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

from google.colab import files
uploaded = files.upload()

<IPython.core.display.HTML object>

Saving aerofit_treadmill.csv to aerofit_treadmill.csv

af = pd.read_csv("aerofit_treadmill.csv")

af

    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]
```

##Analysing basic metrics

```
af.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

af.shape

(180, 9)

af.columns

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

af.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
```

```
af.tail()
```

	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

```
af.describe()
```

	Age	Education	Usage	Fitness	Income
count	180.000000	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	3.455556	3.311111	53719.577778
std	6.943498	1.617055	1.084797	0.958869	16506.684226
min	18.000000	12.000000	2.000000	1.000000	29562.000000
25%	24.000000	14.000000	3.000000	3.000000	44058.750000
50%	26.000000	16.000000	3.000000	3.000000	50596.500000
75%	33.000000	16.000000	4.000000	4.000000	58668.000000
max	50.000000	21.000000	7.000000	5.000000	104581.000000

	Miles
count	180.000000
mean	103.194444
std	51.863605
min	21.000000
25%	66.000000
50%	94.000000

```
75%      114.750000
max      360.000000
```

Observations

Age Demographics: The customer base exhibits a diverse age range, spanning from 18 to 50 years, with an average age of approximately 29 years.

Educational Attainment: Customers showcase a broad spectrum of educational backgrounds, ranging from 12 to 21 years, and the average education duration stands at 15+ years.

Product Usage Patterns: The intended usage frequency of the product varies among customers, ranging from 2 to 7 times per week. On average, customers plan to use the product approximately 3 times per week.

Fitness Levels: The average customer rates their fitness at 3 on a 5-point scale, indicating a moderate level of fitness across the user base.

Income Range: The annual income of customers spans from USD 30,000 to USD 100,000, with an average income of approximately USD 54,000.

Running Goals: Customers set diverse weekly running goals, ranging from 21 to 360 miles, with an average target of 103 miles per week.

Finding Duplicated and Missing values

```
af.duplicated().sum()
```

```
0
```

```
af.duplicated().value_counts()
```

```
False    180
dtype: int64
```

```
af.isnull().sum()
```

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

Observations

There are no duplicate or missing values in the dataset.

Non-Graphical Analysis(Counting Values and Unique attributes)

```
for i in af.columns:
    print('The count of unique values in',i,'are')
    print(af[i].nunique())
```

```
The count of unique values in Product are
3
The count of unique values in Age are
32
The count of unique values in Gender are
2
The count of unique values in Education are
8
The count of unique values in MaritalStatus are
2
The count of unique values in Usage are
6
The count of unique values in Fitness are
5
The count of unique values in Income are
62
The count of unique values in Miles are
37
```

```
for i in af.columns:
    print('The Unique Values in',i,'are')
    print(af[i].value_counts())
```

```
The Unique Values in Product are
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
The Unique Values in Age are
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
The Unique Values in Gender are
Male      104
Female     76
Name: Gender, dtype: int64
The Unique Values in Education are
16    85
14    55
18    23
15     5
13     5
12     3
21     3
20     1
Name: Education, dtype: int64
The Unique Values in MaritalStatus are
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
The Unique Values in Usage are
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
The Unique Values in Fitness are
3    97
5    31
2    26
```

```
4    24
1     2
Name: Fitness, dtype: int64
The Unique Values in Income are
45480    14
52302     9
46617     8
54576     8
53439     8
         ..
65220     1
55713     1
68220     1
30699     1
95508     1
Name: Income, Length: 62, dtype: int64
The Unique Values in Miles are
85     27
95     12
66     10
75     10
47      9
106     9
94      8
113     8
53      7
100     7
180     6
200     6
56      6
64      6
127     5
160     5
42      4
150     4
38      3
74      3
170     3
120     3
103     3
132     2
141     2
280     1
260     1
300     1
240     1
112     1
212     1
80      1
```

```
140     1
21      1
169     1
188     1
360     1
Name: Miles, dtype: int64
```

```
for i in af.columns:
    print('The Unique Values in',i,'are')
    print(af[i].unique())
```

```
The Unique Values in Product are
['KP281' 'KP481' 'KP781']
The Unique Values in Age 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]
The Unique Values in Gender are
['Male' 'Female']
The Unique Values in Education are
[14 15 12 13 16 18 20 21]
The Unique Values in MaritalStatus are
['Single' 'Partnered']
The Unique Values in Usage are
[3 2 4 5 6 7]
The Unique Values in Fitness are
[4 3 2 1 5]
The Unique Values in Income 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]
The Unique Values in Miles 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]
```

```
af["Product"].value_counts()
```

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

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

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

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

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

```
af1= af[["Product","Gender","MaritalStatus"]].melt()
af1.groupby(['variable','value'])[['value']].count()/len(af)
```

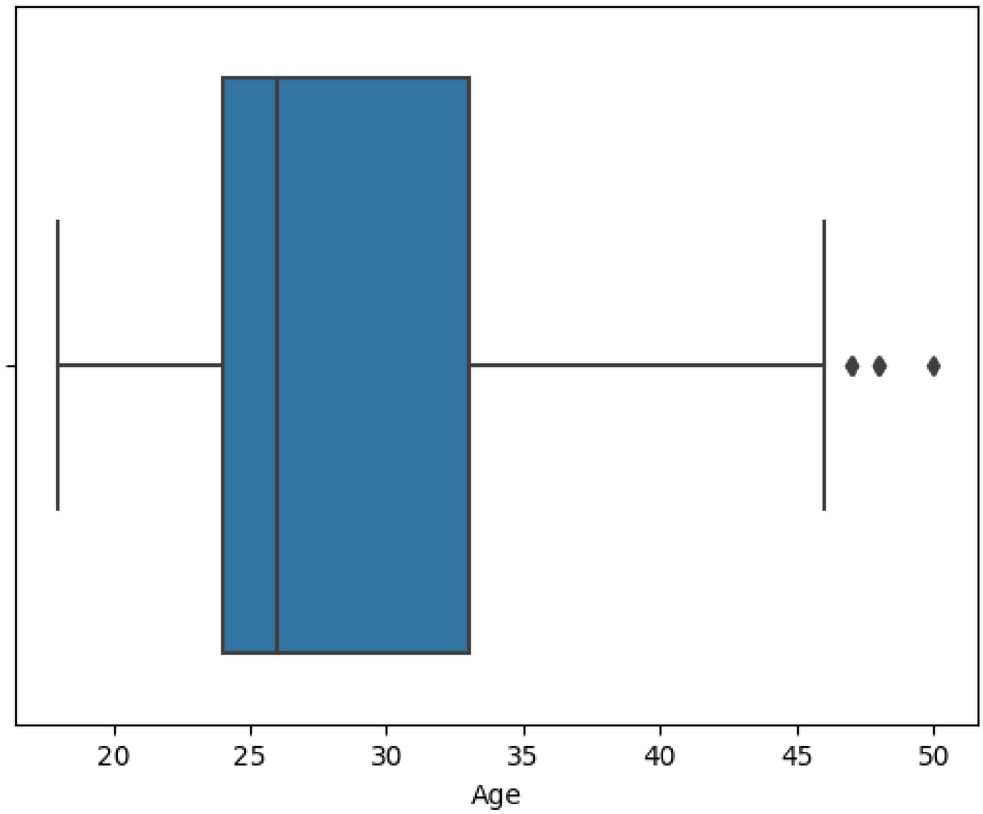
variable	value	value
Gender	Female	0.422222
	Male	0.577778
MaritalStatus	Partnered	0.594444
	Single	0.405556
Product	KP281	0.444444
	KP481	0.333333
	KP781	0.222222

Observations

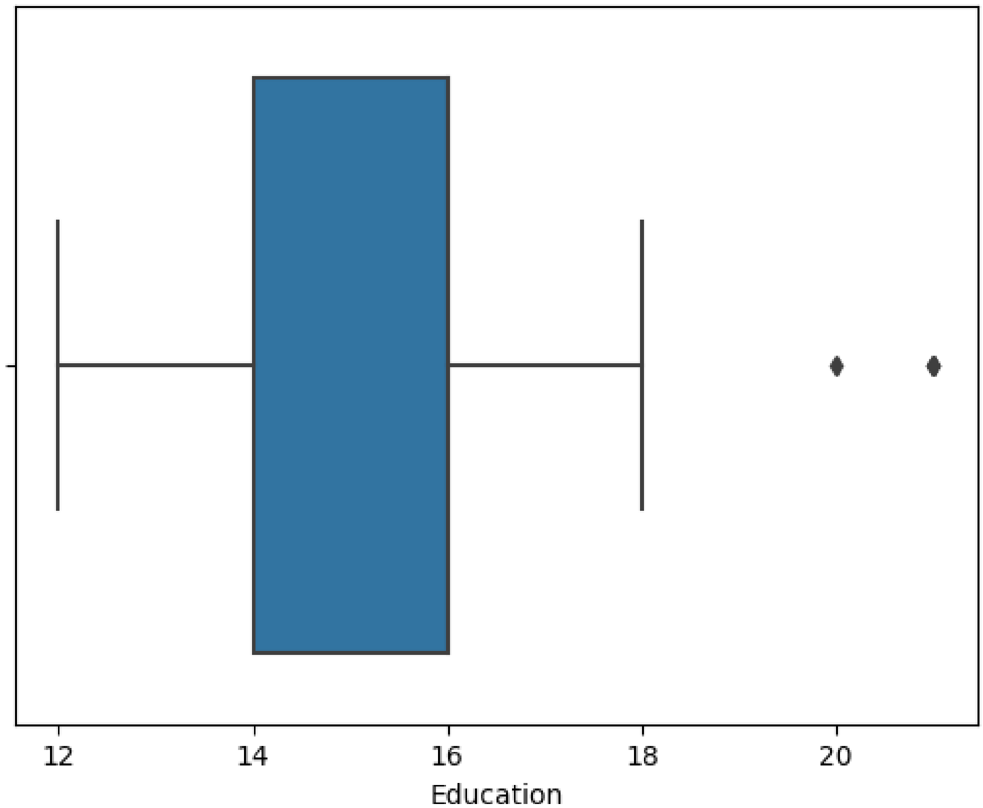
- KP281, KP481, KP781 are the 3 different products
- The KP281 product has emerged as the top-performing product, contributing significantly to the overall sales with an impressive share of approximately 44%.
- There are 32 unique ages.
- 104 Males and 76 Females are in the customers list.The customer base is slightly inclined towards males, constituting around 58% of buyers, while females account for approximately 42%.
- 8 unique set of Educations (14, 15, 12, 13, 16, 18, 20, 21)
- Highest rated Fitness rating is 3
- Most customers usage treadmill atleast 3 days per week
- A majority of buyers, approximately 60%, were married, while the remaining 40% were single. This insight sheds light on the demographic composition of the customer base in terms of marital status.

Outliers Detection for the columns(Age,Education,Income,Miles)

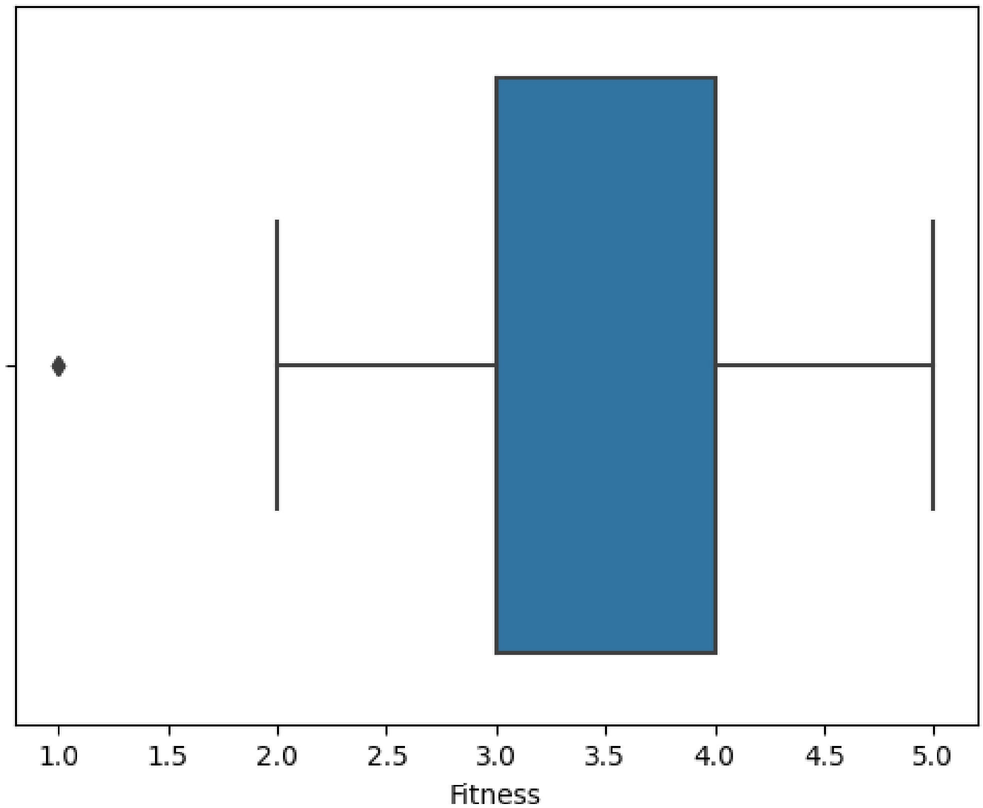
```
sns.boxplot(data =af,x='Age')
plt.show()
```



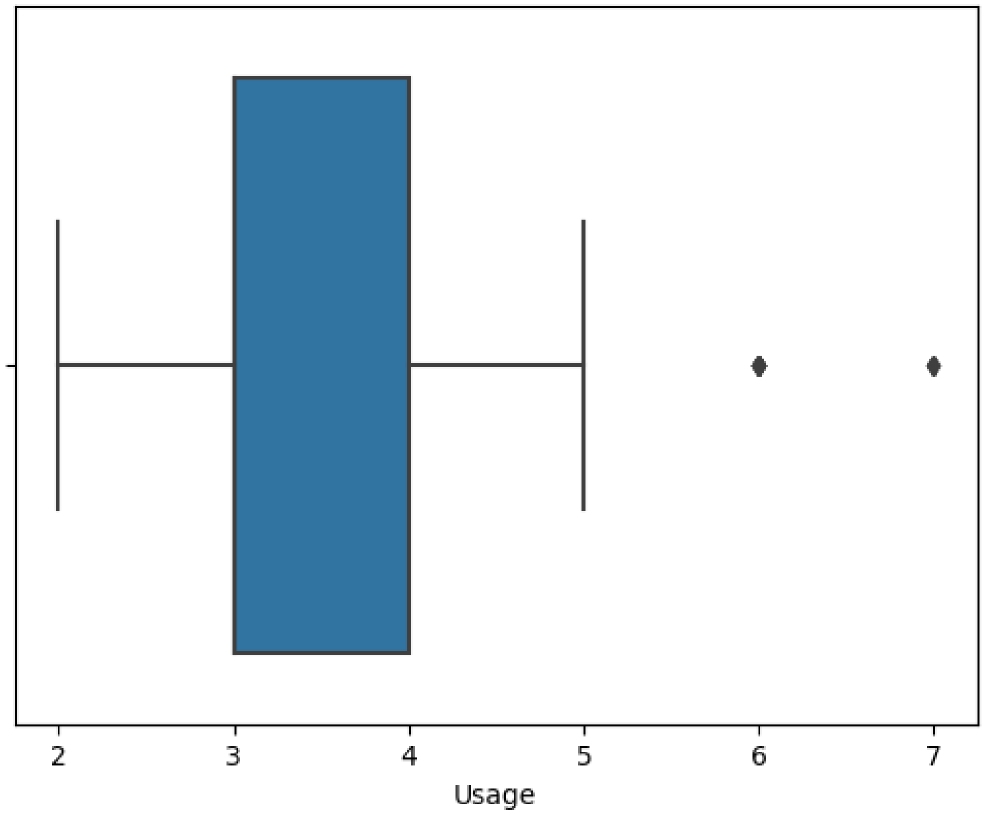

```
sns.boxplot(data =af,x='Education')
plt.show()
```



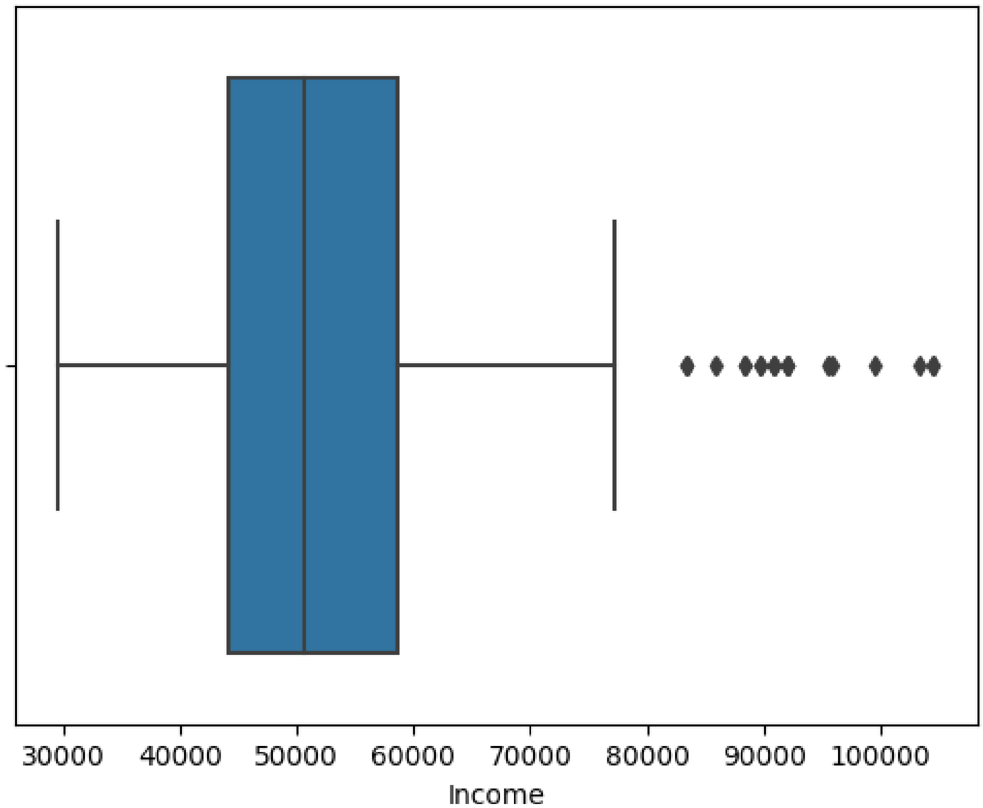

```
sns.boxplot(data=af,x='Fitness')
plt.show()
```




```
sns.boxplot(data=af,x='Usage')
plt.show()
```




```
sns.boxplot(data =af,x="Income")
plt.show()
```



Calculating the IQR for Income column

```
q1 = np.percentile(af["Income"],25)
q3 = np.percentile(af["Income"],75)
IQR = q3-q1
Upper_band = q3+1.5*(IQR)
Lower_band = q1-1.5*(IQR)
Median = af["Income"].median()

print("Q1=", q1)
print("Q3=", q3)
print("IQR=", IQR)
print("Upper band=", Upper_band)
print("Lower band=", Lower_band)
print("Median", af["Income"].median())

Q1= 44058.75
Q3= 58668.0
IQR= 14609.25
Upper band= 80581.875
Lower band= 22144.875
Median 50596.5
```

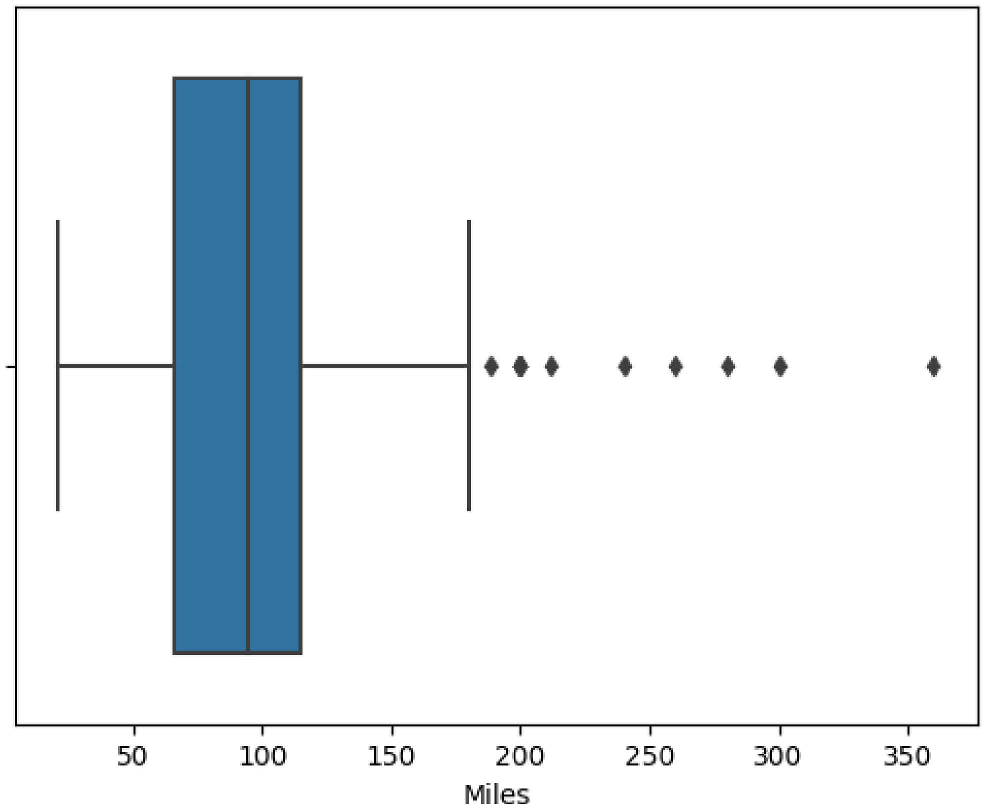
All values above the upper band i.e >80581.875 are outliers.

```
(len(af.loc[af["Income"]>Upper_band])/len(af))*100
```

```
10.555555555555555
```

10.5% values in the Income column are outliers.

```
sns.boxplot(data =af,x="Miles")
plt.show()
```



Calculating the IQR for Miles column

```
q1 = np.percentile(af["Miles"],25)
q3 = np.percentile(af["Miles"],75)
IQR = q3-q1
Upper_band = q3+1.5*(IQR)
Lower_band = q1-1.5*(IQR)

print("Q1=", q1)
print("Q3=", q3)
print("IQR=", IQR)
print("Upper band=", Upper_band)
print("Lower band=", Lower_band)
print("Median", af["Miles"].median())
```

```
Q1= 66.0
Q3= 114.75
IQR= 48.75
Upper band= 187.875
Lower band= -7.125
Median 94.0
```

All values above the upper band i.e >187.875 are outliers.

```
(len(af.loc[af["Miles"]>Upper_band])/len(af))*100
```

```
7.222222222222221
```

7.2% values in the Miles column are outliers.

Observations

- 85% of the customers fall in the age range of 18 to 35. with a median age of 26, suggesting young people showing more interest in the companies products
- 98% of the customers have education more than 13 years highlighting a strong inclination among well-educated individuals to purchase the products. It's plausible that health awareness driven by education could play a pivotal role in this trend.
- Almost 60% of the customers fall in the income group of (40k to 60k) dollars suggesting higher inclination of this income group people towards the products.

Outliers

- As we can see from the box plot, there are 3 outlier's present in the age data.
- There are 2 outliers in the Education column.
- There is 1 outliers in the Fitness column.
- There are 2 outliers in the Usage column.
- As we can see from the box plot, the majority of the outlier's are present in the Income with 10.5% outliers and Miles with 7.2% outliers compared to other parameters.

Statistical Summary

```
af["Product"].value_counts()
```

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

```
(af["Product"].value_counts()/len(af["Product"]))*100
```

```
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: Product, dtype: float64
```

- 44.44% of customers bought KP281 product type
- 33.33% of customers bought KP481 product type
- 22.22% of customers bought KP781 product type

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

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

```
(af["Gender"].value_counts()/len(af["Gender"]))*100
```

```
Male      57.777778
Female    42.222222
Name: Gender, dtype: float64
```

- 57.78% of customers are Male and 42.22% customers are Female

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

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

```
(af["MaritalStatus"].value_counts()/len(af["MaritalStatus"]))*100
```

```
Partnered    59.444444
Single       40.555556
Name: MaritalStatus, dtype: float64
```

- 59.44% of customers are Married/Partnered
- 40.56% of customers are Single

```
(af["Usage"].value_counts()/len(af["Usage"]))*100
```

```
3    38.333333
4    28.888889
2    18.333333
5     9.444444
6     3.888889
7     1.111111
Name: Usage, dtype: float64
```

- Around 39% of customers use 3 days per week
- Less than 2% of customers use 7 days per week

```
af["Fitness"].value_counts()
```

```
3    97
5    31
2    26
4    24
```

```
1      2
Name: Fitness, dtype: int64
```

```
af["Fitness"].describe()
```

```
count    180.000000
mean       3.311111
std        0.958869
min        1.000000
25%        3.000000
50%        3.000000
75%        4.000000
max        5.000000
Name: Fitness, dtype: float64
```

```
rating = (af["Fitness"].value_counts()/len(af["Fitness"]))*100
rating
```

```
3    53.888889
5    17.222222
2    14.444444
4    13.333333
1     1.111111
Name: Fitness, dtype: float64
```

```
fitness_gender_af =af.groupby(["Fitness",'Gender']).size().unstack()
fitness_gender_af
```

```
Gender   Female  Male
Fitness
1             1     1
2            16    10
3            45    52
4             8    16
5             6    25
```

```
(fitness_gender_af/len(af["Gender"]))*100
```

```
Gender      Female       Male
Fitness
1         0.555556   0.555556
2         8.888889   5.555556
3        25.000000  28.888889
4         4.444444   8.888889
5         3.333333  13.888889
```

- More than 53.8% of customers have rated themselves as average in fitness (rated 3). Among this 53% of customers 28.8% are male and 25% are female
- 14% of customers have rated their fitness less than average

- Over 17% of customers have peak fitness ratings

```
af.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
```

```
af.describe(include = "object")
```

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

Insights

Product

- 44.44% of customers bought KP281 product type
- 33.33% of customers bought KP481 product type
- 22.22% of customers bought KP781 product type

Gender

- 57.78% of customers are Male and 42.22% customers are Female

Marital Status

- 59.44% of customers are Married/Partnered
- 40.56% of customers are Single

Usage

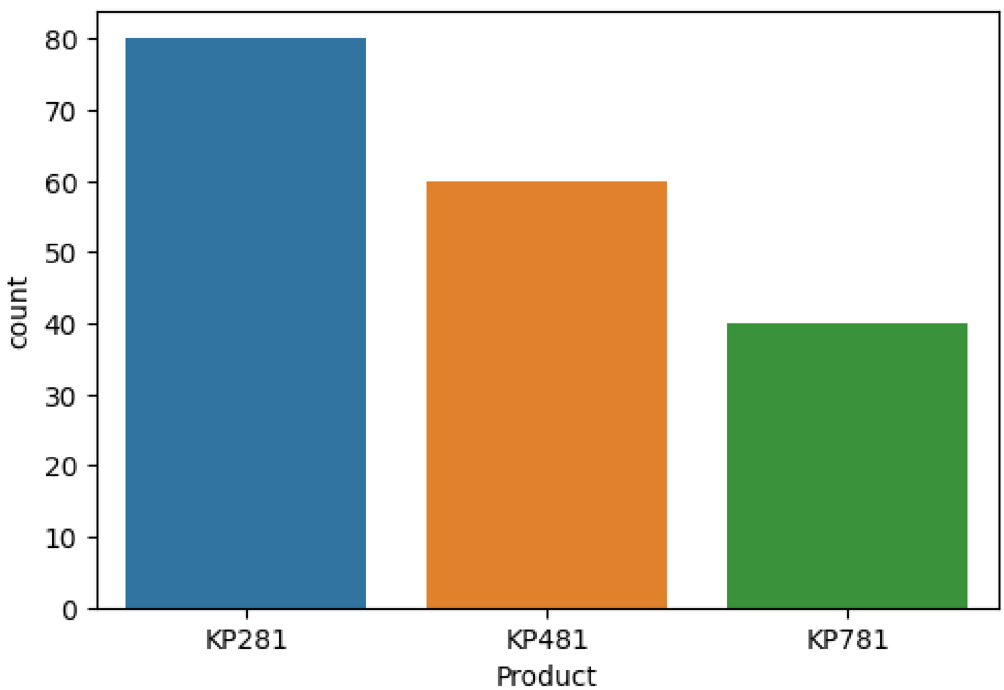
- Around 39% of customers use 3 days per week
- Less than 2% of customers use 7 days per week

Fitness

- More than 53.8% of customers have rated themselves as average in fitness (rated 3). Among this 53% of customers 28.8% are male and 25% are female
- 14% of customers have rated their fitness less than average
- Over 17% of customers have peak fitness ratings

Univariate Analysis

```
plt.figure(figsize=(6,4))
sns.countplot(data=af,x=af['Product'])

plt.show()
```

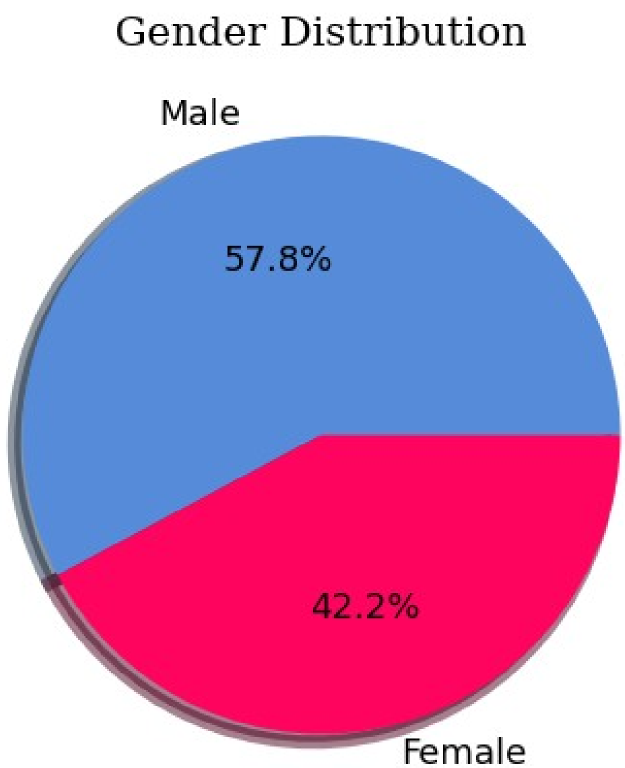


- KP281 is the most commonly purchase product type
- KP481 is the second most top product type purchased
- KP781 is the least purchased product type

```
fig = plt.figure(figsize = (12,5))
gs = fig.add_gridspec(1,2)

ax0 = fig.add_subplot(gs[0,0])

color_map = ["#558bd9", "#ff045f"]
ax0.pie(af['Gender'].value_counts().values,labels =
af['Gender'].value_counts().index,autopct = '%.1f%%',
        shadow = True,colors = color_map,wedgeprops = {'linewidth':
5},textprops={'fontsize': 13, 'color': 'black'})
```

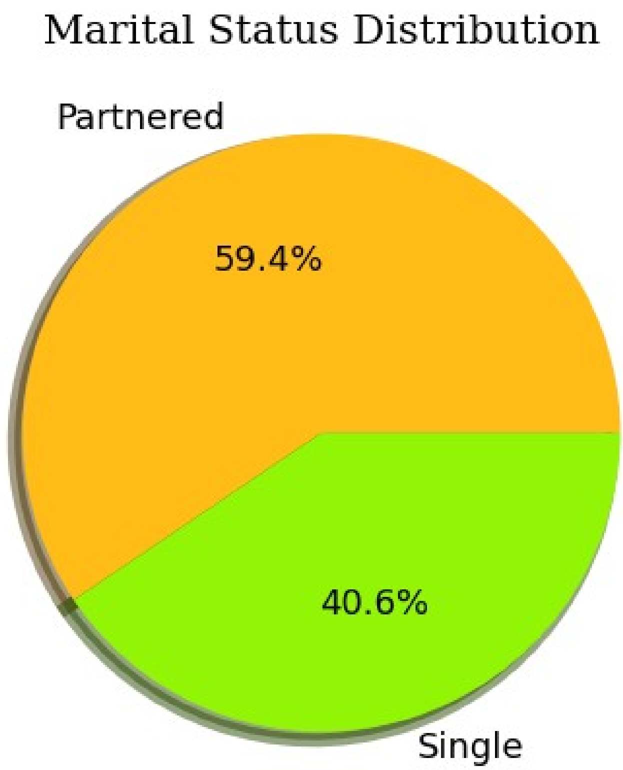
```
ax0.set_title('Gender Distribution',{'font':'serif',
'size':15,'weight':'regular'})
```

```
Text(0.5, 1.0, 'Gender Distribution')
```



- 57.8% products purchased by Males, females are less interested in the product compared to Males.

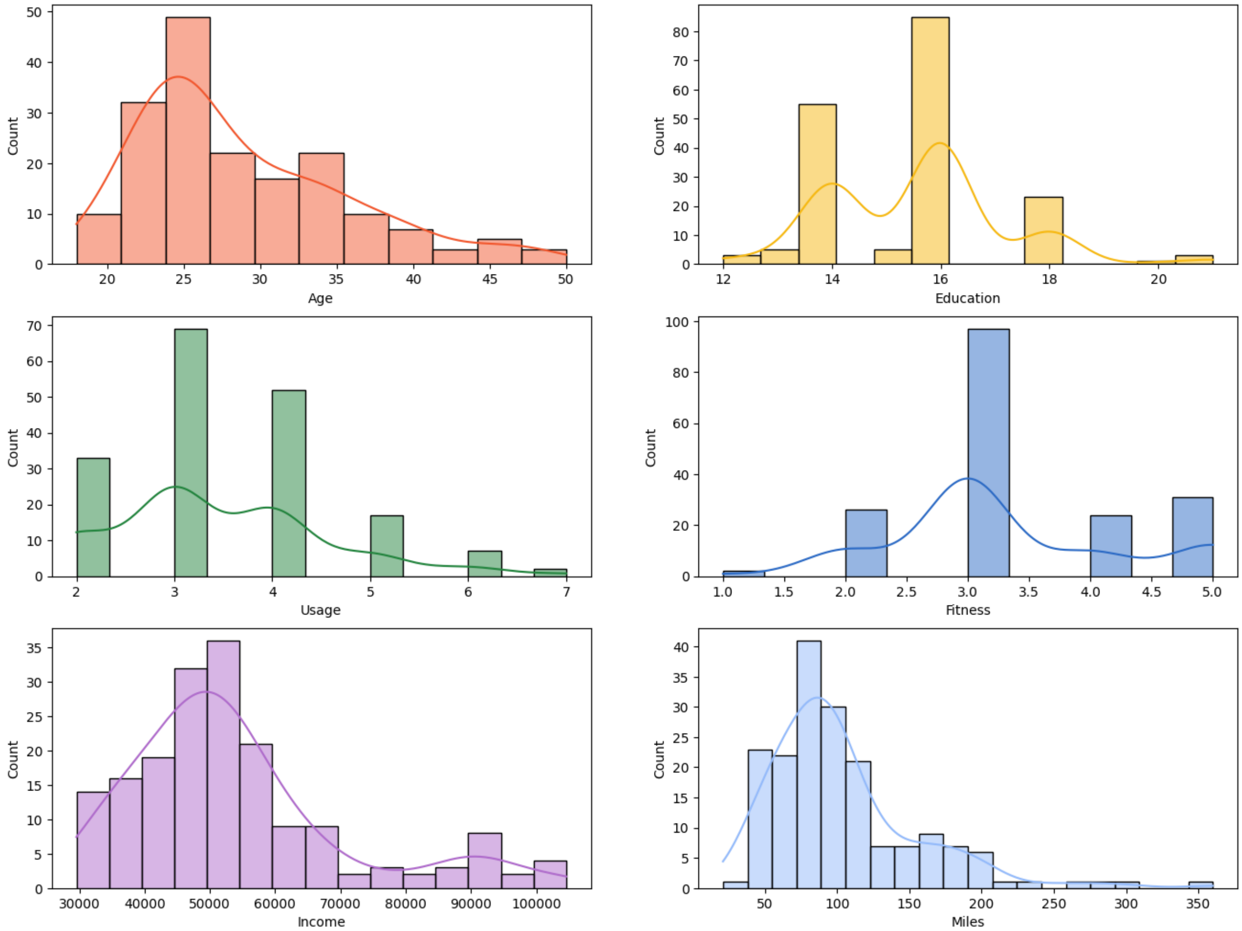
```
fig = plt.figure(figsize = (12,5))
gs = fig.add_gridspec(1,2)

ax1 = fig.add_subplot(gs[0,1])

color_map = ["#ffbb16","#92f505"]
ax1.pie(af['MaritalStatus'].value_counts().values,labels =
af['MaritalStatus'].value_counts().index,autopct = '%.1f%%',
        shadow = True,colors = color_map,wedgeprops = {'linewidth':
5},textprops={'fontsize': 13, 'color': 'black'})

ax1.set_title('Marital Status Distribution',{'font':'serif',
'size':15,'weight':'regular'})

plt.show()
```



- 59.4% products purchased by Married customer category

Distribution of data for the columns (Age,Education,Usage,Fitness,Income,Miles)

```
fig,axis= plt.subplots(3,2 , figsize=(16,12))
sns.histplot(data=af, x="Age" , kde=True , ax=
axis[0,0],color="#f25930")
sns.histplot(data=af, x="Education" , kde=True , ax=
axis[0,1],color="#f6b914")
sns.histplot(data=af, x="Usage" , kde=True , ax=
axis[1,0],color="#24853f")
sns.histplot(data=af, x="Fitness" , kde=True , ax=
axis[1,1],color="#2e6cc6")
sns.histplot(data=af, x="Income" , kde=True , ax=
axis[2,0],color="#b06dcc")
sns.histplot(data=af, x="Miles" , kde=True , ax=
axis[2,1],color="#95bbfa")
plt.show()
```



Usage

- 3 days per week is the most common usage among the customers
- 4 days and 2 days per week is the second and third highest usage among the customers
- Very few customers use product 7 days per week

Fitness

- Over 1.5 density customer population have rated their physical fitness rating as Average i.e 3 rated

Income

- Most of customers who have purchased the product have a average income between 40K to 60K
- More than 35 customers earn 50-55K per year
- More than 30 customers earn 45-50K per year
- More than 20 customers earn 55-60K per year

```
bin_range1 = [17,25,35,45,float('inf')]
bin_labels1 = ['Young Adults', 'Adults', 'Middle Aged Adults',
'Elder']
```

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

Categorizing the values in age column in 4 different buckets:

Young Adult: from 18 - 25

Adults: from 26 - 35

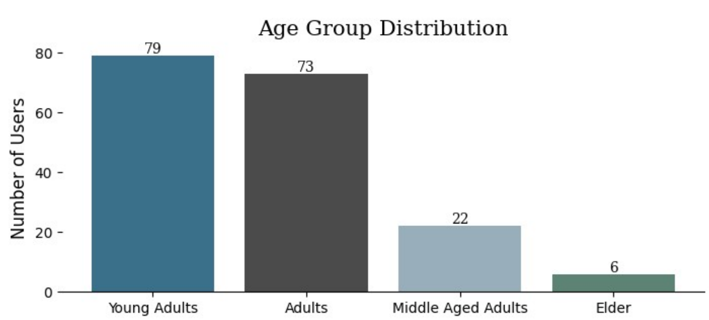
Middle Aged Adults: 36-45

Elder :46 and above

```
af.head()

  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
0    112  Young Adults
1     75  Young Adults
2     66  Young Adults
3     85  Young Adults
4     47  Young Adults

fig = plt.figure(figsize = (15,10))
gs = fig.add_gridspec(2,2,height_ratios=[0.65, 0.35],width_ratios = 
[0.6,0.4])

ax2 = fig.add_subplot(gs[1,0])
temp = af['age_group'].value_counts()
color_map = ["#3A7089", "#4b4b4c",'#99AEBB','#5C8374']
ax2.bar(x=temp.index,height = temp.values,color = color_map,zorder = 
2)

for i in temp.index:
    ax2.text(i,temp[i]+2,temp[i],{'font':'serif','size' : 10},ha = 
'center',va = 'center') #adding the value_counts

for s in ['top','left','right']:      #removing the axis lines
```

```
    ax2.spines[s].set_visible(False)

ax2.set_ylabel('Number of Users',fontweight = 'regular',fontsize = 12)
#adding axis label
ax2.set_xticklabels(temp.index,fontweight = 'regular')

ax2.set_title('Age Group Distribution',{'font':'serif',
'size':15,'weight':'regular'})    #setting title for visual

ax3 = fig.add_subplot(gs[1,1])
age_info = [['Young Adults','44%','18 to 25'],['Adults','41%','26 to
35'],['Middle Aged','12%','36 to 45'],
            ['Elder','3%','Above 45']]
color_2d = [["#3A7089",'#FFFFFF','#FFFFFF'],
["#4b4b4c",'#FFFFFF','#FFFFFF'],['#99AEBB','#FFFFFF','#FFFFFF'],
            ['#5C8374','#FFFFFF','#FFFFFF']]

table = ax3.table(cellText = age_info, cellColours=color_2d,
cellLoc='center',colLabels =['Age','Probability','Group'],
                  colLoc = 'center',bbox =[0, 0, 1, 1])

table.set_fontsize(13)

ax3.axis('off') #removing axis

plt.show()
```

```
<ipython-input-149-9fa53e7b615c>:16: UserWarning: FixedFormatter
should only be used together with FixedLocator
  ax2.set_xticklabels(temp.index,fontweight = 'regular')
```

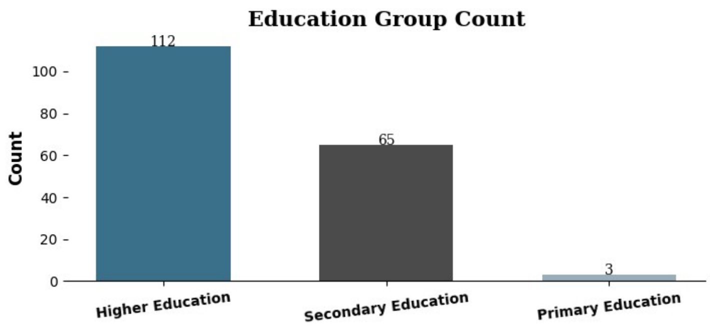


Age	Probability	Group
Young Adults	44%	18 to 25
Adults	41%	26 to 35
Middle Aged	12%	36 to 45
Elder	3%	Above 45

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

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

```python
fig = plt.figure(figsize = (15,10))
gs = fig.add_gridspec(2,2,height_ratios=[0.65, 0.35],width_ratios =
[0.6,0.4])

ax2 = fig.add_subplot(gs[1,0])
temp = af['edu_group'].value_counts()
color_map = ["#3A7089", "#4b4b4c",'#99AEBB']
ax2.bar(x=temp.index,height = temp.values,color = color_map,zorder =
2,width = 0.6)

for i in temp.index:   #adding the value_counts
    ax2.text(i,temp[i]+2,temp[i],{'font':'serif','size' : 10},ha =
'center',va = 'center')

for s in ['top','left','right']:      #removing the axis lines
    ax2.spines[s].set_visible(False)

ax2.set_ylabel('Count',fontweight = 'bold',fontsize = 12) #adding axis
label
ax2.set_xticklabels(temp.index,fontweight = 'bold',rotation = 7)

ax2.set_title('Education Group Count',{'font':'serif',
'size':15,'weight':'bold'}) #setting title for visual

ax3 = fig.add_subplot(gs[1,1])
edu_info = [['Higher','62%','Above 15'],['Secondary','36%','13 to
15'],['Primary','2%','0 to 12']]
color_2d = [["#3A7089",'#FFFFFF','#FFFFFF'],
["#4b4b4c",'#FFFFFF','#FFFFFF'],['#99AEBB','#FFFFFF','#FFFFFF']]

table = ax3.table(cellText = edu_info, cellColours=color_2d,
cellLoc='center',colLabels =['Education','Probability','Years'],
                  colLoc = 'center',bbox =[0, 0, 1, 1])

table.set_fontsize(13)

ax3.axis('off') #removing axis

plt.show()
```

```
<ipython-input-151-2b7d84f6fb2c>:16: UserWarning: FixedFormatter
should only be used together with FixedLocator
  ax2.set_xticklabels(temp.index,fontweight = 'bold',rotation = 7)
```



Education	Probability	Years
Higher	62%	Above 15
Secondary	36%	13 to 15
Primary	2%	0 to 12

```
(af['edu_group'].value_counts()/len(af['Education']))*100
```

```
Higher Education       62.222222
Secondary Education    36.111111
Primary Education       1.666667
Name: edu_group, dtype: float64
```

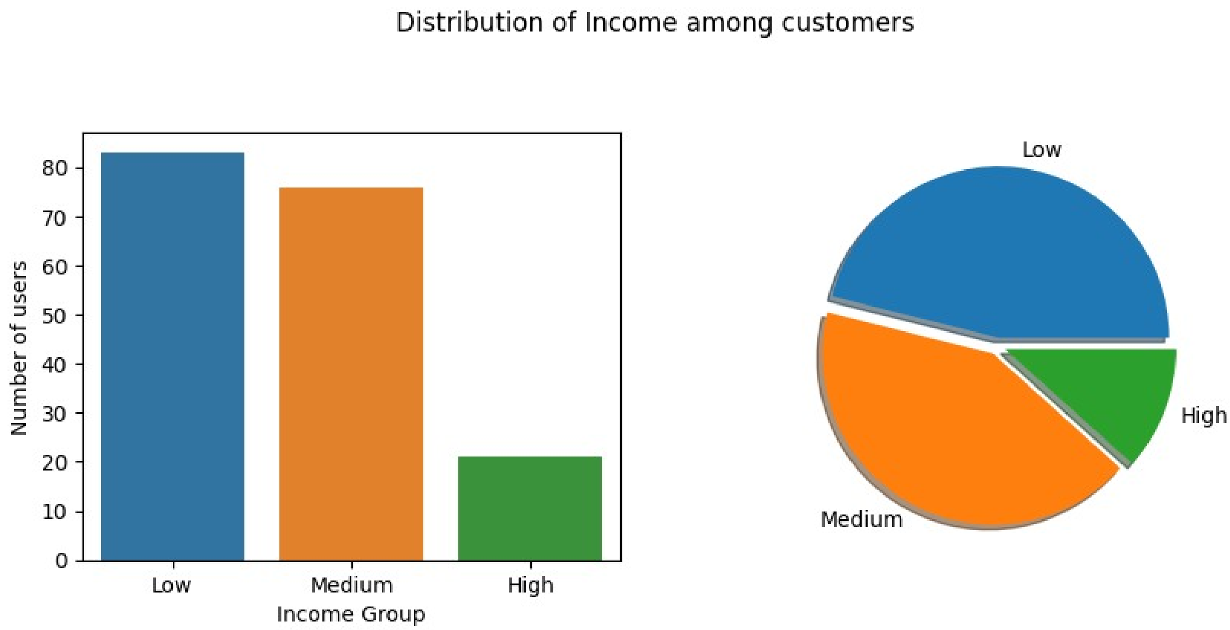
About 98.4% of the Aerofit customers have completed their Secondary and Higher educations.The remaining 1.6% of the group have completed their Primary education (i.e < 12 years).

```
af["Income group"]=pd.cut(af["Income"],bins=[29000,50000,75000,105000],labels=["Low","Medium","High"])
```

Categorizing the values in Income column in 3 different buckets:

- Low: 29000-50000
- Medium: 51000-75000
- High: 76000-105000

```
plt.figure(figsize = (10,8))
plt.subplot(2,2,1)
sns.countplot(data =af,x="Income group")
plt.xlabel("Income Group")
plt.ylabel("Number of users")

plt.subplot(2,2,2)
plt.pie(af["Income group"].value_counts(),labels =af["Income group"].unique(),explode=(0.05,0.05,0.05),shadow =True)
plt.suptitle("Distribution of Income among customers")
plt.show()
```

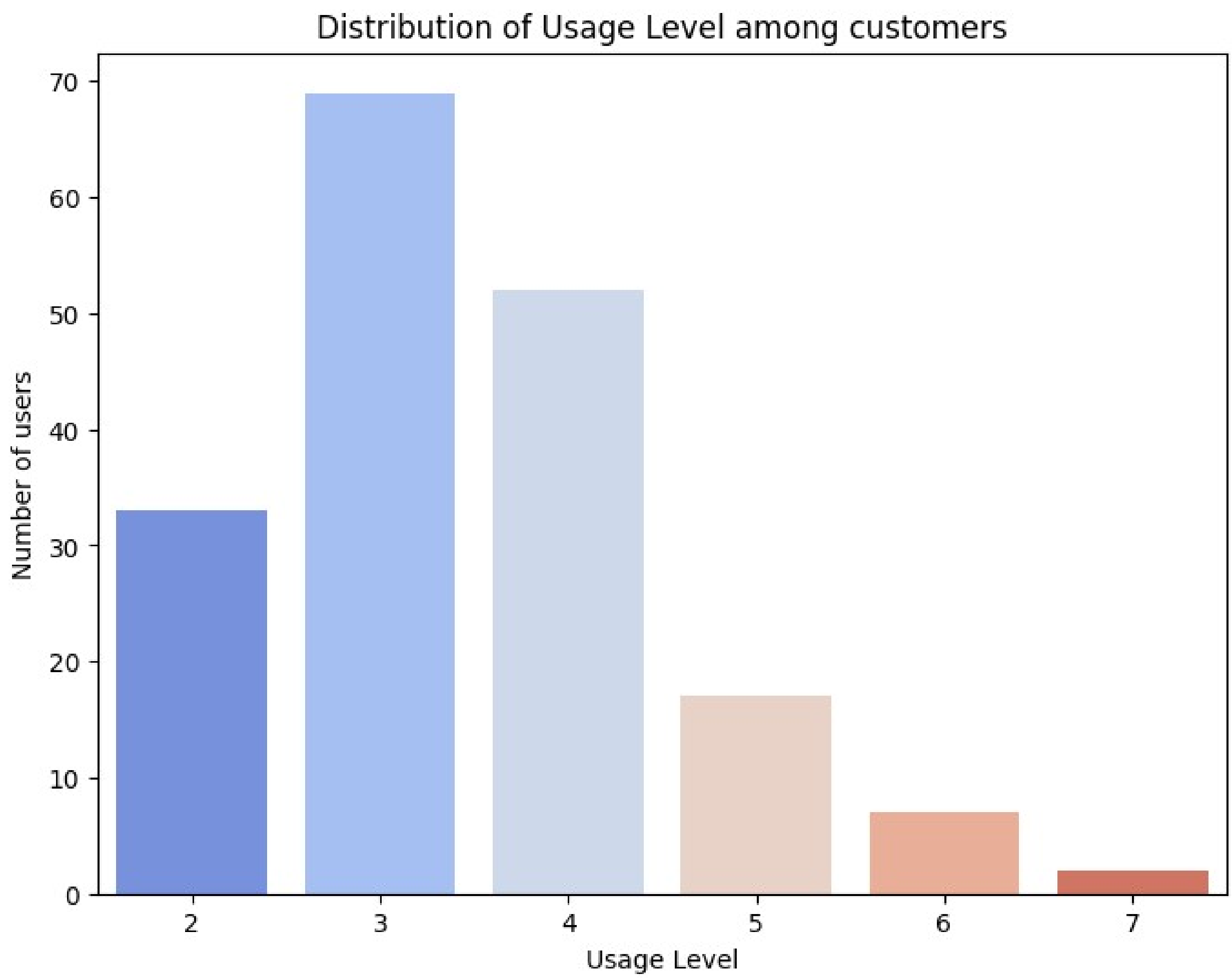



```
(af["Income group"].value_counts()/len(af['Income']))*100
```

```
Low       46.111111
Medium    42.222222
High      11.666667
Name: Income group, dtype: float64
```

About 88% of the Aerofit customers belong to the Low and Medium income group. The remaining 11.66% of the group belong to the high income category (above 75000 USD to $105000 USD).

```
plt.figure(figsize = (8,6))
sns.countplot(data =af,x="Usage", palette ="coolwarm")
plt.xlabel("Usage Level")
plt.ylabel("Number of users")
plt.title("Distribution of Usage Level among customers")
plt.show()
```



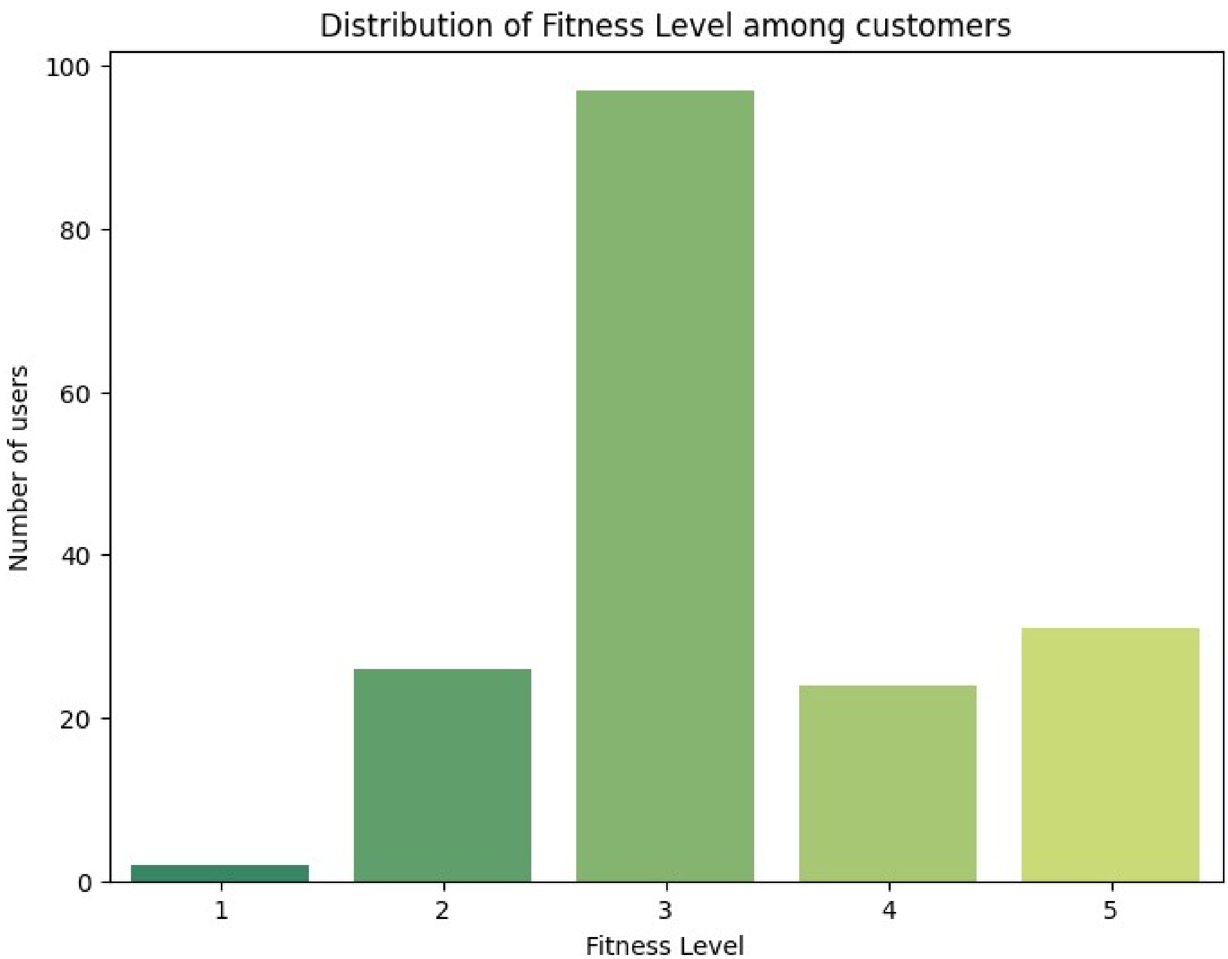

```
(af["Usage"].value_counts()/len(af['Usage']))*100
```

```
3    38.333333
4    28.888889
2    18.333333
5     9.444444
6     3.888889
7     1.111111
Name: Usage, dtype: float64
```

- 3 days per week is the most common usage among the customers which is 38.3% of the customers.
- 4 days and 2 days per week is the second and third highest usage among the customers
- Very few customers use product 7 days per week

```
plt.figure(figsize = (8,6))
sns.countplot(data =af,x="Fitness", palette ="summer")
plt.xlabel("Fitness Level")
plt.ylabel("Number of users")
```

```
plt.title("Distribution of Fitness Level among customers")
plt.show()
```



- More than 90 customers have rated their physical fitness rating as Average i.e 3

Observations

Product

- KP281 is the most commonly purchase product type
- KP481 is the second most top product type purchased
- KP781 is the least purchased product type

Age

- 18 to 35 is the most common customer age group that has purchased the product.

Gender

- 57.8% products purchased by Males, females are less interested in the product compared to Males.

Marital Status

- 59.4% products purchased by Married customer category

Distribution of data

Usage

- 3 days per week is the most common usage among the customers (38.3%)
- 4 days and 2 days per week is the second and third highest usage among the customers
- Very few customers use product 7 days per week

Fitness

- More than 90 customers have rated their physical fitness rating as Average i.e 3 rated

Income

- Most of customers who have purchased the product have a average income between 40K to 60K.
- About 88% of the Aerofit customers belong to the Low and Medium income group. The remaining 11.66% of the group belong to the high income category (above 75000 USD to $105000 USD).

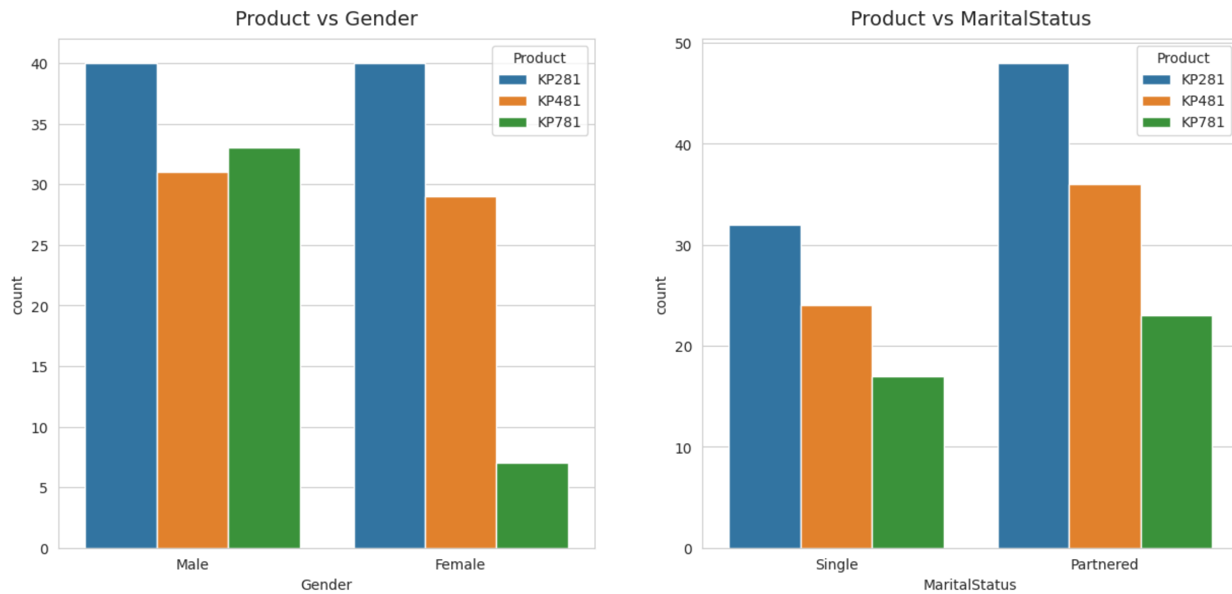
Education

- About 98.4% of the Aerofit customers have completed their Secondary and Higher educations.The remaining 1.6% of the group have completed their Primary education (i.e < 12 years).

Bivariate Analysis

Product vs Gender , Product vs Marital Status

```
sns.set_style(style='whitegrid')
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15, 6.5))
plt.figure(figsize = (8,6))
sns.countplot(data= af,hue= "Product", x= "Gender" ,ax=axs[0])
sns.countplot(data= af,hue= "Product",x= "MaritalStatus",ax=axs[1])
axs[0].set_title("Product vs Gender", pad=10, fontsize=14)
axs[1].set_title("Product vs MaritalStatus", pad=10, fontsize=14)
plt.show()
```




```
<Figure size 800x600 with 0 Axes>
```

```
gender_af =af.groupby(["Gender","Product"]).size().unstack()
gender_af
```

Product	KP281	KP481	KP781
Gender			
Female	40	29	7
Male	40	31	33

```
maritalstatus_af
=af.groupby(["MaritalStatus","Product"]).size().unstack()
maritalstatus_af
```

Product	KP281	KP481	KP781
MaritalStatus			
Partnered	48	36	23
Single	32	24	17

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.

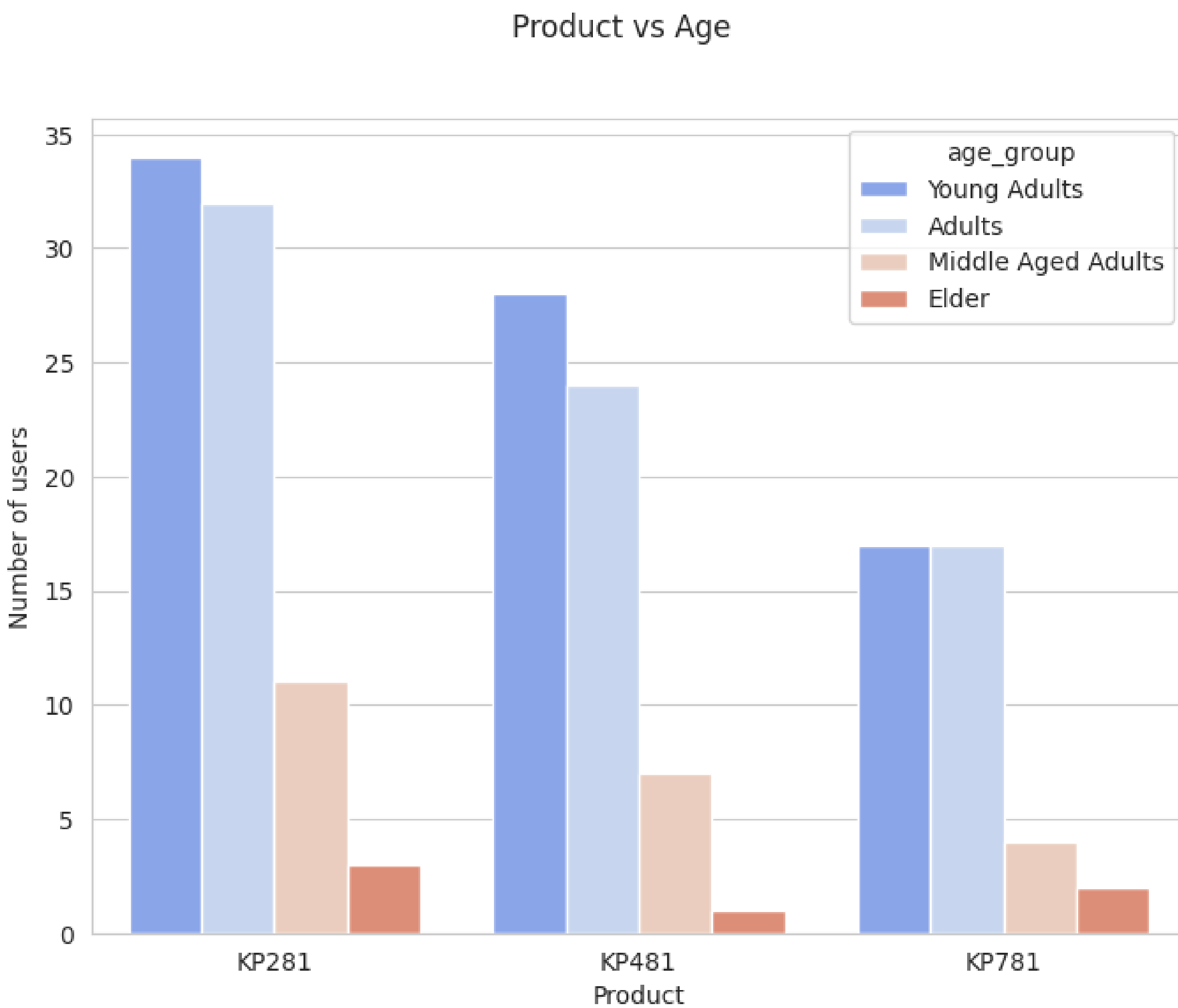
Product vs MaritalStatus

Customer who are Partnered, are more likely to purchase the product.

Product vs Age

```
plt.figure(figsize = (8,6))
sns.countplot(data= af,x= "Product",hue= 'age_group',
palette="coolwarm")
```

```
plt.xlabel("Product")
plt.ylabel("Number of users")
plt.suptitle("Product vs Age")
plt.show()
```




```
age_group_af =af.groupby(['Product','age_group']).size().unstack()
age_group_af
```

age_group Product	Young Adults	Adults	Middle Aged Adults	Elder
KP281	34	32	11	3
KP481	28	24	7	1
KP781	17	17	4	2

```
age_group_af.mean()
```

```
age_group
Young Adults          26.333333
Adults                24.333333
Middle Aged Adults     7.333333
Elder                  2.000000
dtype: float64
```

```
af['Age'].mean()
```

```
28.788888888888888
```

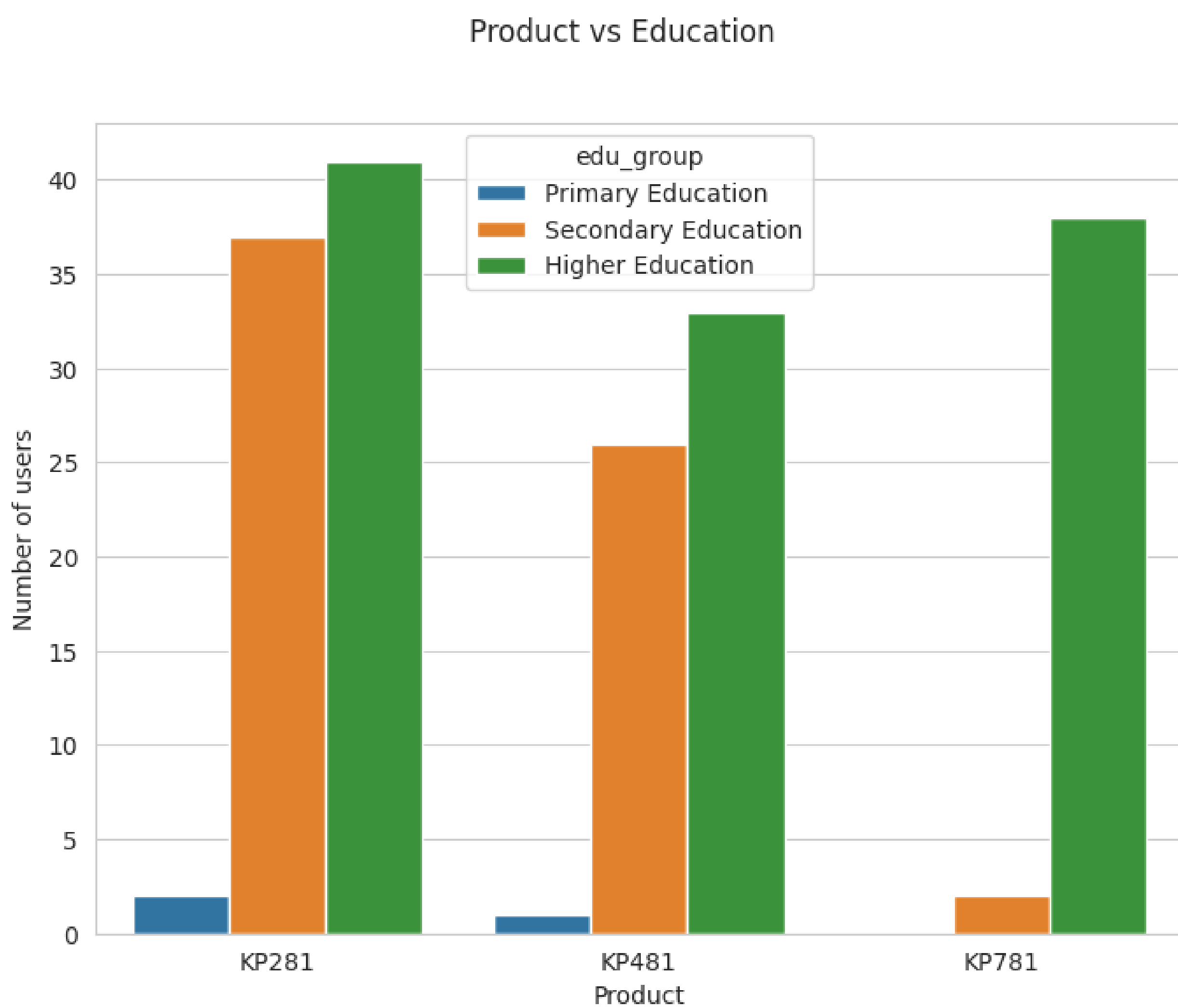
```
af['Age'].median()
```

```
26.0
```

- 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.

Product vs Education

```
plt.figure(figsize = (8,6))
sns.countplot(data= af,hue= 'edu_group',x= "Product")
plt.xlabel("Product")
plt.ylabel("Number of users")
plt.suptitle("Product vs Education")
plt.show()
```



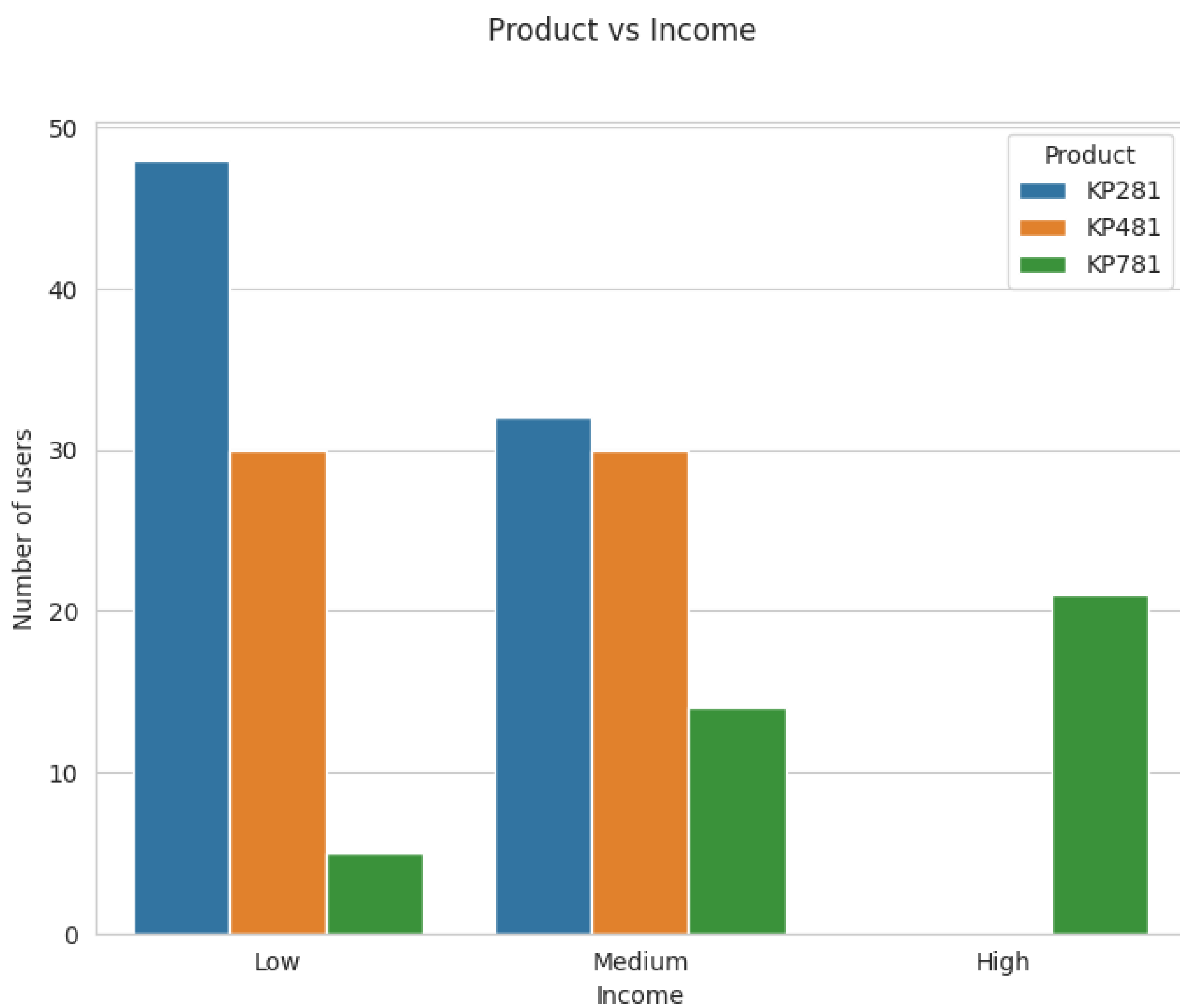

```
edu_group_af =af.groupby(['Product','edu_group']).size().unstack()
edu_group_af

edu_group  Primary Education  Secondary Education  Higher Education
Product
KP281                      2                   37                41
KP481                      1                   26                33
KP781                      0                    2                38
```

- Customers who have completed their higher Education , have more chances to purchase the KP781 product.

Product vs Income

```
plt.figure(figsize = (8,6))
sns.countplot(data= af,x= "Income group",hue= "Product")
plt.xlabel("Income")
plt.ylabel("Number of users")
```

```
plt.suptitle("Product vs Income")
plt.show()
```




```
income_group_af =af.groupby(['Product','Income
group']).size().unstack()
income_group_af
```

Income group	Low	Medium	High
Product			
KP281	48	32	0
KP481	30	30	0
KP781	5	14	21

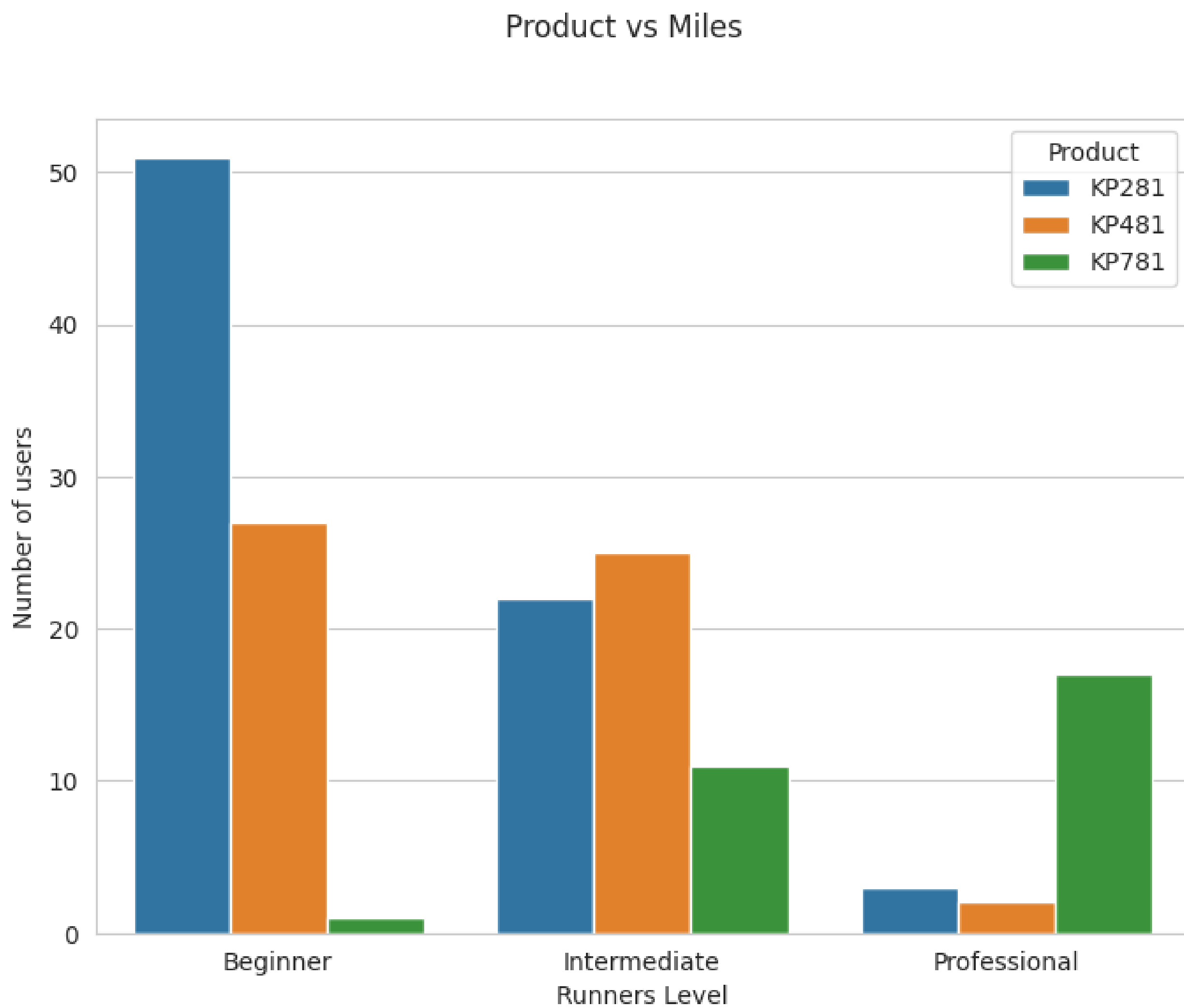
- Higher the Income of the customer (Income >= 60000), higher the chances of the customer to purchase the KP781 product.

Product vs Miles

```
af["Miles"]
```

```
0      112
1       75
2       66
3       85
4       47
      ... 
175    200
176    200
177    160
178    120
179    180
Name: Miles, Length: 180, dtype: int64
```

```
bin_range4 = [45,90,135,180]
bin_labels4 = ['Beginner', 'Intermediate', 'Professional']

af['Mile_group'] = pd.cut(af['Miles'],bins = bin_range4,labels =
bin_labels4)

plt.figure(figsize = (8,6))
sns.countplot(data= af, x= 'Mile_group',hue= "Product")
plt.xlabel("Runners Level")
plt.ylabel("Number of users")
plt.suptitle("Product vs Miles")
plt.show()
```



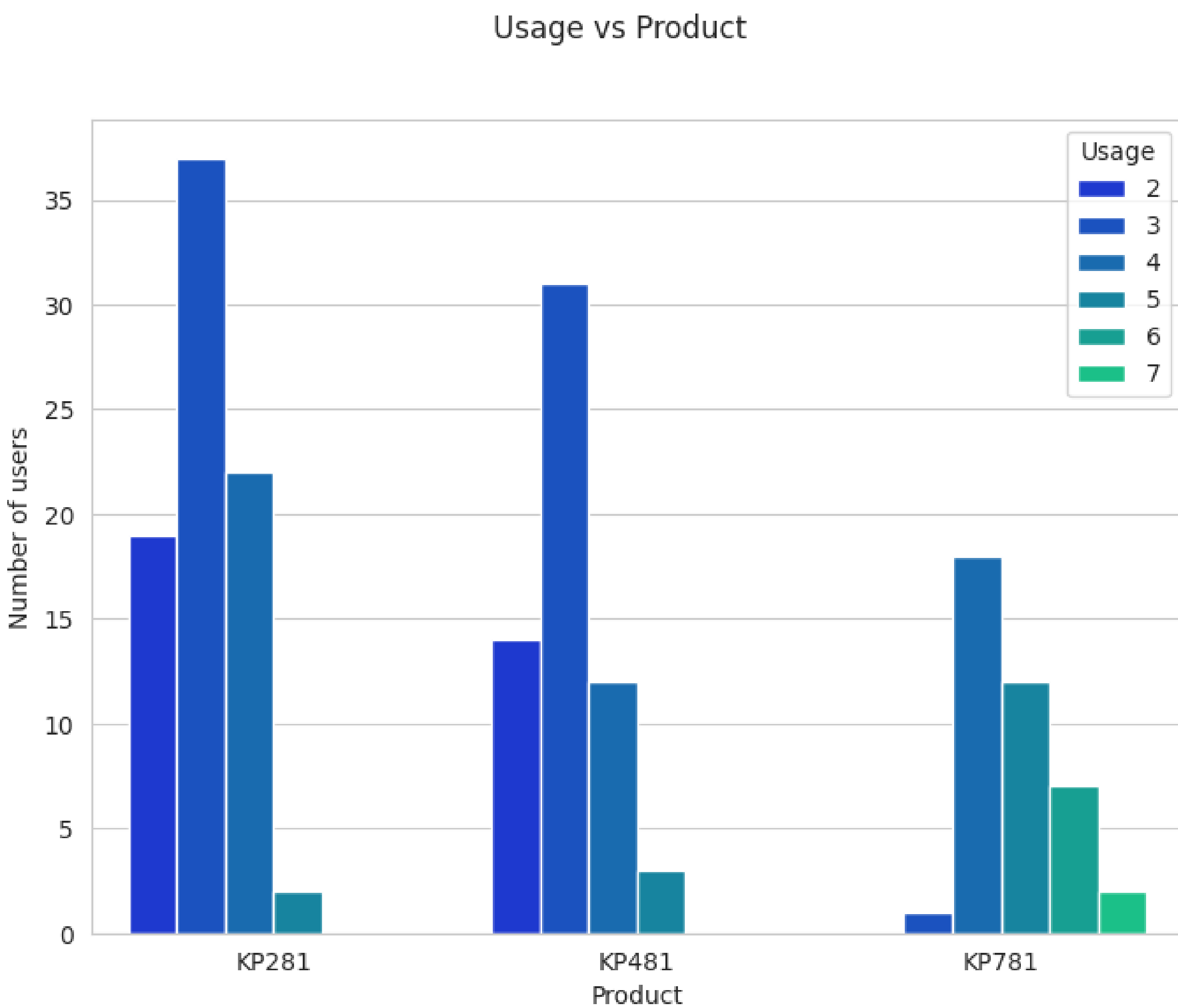

```
Mile_group_af =af.groupby(['Product','Mile_group']).size().unstack()
Mile_group_af
```

Mile_group	Beginner	Intermediate	Professional
Product			
KP281	51	22	3
KP481	27	25	2
KP781	1	11	17

- Beginners are more likely to purchase the KP281.
- Professionals are more likely to purchase the KP781.
- If the customer expects to walk/run greater than 120 Miles per week, it is more likely that the customer will buy KP781 product.

Usage vs Product

```
plt.figure(figsize = (8,6))
sns.countplot(data= af, hue= 'Usage',x= "Product", palette ="winter")
plt.xlabel("Product")
plt.ylabel("Number of users")
plt.suptitle("Usage vs Product")
plt.show()
```



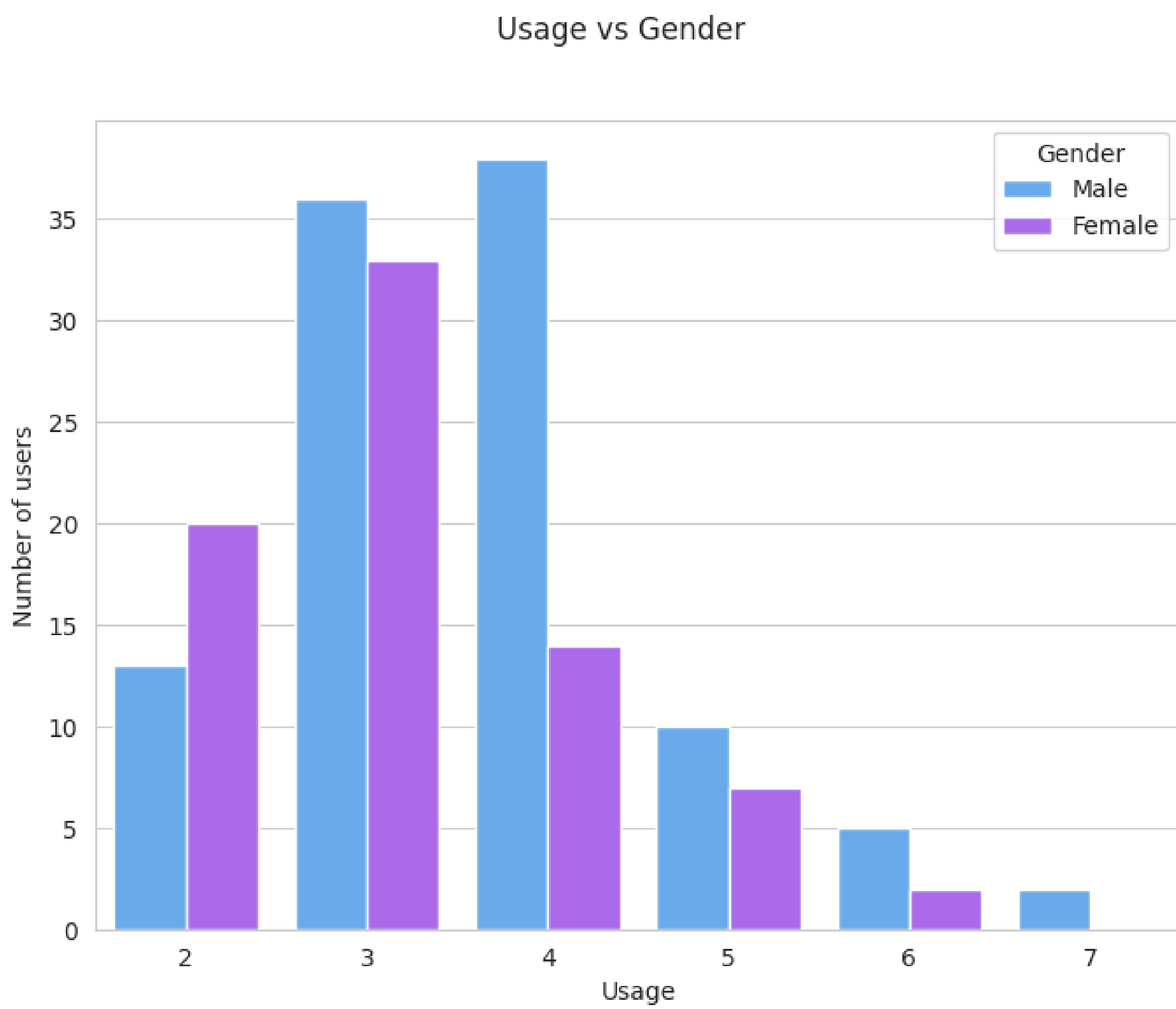

```
usage_group_af =af.groupby(['Product','Usage']).size().unstack()
usage_group_af
```

Usage	2	3	4	5	6	7
Product						
KP281	19.0	37.0	22.0	2.0	NaN	NaN
KP481	14.0	31.0	12.0	3.0	NaN	NaN
KP781	NaN	1.0	18.0	12.0	7.0	2.0

- 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.

Usage vs Gender

```
plt.figure(figsize = (8,6))
sns.countplot(data= af, x= 'Usage',hue= "Gender", palette='cool')
plt.xlabel("Usage")
plt.ylabel("Number of users")
plt.suptitle("Usage vs Gender")
plt.show()
```



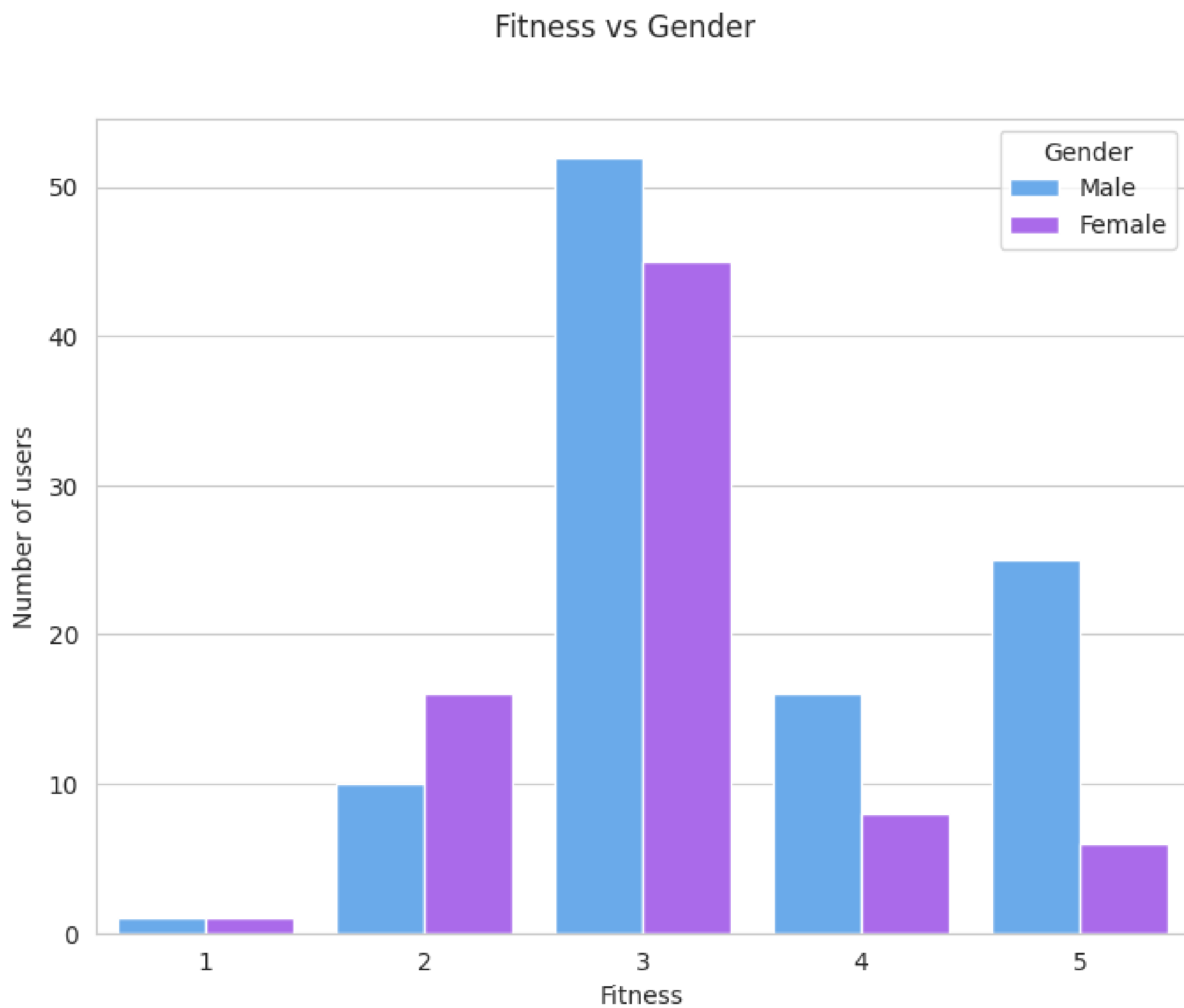

```
usage_gender_af =af.groupby(['Usage',"Gender"]).size().unstack()
usage_gender_af
```

```
Gender  Female  Male
Usage
2         20.0  13.0
3         33.0  36.0
4         14.0  38.0
5          7.0  10.0
6          2.0   5.0
7          NaN   2.0
```

- Among Male and Female genders, Male's usage is 4 days per week Female customers mostly use 3 days per week.
- Only few Male customers use 7 days per week whereas female customer's maximum usage is only 6 days per week.

Fitness vs Gender

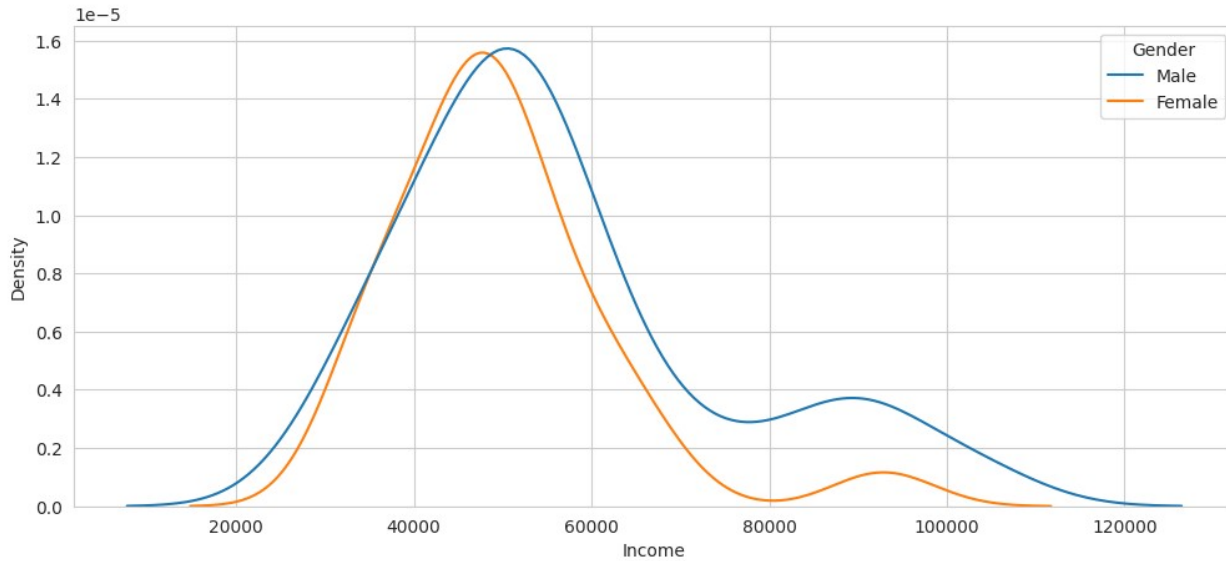
```
plt.figure(figsize = (8,6))
sns.countplot(data= af, x= 'Fitness',hue= "Gender", palette = "cool")
plt.xlabel("Fitness")
plt.ylabel("Number of users")
plt.suptitle("Fitness vs Gender")
plt.show()
```



- About 59% of Males and 50% of Females are found to have the fitness level of 3.
- Among the fitness rating both Male and Female most have rated as average.

Income vs Gender

```
plt.figure(figsize=(12,5))
sns.kdeplot(data=af,x='Income',hue='Gender')
plt.show()
```



- The spike from 40K to around 80K is the most common income per annum of the customers for both the Genders.

Observations

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.

Product vs Education

Customers who have completed their higher Education , have more chances to purchase the KP781 product.

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 are Partnered, are more likely to purchase the product.

Product vs Income

Higher the Income of the customer (Income >= 60000), higher the chances of the customer to purchase the KP781 product.

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.

Product vs Miles

Beginners are more likely to purchase the KP281. Professionals are more likely to purchase the KP781. If the customer expects to walk/run greater than 120 Miles per week, it is more likely that the customer will buy KP781 product.

Usage vs Gender

Among Male and Female genders, Male's usage is 4 days per week Female customers mostly use 3 days per week. Only few Male customers use 7 days per week whereas female customer's maximum usage is only 6 days per week.

Fitness vs Gender

About 59% of Males and 50% of Females are found to have the fitness level of 3. Among the fitness rating both Male and Female most have rated as average.

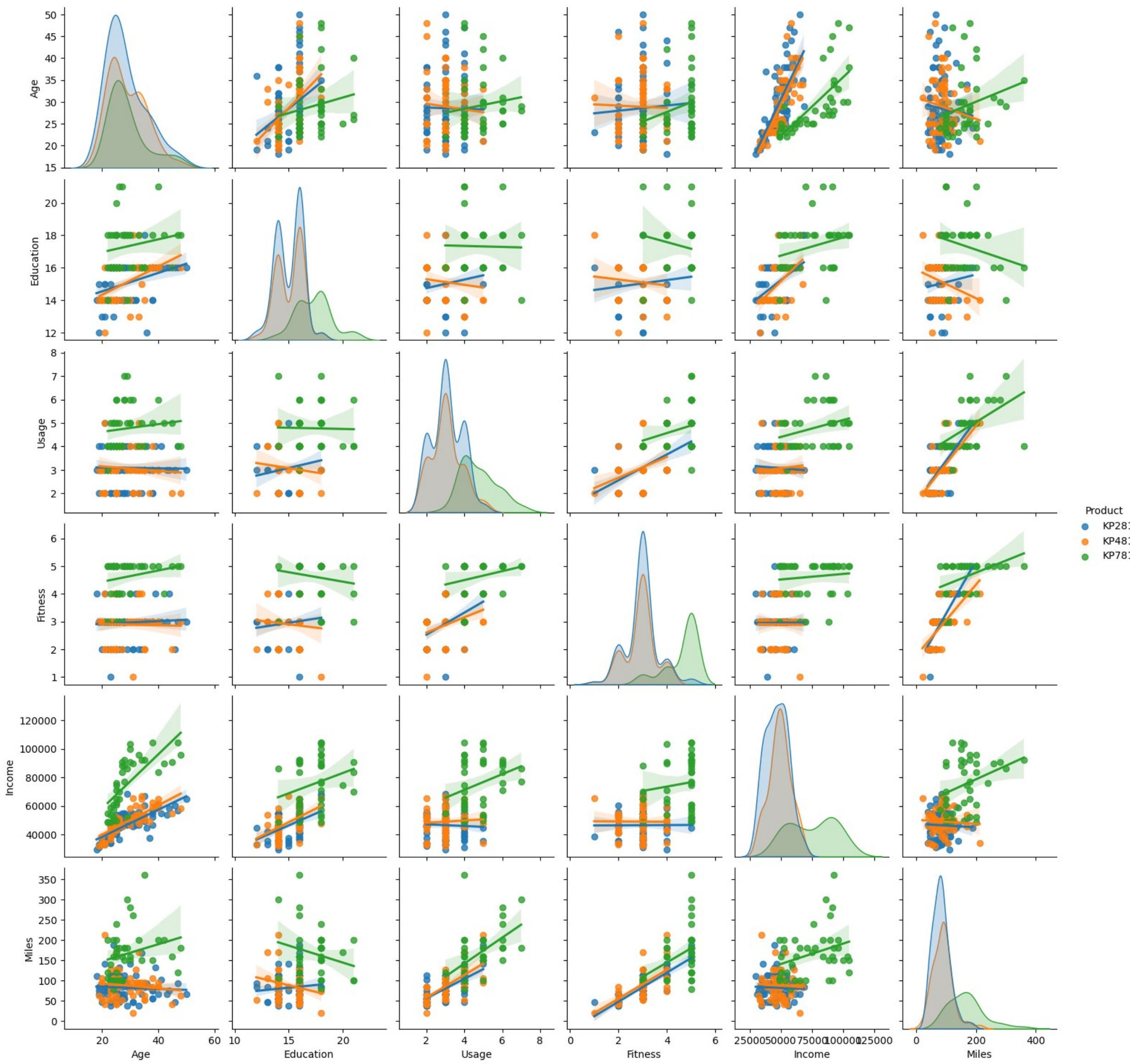
Income vs Gender

The spike from 40K to around 80K is the most common income per annum of the customers.

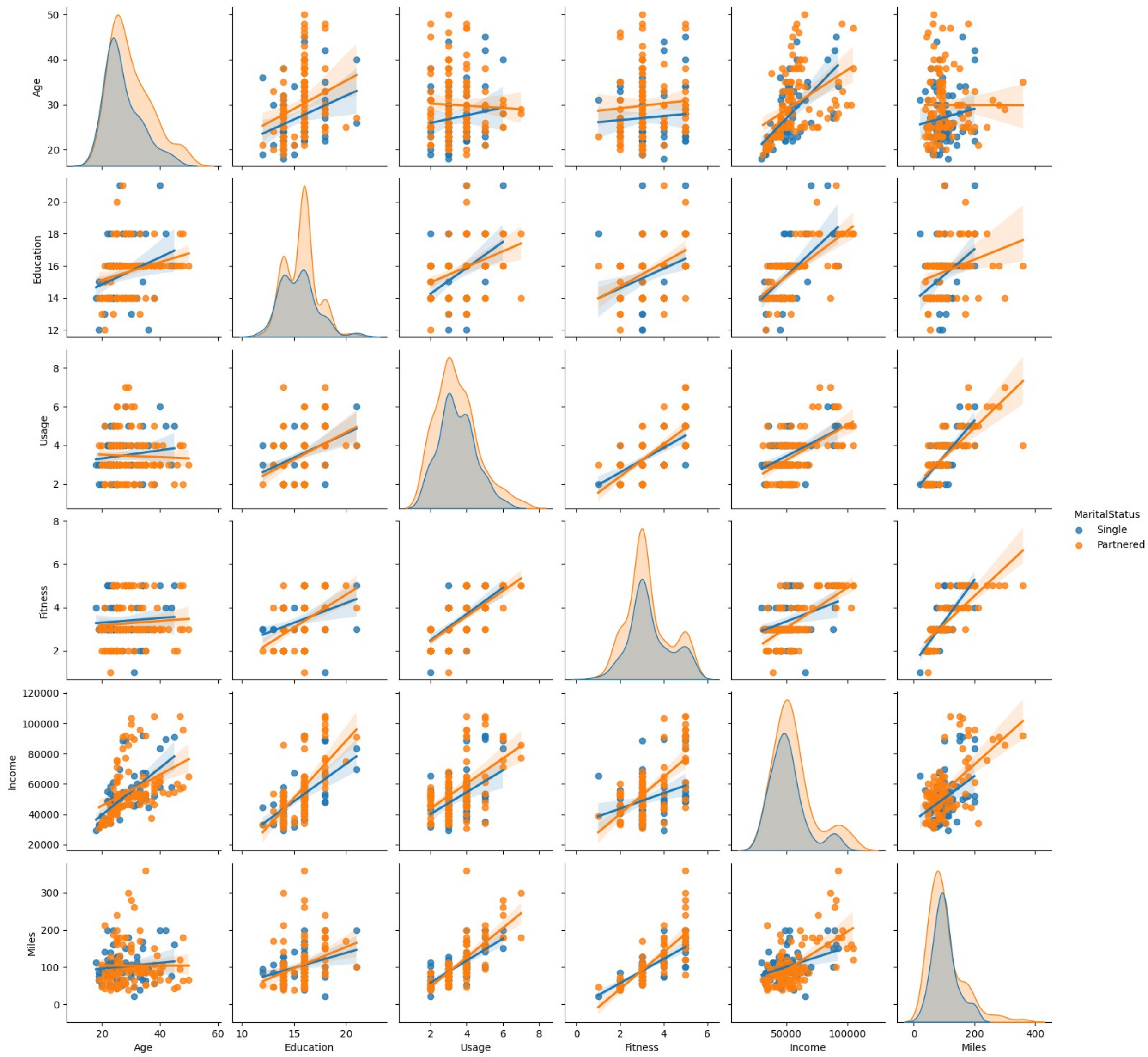
For correlation: Heatmaps, Pairplots

Pair plot

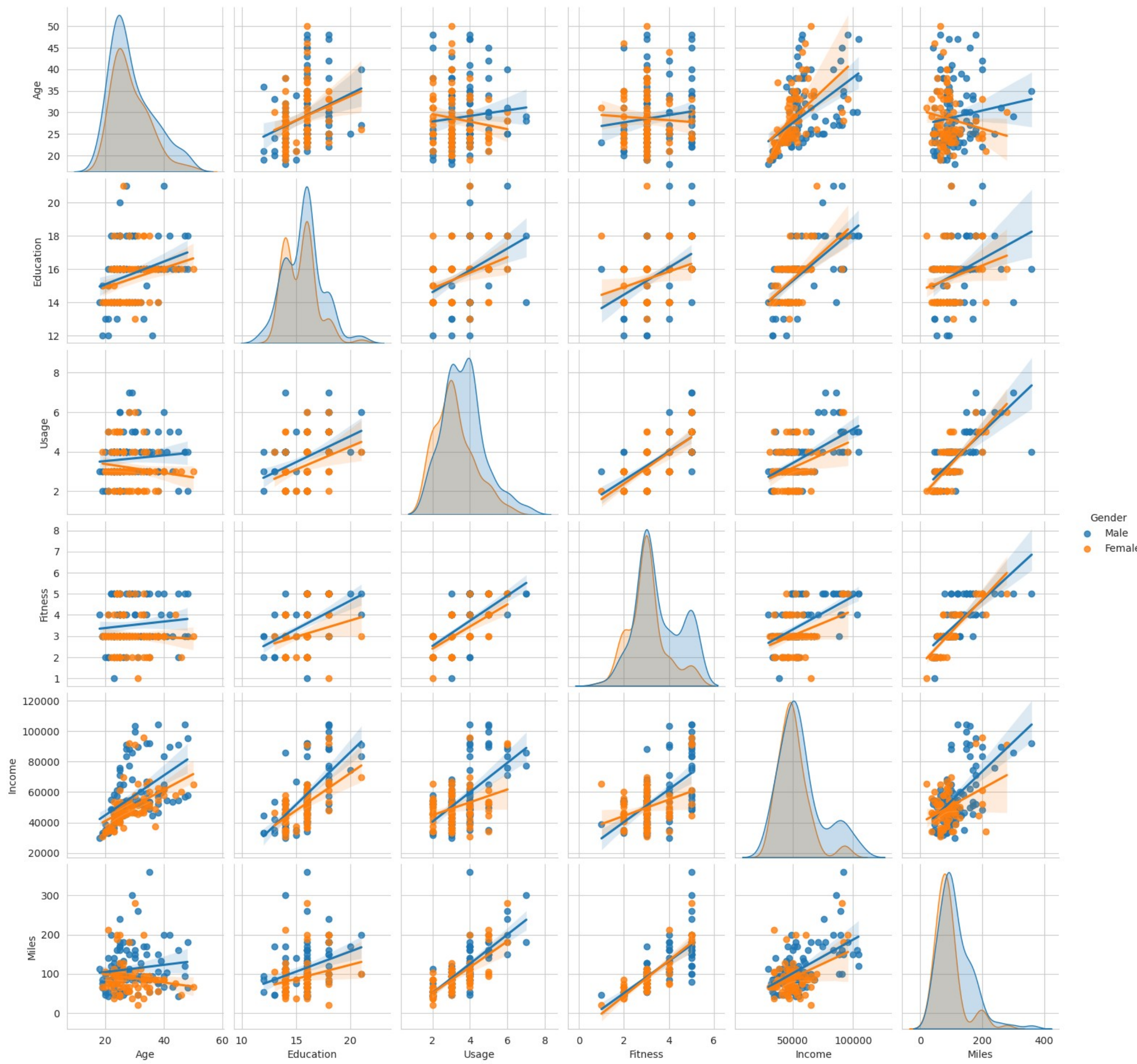
```
sns.pairplot(af,hue='Product',kind='reg')
plt.show()
```



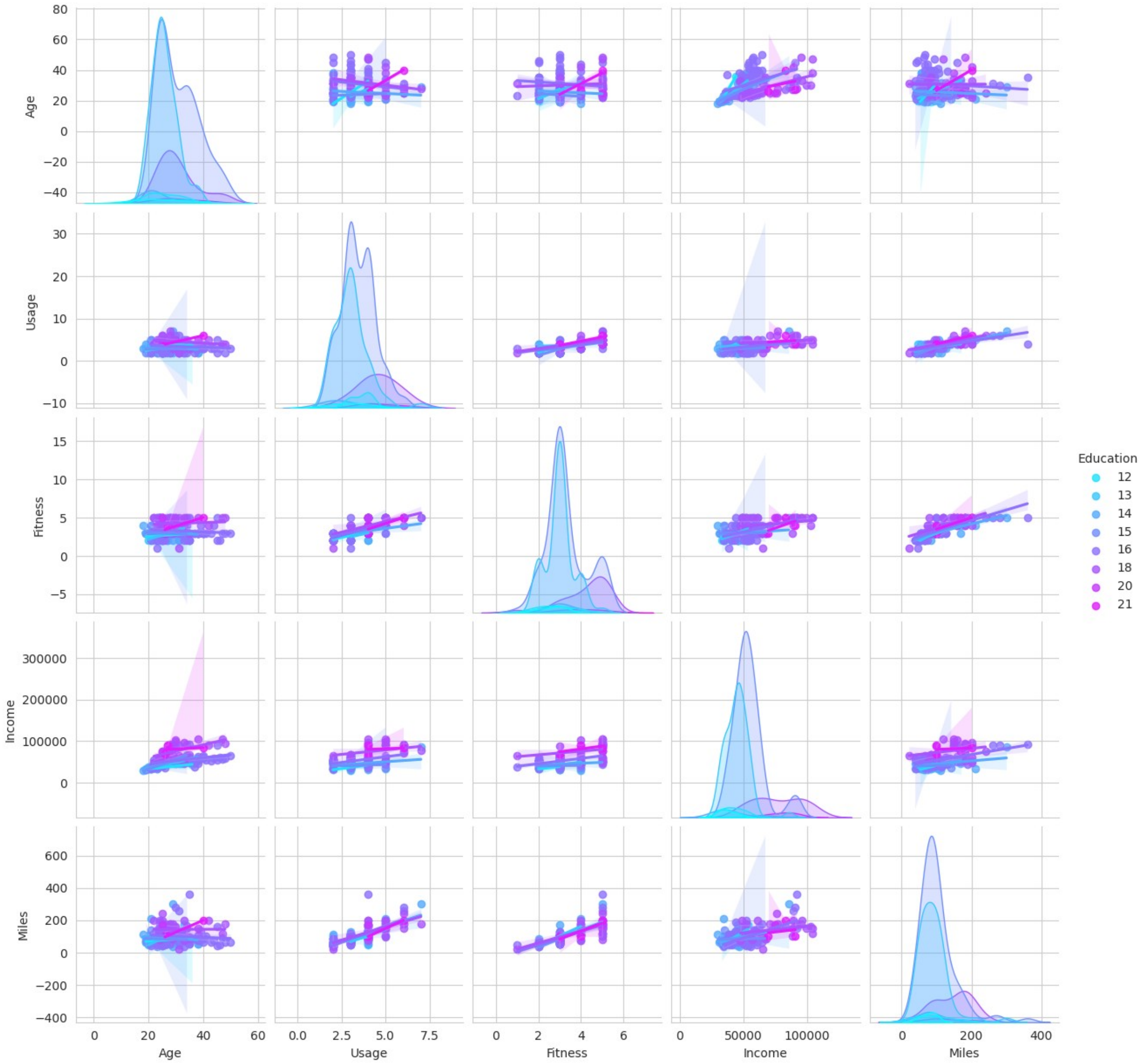

```
sns.pairplot(af,hue='MaritalStatus',kind='reg')
plt.show()
```




```
sns.pairplot(af,hue='Gender',kind='reg')
plt.show()
```




```
sns.pairplot(af,hue='Education',kind='reg', palette ="cool")
plt.show()
```



Insights

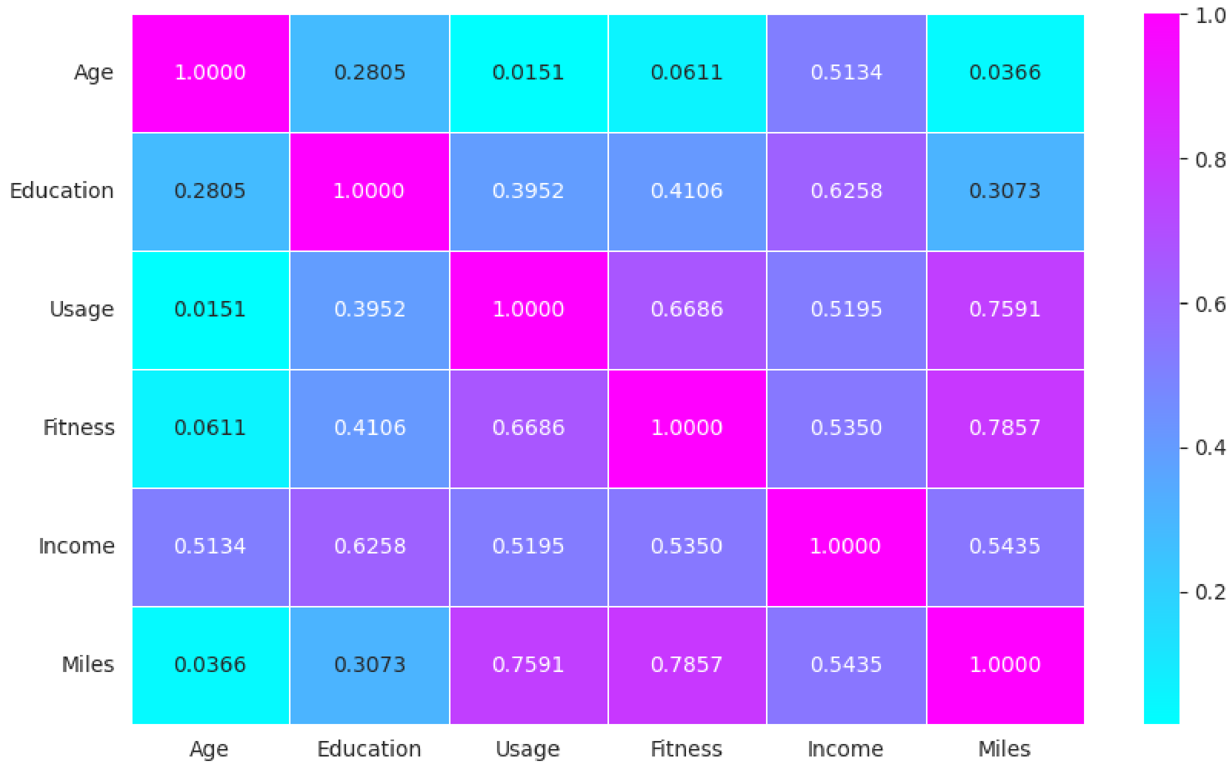
- It's clear that Age and Income go up together, showing a positive connection. The heatmap also indicates a strong link between them.
- Education and Income are closely connected, as expected. Education also shows a noticeable link with Fitness rating and Treadmill Usage.
- The more a treadmill is used, the more it relates to Fitness and Miles. Higher usage means higher fitness and mileage.

Heat Map

```
plt.figure(figsize=(10,6))
ax =
```

```
sns.heatmap(af.corr(),annot=True,fmt='.4f',linewidths=.5,cmap='cool')
plt.yticks(rotation=0)
plt.show()
```

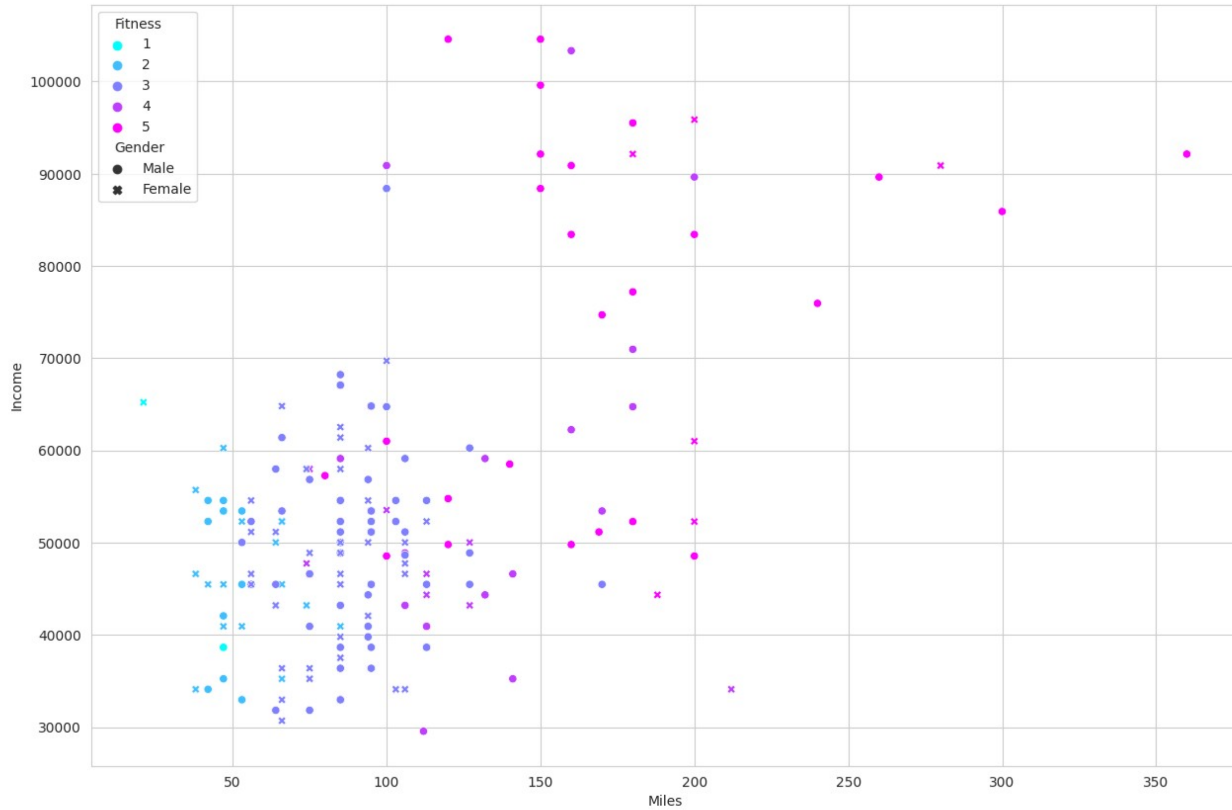
```
<ipython-input-218-9e90c4f465a9>:2: FutureWarning: The default value
of numeric_only in DataFrame.corr is deprecated. In a future version,
it will default to False. Select only valid columns or specify the
value of numeric_only to silence this warning.
  ax =
sns.heatmap(af.corr(),annot=True,fmt='.4f',linewidths=.5,cmap='cool')
```



Scatter Plot

```
plt.figure(figsize=(15,10))
sns.scatterplot(x='Miles',y='Income',data=af,hue='Fitness',style='Gend
er',palette='cool')
```

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



- The Scatter Plot gives a general idea about customers' income and their exercise habits. Most customers have a fitness level between 3 and 4. The plot suggests that people who run more miles tend to have a higher fitness level. There are only a few customers who both earn a lot and run more miles.

Correlations

- b/w Age and Miles is 0.03
- b/w Education and Income is 0.62
- b/w Usage and Fitness is 0.66
- b/w Fitness and Age is 0.06
- b/w Income and Usage is 0.51
- b/w Miles and Age is 0.03

Observations

- It's clear that Age and Income go up together, showing a positive connection. The heatmap also indicates a strong link between them.
- Education and Income are closely connected, as expected. Education also shows a noticeable link with Fitness rating and Treadmill Usage.
- The more a treadmill is used, the more it relates to Fitness and Miles. Higher usage means higher fitness and mileage.

- The Scatter Plot gives a general idea about customers' income and their exercise habits. Most customers have a fitness level between 3 and 4. The plot suggests that people who run more miles tend to have a higher fitness level. There are only a few customers who both earn a lot and run more miles.

Descriptive Statistics (Conditional and Marginal Probability)

```
#Probability of product purchase with respect to Gender

pd.crosstab([af.Product],af.Gender,margins=True)
```

Gender	Female	Male	All
Product			
KP281	40	40	80
KP481	29	31	60
KP781	7	33	40
All	76	104	180

```
np.round((pd.crosstab([af.Product],af.Gender,margins=True)/180)*100,2)
```

Gender	Female	Male	All
Product			
KP281	22.22	22.22	44.44
KP481	16.11	17.22	33.33
KP781	3.89	18.33	22.22
All	42.22	57.78	100.00

The Probability of a treadmill being purchased by a female is 42.2%.

The conditional probability of purchasing the treadmill model given that the customer is female is

- For model KP281 - 22.2%
- For model KP481 - 16.1%
- For model KP781 - 3.8%

The Probability of a treadmill being purchased by a male is 57.7%.

The conditional probability of purchasing the treadmill model given that the customer is male is -

- For model KP281 - 22.2%
- For model KP481 - 17.2%
- For model KP781 - 18.3%

```
#Probability of product purchase with respect to Age

af['age_group'].value_counts()
```

```
Young Adults          79
Adults                73
Middle Aged Adults    22
Elder                  6
Name: age_group, dtype: int64
```

```
pd.crosstab(index=af.Product,columns=af['age_group'],margins=True)
```

age_group	Young Adults	Adults	Middle Aged Adults	Elder	All
Product					
KP281	34	32	11	3	80
KP481	28	24	7	1	60
KP781	17	17	4	2	40
All	79	73	22	6	180

```
np.round((pd.crosstab(index=af.Product,columns=af['age_group'],margins
=True)/180)*100,2)
```

age_group	Young Adults	Adults	Middle Aged Adults	Elder	All
Product					
KP281	18.89	17.78	6.11	1.67	44.44
KP481	15.56	13.33	3.89	0.56	33.33
KP781	9.44	9.44	2.22	1.11	22.22
All	43.89	40.56	12.22	3.33	100.00

The Probability of a treadmill being purchased by a Young Adult(18-25) is 43.8%.

The conditional probability of purchasing the treadmill model given that the customer is Young Adult is

- For model KP281 - 18.8%
- For model KP481 - 15.5%
- For model KP781 - 9.4%

The Probability of a treadmill being purchased by a Adult(26-35) is 40.5%.

The conditional probability of purchasing the treadmill model given that the customer is Adult is -

- For model KP281 - 17.7%
- For model KP481 - 13.3%
- For model KP781 - 9.4%

The Probability of a treadmill being purchased by a Middle Aged(36-45) is 12.2%.

The conditional probability of purchasing the treadmill model given that the customer is Middle Aged Adult is -

- For model KP281 - 6.1%
- For model KP481 - 3.8%
- For model KP781 - 2.2%

The Probability of a treadmill being purchased by a Elder(Above 45) is only 3.3%.

The conditional probability of purchasing the treadmill model given that the customer is Elder is -

- For model KP281 - 1.6%
- For model KP481 - 0.5%
- For model KP781 - 1.1%

```
#Probability of product purchase with respect to Education

pd.crosstab(index=af.Product,columns=af['edu_group'],margins=True)
```

edu_group	Primary Education	Secondary Education	Higher Education	All
Product				
KP281	2	37	41	80
KP481	1	26	33	60
KP781	0	2	38	40
All	3	65	112	180

```
np.round((pd.crosstab(index=af.Product,columns=af['edu_group'],margins
=True)/180)*100,2)
```

edu_group	Primary Education	Secondary Education	Higher Education	All
Product				
KP281	1.11	20.56	22.78	44.44
KP481	0.56	14.44	18.33	33.33
KP781	0.00	1.11	21.11	22.22
All	1.67	36.11	62.22	100.00

The Probability of a treadmill being purchased by a customer with Higher Education(Above 15 Years) is 62.2%.

The conditional probability of purchasing the treadmill model given that the customer has Higher Education is-

- For KP281 - 22.7%
- For KP481 - 18.3%
- For KP781 - 21.1%

The Probability of a treadmill being purchased by a customer with Secondary Education(13-15 yrs) is 36.1%.

The conditional probability of purchasing the treadmill model given that the customer has Secondary Education is -

- For KP281 - 20.5%
- For KP481 - 14.4%
- For KP781 - 1.1%

The Probability of a treadmill being purchased by a customer with Primary Education(13-15 yrs) is 1.6%.

The conditional probability of purchasing the treadmill model given that the customer has Secondary Education is -

- For KP281 - 1.1%
- For KP481 - 0.5%

```
#Probability of product purchase with respect to Income

pd.crosstab(index=af.Product,columns=af['Income group'],margins=True)
```

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

```
np.round((pd.crosstab(index=af.Product,columns=af['Income group'],margins=True)/180)*100,2)
```

Income group	Low	Medium	High	All
Product				
KP281	26.67	17.78	0.00	44.44
KP481	16.67	16.67	0.00	33.33
KP781	2.78	7.78	11.67	22.22
All	46.11	42.22	11.67	100.00

The Probability of a treadmill being purchased by a customer with Low Income is 46.1%.

The conditional probability of purchasing the treadmill model given that the customer has Low Income is -

- For model KP281 - 26.6%
- For model KP481 - 16.6%
- For model KP781 - 2.7%

The Probability of a treadmill being purchased by a customer with Medium Income is 42.2%.

The conditional probability of purchasing the treadmill model given that the customer has Moderate Income is -

- For model KP281 - 17.7%
- For model KP481 - 16.6%
- For model KP781 - 7.7%

The Probability of a treadmill being purchased by a customer with High Income(60k - 80k) is 11.6%

The conditional probability of purchasing the treadmill model given that the customer has High Income is -

- For model KP281 - 0%
- For model KP481 - 0%
- For model KP781 - 11.6%

```
#Probability of product purchase with respect to Marital Status

pd.crosstab(index=af.Product,columns=af['MaritalStatus'],margins=True)
```

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

```
np.round((pd.crosstab(index=af.Product,columns=af['MaritalStatus'],margins=True)/180)*100,2)
```

MaritalStatus	Partnered	Single	All
Product			
KP281	26.67	17.78	44.44
KP481	20.00	13.33	33.33
KP781	12.78	9.44	22.22
All	59.44	40.56	100.00

The Probability of a treadmill being purchased by a Married Customer is 59.4%.

The conditional probability of purchasing the treadmill model given that the customer is Married is

- For model KP281 - 26.6%
- For model KP481 - 20%
- For model KP781 - 12.7%

The Probability of a treadmill being purchased by a Single Customer is 40.5%.

The conditional probability of purchasing the treadmill model given that the customer is Unmarried is -

- For model KP281 - 17.7%
- For model KP481 - 13.3%
- For model KP781 - 9.4%

```python
#Probability of product purchase with respect to Weekly Usage

pd.crosstab(index=af.Product,columns=af['Usage'],margins=True)
```

Usage	2	3	4	5	6	7	All
Product							
KP281	19	37	22	2	0	0	80
KP481	14	31	12	3	0	0	60
KP781	0	1	18	12	7	2	40
All	33	69	52	17	7	2	180

```python
np.round((pd.crosstab(index=af.Product,columns=af['Usage'],margins=True)/180)*100,2)
```

Usage	2	3	4	5	6	7	All
Product							
KP281	10.56	20.56	12.22	1.11	0.00	0.00	44.44
KP481	7.78	17.22	6.67	1.67	0.00	0.00	33.33
KP781	0.00	0.56	10.00	6.67	3.89	1.11	22.22
All	18.33	38.33	28.89	9.44	3.89	1.11	100.00

The Probability of a treadmill being purchased by a customer with Usage 3 per week is 38.3%.

The conditional probability of purchasing the treadmill model given that the customer has Usage 3 per week is -

- For model KP281 - 20.5%
- For model KP481 - 17.2%
- For model KP781 - 0.5%

The Probability of a treadmill being purchased by a customer with Usage 4 per week is 28.8%.

The conditional probability of purchasing the treadmill model given that the customer has Usage 4 per week is -

- For model KP281 - 12.2%
- For model KP481 - 6.6%
- For model KP781 - 10%

The Probability of a treadmill being purchased by a customer with Usage 2 per week is 18.3%

The conditional probability of purchasing the treadmill model given that the customer has Usage 2 per week is -

- For model KP281 - 10.5%
- For model KP481 - 7.7%
- For model KP781 - 0%

```python
#Probability of product purchase with respect to Fitness

pd.crosstab(index=af.Product,columns= af['Fitness'],margins=True)
```

Fitness	1	2	3	4	5	All
Product						
KP281	1	14	54	9	2	80
KP481	1	12	39	8	0	60
KP781	0	0	4	7	29	40
All	2	26	97	24	31	180

```
np.round((pd.crosstab(index=af.Product,columns=
af['Fitness'],margins=True)/180)*100,2)
```

Fitness	1	2	3	4	5	All
Product						
KP281	0.56	7.78	30.00	5.00	1.11	44.44
KP481	0.56	6.67	21.67	4.44	0.00	33.33
KP781	0.00	0.00	2.22	3.89	16.11	22.22
All	1.11	14.44	53.89	13.33	17.22	100.00

The Probability of a treadmill being purchased by a customer with Average(3) Fitness is 53.8%.

The conditional probability of purchasing the treadmill model given that the customer has Average Fitness is -

- For model KP281 - 30%
- For model KP481 - 21.6%
- For model KP781 - 2.2%

The Probability of a treadmill being purchased by a customer with low(2) Fitness is only 14.4%.

The Probability of a treadmill being purchased by a customer with High(4) Fitness is only 13.3%.

The Probability of a treadmill being purchased by a customer with Very High(5) Fitness is only 17.2%.

The Probability of a treadmill being purchased by a customer with very low(1) Fitness is only 1.1%.

```
#Probability of product purchase with respect to Miles

pd.crosstab(index=af.Product,columns= af['Mile_group'],margins=True)
```

Mile_group	Beginner	Intermediate	Professional	All
Product				
KP281	51	22	3	76
KP481	27	25	2	54
KP781	1	11	17	29
All	79	58	22	159

```
np.round((pd.crosstab(index=af.Product,columns=
af['Mile_group'],margins=True)/180)*100,2)
```

```
Mile_group  Beginner  Intermediate  Professional    All
Product
KP281          28.33         12.22          1.67  42.22
KP481          15.00         13.89          1.11  30.00
KP781           0.56          6.11          9.44  16.11
All            43.89         32.22         12.22  88.33
```

The Probability of a treadmill being purchased by a customer in Beginner level is 43.8%.

The conditional probability of purchasing the treadmill model given that the customer is in Beginner level is -

- For model KP281 - 28.3%
- For model KP481 - 15%
- For model KP781 - 0.5%

The Probability of a treadmill being purchased by a customer in Beginner level is 32.2%.

The conditional probability of purchasing the treadmill model given that the customer is in Intermediate level is -

- For model KP281 - 12.2%
- For model KP481 - 13.8%
- For model KP781 - 6.1%

The Probability of a treadmill being purchased by a customer in Professional level is 12.2%.

The conditional probability of purchasing the treadmill model given that the customer is in Professional level is -

- For model KP281 - 1.6%
- For model KP481 - 1.1%
- For model KP781 - 9.4%

Customer Profiling

The chances of buying different treadmills are as follows:

- KP281 (44%)
- KP481 (33%)
- KP781 (22%)

For the KP281:

- Customers are usually aged 18 to 35, with a few aged 35 to 50.
- They have at least 13 years of education.
- Annual income is less than USD 60,000.
- Weekly usage is 2 to 4 times.
- Fitness level ranges from 2 to 4.
- Weekly running mileage is 50 to 100 miles.

For the KP481:

- Customers are mainly aged 18 to 35, with some aged 35 to 50.
- They have at least 13 years of education.
- Annual income is between USD 40,000 to USD 80,000.
- Weekly usage is 2 to 4 times.
- Fitness level ranges from 2 to 4.
- Weekly running mileage is 50 to 200 miles.

For the KP781:

- Customers are male.
- Age ranges from 18 to 35.
- They have at least 15 years of education.
- Annual income is USD 80,000 and above.
- Weekly usage is 4 to 7 times.
- Fitness level ranges from 3 to 5.
- Weekly running mileage is 100 miles and above.

Recommendations

For KP781:

- The KP781 model is selling a lot more to men (82%) than women (18%). To increase sales to women, we suggest. Females who prefer exercising equipment are very low here. To increase sales to women, we should run a marketing campaign on special deals and trials made just for them.
- As KP781 provides more features and functionalities,This model should be marketed by influencers and other international athletes portraying that KP781 is specially made for professionals and athletes.
- The Age group of above 40 years should be targeted to recommend Product KP781.

For KP281 and KP481:

- It's a good idea to set reasonable prices for both the KP281 and KP481 models.
- By introducing flexible payment plans and options, customers can pay in installments over a few months, making it easier for customers with lower budgets.

User-Friendly App Integration: Make an easy-to-use app that connects with the product through mobile. This app can track how much users run each week, give them feedback on their progress, and suggest workouts based on their fitness level and goals. This will make using the treadmill more enjoyable and keep users engaged.

Providing customer support through call or email and recommend users to upgrade from lower versions to next level versions after consistent usages.