 is the second most popular product among the customers.
> > - Average income of customers buying KP481 is 49K.
> > - This product is preffered among married customers prefer this product.
> > - Average distance covered on this product is from 72 to 131 miles per week.
> > - The age range of KP481 treadmill customers stands roughly between 25-36 years.
> > - Percentage of female customer buying KP481 is significantly higher than male.

> > KP781
> >
> > - Due to the High Price, this is the least preferred product.
> > - Customers use this variant mostly around 5 to 6 times a week at least.
> > - Customers using this variant run more.
> > - This product is preferred by customers where the correlation between Education and Income is High.
> > - Average Income of KP781 buyers are approximately over 75K per annum
> > - Percentage of Male customer buying Product KP781(31.73%) is way more than female(9.21%).
> > - Partnered Female bought KP781 treadmill compared to Partnered Male.

> > ## Recommendations

1) Price of KP781 being higher than the other variant, high income groups need to be targetted.

2) As KP281 & KP481 are budget treadmills, so aerofit need to increase their sale with respect to this equipments so as to increase profit & revenue.

3) Usage of female customers is low as compared to men. so aerofit need to give some special offers, so that more women tend to buy the treadmill of any category.

4) Treadmill KP781 will always get good revenue & profit if aerofit launches it in the sports event as it will be readily sold out due to the huge requirement in various kind of sports.

5) Target the Age group above 40 years to recommend Product KP781.

6) Aerofit can do online marketing to increase their sale with respect to every equipment especially for KP781 which is lower sold out equipment.

7) Aerofit should upgrade or provide instant customer service if there is problem with equipment installation or working.

8) Aerofit should think to expand their business in future years to launch new equipment other than the current equipments so as to increase the sale.