business-case-aerofit

December 7, 2023

1 Business Case: Aerofit - Descriptive Statistics & Probability

---

---

2 About Aerofit

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.

3 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.

4 1. Defining Problem Statement and Analysing basic metrics

Import Libraries

Importing the libraries we need

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

5 Loading The Dataset

[2]:
```
df = pd.read_csv("aerofit_treadmill.csv")
df
```

[2]:
```
    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]
```

6 Basic Analysis

7 Shape of Data

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

[3]: (180, 9)

Analysis

1. The shape of Dataframe is 180 * 9
2. No. of rows = 180
3. No. of columns = 9

Columns in Dataframe

[4]:
```
df.columns
```

[4]:
```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

Let's check the first 5 data

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

[5]:

	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

Let's check the last 5 data

[6]:
```
df.tail()
```

[6]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	\
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
175	200
176	200
177	160
178	120
179	180

Data type of all attributes(Columns)

[7]:
```
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
```

Statistical Summary of object type columns

[8]:
```python
# statisctical summary of object type columns

df.describe(include ='object').T
```

[8]:

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

Insights

1. **Product** - Over the past three months, the KP281 product demonstrated the highest sales performance among the three products, accounting for approximately 44% of total sales.
2. **Gender** - Based on the data of last 3 months, around 58% of the buyers were Male and 42% were female
3. **Marital Status** - Based on the data of last 3 months, around 60% of the buyers were Married and 40% were single

Statistical Summary of Numeric columns

[9]:
```python
# statisctical summary of numerical data type columns

df.describe().T
```

[9]:

	count	mean	std	min	25%	50%	\
Age	180.0	28.788889	6.943498	18.0	24.00	26.0	
Education	180.0	15.572222	1.617055	12.0	14.00	16.0	
Usage	180.0	3.455556	1.084797	2.0	3.00	3.0	
Fitness	180.0	3.311111	0.958869	1.0	3.00	3.0	
Income	180.0	53719.577778	16506.684226	29562.0	44058.75	50596.5	
Miles	180.0	103.194444	51.863605	21.0	66.00	94.0	

	75%	max
Age	33.00	50.0
Education	16.00	21.0
Usage	4.00	7.0
Fitness	4.00	5.0

```
Income     58668.00  104581.0
Miles        114.75     360.0
```

Insights

1. **Age** - The age range of customers spans from 18 to 50 year, with an average age of 29 years.
2. **Education** - Customer education levels vary between 12 and 21 years, with an average education duration of 16 years.
3. **Usage** - Customers intend to utilize the product anywhere from 2 to 7 times per week, with an average usage frequency of 3 times per week.
4. **Fitness** - On average, customers have rated their fitness at 3 on a 5-point scale, reflecting a moderate level of fitness.
5. **Income** - The annual income of customers falls within the range of USD 30,000 to USD 100,000, with an average income of approximately USD 54,000.
6. **Miles** - Customers' weekly running goals range from 21 to 360 miles, with an average target of 103 miles per week.
7. Minimum & Maximum age of the person is 18 & 50, mean is 28.79 and 75% of persons have age less than or equal to 33.
8. Most of the people are having 16 years of education i.e. 75% of persons are having education <= 16 years.
9. Out of 180 data points, 104's gender is Male and rest are the female.
10. Standard deviation for Income & Miles is very high. These variables might have the outliers in it.
11. There are 180 rows and 9 columns.

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

Duplicate Detection

[10]:
```
df.duplicated().value_counts()
```

[10]:
```
False    180
dtype: int64
```

Insights

1. There are no duplicate entries in the dataset.

Value Count check for Columns

9 Product Column

Unique

[11]:
```
df["Product"].unique()
```

[11]:
```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

Insight

Aerofit produces three treadmill models **KP281, KP481, KP781**.

[12]:
```
df["Product"].nunique()
```

[12]:
```
3
```

Insight > There are 3 unique products available in the dataset.

Value Count

[13]:
```
product_count=df["Product"].value_counts(normalize = True) * 100
product_count.round(2)
```

[13]:
```
KP281    44.44
KP481    33.33
KP781    22.22
Name: Product, dtype: float64
```

Insight

Among the users, 44.44% prefer using KP281 treadmill, while 33.33% opt for the KP481 treadmill and only 22.22% of user favour the KP781 treadmill.

[14]:
```
df.head()
```

[14]:

	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

10 Gender Column

Unique

[15]:
```
df["Gender"].unique()
```

[15]:
```
array(['Male', 'Female'], dtype=object)
```

[16]:
```
df["Gender"].nunique()
```

[16]:
```
2
```

Insight

Data represents details for only male and female.

Value Count

[17]:
```
gender_count=df["Gender"].value_counts(normalize=True)*100
gender_count.round(2)
```

[17]:
```
Male      57.78
Female    42.22
Name: Gender, dtype: float64
```

[18]:
```
df['Gender'].value_counts()
```

[18]:
```
Male      104
Female     76
Name: Gender, dtype: int64
```

Insight

Aerofit data has 57.78%(104) male and 42.22% (76) female available in the dataset.

11 MaritalStatus Column

Unique

[19]:
```
df["MaritalStatus"].unique()
```

[19]:
```
array(['Single', 'Partnered'], dtype=object)
```

[20]:
```
df["MaritalStatus"].nunique()
```

[20]:
```
2
```

Insight

Dataset have data only for Single and Partnered.

Value Count

[21]:
```
Maritalstatus_count=df["MaritalStatus"].value_counts(normalize=True)*100
Maritalstatus_count.round(2)
```

[21]:
```
Partnered    59.44
Single       40.56
Name: MaritalStatus, dtype: float64
```

[22]:
```
df["MaritalStatus"].value_counts()
```

[22]:
```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

Insight

59.44% of Aerofit customers are married while the remaining 40.56% are single.

Unique Values check for all columns

[23]:
```
# checking the unique values for columns
for i in df.columns:
    print('Unique Values in',i,'column are :-')
    print(df[i].unique())
    print('='*70)
```

```
Unique Values in Product column are :-
['KP281' 'KP481' 'KP781']
======================================================================
Unique Values in Age column are :-
[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]
======================================================================
Unique Values in Gender column are :-
['Male' 'Female']
======================================================================
Unique Values in Education column are :-
[14 15 12 13 16 18 20 21]
======================================================================
Unique Values in MaritalStatus column are :-
['Single' 'Partnered']
======================================================================
Unique Values in Usage column are :-
[3 2 4 5 6 7]
======================================================================
Unique Values in Fitness column are :-
[4 3 2 1 5]
======================================================================
Unique Values in Income column are :-
[ 29562  31836  30699  32973  35247  37521  36384  38658  40932  34110
  39795  42069  44343  45480  46617  48891  53439  43206  52302  51165
  50028  54576  68220  55713  60261  67083  56850  59124  61398  57987
  64809  47754  65220  62535  48658  54781  48556  58516  53536  61006
  57271  52291  49801  62251  64741  70966  75946  74701  69721  83416
  88396  90886  92131  77191  52290  85906 103336  99601  89641  95866
 104581  95508]
======================================================================
Unique Values in Miles column are :-
[112  75  66  85  47 141 103  94 113  38 188  56 132 169  64  53 106  95
```

```
 212  42 127  74 170  21 120 200 140 100  80 160 180 240 150 300 280 260
 360]
======================================================================
```

Insights

1. The dataset does not contain any abnormal values.

12 Data Pre-processing

13 Missing Value and Outliers Detection

Handling Mising Values

[24]:
```
df.isnull().sum()
```

[24]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

Insight

There is no missing value present in dataset.

Handling Outliers

Income Column

[25]:
```
df['Income'].describe()
```

[25]:
```
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
```

To find outliers in Income column we need to use **Boxplot** here. but before using boxplot we need to find these 5 points:

1. Q3 - Upper Quartile

2. Q1 - Lower Quartile
3. Median
4. Upper Bound
5. Lower Bound

[26]:
```
q1 = np.percentile(df['Income'],25)
q3 = np.percentile(df['Income'],75)
print( 'q1 =', q1)
print( 'q3 =', q3)
```

```
q1 = 44058.75
q3 = 58668.0
```

Insight

we get

Q1 = 44058.75

Q3 = 58668.0

[27]:
```
#to find upper bound and lower bound we need to find IQR (inter quartile range)

IQR = q3 - q1
IQR
```

[27]: 14609.25

Insight

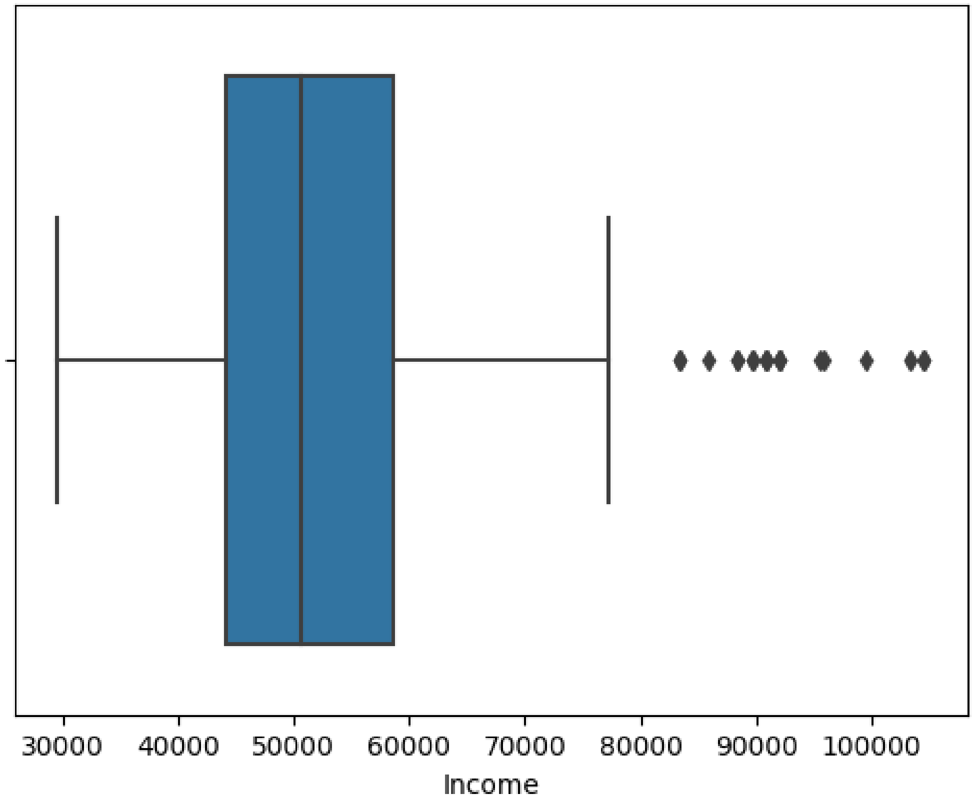
we get

IQR = 14609.25

[28]:
```
upper_bound = q3 + 1.5 * IQR
lower_bound = q1 - 1.5 * IQR
print('Upper Bound =', upper_bound)
print('Lpper Bound =', lower_bound)
print('Median =', df['Income'].median())
```

```
Upper Bound = 80581.875
Lpper Bound = 22144.875
Median = 50596.5
```

[29]:
```
sns.boxplot(data = df, x = 'Income')
plt.show()
```



- As we see there are Outliers in the 'Income' Column
- All values > 80581.75 (Upper Bound) are outliers in the 'Income' Column

[30]:
```
(len(df.loc[df['Income'] > upper_bound]) / len(df)) * 100
```

[30]:
```
10.555555555555555
```

Insight

10.5% in Income column are outliers but we choose not to drop them as these values may required to draw some valuable insights and it may be useful for customer profiling.

Miles Column

[31]:
```
df['Miles'].describe()
```

[31]:
```
count    180.000000
mean     103.194444
std       51.863605
min       21.000000
25%       66.000000
50%       94.000000
75%      114.750000
```

```
max      360.000000
Name: Miles, dtype: float64
```

[32]:

```
q1 = np.percentile(df['Miles'],25)
q3 = np.percentile(df['Miles'],75)
IQR = q3 - q1
print( 'q1 =', q1)
print( 'q3 =', q3)
print('IQR = ', IQR)
```

```
q1 = 66.0
q3 = 114.75
IQR =  48.75
```

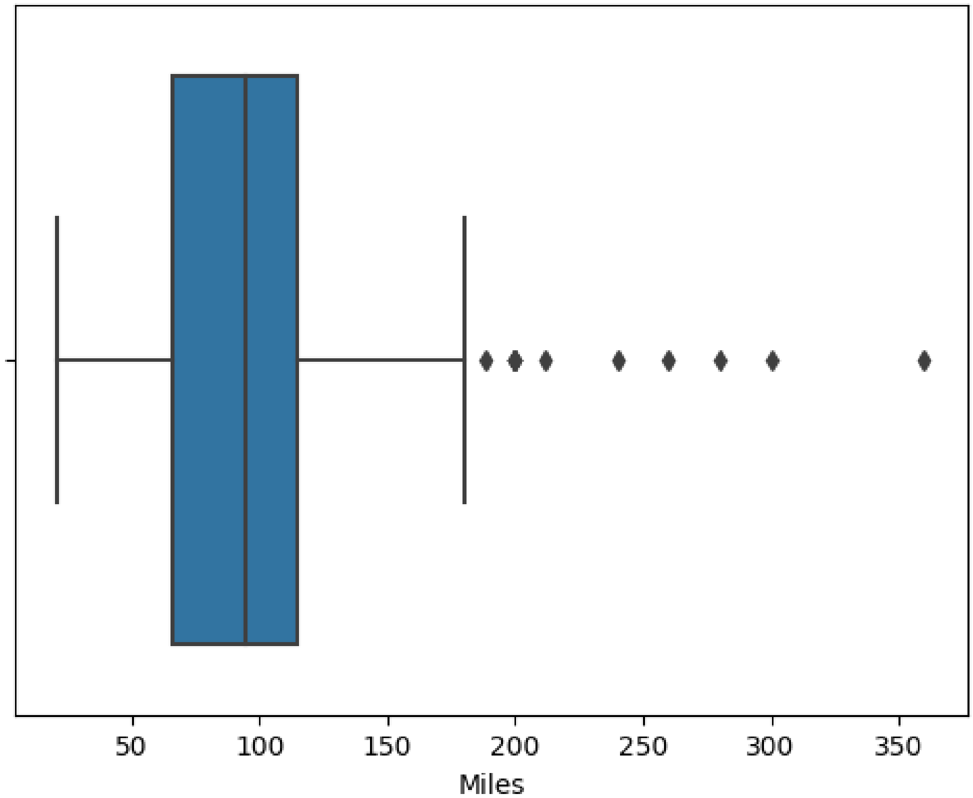
[33]:

```
upper_bound = q3 + 1.5 * IQR
lower_bound = q1 - 1.5 * IQR
print('Upper Bound =', upper_bound)
print('Lpper Bound =', lower_bound)
print('Median =', df['Miles'].median())
```

```
Upper Bound = 187.875
Lpper Bound = -7.125
Median = 94.0
```

[34]:

```
sns.boxplot(data = df, x = 'Miles')
plt.show()
```



- As we see there are Outliers in the 'Miles' Column
- All values > 187.875 (Upper Bound) are outliers in the 'Miles' Column

[35]:
```
(len(df.loc[df['Miles'] > upper_bound]) / len(df)) * 100
```

[35]:
```
7.222222222222221
```

Insight

7.22% in Miles column are outliers but we choose not to drop them as these values may required to draw some valuable insights and it may be useful for customer profiling.

14 Outlier detection using the Z-Score

what is Z score?

Z scores are: z = (x - mean)/std, so values in each row (column) will get the mean of the row (column) subtracted, then divided by the standard deviation of the row (column). This ensures that each row (column) has mean of 0 and variance of 1.

- We can detect outliers in numeric column using the z-score.

- if the Z-score of a data point is more then 3, it inicates the that the data point is quite different from the other data points. Such a data can be a outlier.
- Z score = (x - mean)/std.devation

[36]:
```python
outliers = {}
for col in df.select_dtypes(include = np.number):
  #finding Z-score for each value in a column
  z_score = np.abs((df[col] - df[col].mean())) / df[col].std()

  #if the z score of a value ia greater than 3 then the value is outlier
  column_outliers = df[z_score > 3][col]

  outliers[col] = column_outliers

for col, outliers_values in outliers.items():
  print(f"Outliers for {col} column")
  print(outliers_values)
  print()
```

```
Outliers for Age column
79    50
Name: Age, dtype: int64

Outliers for Education column
157    21
161    21
175    21
Name: Education, dtype: int64

Outliers for Usage column
163    7
166    7
Name: Usage, dtype: int64

Outliers for Fitness column
Series([], Name: Fitness, dtype: int64)

Outliers for Income column
168    103336
174    104581
178    104581
Name: Income, dtype: int64

Outliers for Miles column
166    300
167    280
170    260
```

```
173    360
Name: Miles, dtype: int64
```

Insight

- The absence of outliers in the 'Fitness' column suggest that all customer fallwithin a reasonable range of self-rated fitness levels.
- The outliers in the 'Income' column indicates that a few customers have much higher Incomes compared to the rest.
- The outliers in the 'Miles' column suggest that some customer expect to walk or run significantly more miles per week than others.

15 Adding new columns for better analysis

Creating New Column and Categorizing values in ***Age,Education,Income*** and Miles to different classes for better visualization

Age Column

Categorizing the values in age column in 4 different buckets:

1. Young Adult: from 18 - 25
2. Adults: from 26 - 35
3. Middle Aged Adults: 36-45
4. Elder :46 and above

Education Column

Categorizing the values in education column in 3 different buckets:

1. Primary Education: upto 12
2. Secondary Education: 13 to 15
3. Higher Education: 16 and above

Income Column

Categorizing the values in Income column in 4 different buckets:

1. Low Income - Upto 40,000
2. Moderate Income - 40,000 to 60,000
3. High Income - 60,000 to 80,000
4. Very High Income - Above 80,000

Miles column

Categorizing the values in miles column in 4 different buckets:

1. Light Activity - Upto 50 miles

2. Moderate Activity - 51 to 100 miles
3. Active Lifestyle - 101 to 200 miles
4. Fitness Enthusiast - Above 200 miles

[37]:
```
#binning the age values into categories
bin_range1 = [17,25,35,45,float('inf')]
bin_labels1 = ['Young Adults', 'Adults', 'Middle Aged Adults', 'Elder']

df['age_group'] = pd.cut(df['Age'],bins = bin_range1,labels = bin_labels1)

#binning the education values into categories
bin_range2 = [0,12,15,float('inf')]
bin_labels2 = ['Primary Education', 'Secondary Education', 'Higher Education']

df['edu_group'] = pd.cut(df['Education'],bins = bin_range2,labels = bin_labels2)

#binning the income values into categories
bin_range3 = [0,40000,60000,80000,float('inf')]
bin_labels3 = ['Low Income','Moderate Income','High Income','Very High Income']

df['income_group'] = pd.cut(df['Income'],bins = bin_range3,labels = bin_labels3)

#binning the miles values into categories
bin_range4 = [0,50,100,200,float('inf')]
bin_labels4 = ['Light Activity', 'Moderate Activity', 'Active Lifestyle',␣
↪'Fitness Enthusiast ']

df['miles_group'] = pd.cut(df['Miles'],bins = bin_range4,labels = bin_labels4)
```

[38]:
```
df.head()
```

[38]:
```
  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

   Miles     age_group            edu_group income_group        miles_group
0    112  Young Adults  Secondary Education   Low Income   Active Lifestyle
1     75  Young Adults  Secondary Education   Low Income  Moderate Activity
2     66  Young Adults  Secondary Education   Low Income  Moderate Activity
3     85  Young Adults    Primary Education   Low Income  Moderate Activity
4     47  Young Adults  Secondary Education   Low Income     Light Activity
```

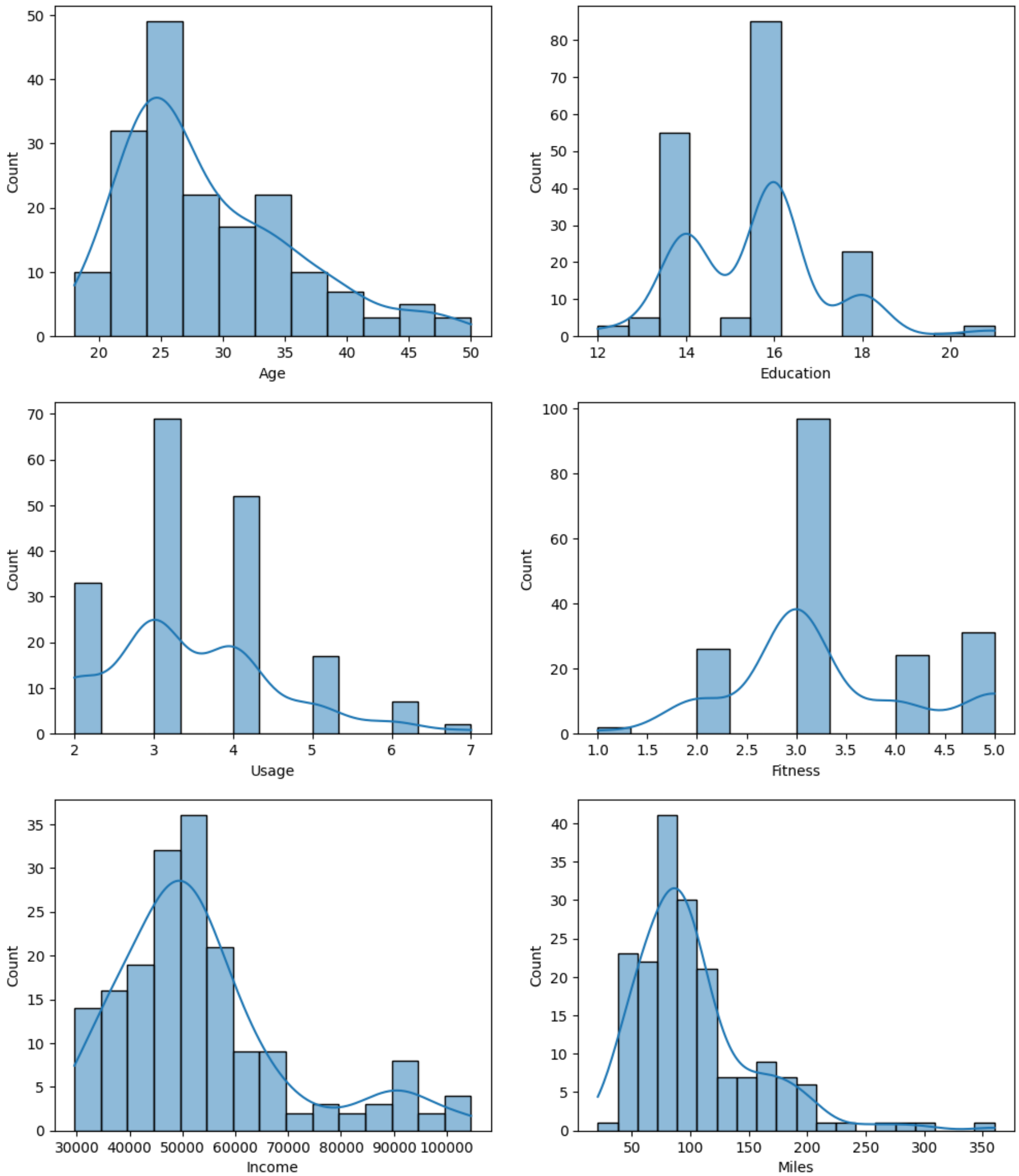
16 3. Visual Analysis - Univariate & Bivariate

17 Univariate Analysis

18 For continuous variable(s):

Understanding the distribution of the data for the quantitative attributes: 1. Age 2. Education 3. Usage 4. Fitness 5. Income 6. Miles

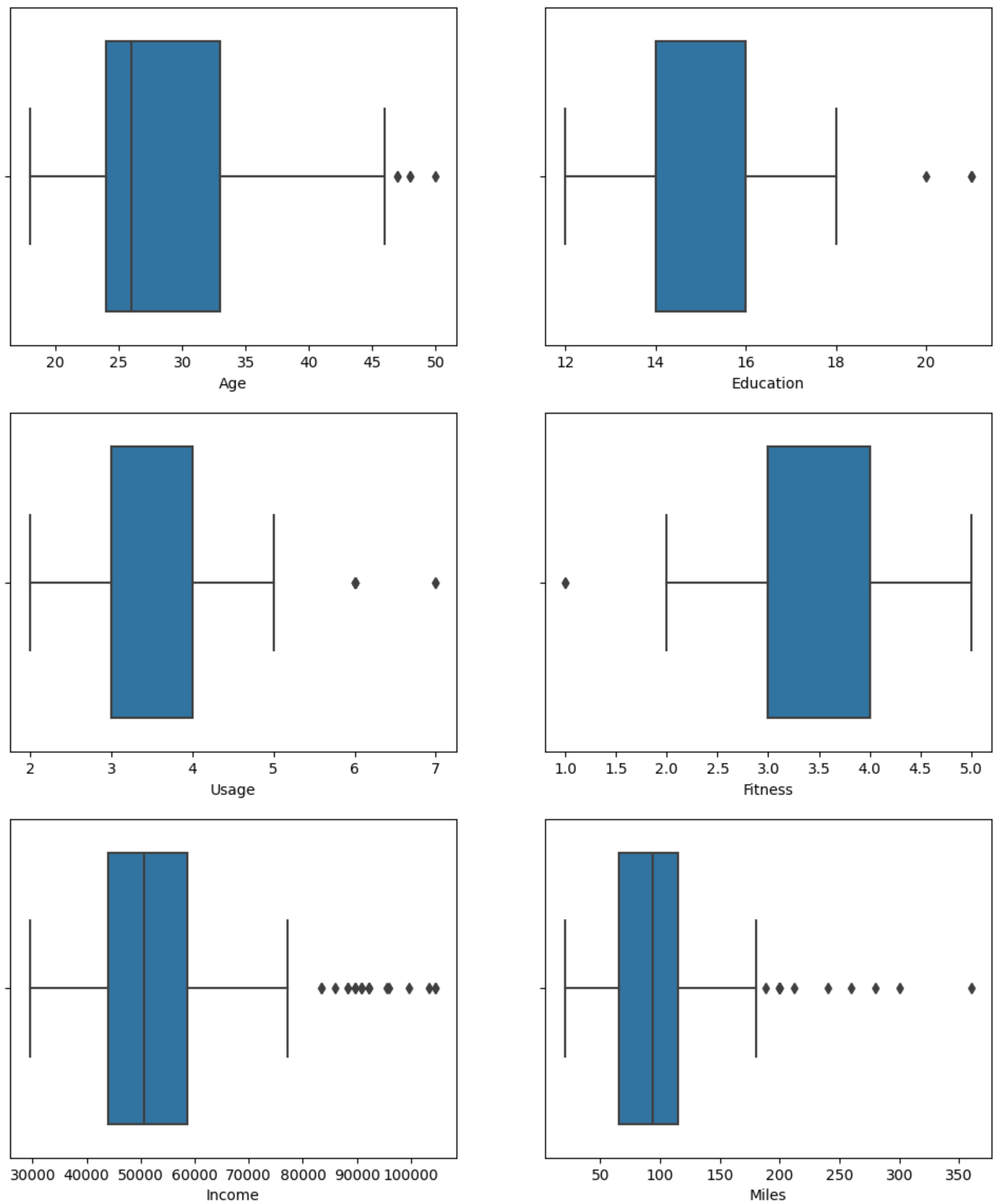
[39]:
```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))
fig.subplots_adjust(top=1.2)

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()
```



Outliers detection using BoxPlots

[40]:
```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))
fig.subplots_adjust(top=1.2)

sns.boxplot(data=df, x="Age", ax=axis[0,0])
sns.boxplot(data=df, x="Education", ax=axis[0,1])
sns.boxplot(data=df, x="Usage", ax=axis[1,0])
```

```
sns.boxplot(data=df, x="Fitness", ax=axis[1,1])
sns.boxplot(data=df, x="Income", ax=axis[2,0])
sns.boxplot(data=df, x="Miles" , ax=axis[2,1])
plt.show()
```



Insight

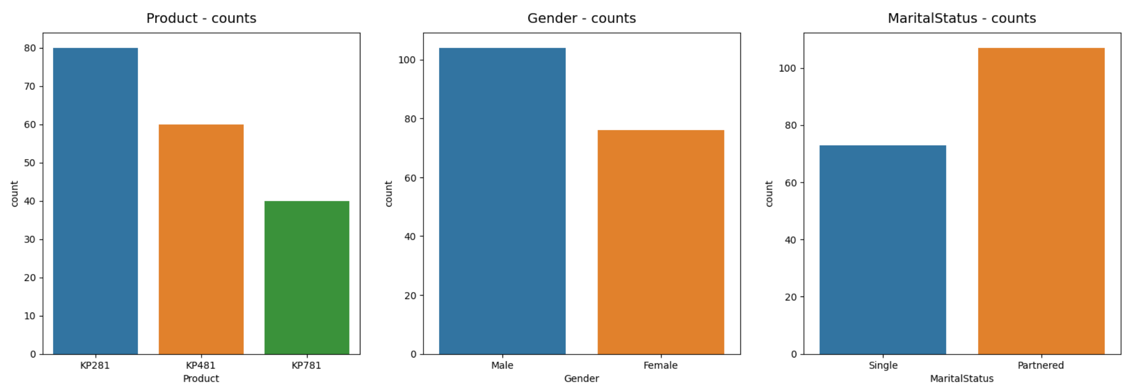
Even from the boxplots it is quite clear that:

1. Age, Education and Usage are having very few outliers.
2. While Income and Miles are having more outliers.

Understanding the distribution of the data for the qualitative attributes:

1. Product
2. Gender
3. Marital Status

[41]:
```
fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(20, 6))
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()
```



Insight

1. KP281 is the most frequent product.
2. Thare are more Males in the data than Females.
3. More Partnered persons are there in the data.

To be precise - normalized count for each variable is shown below

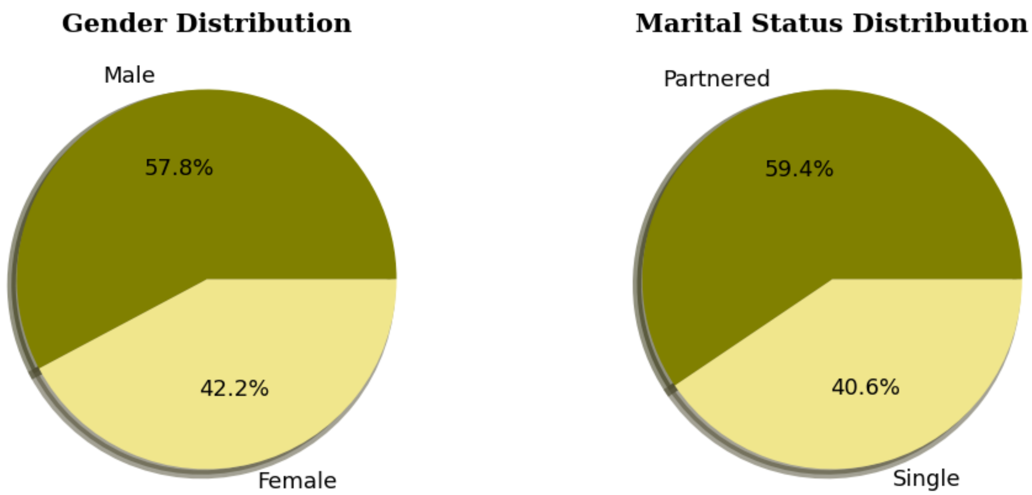
[42]:
```
df1 = df[['Product', 'Gender', 'MaritalStatus']].melt()
df1.groupby(['variable', 'value'])[['value']].count() / len(df)
```

[42]:
```
                             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
```

Gender and Marital Status Disribution

[43]:
```python
#setting the plot style
fig = plt.figure(figsize = (12,5))
gs = fig.add_gridspec(1,2)

                                        # creating pie chart for gender␣
 ↪disribution
ax0 = fig.add_subplot(gs[0,0])

color_map = ['#808000',"#F0E68C"]
ax0.pie(df['Gender'].value_counts().values,labels = df['Gender'].value_counts().
 ↪index,autopct = '%.1f%%',
        shadow = True,colors = color_map,wedgeprops = {'linewidth':␣
 ↪5},textprops={'fontsize': 13, 'color': 'black'})

#setting title for visual
ax0.set_title('Gender Distribution',{'font':'serif', 'size':15,'weight':'bold'})

                                        # creating pie chart for marital status
ax1 = fig.add_subplot(gs[0,1])

color_map = ['#808000',"#F0E68C"]
ax1.pie(df['MaritalStatus'].value_counts().values,labels = df['MaritalStatus'].
 ↪value_counts().index,autopct = '%.1f%%',
        shadow = True,colors = color_map,wedgeprops = {'linewidth':␣
 ↪5},textprops={'fontsize': 13, 'color': 'black'})

#setting title for visual
ax1.set_title('Marital Status Distribution',{'font':'serif', 'size':15,'weight':
 ↪'bold'})

plt.show()
```



Insight

- **Product**
  1. 44.44% of the customers have purchased KP2821 product.
  2. 33.33% of the customers have purchased KP481 product.
  3. 22.22% of the customers have purchased KP781 product.
- **Gender**
  1. 57.78% of the customers are Male.
- **MaritalStatus**
  1. 59.44% of the customers are Partnered.

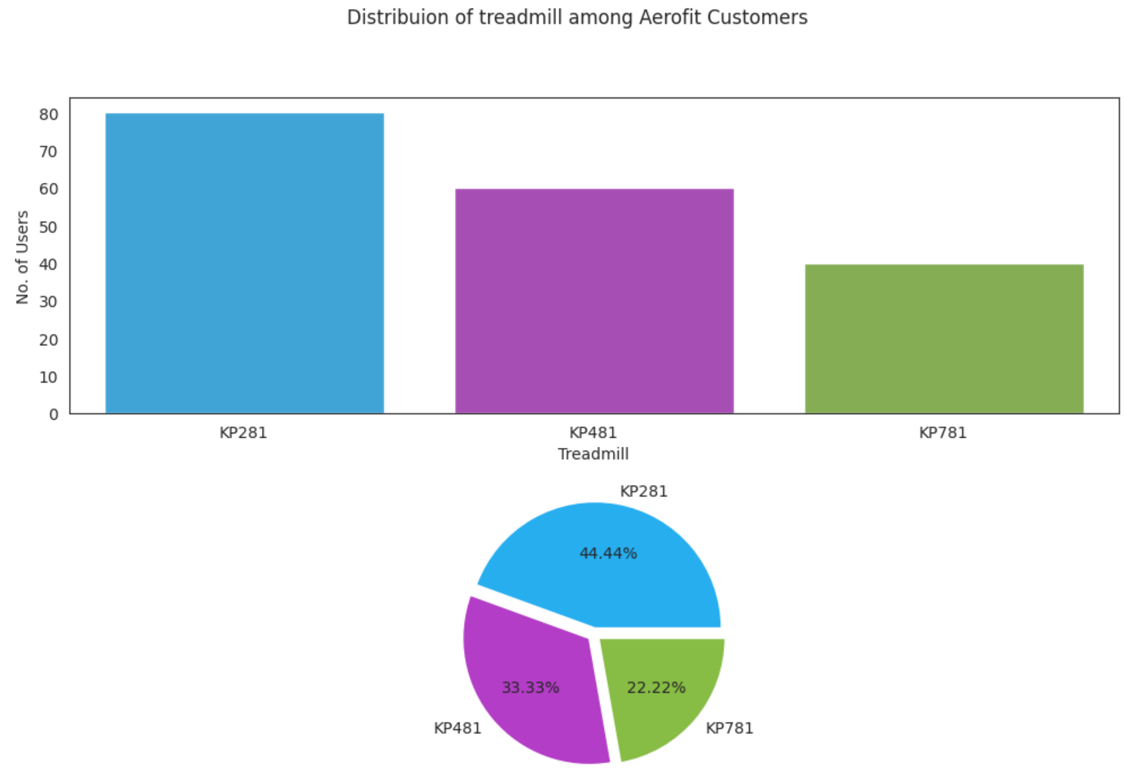
19 Bivariate Analysis

20 For categorical variable(s):

21 Distribution of treadmills among Aerofit Customers

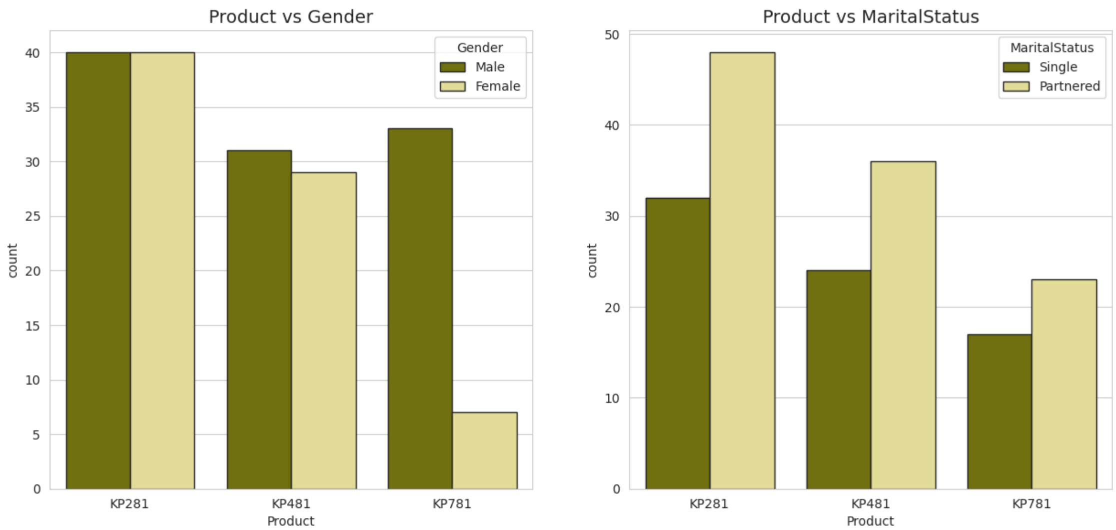
[76]:
```
plt.figure(figsize=(12,8))
plt.subplot(2,1,1)
sns.countplot(data=df, x = df['Product'], palette =␣
 ↪['#27aeef','#b33dc6','#87bc45'])
plt.xlabel('Treadmill')
plt.ylabel('No. of Users')
plt.subplot(2,1,2)
```

```
plt.pie(df['Product'].value_counts(), labels = df['Product'].
 ↪unique(),explode=(0.05, 0.05,0.05),autopct='%1.
 ↪2f%%',colors=['#27aeef','#b33dc6','#87bc45'])
plt.suptitle('Distribuion of treadmill among Aerofit Customers')
plt.show()
```



Checking if features - Gender or MaritalStatus have any effect on the product purchased.

[45]:
```
sns.set_style(style='whitegrid')
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15, 6.5))
sns.countplot(data=df, x='Product', hue='Gender', edgecolor="0.15",␣
 ↪palette=['#808000',"#F0E68C"], ax=axs[0])
sns.countplot(data=df, x='Product', hue='MaritalStatus', edgecolor="0.15",␣
 ↪palette=['#808000',"#F0E68C"], ax=axs[1])
axs[0].set_title("Product vs Gender", fontsize=14)
axs[1].set_title("Product vs MaritalStatus", fontsize=14)
plt.show()
```



Insight

Product vs Gender

- Equal number of males and females have purchased KP281 product and Almost same for the product KP481
- Most of the Male customers have purchased the KP781 product.

Product vs MaritalStatus

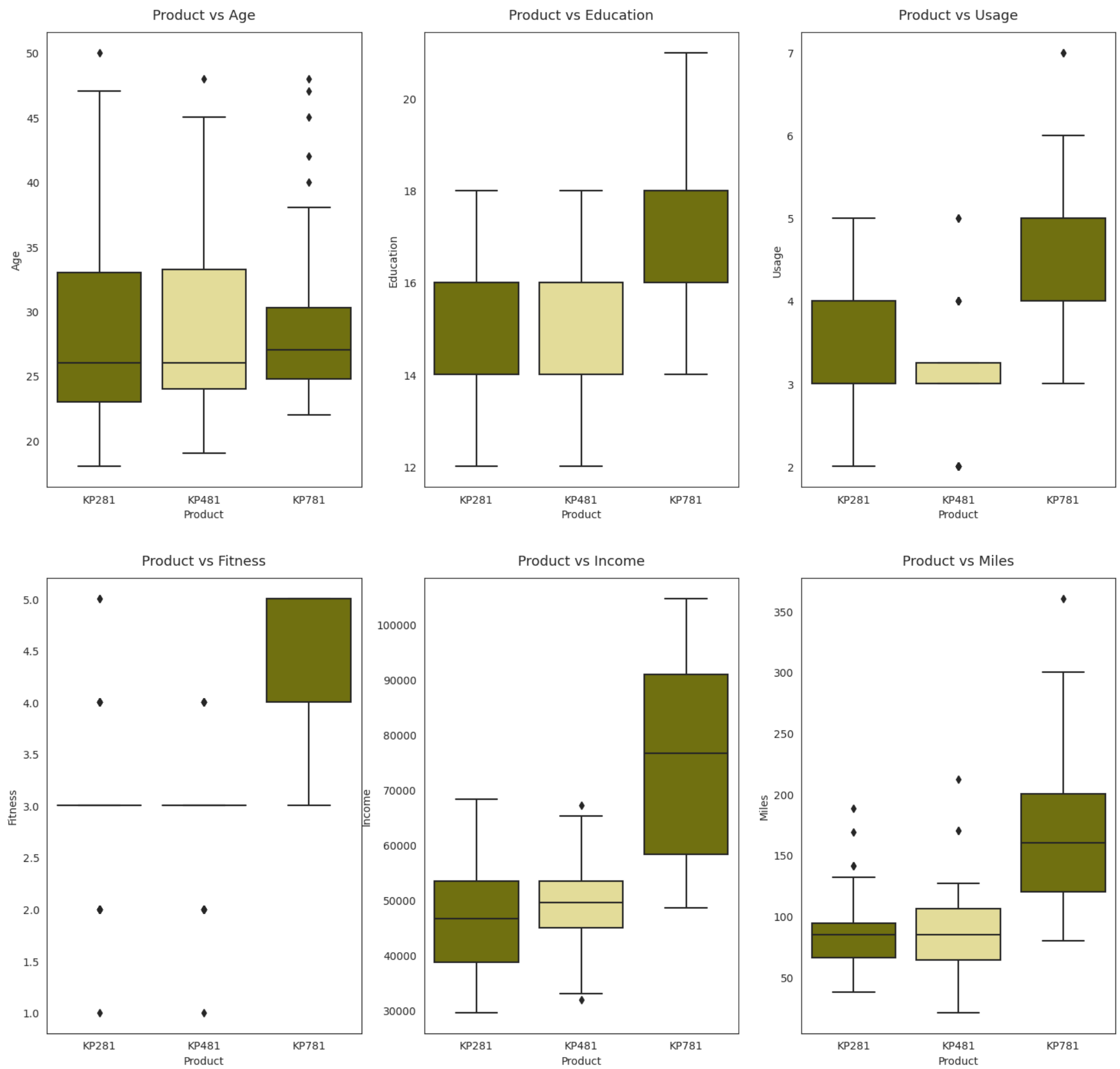
- Customer who is Partnered, is more likely to purchase the product.

Checking if following features have any effect on the product purchased:

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

[46]:
```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(18, 12))
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=['#808000',"#F0E68C"])
```

```
        axs[i,j].set_title(f"Product vs {attrs[count]}", pad=12, fontsize=13)
        count += 1
```



Insights

1. Product vs Age

- Customers purchasing products KP281 & KP481 are having same Age median value.
- 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 Education less than 16 have equal chances of purchasing KP281 or KP481.

3. **Product vs Usage**
   - 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 purchasing KP281 or KP481.
4. **Product vs Fitness**
   - The more the customer is fit (fitness >= 3), higher the chances of the customer to purchase 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.

22 3.3 For Correlation: Heatmaps, Pairplots

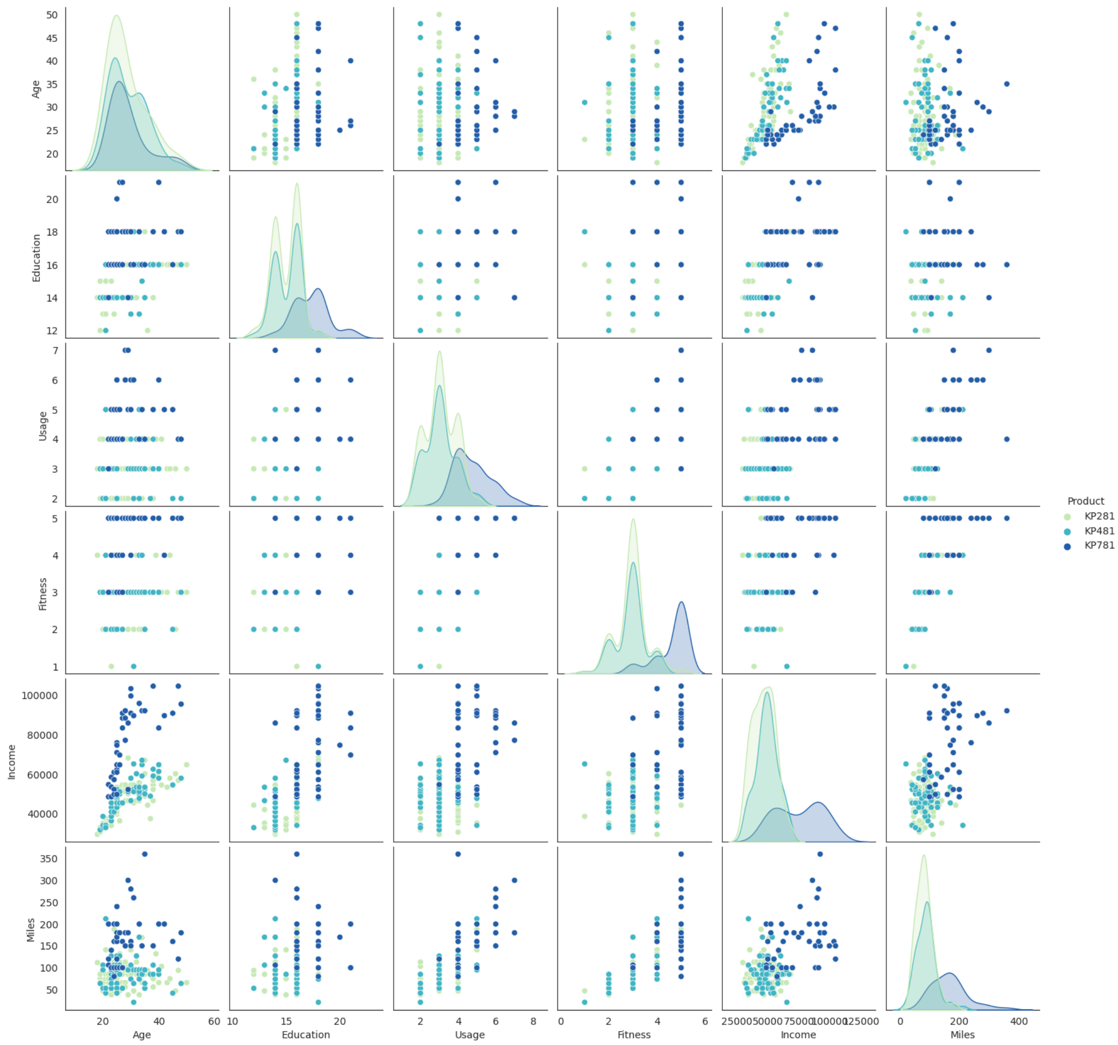
Correlation between Variables

23 3.3.1 Pairplot

A pairplot plot a pairwise relationships in a dataset. The pairplot function creates a grid of Axes such that each variable in data will by shared in the y-axis across a single row and in the x-axis across a single column.

[47]:
```
df_copy = df
```

[48]:
```
sns.pairplot(df_copy, hue ='Product', palette= 'YlGnBu')
plt.show()
```



24 3.3.2 Heatmap

A heatmap is a plot of rectangular data as a color-encoded matrix. As parameter it takes a 2D dataset. That dataset can be coerced into an ndarray. This is a great way to visualize data, because it can show the relation between variabels including time.

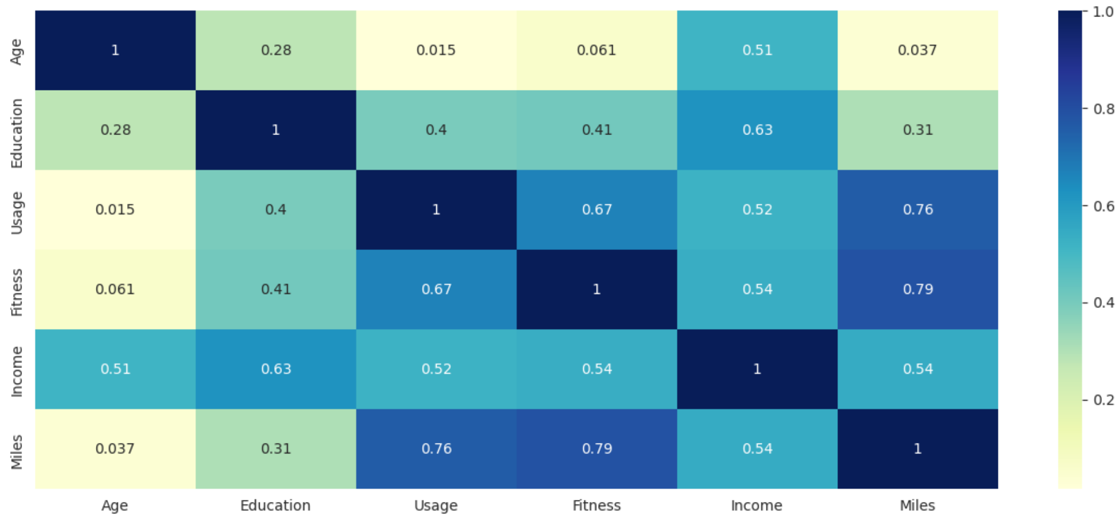
[49]:
```
# First we need to convert object into int datatype for usage and fitness␣
↪columns

df_copy['Usage'] = df_copy['Usage'].astype('int')
df_copy['Fitness'] = df_copy['Fitness'].astype('int')

df_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 13 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
 9   age_group      180 non-null    category
 10  edu_group      180 non-null    category
 11  income_group   180 non-null    category
 12  miles_group    180 non-null    category
dtypes: category(4), int64(6), object(3)
memory usage: 14.2+ KB
```

[50]:

```
corr_mat = df_copy.corr()

plt.figure(figsize=(15,6))

sns.heatmap(corr_mat,annot = True, cmap="YlGnBu")

plt.show()
```

```
<ipython-input-50-ba0b4211d231>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  corr_mat = df_copy.corr()
```



Insights

1. From the pair plot we can see Age and Income are positively correlated and heatmap also suggests a strong correlation betwwen them
2. Eductaion and Income are highly correlated as its obvious. Eductation also has significatnt correlation between Fitness rating and Usage of the treadmill.
3. Usage is highly correlated with Fitness and Miles as more the usage more the fitness and mileage.
4. Age and Education : There ia a positive correlation of approximately 0.28 between Age and Eductaion. this indicates that as the customers age increases their education level tends to be higher.
5. Age and Income : There is a moderate positive correlation of appoximately 0.51 between Age and Income.this suggest that as the customers age increases their income tends to be higher.
6. Education and Income : There is relatively strong positive correlation of approximately 0.63 between Education and Income.this suggested that the customers with higher education tends to have higher income.
7. Usage and Fitness : There is relatively strong positive correlation of approximately 0.67 between Usage and Fitness.this indicates that customer who plan to use the treadmill more frequently tends to have high fitness level.
8. Fitness and Miles : There is strong positive correlation of approximately 0.79 between Fitness and Miles.this indicates that customers with high fitness level except to walk/run more miles per week.
9. Age and Fitness : There is a weak positive correlation of approximately 0.06 between age and fitness. Similar correlation can be observed with age and usage as well as Age and Miles.

25 4. Missing Value & Outlier Detection

26 Handling Mising Values

[51]:
```
df.isnull().sum()
```

[51]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
age_group        0
edu_group        0
income_group     0
miles_group      0
dtype: int64
```

Insight

There is no missing value present in dataset.

27 Handling Outliers

Income Column

[52]:
```
df["Income"].describe()
```

[52]:
```
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
```

To find outliers in Income column we need to use **Boxplot** here. but before using boxplot we need to find these 5 points:

1. Q3 - Upper Quartile
2. Q1 - Lower Quartile
3. Median

4. Upper Bound
5. Lower Bound

[53]:
```
q1 = np.percentile(df['Income'],25)
q3 = np.percentile(df['Income'],75)
print( 'q1 =', q1)
print( 'q3 =', q3)
```

```
q1 = 44058.75
q3 = 58668.0
```

Insight

we get

Q1 = 44058.75

Q3 = 58668.0

[54]:
```
#to find upper bound and lower bound we need to find IQR (inter quartile range)

IQR = q3 - q1
IQR
```

[54]:
```
14609.25
```

Insight

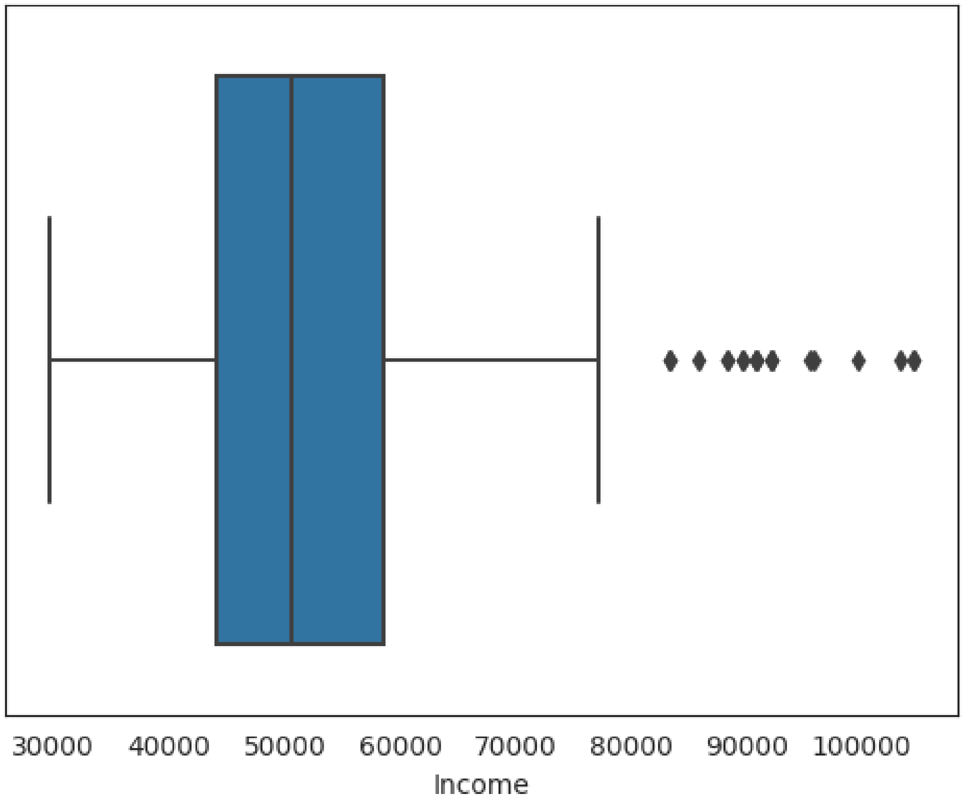
we get

IQR = 14609.25

[55]:
```
upper_bound = q3 + 1.5 * IQR
lower_bound = q1 - 1.5 * IQR
print('Upper Bound =', upper_bound)
print('Lpper Bound =', lower_bound)
print('Median =', df['Income'].median())
```

```
Upper Bound = 80581.875
Lpper Bound = 22144.875
Median = 50596.5
```

[56]:
```
sns.boxplot(data = df, x = 'Income')
plt.show()
```



- As we see there are Outliers in the 'Income' Column
- All values > 80581.75 (Upper Bound) are outliers in the 'Income' Column

[57]:
```
(len(df.loc[df['Income'] > upper_bound]) / len(df)) * 100
```

[57]:
```
10.555555555555555
```

Insight

10.5% in Income column are outliers but we choose not to drop them as these values may required to draw some valuable insights and it may be useful for customer profiling.

Miles Column

[58]:
```
df["Miles"].describe()
```

[58]:
```
count    180.000000
mean     103.194444
std       51.863605
min       21.000000
25%       66.000000
50%       94.000000
75%      114.750000
```

```
max      360.000000
Name: Miles, dtype: float64
```

To find outliers in Income column we need to use **Boxplot** here. but before using boxplot we need to find these 5 points:

1. Q3 - Upper Quartile
2. Q1 - Lower Quartile
3. Median
4. Upper Bound
5. Lower Bound

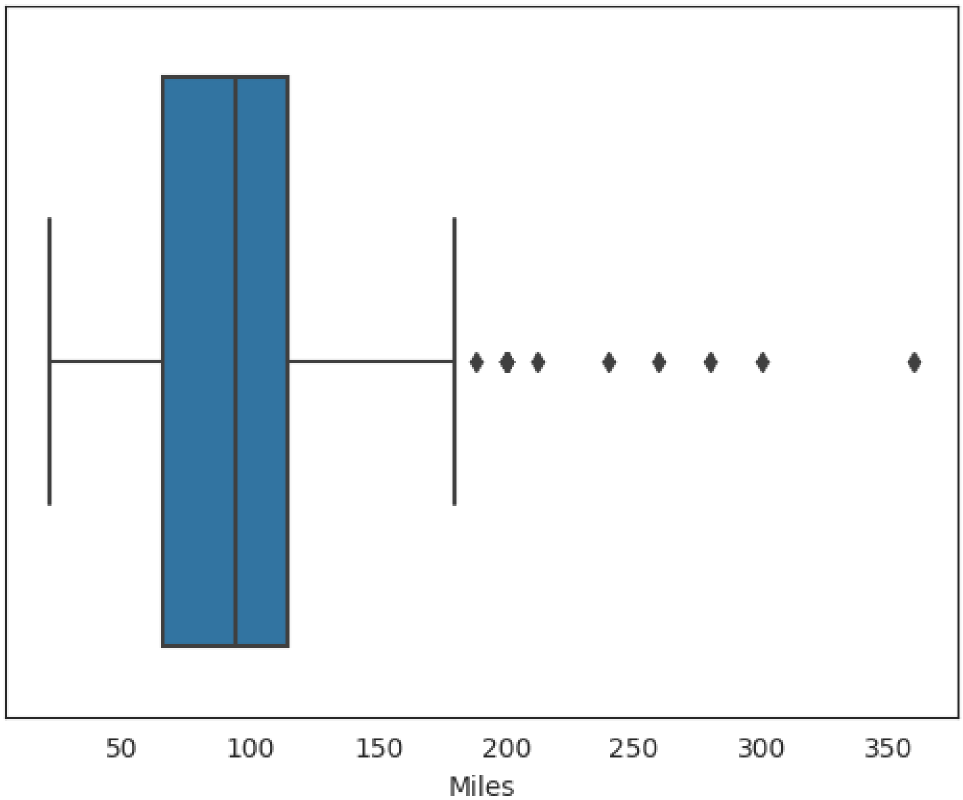
[59]:
```
q1 = np.percentile(df['Miles'],25)
q3 = np.percentile(df['Miles'],75)
IQR = q3 - q1
print( 'q1 =', q1)
print( 'q3 =', q3)
print('IQR = ', IQR)
```

```
q1 = 66.0
q3 = 114.75
IQR =  48.75
```

[60]:
```
upper_bound = q3 + 1.5 * IQR
lower_bound = q1 - 1.5 * IQR
print('Upper Bound =', upper_bound)
print('Lpper Bound =', lower_bound)
print('Median =', df['Miles'].median())
```

```
Upper Bound = 187.875
Lpper Bound = -7.125
Median = 94.0
```

[61]:
```
sns.boxplot(data = df, x = 'Miles')
plt.show()
```



- As we see there are Outliers in the 'Miles' Column
- All values > 187.875 (Upper Bound) are outliers in the 'Miles' Column

[62]:
```
(len(df.loc[df['Miles'] > upper_bound]) / len(df)) * 100
```

[62]:
```
7.222222222222221
```

Insight

7.22% in Miles column are outliers but we choose not to drop them as these values may required to draw some valuable insights and it may be useful for customer profiling.

28 Outlier detection using the Z-Score

what is Z score?

Z scores are: z = (x - mean)/std, so values in each row (column) will get the mean of the row (column) subtracted, then divided by the standard deviation of the row (column). This ensures that each row (column) has mean of 0 and variance of 1.

- We can detect outliers in numeric column using the z-score.

- if the Z-score of a data point is more then 3, it indicates that the data point is quite different from the other data points. Such a data can be a outlier.
- Z score = (x - mean)/std.devation

[63]:
```
outliers = {}
for col in df.select_dtypes(include = np.number):
  #finding Z-score for each value in a column
  z_score = np.abs((df[col] - df[col].mean())) / df[col].std()

  #if the z score of a value ia greater than 3 then the value is outlier
  column_outliers = df[z_score > 3][col]

  outliers[col] = column_outliers

for col, outliers_values in outliers.items():
  print(f"Outliers for {col} column")
  print(outliers_values)
  print()
```

```
Outliers for Age column
79    50
Name: Age, dtype: int64

Outliers for Education column
157    21
161    21
175    21
Name: Education, dtype: int64

Outliers for Usage column
163    7
166    7
Name: Usage, dtype: int64

Outliers for Fitness column
Series([], Name: Fitness, dtype: int64)

Outliers for Income column
168    103336
174    104581
178    104581
Name: Income, dtype: int64

Outliers for Miles column
166    300
167    280
170    260
```

```
173    360
Name: Miles, dtype: int64
```

Insight

- The absence of outliers in the 'Fitness' column suggest that all customer fallwithin a reasonable range of self-rated fitness levels.
- The outliers in the 'Income' column indicates that a few customers have much higher Incomes compared to the rest.
- The outliers in the 'Miles' column suggest that some customer expect to walk or run significantly more miles per week than others.

29 5. Computing Probability - Marginal, Conditional Probability

5.1 Probability of product purchase w.r.t. gender

[64]:
```
pd.crosstab(index =df['Product'],columns = df['Gender'],margins =
 ↪True,normalize = True ).round(2)
```

[64]:

Gender Product	Female	Male	All
KP281	0.22	0.22	0.44
KP481	0.16	0.17	0.33
KP781	0.04	0.18	0.22
All	0.42	0.58	1.00

Insights

1. The Probability of a treadmill being purchased by a female is 42%.
   - The conditional probability of purchasing the treadmill model given that the customer is female is
   - For Treadmill model KP281 - 22%
   - For Treadmill model KP481 - 16%
   - For Treadmill model KP781 - 4%
2. The Probability of a treadmill being purchased by a male is 58%.
   - The conditional probability of purchasing the treadmill model given that the customer is male is -
   - For Treadmill model KP281 - 22%
   - For Treadmill model KP481 - 17%
   - For Treadmill model KP781 - 18%

5.2 Probability of product purchase w.r.t. Age

[65]:
```
pd.crosstab(index =df['Product'],columns = df['age_group'],margins =␣
↪True,normalize = True ).round(2)
```

[65]:

age_group	Young Adults	Adults	Middle Aged Adults	Elder	All
Product					
KP281	0.19	0.18	0.06	0.02	0.44
KP481	0.16	0.13	0.04	0.01	0.33
KP781	0.09	0.09	0.02	0.01	0.22
All	0.44	0.41	0.12	0.03	1.00

Insight

1. The Probability of a treadmill being purchased by a Young Adult(18-25) is 44%.
   - The conditional probability of purchasing the treadmill model given that the customer is Young Adult is
   - For Treadmill model KP281 - 19%
   - For Treadmill model KP481 - 16%
   - For Treadmill model KP781 - 9%
2. The Probability of a treadmill being purchased by a Adult(26-35) is 41%.
   - The conditional probability of purchasing the treadmill model given that the customer is Adult is -
   - For Treadmill model KP281 - 18%
   - For Treadmill model KP481 - 13%
   - For Treadmill model KP781 - 9%
3. The Probability of a treadmill being purchased by a Middle Aged(36-45) is 12%.
4. The Probability of a treadmill being purchased by a Elder(Above 45) is only 3%.

5.3 Probability of product purchase w.r.t. Income

[66]:
```
pd.crosstab(index =df['Product'],columns = df['income_group'],margins =␣
↪True,normalize = True ).round(2)
```

[66]:

income_group	Low Income	Moderate Income	High Income	Very High Income	All
Product					
KP281	0.13	0.28	0.03	0.00	0.44
KP481	0.05	0.24	0.04	0.00	0.33
KP781	0.00	0.06	0.06	0.11	0.22
All	0.18	0.59	0.13	0.11	1.00

Insight

1. The Probability of a treadmill being purchased by a customer with Low Income(<40k) is 18%.

- The conditional probability of purchasing the treadmill model given that the customer has Low Income is -
- For Treadmill model KP281 - 13%
- For Treadmill model KP481 - 5%
- For Treadmill model KP781 - 0%

2. The Probability of a treadmill being purchased by a customer with Moderate Income(40k - 60k) is 59%.
   - The conditional probability of purchasing the treadmill model given that the customer has Moderate Income is -
   - For Treadmill model KP281 - 28%
   - For Treadmill model KP481 - 24%
   - For Treadmill model KP781 - 6%
3. The Probability of a treadmill being purchased by a customer with High Income(60k - 80k) is 13%
   - The conditional probability of purchasing the treadmill model given that the customer has High Income is -
   - For Treadmill model KP281 - 3%
   - For Treadmill model KP481 - 4%
   - For Treadmill model KP781 - 6%
4. The Probability of a treadmill being purchased by a customer with Very High Income(>80k) is 11%
   - The conditional probability of purchasing the treadmill model given that the customer has High Income is -
   - For Treadmill model KP281 - 0%
   - For Treadmill model KP481 - 0%
   - For Treadmill model KP781 - 11%

5.4 Probability of product purchase w.r.t. Education level

[67]:
```
pd.crosstab(index =df['Product'],columns = df['edu_group'],margins =
↪True,normalize = True ).round(2)
```

[67]:

edu_group	Primary Education	Secondary Education	Higher Education	All
Product				
KP281	0.01	0.21	0.23	0.44
KP481	0.01	0.14	0.18	0.33
KP781	0.00	0.01	0.21	0.22
All	0.02	0.36	0.62	1.00

Insight

1. The Probability of a treadmill being purchased by a customer with Higher Education(Above 15 Years) is 62%.

   - The conditional probability of purchasing the treadmill model given that the customer has Higher Education is
   - For Treadmill model KP281 - 23%
   - For Treadmill model KP481 - 18%
   - For Treadmill model KP781 - 21%

2. The Probability of a treadmill being purchased by a customer with Secondary Education(13-15 yrs) is 36%.

   - The conditional probability of purchasing the treadmill model given that the customer has Secondary Education is -
   - For Treadmill model KP281 - 21%
   - For Treadmill model KP481 - 14%
   - For Treadmill model KP781 - 1%

3. The Probability of a treadmill being purchased by a customer with Primary Education(0 to 12 yrs) is only 2%.

5.5 Probability of product purchase w.r.t. Marital Status

[68]:
```
pd.crosstab(index =df['Product'],columns = df['MaritalStatus'],margins =␣
↪True,normalize = True ).round(2)
```

[68]:

MaritalStatus	Partnered	Single	All
Product			
KP281	0.27	0.18	0.44
KP481	0.20	0.13	0.33
KP781	0.13	0.09	0.22
All	0.59	0.41	1.00

Insight

1. The Probability of a treadmill being purchased by a Married Customer is 59%.

   - The conditional probability of purchasing the treadmill model given that the customer is Married is
   - For Treadmill model KP281 - 27%
   - For Treadmill model KP481 - 20%
   - For Treadmill model KP781 - 13%

2. The Probability of a treadmill being purchased by a Unmarried Customer is 41%.

- The conditional probability of purchasing the treadmill model given that the customer is Unmarried is -
- For Treadmill model KP281 - 18%
- For Treadmill model KP481 - 13%
- For Treadmill model KP781 - 9%

5.6 Probability of product purchase w.r.t. Weekly Usage

[69]:
```
pd.crosstab(index =df['Product'],columns = df['Usage'],margins = True,normalize␣
↪= True ).round(2)
```

[69]:
```
Usage       2     3     4     5     6     7   All
Product
KP281    0.11  0.21  0.12  0.01  0.00  0.00  0.44
KP481    0.08  0.17  0.07  0.02  0.00  0.00  0.33
KP781    0.00  0.01  0.10  0.07  0.04  0.01  0.22
All      0.18  0.38  0.29  0.09  0.04  0.01  1.00
```

Insight

1. The Probability of a treadmill being purchased by a customer with Usage 3 per week is 38%.
   - The conditional probability of purchasing the treadmill model given that the customer has Usage 3 per week is -
   - For Treadmill model KP281 - 21%
   - For Treadmill model KP481 - 17%
   - For Treadmill model KP781 - 1%
2. The Probability of a treadmill being purchased by a customer with Usage 4 per week is 29%.
   - The conditional probability of purchasing the treadmill model given that the customer has Usage 4 per week is -
   - For Treadmill model KP281 - 12%
   - For Treadmill model KP481 - 7%
   - For Treadmill model KP781 - 10%
3. The Probability of a treadmill being purchased by a customer with Usage 2 per week is 18%
   - The conditional probability of purchasing the treadmill model given that the customer has Usage 2 per week is -
   - For Treadmill model KP281 - 11%
   - For Treadmill model KP481 - 8%
   - For Treadmill model KP781 - 0%

5.7 Probability of product purchase w.r.t. Customer Fitness

[70]:
```
pd.crosstab(index =df['Product'],columns = df['Fitness'],margins =␣
↪True,normalize = True ).round(2)
```

[70]:
```
Fitness     1     2     3     4     5   All
Product
KP281    0.01  0.08  0.30  0.05  0.01  0.44
KP481    0.01  0.07  0.22  0.04  0.00  0.33
KP781    0.00  0.00  0.02  0.04  0.16  0.22
All      0.01  0.14  0.54  0.13  0.17  1.00
```

Insight

1. The Probability of a treadmill being purchased by a customer with Average(3) Fitness is 54%
    - The conditional probability of purchasing the treadmill model given that the customer has Average Fitness is -
    - For Treadmill model KP281 - 30%
    - For Treadmill model KP481 - 22%
    - For Treadmill model KP781 - 2%
2. The Probability of a treadmill being purchased by a customer with Fitness of 2,4,5 is almost 15%.
3. The Probability of a treadmill being purchased by a customer with very low(1) Fitness is only 1%.

5.8 Probability of product purchase w.r.t. weekly mileage

[71]:
```
pd.crosstab(index =df['Product'],columns = df['miles_group'],margins =␣
↪True,normalize = True ).round(2)
```

[71]:
```
miles_group  Light Activity  Moderate Activity  Active Lifestyle  \
Product
KP281                  0.07               0.28              0.10
KP481                  0.03               0.22              0.08
KP781                  0.00               0.04              0.15
All                    0.09               0.54              0.33

miles_group  Fitness Enthusiast   All
Product
KP281                      0.00  0.44
KP481                      0.01  0.33
KP781                      0.03  0.22
All                        0.03  1.00
```

Insight

1. The Probability of a treadmill being purchased by a customer with lifestyle of Light Activity(0 to 50 miles/week) is 9%.

  - The conditional probability of purchasing the treadmill model given that the customer has Light Activity Lifestyle is -
  - For Treadmill model KP281 - 7%
  - For Treadmill model KP481 - 3%
  - For Treadmill model KP781 - 0%

2. The Probability of a treadmill being purchased by a customer with lifestyle of Moderate Activity(51 to 100 miles/week) is 54%.
   - The conditional probability of purchasing the treadmill model given that the customer with lifestyle of Moderate Activity is -
   - For Treadmill model KP281 - 28%
   - For Treadmill model KP481 - 22%
   - For Treadmill model KP781 - 4%
3. The Probability of a treadmill being purchased by a customer has Active Lifestyle(100 to 200 miles/week) is 33%.
   - The conditional probability of purchasing the treadmill model given that the customer has Active Lifestyle is -
   - For Treadmill model KP281 - 10%
   - For Treadmill model KP481 - 8%
   - For Treadmill model KP781 - 15%
4. The Probability of a treadmill being purchased by a customer who is Fitness Enthusiast(>200 miles/week) is 3% only.

30 6. Customer Profiling

Based on above analysis

- Probability of purchase of KP281 = 44%
- Probability of purchase of KP481 = 33%
- Probability of purchase of KP781 = 22%
- Customer Profile for KP281 Treadmill: > * Age of customer mainly between 18 to 35 years with few between 35 to 50 years > * Education level of customer 13 years and above > * Annual Income of customer below USD 60,000 > * Weekly Usage - 2 to 4 times > * Fitness Scale - 2 to 4 > * Weekly Running Mileage - 50 to 100 miles
- Customer Profile for KP481 Treadmill: > * Age of customer mainly between 18 to 35 years with few between 35 to 50 years > * Education level of customer 13 years and above > * Annual Income of customer between USD 40,000 to USD 80,000 > * Weekly Usage - 2 to 4 times > * Fitness Scale - 2 to 4 > * Weekly Running Mileage - 50 to 200 miles

- Customer Profile for KP781 Treadmill: > * Gender - Male > * Age of customer between 18 to 35 years > * Education level of customer 15 years and above > * Annual Income of customer USD 80,000 and above > * Weekly Usage - 4 to 7 times > * Fitness Scale - 3 to 5 > * Weekly Running Mileage - 100 miles and above

31 7. Business Insights based on Non-Graphical and Visual Analysis

1. Among the Users, 44.44% prefer using the KP281 treadmill, while 33.33% opt for the KP481 treadmill, and only 22.22% of user favor the KP781 treadmill.
2. KP281, being an entry level and more affordable compared with others, is the preffered choice among the majority of customers.
3. 33.33% of customers favor the KP481 treadmill, drawn by its ideal fit for mid-level runner and its excellent value for money offering.
4. KP781 treadmill, being more advanced and costlier than the other two options, is chosen by only 22.2% of customers
5. Aerofit has 57.78% male customers and 42.22% female customers.
6. Among male customers, 38.5% prefer KP281 as an entry-level and cost-effective option Meanwhile, 29.8% opt for KP481 due to its value for money proposition, and 31.7% favor KP781 for its advanced features
7. Among female customers, 52.6% prefer KP281 as an entry-level and cost-effective option. Additionally, 38.2% opt for KP481 due to its value for money proposition, while only 9.2% favor KP781 due to its higher cost compared to the other two options
8. Probablity of female customers buying KP781 is 4% which is very low.
9. Both female and male customers equally prefers KP281 with probablity 22.2%
10. Probablity of male customers buying KP481 is 17%
11. Probablity of female customers buying KP481 is 16% which is also good 12.59,4% of Aerofit customers are married, while remaining 40.56% we single.
12. Married customers have a higher frequency of purchasing all treadmills compared to single customers.
13. The trend observed among both married and single customers reflects that KP281, being an entry-level treadmill, is the most frequently purchased option, while KP781, due to its higher cost, remains the least popular choice for both customer groups.
14. The purchase frequency for both married and single customers follows the trend of KP281> KP481 > KP781, with KP281 being the most frequently purchased treadmill and KP781 being the least frequently purchased one.
15. The probability of single customers purchasing each of the treadmills is lower compared to that of married customers.
16. Most of the Aerofit customer falls under young age-group (18-29).

17. 27.78 % of middle-aged(30-39) users prefer to use the Aerofit Treadmills
18. 9.4% of users in the old (40-50) age group prefer purchasing Aerofit treadmills.
19. Among young customers, the purchase distribution tor Aerofit treadmills is as follows: 46.9% prefer KP281, 29.2% prefer KP481, and the remaining 23.9% prefer KP781.
20. Among middle-aged customers, surprisingly 44% prefer KP481 over the other two treadmills while 40% prefer KP281 and only 16% prefer KP781.
21. Among old customers, 41.2% prefer KP281, while 29,4% prefer both KP481 and KP781
22. The probability of young customers buying the KP281 treadmill is 29%, while the probability of buying the KP481 treadmillis 18%, and the probability of buying the KP781 treadmill is 15%.
23. The probability of middle-aged customers buying the KP281 treadmill is 11%, while the probability of buying the KP481 treadmill is 12%, and the probability of buying the KP781 treadmill is 4%.
24. The probability of old customers buying the KP281 treadmill is 4%, while the probability of buying the KP481 treadmillis 3%, and the probability of buying the KP781 treadmillis 3%.
25. The probability of old customers purchasing each of the treadmills is lower compared to that of other age-group customers
26. Approximately 88% of Aerofit customers belong to the low-income (29000-50000 USD) and medium-income (51000-75000 USD) groups.Remaining 11.67% blongs to high income group (above 75000 usd).
27. Due to its price of 2500 USD, the probability of customers belonging to the low-income and middle-income groups buying the KP781 treadmill is low compared to customers in the high-income group who can afford this higher-priced treadmill
28. Customers belonging to the high-income group exclusively prefer KP781 due to its advanced features and higher cost compared to the other two treadmills.
29. Customers with 14-16 years of education prefer the KP281 and KP481 treadmills. However, among all treadmills, the majority of customers with 16-18 years of education prefer the KP781 treadmill.
30. Customers who run 60-100 miles per week prefer the KP281 treadmill while mid runners who run 60-120 miles per week opt for the KP481. On the other hand, hardcore runners who run 120-200 miles per week prefer the KP781 treadmill due to its advanced features.
31. Customers who use treadmills 3 times a week prefer both KP281 and KP481. However, customers who use treadmills 4-5 times a week favor the KP781 treadmill.
32. Customers with fitness level 3 prefer both KP281 and KP481 treadmills, while customers with fitness level 5 predominantly use the most advanced KP781 treadmill

32 8. Recommendations

Actionable Insight: Among the users, 44.44% prefer using the KP281 treadmill, while 33.33% opt for the KP481 treadmill, and only 22.22% of users favor the KP781 tread-

mill. 1. Emphasize the budget-friendly nature of the KP281 treadmill to attract more customers. 2. Highlight the key features of the KP281 that make it a great entry-level option for fitness enthusiasts. 3. Provide special offers or discounts to further entice customers looking for a cost-effective option. 4. Engage with fitness communities online to showcase the KP281's appeal to beginners. 5. Focus marketing efforts on reaching out to mid-level runners, emphasizing how the KP481 is tailored to meet their specific fitness needs and goals. 6. Showcase the competitive pricing and the outstanding features of the KP481 that make it a N COst- effective choice for customers. 7. Launch targeted marketing campad gns to increase awareness and interest in the KP781 among potential customers who may value its advanced capabilities. Utilize various channels such as social media, fitness forums, and influencer collaborations. 8. Emphasize the unique features and benefits of the KP781 to justify its higher price. Highlight its advanced functionalities and how they enhance the workout experience, making it worth the Investment.

Actionable Insight: The probability of female customers buying each of the treadmills compared to male customers is 42%: 1. Create targeted advertisements and promotions that appeal to women, showcasing how fitness can positively impact their lives.

2. Showcase the female-friendly features and benefits of Aerofit treadmills to attract more female customers.
3. Offer a diverse selection of treadmill models that cater to various fitness levels and preferences. Actionable Insight: The probability of female customers buying the KP781 treadmill is 4%, which is significantly lower compared to that of male customers : Offer special incentives and discounts exclusively for female customers interested in purchasing the KP781 treadmill This could include limited-time promotions, personalized offers, or package deals to make the treadmill more appealing and accessible to this customer segment. By providing targeted incentives, it can encourage more female customers to consider and invest in the KP781.

Actionable Insight:The probability of single customers purchasing each of the treadmills is lower compared to that of married customers: 1. Appoint Virat Kohli as the brand ambassador for Aerofit, promoting the brand's values of fitness, health, and well-being. Virat's association with Aerofit will resonate with single customers, inspiring them to prioritize their fitness goals and consider Aerofit treadmills as a valuable addition to their fitness routines.

2. Introduce exclusive offers and discounts for single customers as part of the collaboration with Virat Kohli This can include special bundles, personalized packages, or limited-time promotions, providing added incentives for single customers to choose Aerofit treadmills.

3. Organize virtual fitness challenges or competitions, endorsed by Virat Kohli, to engage single customers and encourage them to participate in fitness activities with Aerofit treadmills. Prizes and recognition for participants can further boost motivation and engagement.

Actionable Insight:The probability of old customers purchasing each of the treadmills is lower compared to that of other age-group customers:

> Offer personalized assistance to help customers aged 40-50 select the ideal treadmill model, providing them with the tools to maintain an active and healthy lifestyle. With Aerofit's expert guidance, customers can feel confident and motivated to make the most of their treadmills effectively.

Actionable Insight:Due to its price of 2500 USD, the probability .__ nf of customers customers belonging belonging to the low-income and middle-income groups buying the KP781 treadmillis low compared to customers in the high-income group.

1. Introduce tailored discounts and incentives exclusively for customers belonging to the Ow and middle- income groups. These offers can include limited- time promotions, cashback rewards, or bundle deals, making the KP781 treadmill more affordable and enticing for this target audience.
2. Provide convenient EMI (Equated Monthly Installment) payment options for the KP781 treadmill. This will allow low and middle-income customers to spread the cost over several months, easing their financial burden and making the purchase more manageable.