AeroFit-MySolution

September 15, 2022

1 Problem Statement

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.

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

2 1. Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset

[2]:
```
df = pd.read_csv("aerofit_treadmill.txt", sep=",")
df.head()
```

[2]:

	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

[3]:
```
df.shape
```

[3]:
```
(180, 9)
```

2.0.1 Total 180 customers and 9 characterstics

[4]:
```
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
```

2.0.2 Product, Gender and MaritalStatus have Categorical data

[5]:
```
df.nunique()
```

[5]:
```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

[6]:
```
print(df.Product.value_counts()/len(df))
print(df.Gender.value_counts()/len(df))
print(df.MaritalStatus.value_counts()/len(df))
```

```
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: Product, dtype: float64
Male      0.577778
Female    0.422222
Name: Gender, dtype: float64
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
```

2.0.3 Categorical data, Product has 3 types, Gender and MaritalStatus has 2 unique values.

2.0.4 Most users, around 44% uses KP281 machine

2.0.5 More male customers, approx. 57%

2.0.6 40.55% customers are single

[7]:
```
df.isna().sum()
```

[7]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

2.0.7 There are no missing values in the data

[8]:
```
df.describe(include="all")
```

[8]:
```
       Product         Age Gender   Education MaritalStatus       Usage  \
count      180  180.000000    180  180.000000           180  180.000000
unique       3         NaN      2         NaN             2         NaN
top      KP281         NaN   Male         NaN     Partnered         NaN
freq        80         NaN    104         NaN           107         NaN
mean       NaN   28.788889    NaN   15.572222           NaN    3.455556
std        NaN    6.943498    NaN    1.617055           NaN    1.084797
min        NaN   18.000000    NaN   12.000000           NaN    2.000000
25%        NaN   24.000000    NaN   14.000000           NaN    3.000000
50%        NaN   26.000000    NaN   16.000000           NaN    3.000000
75%        NaN   33.000000    NaN   16.000000           NaN    4.000000
max        NaN   50.000000    NaN   21.000000           NaN    7.000000

           Fitness         Income       Miles
count   180.000000     180.000000  180.000000
unique         NaN            NaN         NaN
top            NaN            NaN         NaN
freq           NaN            NaN         NaN
mean      3.311111   53719.577778  103.194444
std       0.958869   16506.684226   51.863605
min       1.000000   29562.000000   21.000000
25%       3.000000   44058.750000   66.000000
50%       3.000000   50596.500000   94.000000
```

```
75%        4.000000   58668.000000  114.750000
max        5.000000  104581.000000  360.000000
```

2.0.8 Statistical Summary

2.0.9 -KP281 is most popular product with frequency 80

2.0.10 -Mean for Age is 28.79 and Median Age is 26

2.0.11 -Aerofit Products has more Male customers with frequency 104

2.0.12 -Mean Education is 15.57 while Median Education is 16

2.0.13 -Most of the Customers have partners with frequency 107

2.0.14 -Mean usage of this products is 3.45 and median is 3 days/week

2.0.15 -Mean fitness is 3.45 and Median is 3

2.0.16 -Mean Income is 53719.577778. Standard deviation of Income is high so it may contain outliers

2.0.17 -Miles per week has mean value 103.19 and median value is 94

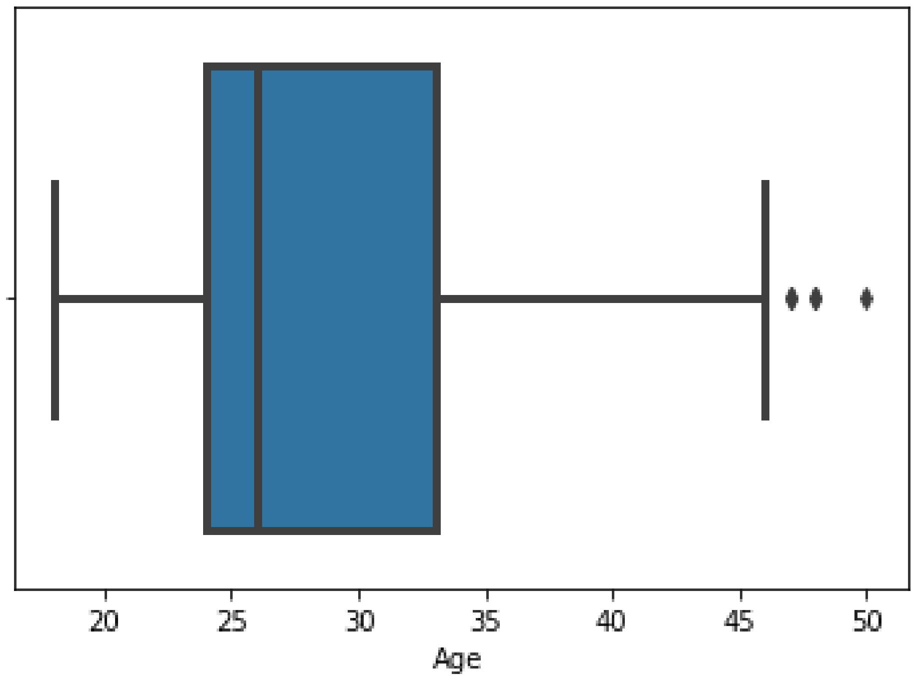
3 Question 2. Detect Outliers (using boxplot, "describe" method by checking the difference between mean and median)

[9]:
```
df.Age.describe()
```

[9]:
```
count    180.000000
mean      28.788889
std        6.943498
min       18.000000
25%       24.000000
50%       26.000000
75%       33.000000
max       50.000000
Name: Age, dtype: float64
```

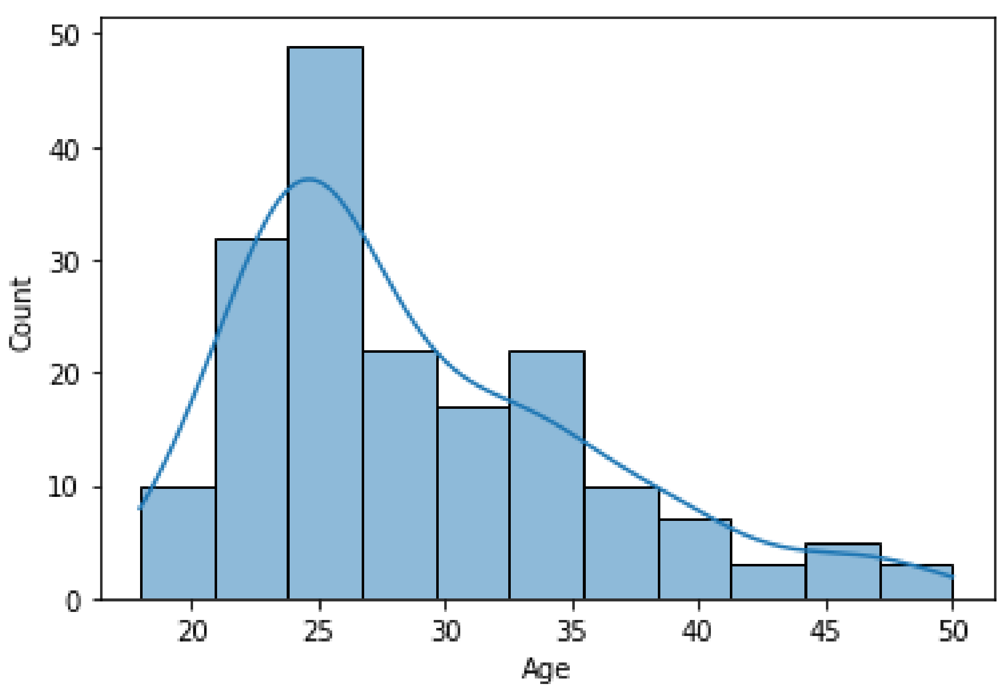
[10]:
```
sns.boxplot(x=df.Age, linewidth=3)
```

[10]:
```
<AxesSubplot:xlabel='Age'>
```



[11]:
```
sns.histplot(x=df.Age, kde=True)
```

[11]: <AxesSubplot:xlabel='Age', ylabel='Count'>



3.0.1 For age, In Boxplot Median(26) is to the left of Mean(28.788), so it is right skewed. Same can be seen using histogram and KDE. Some outliers are also visible

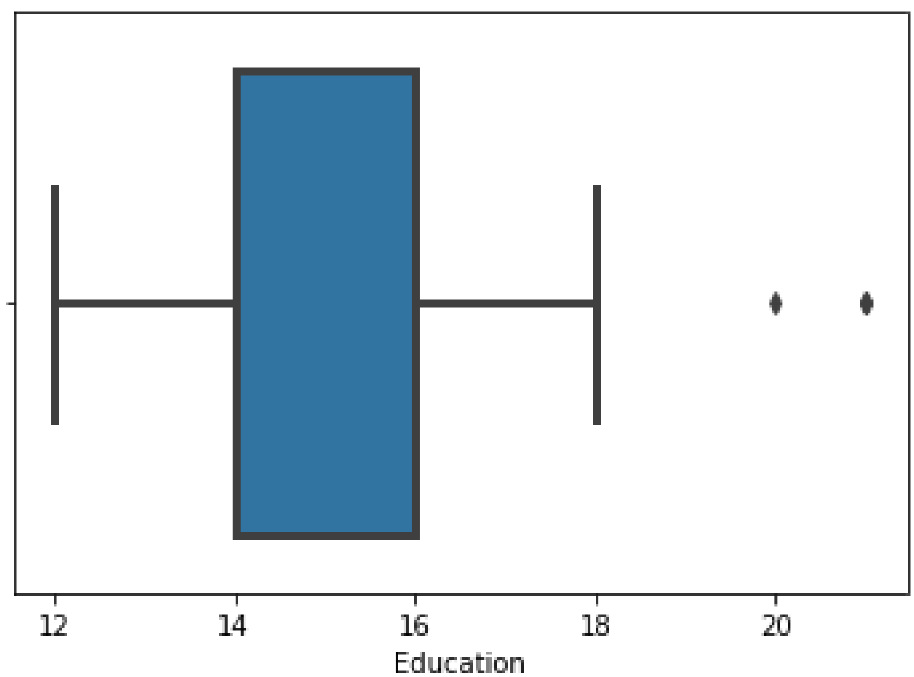
3.0.2 Most of the customers buying tredmills have age in the range 24 to 33

[12]:
```
df.Education.describe()
```

[12]:
```
count    180.000000
mean      15.572222
std        1.617055
min       12.000000
25%       14.000000
50%       16.000000
75%       16.000000
max       21.000000
Name: Education, dtype: float64
```

[13]:
```
sns.boxplot(x=df.Education, linewidth=3)
```

[13]: <AxesSubplot:xlabel='Education'>

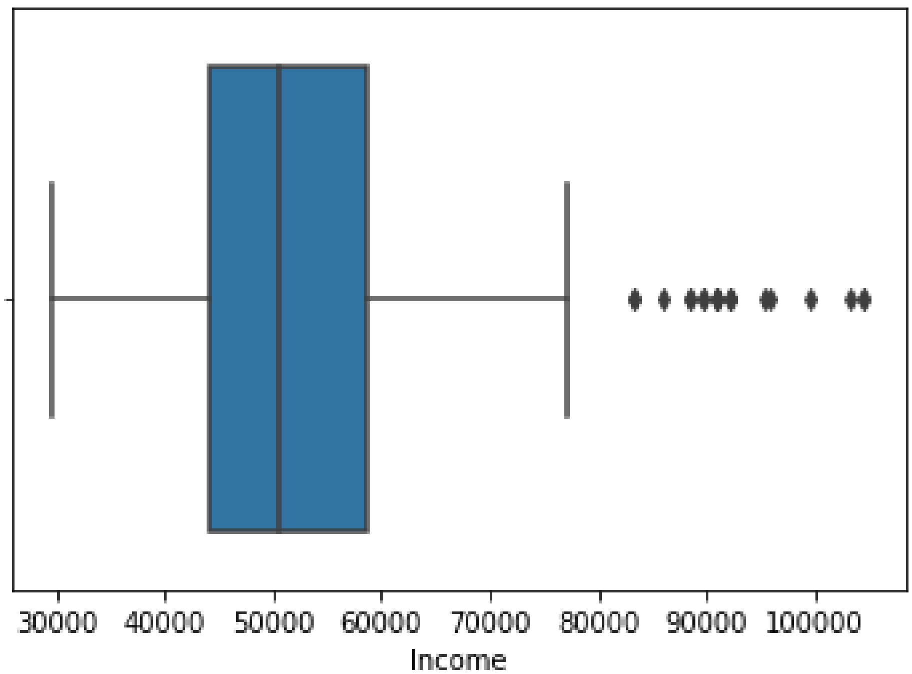


[14]:
```
df.Income.describe()
```

[14]:
```
count       180.000000
mean      53719.577778
std       16506.684226
min       29562.000000
25%       44058.750000
50%       50596.500000
75%       58668.000000
max      104581.000000
Name: Income, dtype: float64
```

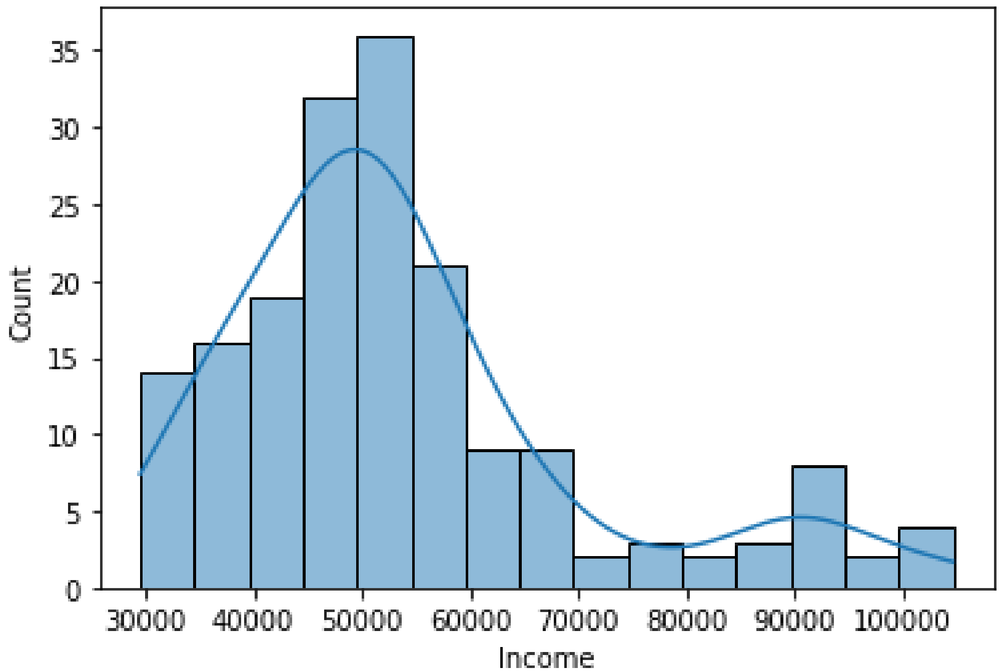
[15]:
```
sns.boxplot(x=df.Income)
```

[15]:
```
<AxesSubplot:xlabel='Income'>
```



[16]:
```
sns.histplot(x=df.Income, kde=True)
```

[16]:
```
<AxesSubplot:xlabel='Income', ylabel='Count'>
```



3.0.3 Income, has a lot of outliers. Median is less than mean, right skewed

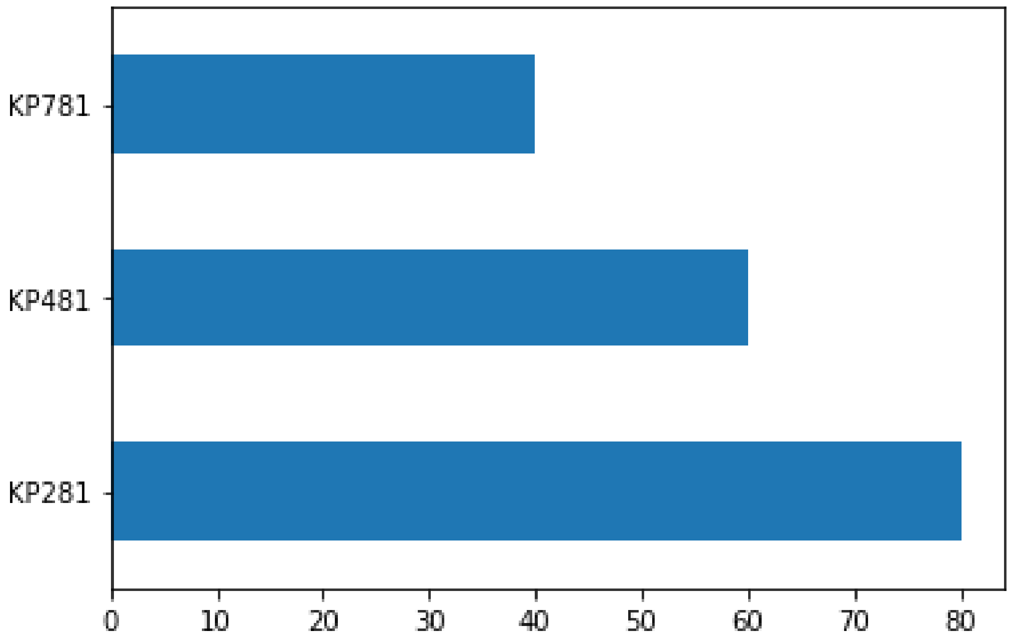
[17]:
```
print(df.Product.unique())
print(df.Gender.unique())
print(df.MaritalStatus.unique())
```

```
['KP281' 'KP481' 'KP781']
['Male' 'Female']
['Single' 'Partnered']
```

3.1 UniVariate Analysis

[18]:
```
df.Product.value_counts().plot(kind='barh')
```

[18]: <AxesSubplot:>



[19]:
```
df.Product.value_counts()/len(df)
```

[19]:
```
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: Product, dtype: float64
```

3.1.1 KP281 is the most used product, having percentage of 44% among all

[20]:
```
df.Gender.value_counts()/len(df)*100
```

[20]:
```
Male      57.777778
Female    42.222222
Name: Gender, dtype: float64
```

3.2 57% of the customers are Male

[21]:
```
df.MaritalStatus.value_counts()/len(df)
```

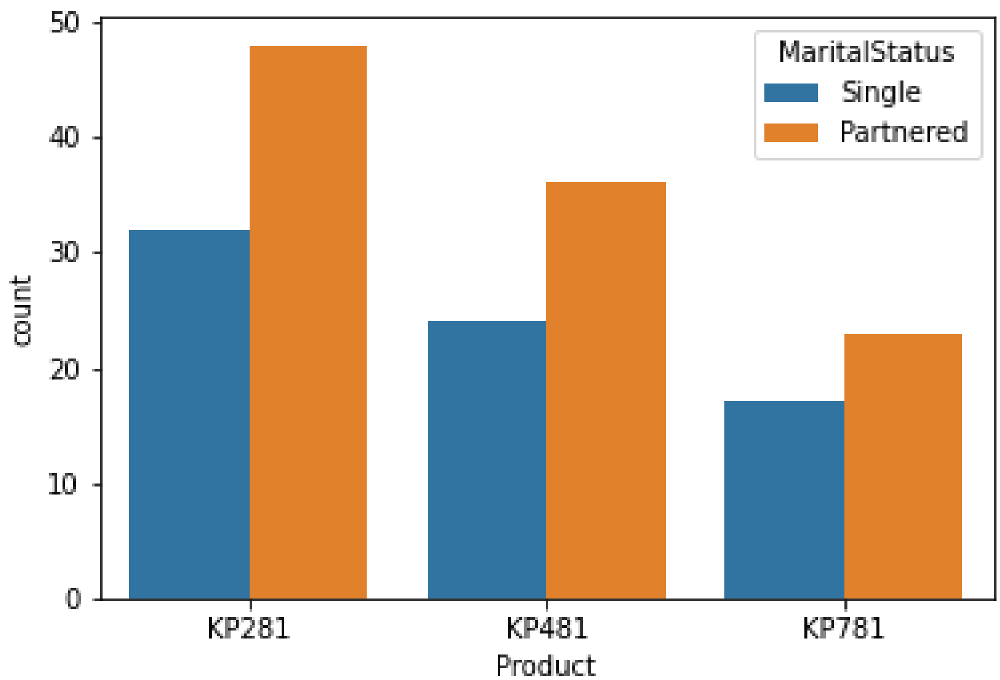
[21]:
```
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
```

3.2.1 Around 40.55% of the customers are Single and 59.44% have partners

4 Question3. Check if features like marital status, age have any effect on the product purchased (using countplot, histplots, boxplots etc)

[22]:
```
sns.countplot(hue=df.MaritalStatus,  x=df.Product)
```

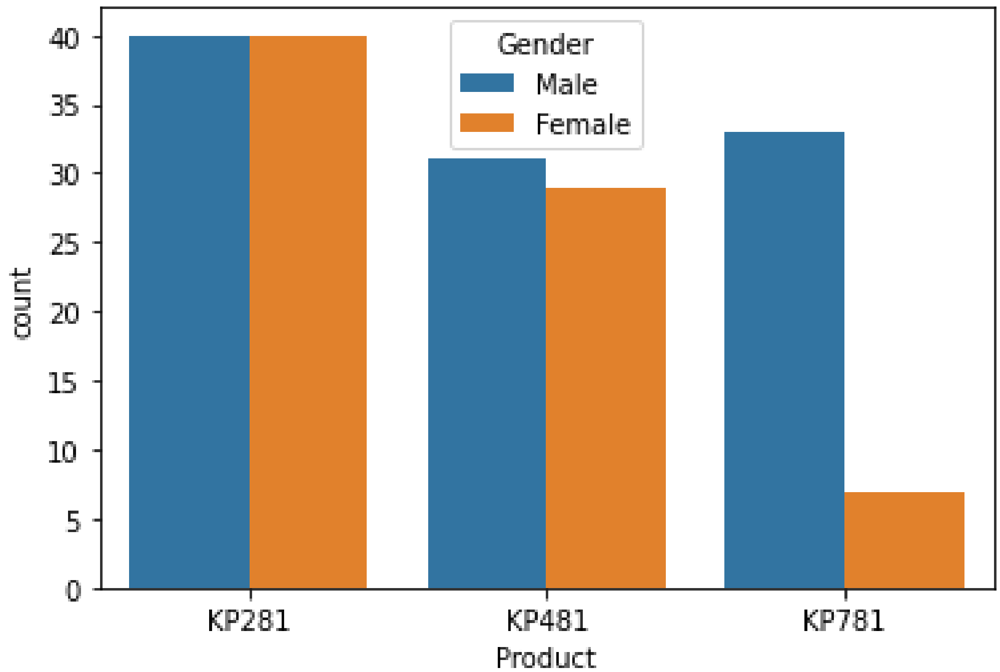
[22]: <AxesSubplot:xlabel='Product', ylabel='count'>



4.0.1 For each product, customers who are single are less

[23]:
```
sns.countplot(hue=df.Gender,  x=df.Product)
```

[23]: <AxesSubplot:xlabel='Product', ylabel='count'>



[24]:
```
df.groupby(["Product", "Gender"]).size()
```

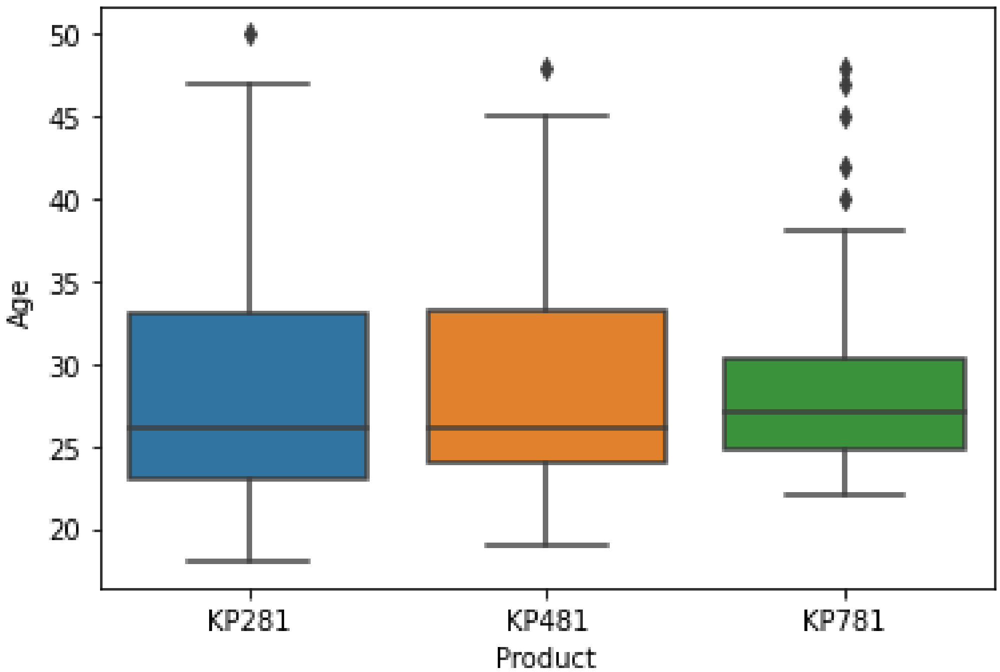
[24]:
```
Product  Gender
KP281    Female    40
         Male      40
KP481    Female    29
         Male      31
KP781    Female     7
         Male      33
dtype: int64
```

4.0.2 For KP781, High number of Male customers can be seen.

For other models, male and female customers are almost same

[25]:
```
sns.boxplot(x=df.Product, y=df.Age)
```

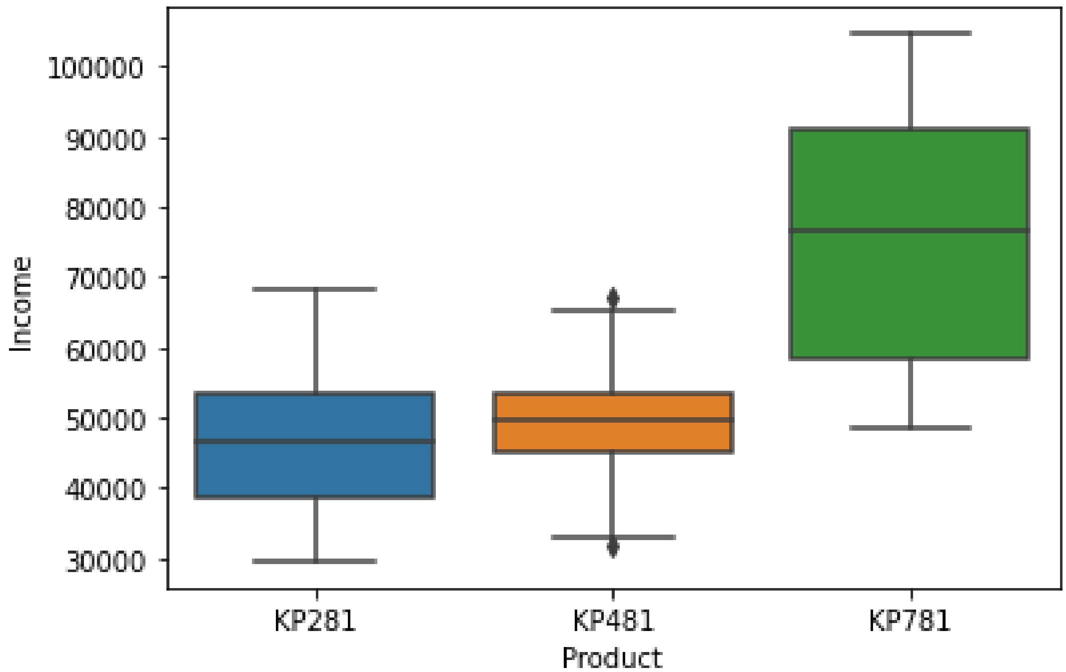
[25]:
```
<AxesSubplot:xlabel='Product', ylabel='Age'>
```



4.0.3 Customers who are purchasing KP281 and KP481 have almost same median age, around 26

[26]:
```
sns.boxplot(x=df.Product, y=df.Income)
```

[26]: <AxesSubplot:xlabel='Product', ylabel='Income'>



4.0.4 Customers having income greater than approx. 59K dollars are more likely to buy KP781 while other customers have more chances to go for KP281 orKP481 treadmill

5 Question4. Representing the marginal probability like - what percent of customers have purchased KP281, KP481, or KP781 in a table (can use pandas.crosstab here)

[27]:
```
x = pd.DataFrame(df.Product.value_counts()/len(df))
x.reset_index(inplace=True)
x.columns = ["Product", "Marginal Prob"]
x
```

[27]:

	Product	Marginal Prob
0	KP281	0.444444
1	KP481	0.333333
2	KP781	0.222222

[28]:
```
pd.crosstab(df.Gender, df.Product, normalize='index', margins=True)
```

[28]:

Product	KP281	KP481	KP781
Gender			
Female	0.526316	0.381579	0.092105
Male	0.384615	0.298077	0.317308

```
All      0.444444  0.333333  0.222222
```

[29]:
```
pd.crosstab(df.MaritalStatus, df.Product, normalize='index', margins=True)
```

[29]:
```
Product           KP281     KP481     KP781
MaritalStatus
Partnered      0.448598  0.336449  0.214953
Single         0.438356  0.328767  0.232877
All            0.444444  0.333333  0.222222
```

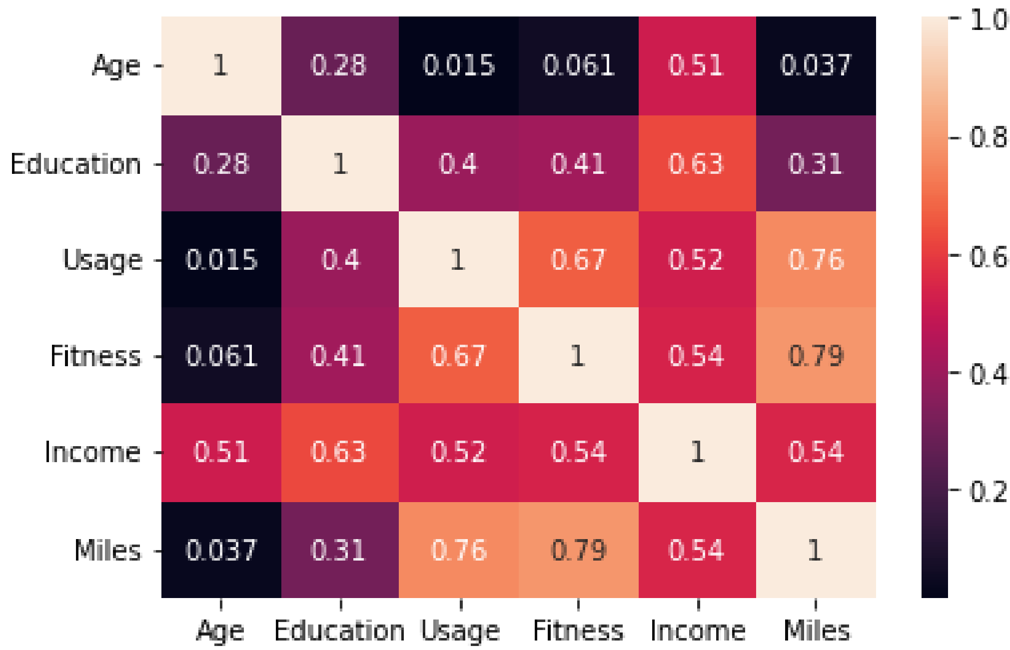
6 Question5. Check correlation among different factors using heat maps or pair plots.

[30]:
```
df.corr()
```

[30]:
```
                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
```

[31]:
```
sns.heatmap(df.corr(), annot=True,  fmt=".2g")
```

[31]: <AxesSubplot:>



6.0.1 Income is highly correlated to Education, Fitness, Age, Usage and Miles

6.0.2 Education has high correlation with Income as well as Fitness

6.0.3 Usage is highly correlated with Miles, Income and Fitness

6.0.4 Fitness is highly correlated with Miles, usage and Income

6.0.5 Miles are highly correlated with Usage and fitness

7 Question6. With all the above steps you can answer questions like: What is the probability of a male customer buying a KP781 treadmill?

[32]:
```
pd.crosstab(index=df.Gender,columns=df.Product, margins=True)
```

[32]:
```
Product  KP281  KP481  KP781  All
Gender
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180
```

[33]:
```
### Male customer buying a KP781 treamill = 33/104, as out of 104 Male␣
↪customers, 33 bought KP781
33/104
```

[33]:
```
0.3173076923076923
```

8 Question7. Customer Profiling - Categorization of users.

From earlier observations

1. KP281
    - Approx. 44% customers bought this product
    - Male and Female equally bought this product
    - Mostly customers who had partners bought this
2. KP481
    - Approx. 33% customers purchased this product.
    - Males bought this product slightly more than Females
    - Most customers had partners
3. KP781
    - Approx. 22% customers purchased this.
    - Males customers were very high in comparision to female customers
    - Most of the customers had partners

9 Question8. Probability- marginal, conditional probability.

[34]:
```
marginal = pd.DataFrame(df.Product.value_counts()/len(df))
marginal.reset_index(inplace=True)
marginal.columns=["Product", "Marginal Probablity"]
marginal
```

[34]:

	Product	Marginal Probablity
0	KP281	0.444444
1	KP481	0.333333
2	KP781	0.222222

[35]:
```
pd.crosstab(index=df.MaritalStatus,columns=df.Product, margins=True)
```

[35]:

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

[36]:
```
pd.crosstab(index=df.MaritalStatus,columns=df.Product, margins=True,␣
↪normalize='index')
```

[36]:

Product	KP281	KP481	KP781
MaritalStatus			
Partnered	0.448598	0.336449	0.214953
Single	0.438356	0.328767	0.232877
All	0.444444	0.333333	0.222222

9.0.1 Conditional Probablities of each product given Marital Status

9.0.2 P(KP281|Partnered) = 0.44

9.0.3 P(KP481|Partnered) = 0.33

9.0.4 P(KP781|Partnered) = 0.21

9.0.5 P(KP281|Single) = 0.43

9.0.6 P(KP481|Single) = 0.32

9.0.7 P(KP781|Single) = 0.23

10 Recommendations

1. For KP781, female customers were very less and most users were male. Some offers can be provided to attract female customers.
2. Across all the products, customers who were single were less. Some fitness campaigns can be run in Universities to make them aware about fitness
3. KP281 and KP481 had customers with less income. Their cost is also less so these can be markted as budget models and they can attract even more middle class customers.

4. As KP781 is expensive and has less customers and even less female customers. To attract females and more customers, it's extra features and benefits should be advertised properly

```
[ ]:
```