Aerofit--Descriptive- Statistics-Probability

Details:

Name : Tanmoy Basuli

Email-Id : basuli575tanmoy@gmail.com

Introduction

In this study-case, I'll give an Exploratory Data Analysis of the Aerofit dataset given by scaler. I will explore the data and hopefully bring some insights.

- We will try to find some insights on below,
  - Univariate Analysis
  - Bivariate Analysis
- We will see Conditional, Joint and Marginal probability among products.
- Will try to create Customer Profiling - Categorization of users.
- Will see correlation among different factors using heat maps or pair plots.

For visualizations I used, seaborn, pyplot, plotly. Some of the visuals interactive, and some of it static. But there's a lot improve. Feedback is always welcome.

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.

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.

Dataset: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749

The Outline of this notebook is as follows.

1. Basic Data Exploration
   - Feature Exploration
   - Summary Statistics
2. Data Cleaning
   - Null Value Analysis
   - Checking Duplicate Values

3. Exploratory data analysis (What is the Story Of Data)

Importing Libraries and Loading the Dataset

```
# Import Relevant Packages
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
# Load the data set

df = pd.read_csv('aerofit_treadmill.csv')
```

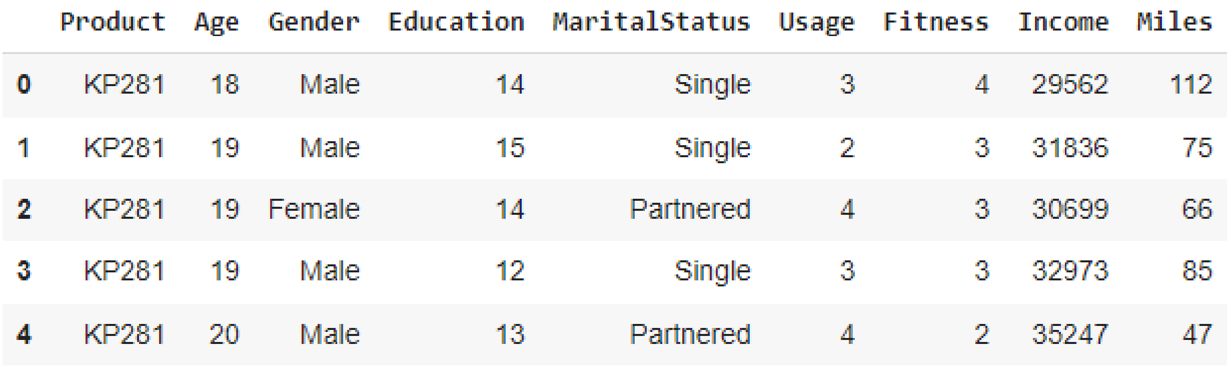
Basic Data Exploration

1. Feature Exploration
2. Summary Statistics

1. Feature Exploration

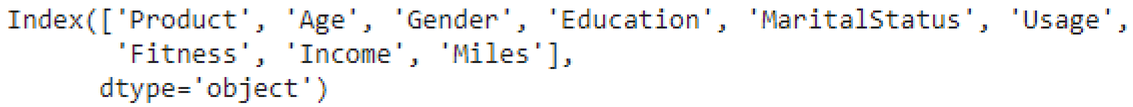
First, let us look at a quick peek of what the first five rows in the data has in store for us and what features we have.

```
# First five rows of the dataset
df.head()
```



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

```
#Look what all are the columns I have
df.columns
```



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

Features & Descriptions:

In this dataset we have,

The company collected the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months. The dataset has the following features:

Product Purchased: KP281, KP481, or KP781

Age: In years

Gender: Male/Female

Education: In years

MaritalStatus: Single or partnered.

Usage: The average number of times the customer plans to use the treadmill each week.

Income: Annual income (in $)

Fitness: Self-rated fitness on a 1-to-5 scales, 1 is the poor shape and 5 is the excellent shape.

Miles: The average number of miles the customer expects to walk/run each week

Next, let us look at how large the data is:

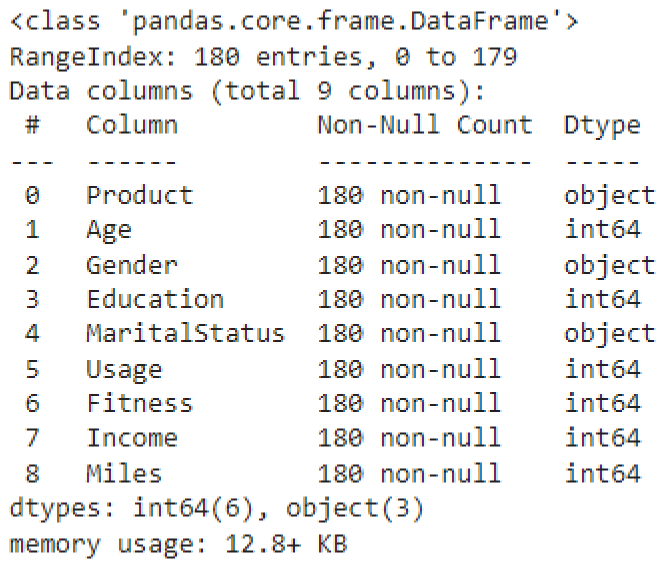
```
#Size of the data
df.shape
```

```
(180, 9)
```

We have 180 Entity from 10 Features.

Now, Let's look What types of data we have:

```
# Type of all Data
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
```

Observation (Data Type)

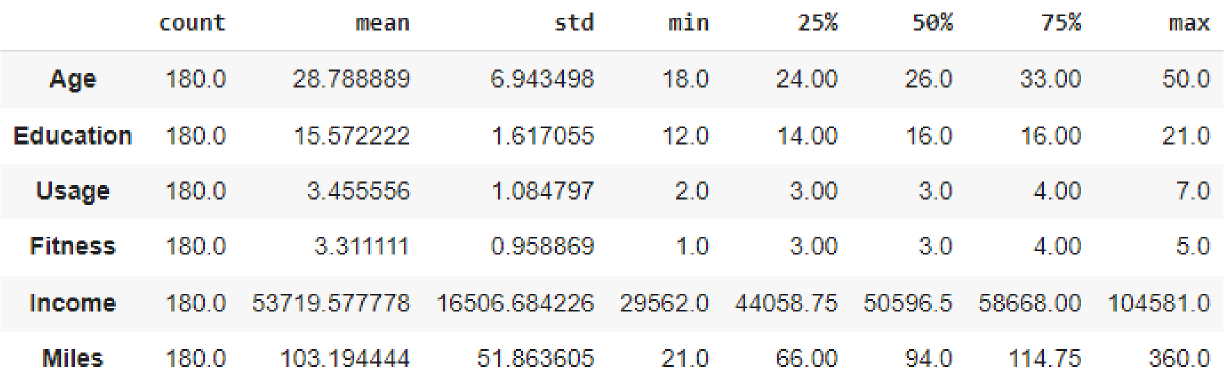
1. Data type of Product is object (string).
2. Data type of Age is int64.
3. Data type of Gender is object (string).
4. Data type of Education is int64.
5. Data type of MaritalStatus is object (string).
6. Data type of Usage is int64.
7. Data type of Fitness is int64.
8. Data type of Income is int64.
9. Data type of Miles is int64.

We can easily find out from the above showcase, we have:

- 3 Categorical Feature
- 6 Numeric Feature

2. Summary Statistics(Non-Graphical Analysis: Value counts and unique attributes)
Here we can see basic statistics in the data.

```
# Summary statistics for numerical features
numerical_features = df.select_dtypes(include='number')
# We have only 'release_year' as a numeric feature
numerical_features.describe().T
```



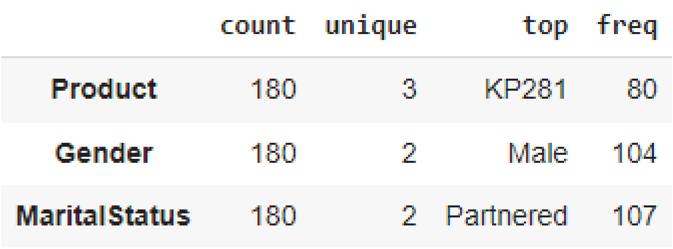
	count	mean	std	min	25%	50%	75%	max
Age	180.0	28.788889	6.943498	18.0	24.00	26.0	33.00	50.0
Education	180.0	15.572222	1.617055	12.0	14.00	16.0	16.00	21.0
Usage	180.0	3.455556	1.084797	2.0	3.00	3.0	4.00	7.0
Fitness	180.0	3.311111	0.958869	1.0	3.00	3.0	4.00	5.0
Income	180.0	53719.577778	16506.684226	29562.0	44058.75	50596.5	58668.00	104581.0
Miles	180.0	103.194444	51.863605	21.0	66.00	94.0	114.75	360.0

Observations from Descriptive Statistics (Numerical)

1. Age: Median age of the customer(s) is 26 years, having maximum age of 50 years and minimum age of 18 years.
2. Education (Years): Median education years of the customer(s) is 16 years, with maximum education years is 21, and minimum years is 12.
3. Usage (Per week) : Median usage of treadmill is 3 times per week, with maximum 7 times per week and minimum 2 times per week.
4. Fitness (1-5) : Median fitness rating of customer(s) is 3 (moderately fit) and mean fitness roughly lies around the median.
5. Income ($): Median income of customer(s) is 50.5K annually. Maximum income is 104K annually, and minimum income is 29.5K.
6. Miles: Median distance travelled (walk/run) by customer(s) is 94. Maximum distance travelled is 114.75 and minimum is 21.

Now we will look into categorical data in our dataset.

```
# Summary statistics for categorical features
categorical_features = df.select_dtypes(include='object')
categorical_features.describe().T
```



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

We can see count along with unique value, top of those categories and frequency of the objective data types.

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

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

```
df['Education'].unique()
```

```
array([14, 15, 12, 13, 16, 18, 20, 21])
```

```
df['Income'].unique()
```

```
array([ 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])
```

```
df['Miles'].unique()
```

```
array([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])
```

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

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

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

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

```
df['Product'].value_counts(normalize=True)
```

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

```
df['Gender'].value_counts(normalize=True)
```

```
Male      0.577778
Female    0.422222
Name: Gender, dtype: float64
```

```
df['MaritalStatus'].value_counts(normalize=True)
```

```
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
```

```
df['Usage'].value_counts(normalize=True)
```

```
3    0.383333
4    0.288889
2    0.183333
5    0.094444
6    0.038889
7    0.011111
Name: Usage, dtype: float64
```

```
df['Fitness'].value_counts(normalize=True)
```

```
3    0.538889
5    0.172222
2    0.144444
4    0.133333
1    0.011111
Name: Fitness, dtype: float64
```

General Observations

1. Product: Most commonly product purchased is KP281, followed by KP481, and KP781 respectively.
2. Gender: Male is the most common gender who purchased more of the aerofit products.
3. MartialStatus: Couples purchased more products compare to Single people. Maybe Couple Goals!
4. Usage: Fair amount (38.3%) of people have reported usage of treadmills 3 times per week, followed by 4 times per week, and 2 times per week respectively.
5. Fitness: More than 50% customers have given self-rating of 3, followed by 5 and 2.

Now will check any None/Null values is there or not:

```
# Checking null value if we have
df.isnull().values.any()
```

```
False
```

```
# Controlling null values again
df.isnull().sum()
```

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

We can easily found there is no null values is present over the whole dataset.

Now will check for duplicated value present or not:

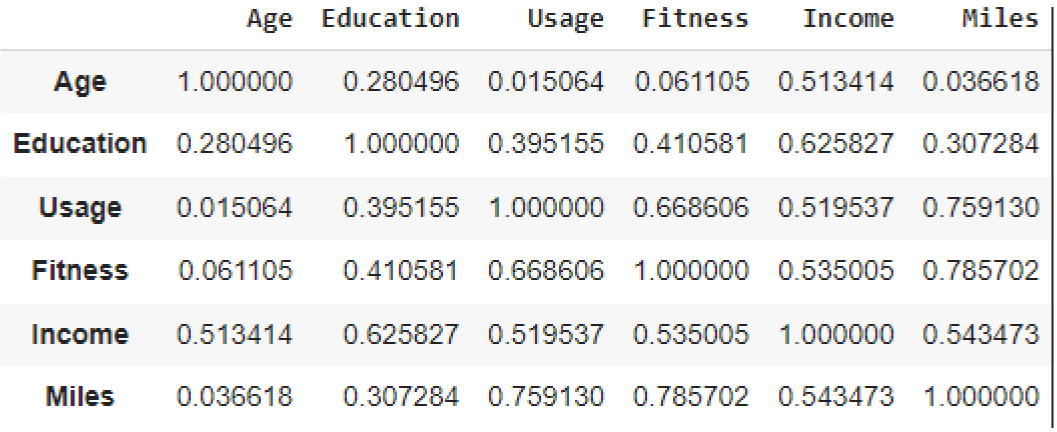
```
# Check Duplicate value
df.duplicated().sum()
```

```
0
```

Visual Analysis

Correlation

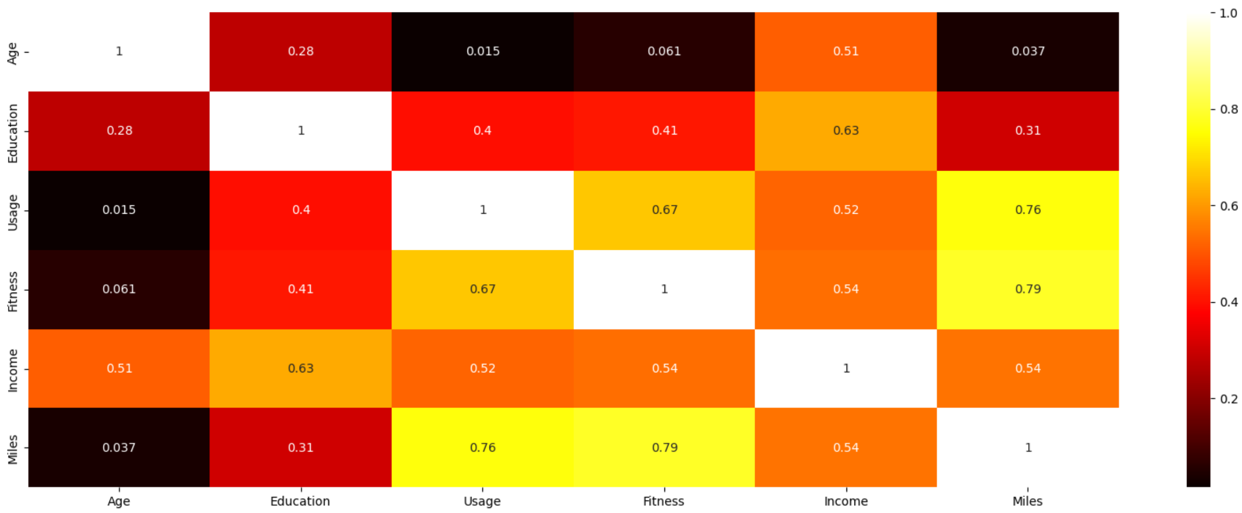
```
newdf = df.corr()
newdf
```



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

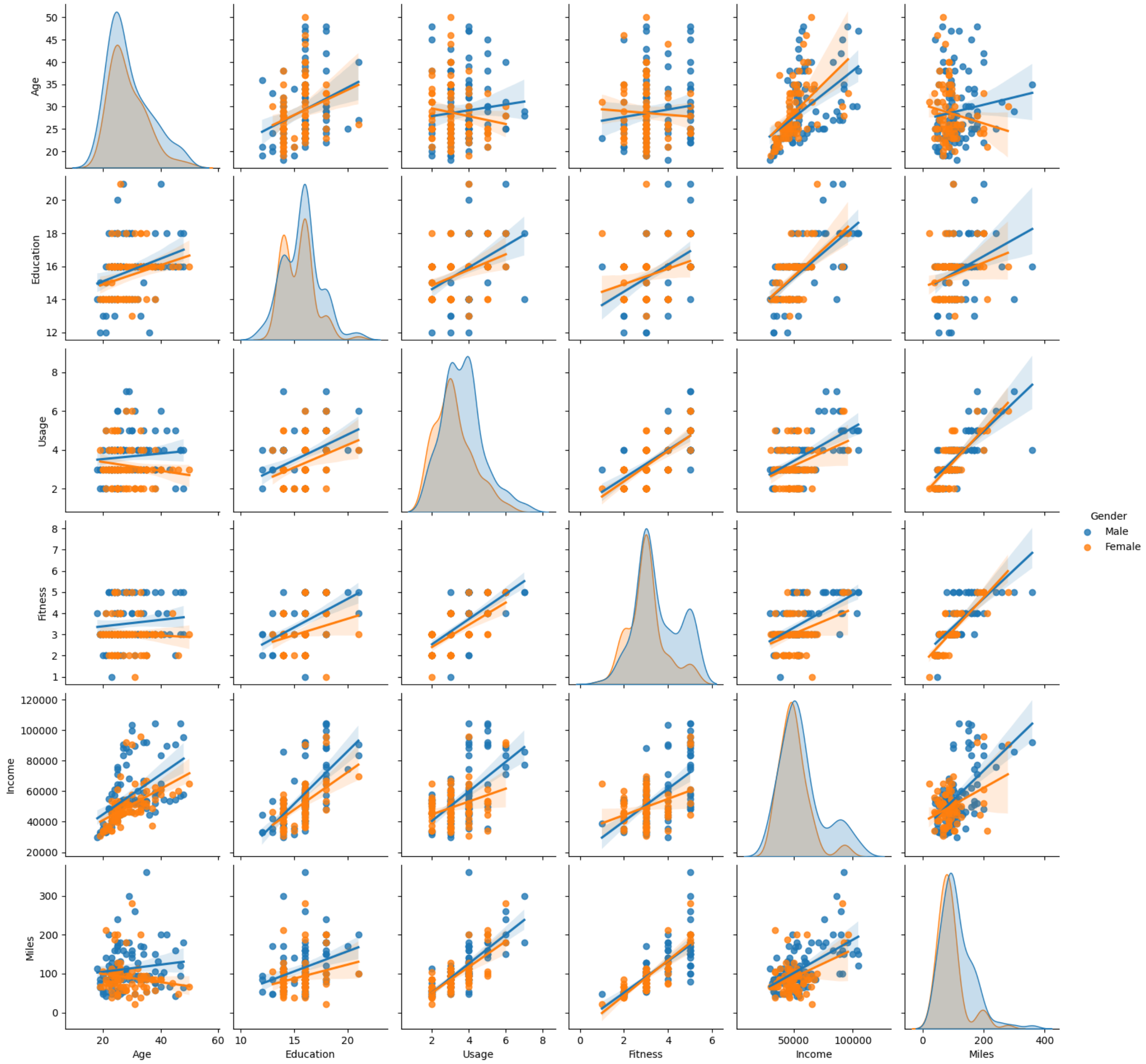
Now let’s try to plot through heatmap:

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,7
sns.heatmap(newdf, annot=True, cmap='hot')
plt.show()
```



Now will check through pairplot:

```
# pairplot
rcParams['figure.figsize'] = 20, 7
sns.pairplot(df,  hue='Gender', kind='reg')
plt.show()
```



Observations

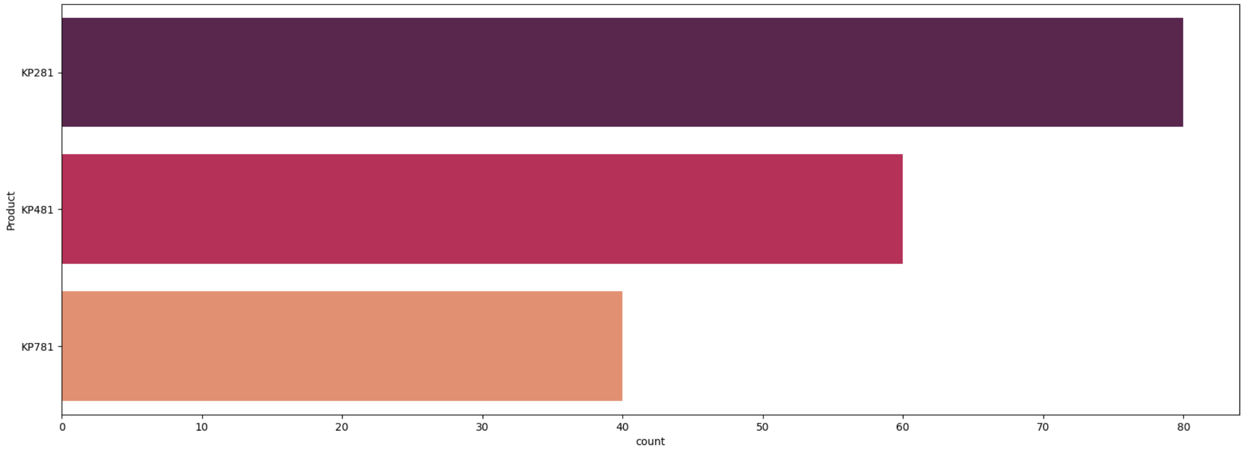
Here Pearson co-efficient is used to evalute the correlation between numerical data points. Pearson evaluates the linear relationship between data points.

Noting down the observations which are higher than 0.5.

1. Correlation between Age & Income is 0.51
2. Correlation between Education & Income is 0.63.
3. Correlation between Usage & Fitness is 0.67.
4. Correlation between Usage & Income is 0.52.
5. Correlation between Usage & Miles is 0.76.
6. Correlation between Fitness & Income is 0.54.
7. Correlation between Fitness & Miles is 0.79.
8. Correlation between Income & Miles is 0.54.

Univariate Analysis

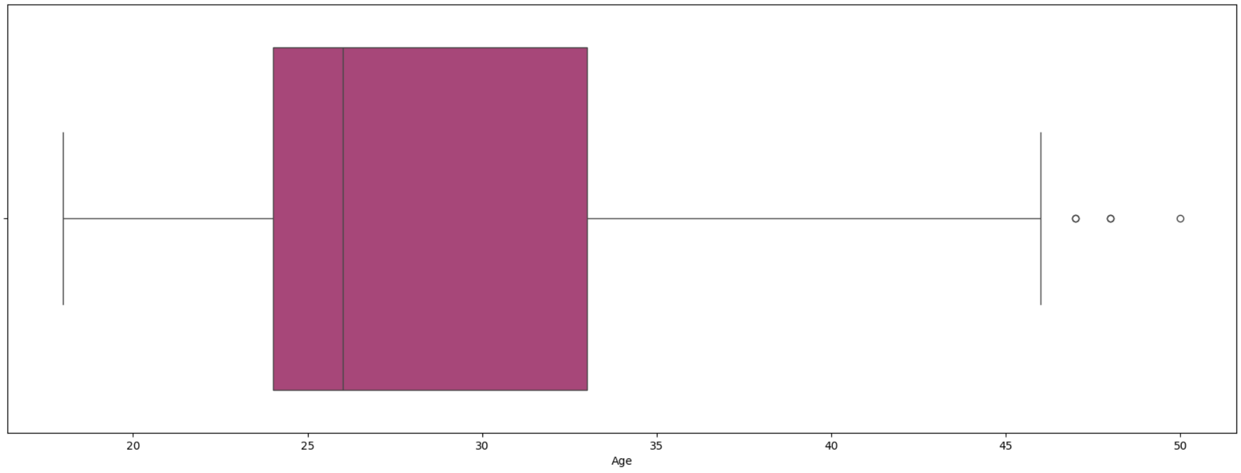
```
# Base on Product
sns.countplot(data = df, y = 'Product', palette='rocket')
plt.show()
```



Observations

1. KP281 is the most purchased product having the count of 80.
2. KP481 is the second most purchased product having the count of 60.
3. Lastly KP781 is the purchased product having the count of 40.

```
# Base on Age
sns.boxplot(data=df, x='Age', palette='magma')
plt.show()
```



```
Q3, Q1 = np.percentile(df['Age'], [75,25])
Q3, Q1
```

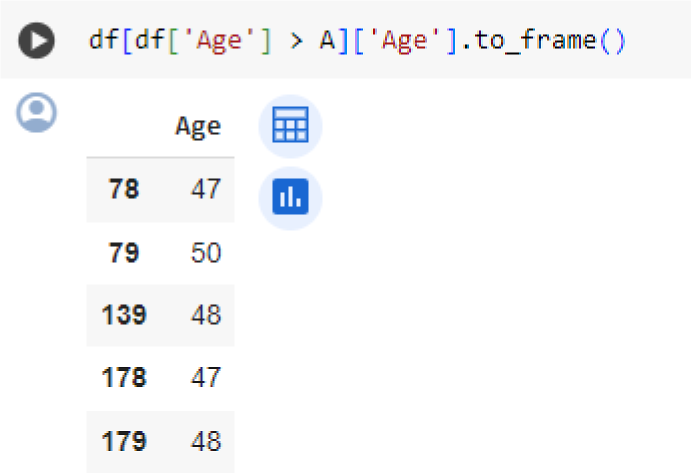
(33.0, 24.0)

```
IQR = Q3 - Q1
print("Inter Quartile Range (IQR) of Age is", IQR)
```

```
Inter Quartile Range (IQR) of Age is 9.0
```

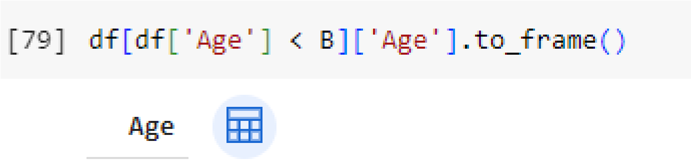
```
A = (Q3 + (1.5 * IQR))
B = (Q1 - (1.5 * IQR))
A,B
```

(46.5, 10.5)



```
df[df['Age'] > A]['Age'].to_frame()
```

	Age
78	47
79	50
139	48
178	47
179	48



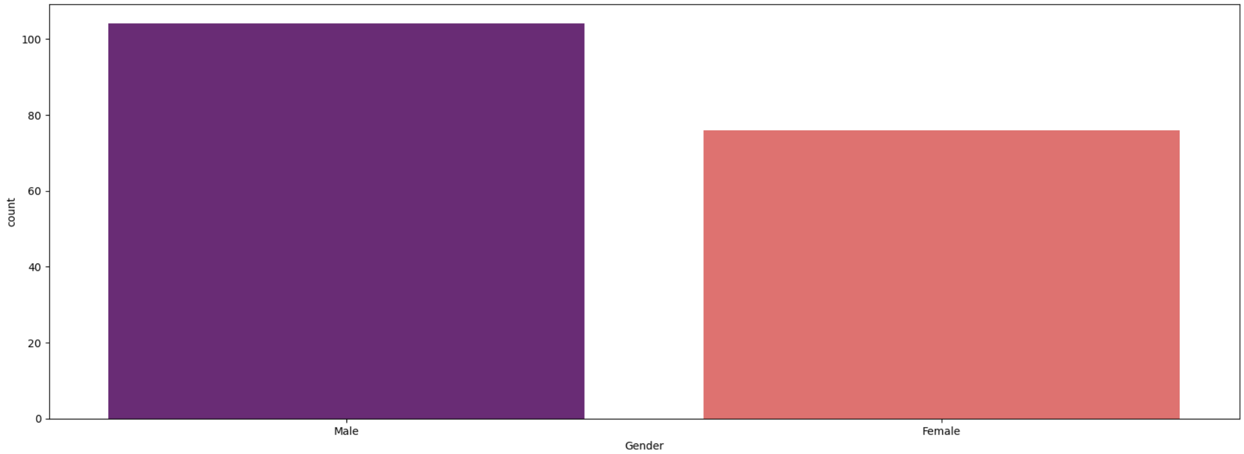
```
[79] df[df['Age'] < B]['Age'].to_frame()
```

	Age

Observations

1. Most common age range is roughly between 23 - 33.
2. Difference between 25th and 75th percentile is 9 years.
3. There are few data points whose age is more than 46 years (Outlier), i.e. 47, 48 and 50.

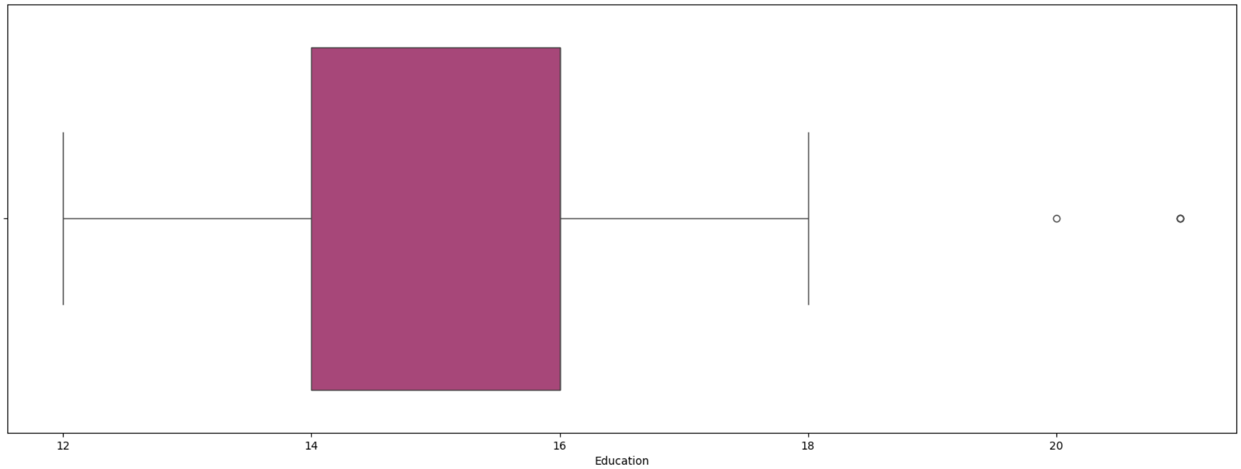
```
# Base on Gender
sns.countplot(data=df, x="Gender", palette='magma')
plt.show()
```

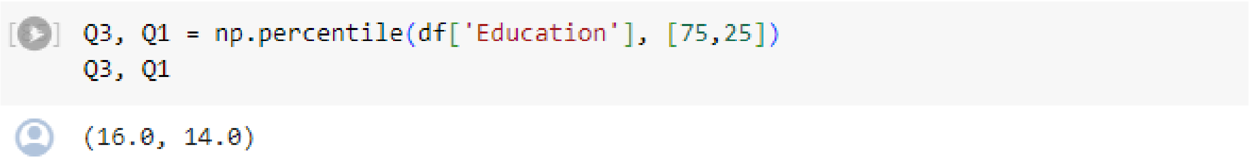


Observations

1. Male is the most frequent buyer of the treadmills with count more than 100.
2. Female is the second most frequent buyer with count of roughly 75.

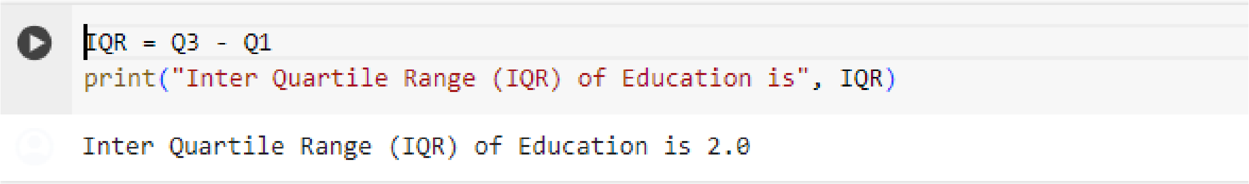
```
# Base on Education
sns.boxplot(data=df, x='Education', palette='magma')
plt.show()
```





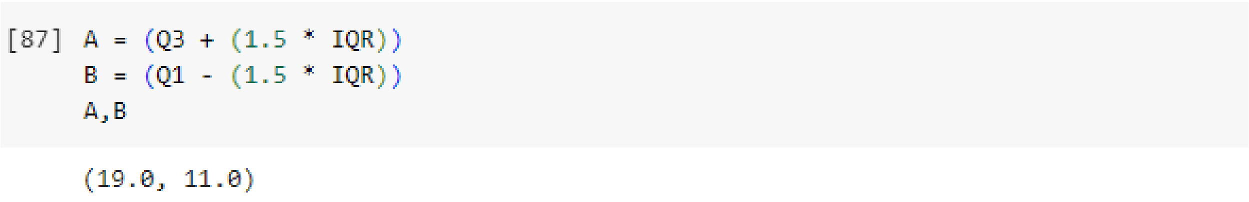
```
[ ] Q3, Q1 = np.percentile(df['Education'], [75,25])
    Q3, Q1
```

```
(16.0, 14.0)
```



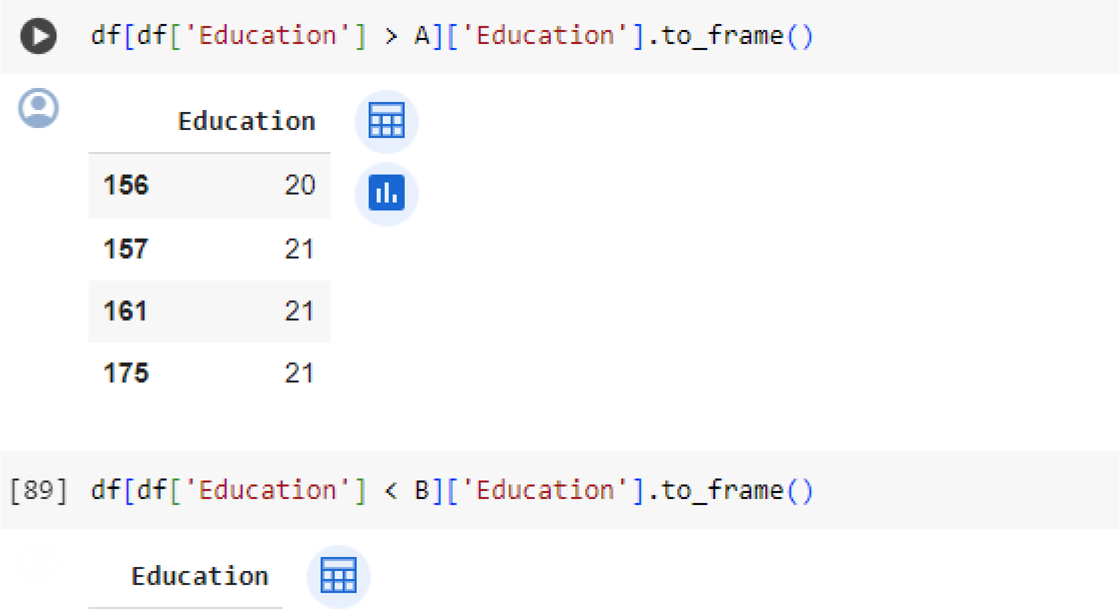
```
IQR = Q3 - Q1
print("Inter Quartile Range (IQR) of Education is", IQR)
```

```
Inter Quartile Range (IQR) of Education is 2.0
```



```
[87] A = (Q3 + (1.5 * IQR))
     B = (Q1 - (1.5 * IQR))
     A,B
```

```
(19.0, 11.0)
```



```
df[df['Education'] > A]['Education'].to_frame()
```

	Education
156	20
157	21
161	21
175	21

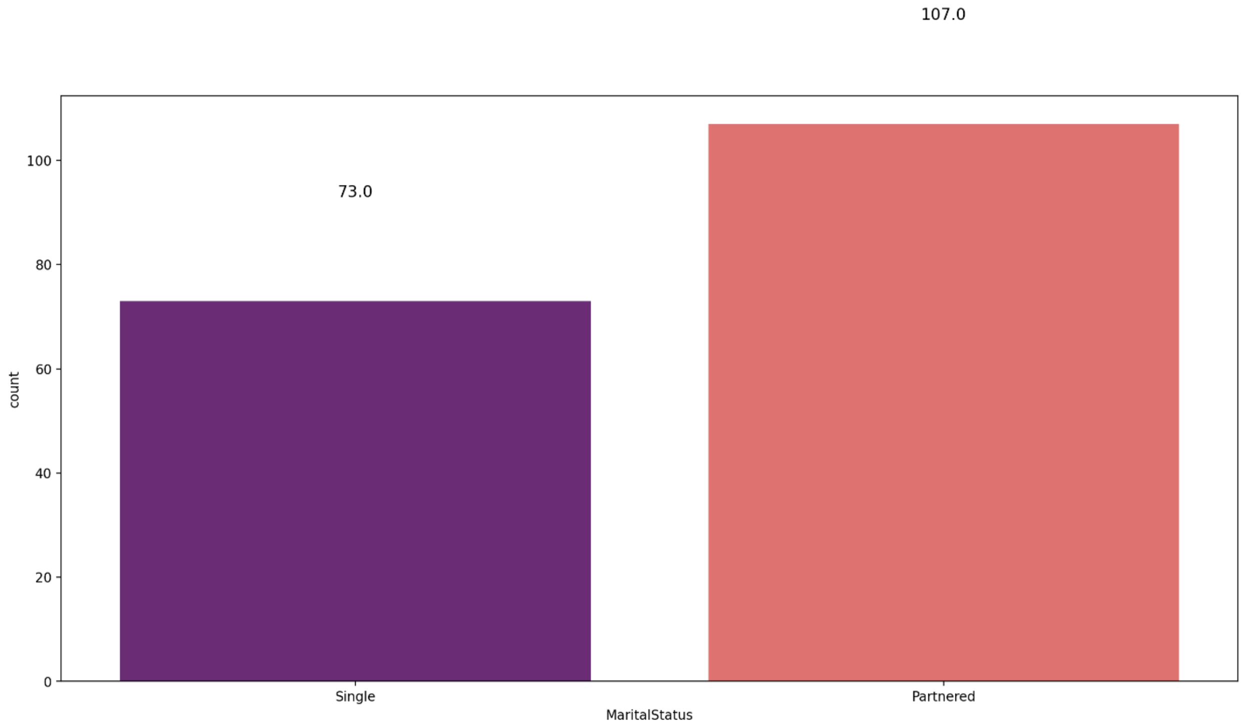
```
[89] df[df['Education'] < B]['Education'].to_frame()
```

	Education

Observations

1. Majority of the people have education between 14-16 years.
2. Difference between 25th and 75th percentile is 2 years.
3. There are couple of data who have education more than 20 years (Outlier). i.e. 20 and 21.

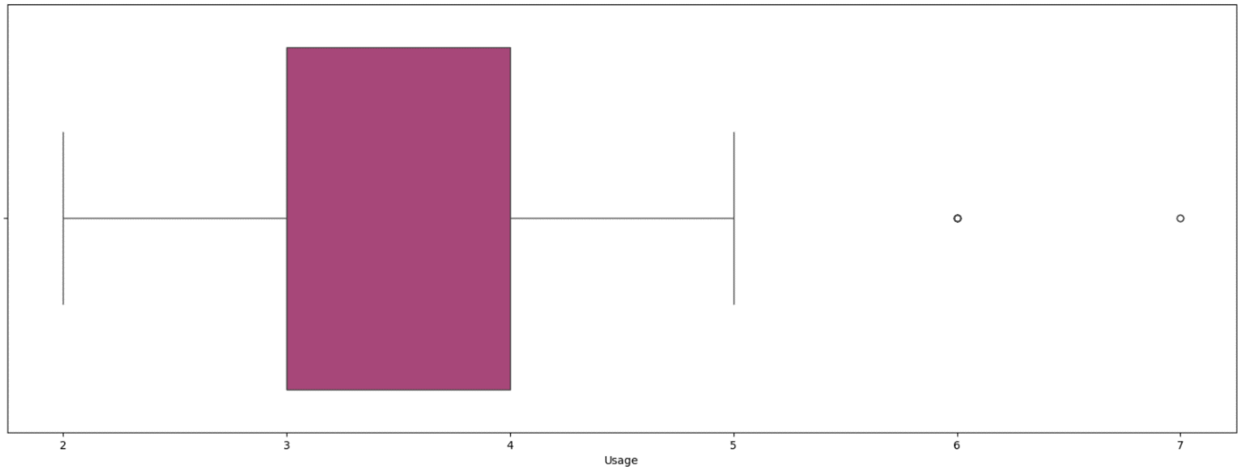
```
# Base on MaritalStatus
plt.figure(figsize = (16,8), dpi = 200)
plot = sns.countplot(data=df, x="MaritalStatus", palette='magma')

for i in plot.patches:
  height = i.get_height()
  plot.text(i.get_x() + i.get_width()/2, height + 20, height, ha =
'center', fontsize=12)
plt.show()
```

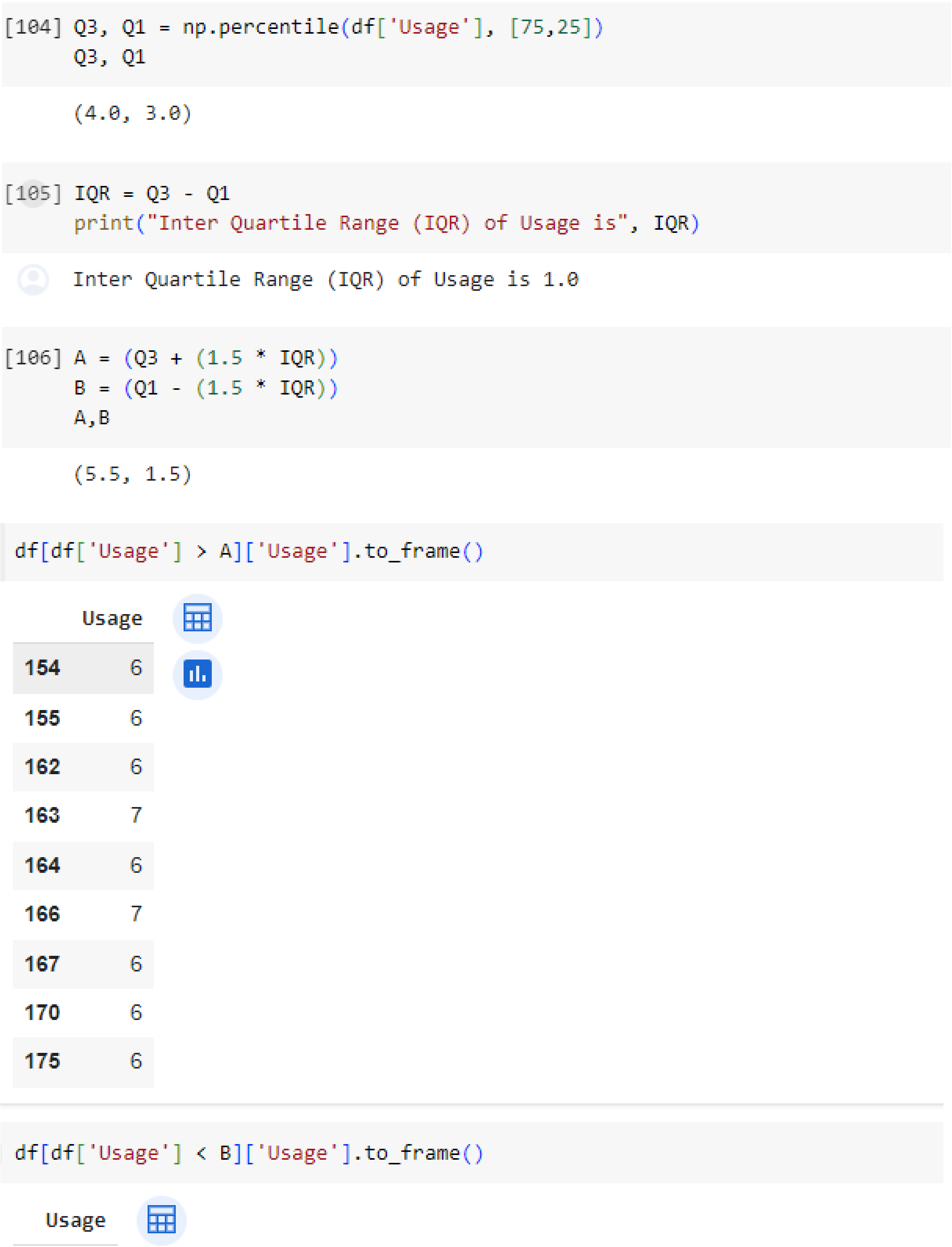


Observations

1. Couples are the most frequent buyers of the treadmill with count looks like 107.
2. Singles are the 2nd most frequent buyers of the treadmill with count of 73.

```
# Base on Usage
sns.boxplot(data=df, x='Usage', palette='magma')
plt.show()
```





```
[104] Q3, Q1 = np.percentile(df['Usage'], [75,25])
      Q3, Q1
```

```
(4.0, 3.0)
```

```
[105] IQR = Q3 - Q1
      print("Inter Quartile Range (IQR) of Usage is", IQR)
```

```
Inter Quartile Range (IQR) of Usage is 1.0
```

```
[106] A = (Q3 + (1.5 * IQR))
      B = (Q1 - (1.5 * IQR))
      A,B
```

```
(5.5, 1.5)
```

```
df[df['Usage'] > A]['Usage'].to_frame()
```

	Usage
154	6
155	6
162	6
163	7
164	6
166	7
167	6
170	6
175	6

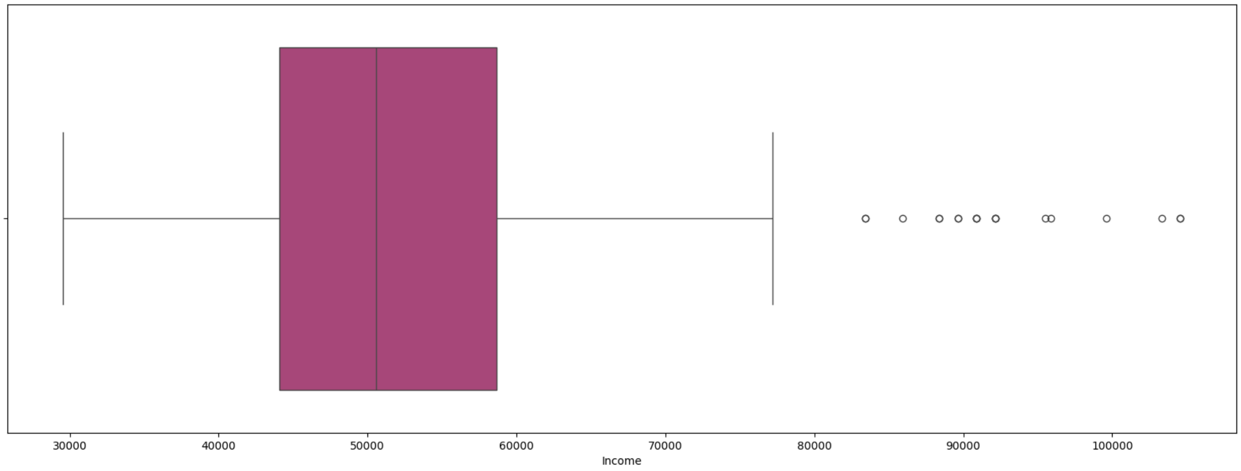
```
df[df['Usage'] < B]['Usage'].to_frame()
```

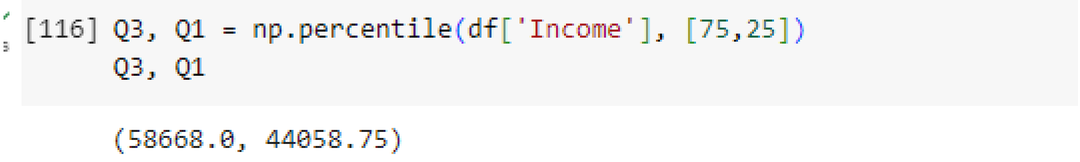
Usage

Observations

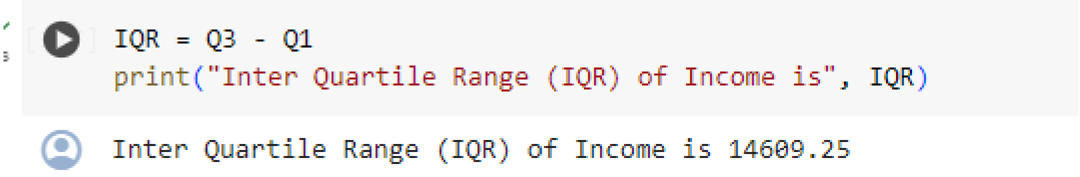
1. Most of the customers use treadmill 3 - 4 times per week.
2. Very few people walk/run on treadmill 6 - 7 times per week (Outlier).
3. Difference between 25th and 75th percentile is 1.0.
4. Overall, it looks like very few people are regular in their workouts, while others are working-out casually.

```
# Base on Income
sns.boxplot(data=df, x='Income', palette='magma')
plt.show()
```

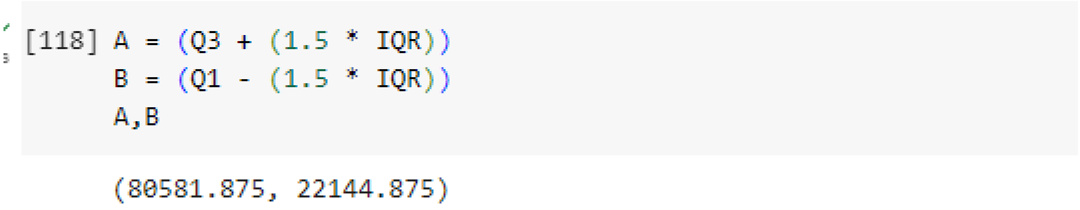




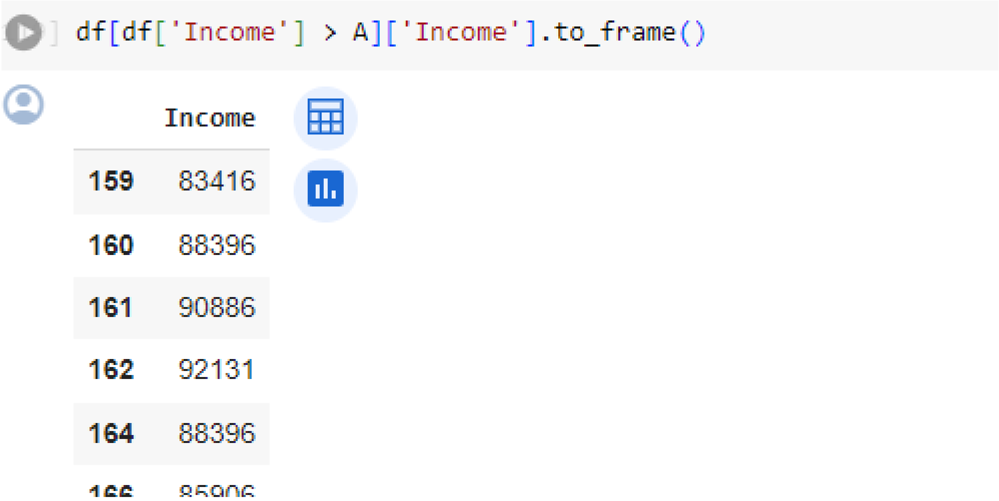
```
[116] Q3, Q1 = np.percentile(df['Income'], [75,25])
      Q3, Q1

      (58668.0, 44058.75)
```



```
IQR = Q3 - Q1
print("Inter Quartile Range (IQR) of Income is", IQR)

Inter Quartile Range (IQR) of Income is 14609.25
```



```
[118] A = (Q3 + (1.5 * IQR))
      B = (Q1 - (1.5 * IQR))
      A,B

      (80581.875, 22144.875)
```



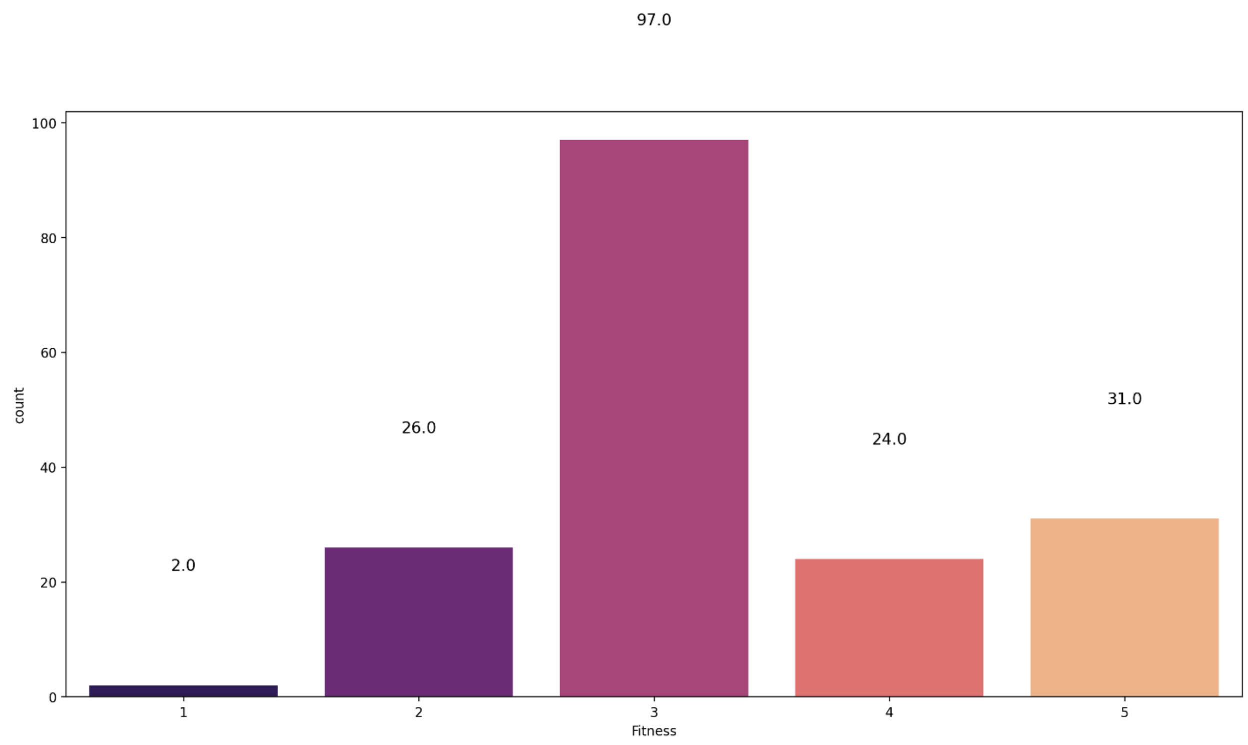
```
df[df['Income'] > A]['Income'].to_frame()
```

```
     Income
159   83416
160   88396
161   90886
162   92131
164   88396
166   85906
```

Observations

1. Most of the customers have income between 45K$ - 60K$.
2. Very few people have income more than roughly 85K$ - 104K$.
3. Difference between 25th and 75th percentile is 14609$.

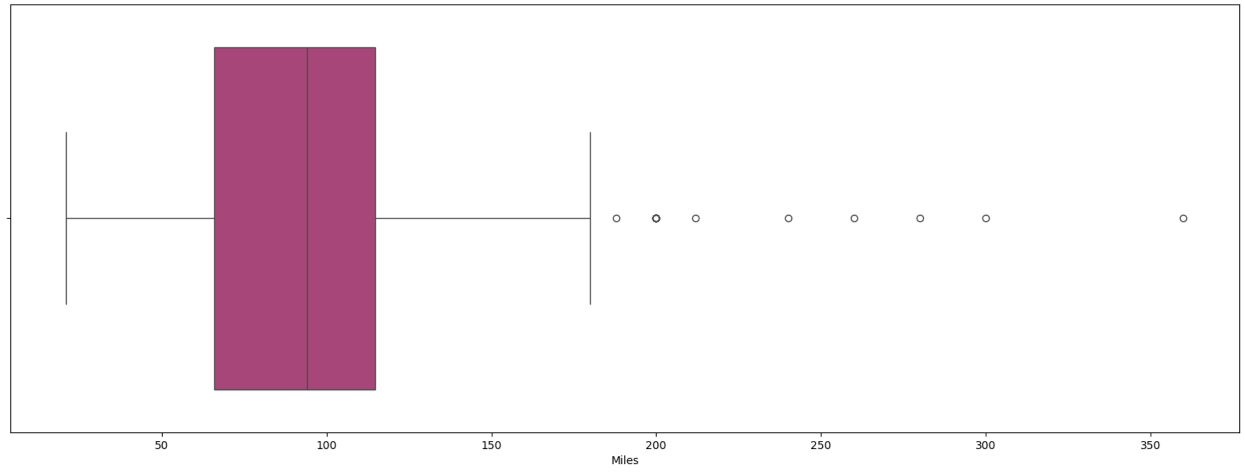
```
# Base on Fitness
plt.figure(figsize = (16,8), dpi = 200)
plot = sns.countplot(data=df, x="Fitness", palette='magma')

for i in plot.patches:
  height = i.get_height()
  plot.text(i.get_x() + i.get_width()/2, height + 20, height, ha =
'center', fontsize=12)
plt.show()
```

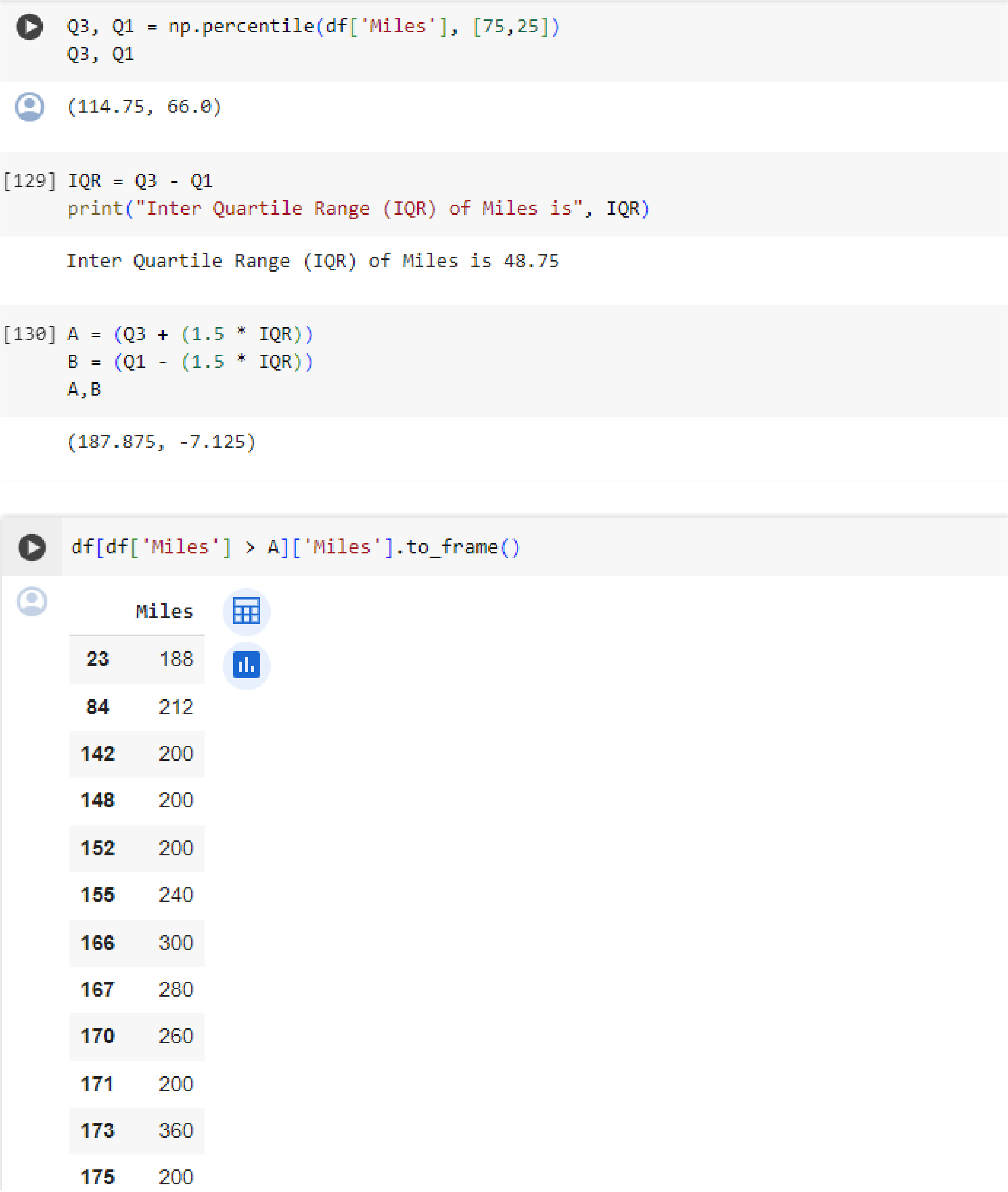


Observations

1. Customers who purchased treadmills are moderately fit (Rating - 3).
2. Distribution of customers rating of 2 & 4 are roughly same.
3. Value count of customers of fitness rating 5 is slightly more than 2 and 4.

```
# Base on Miles
sns.boxplot(data=df, x='Miles', palette='magma')
plt.show()
```





```
Q3, Q1 = np.percentile(df['Miles'], [75,25])
Q3, Q1
```

```
(114.75, 66.0)
```

```
[129] IQR = Q3 - Q1
      print("Inter Quartile Range (IQR) of Miles is", IQR)
```

```
Inter Quartile Range (IQR) of Miles is 48.75
```

```
[130] A = (Q3 + (1.5 * IQR))
      B = (Q1 - (1.5 * IQR))
      A,B
```

```
(187.875, -7.125)
```

```
df[df['Miles'] > A]['Miles'].to_frame()
```

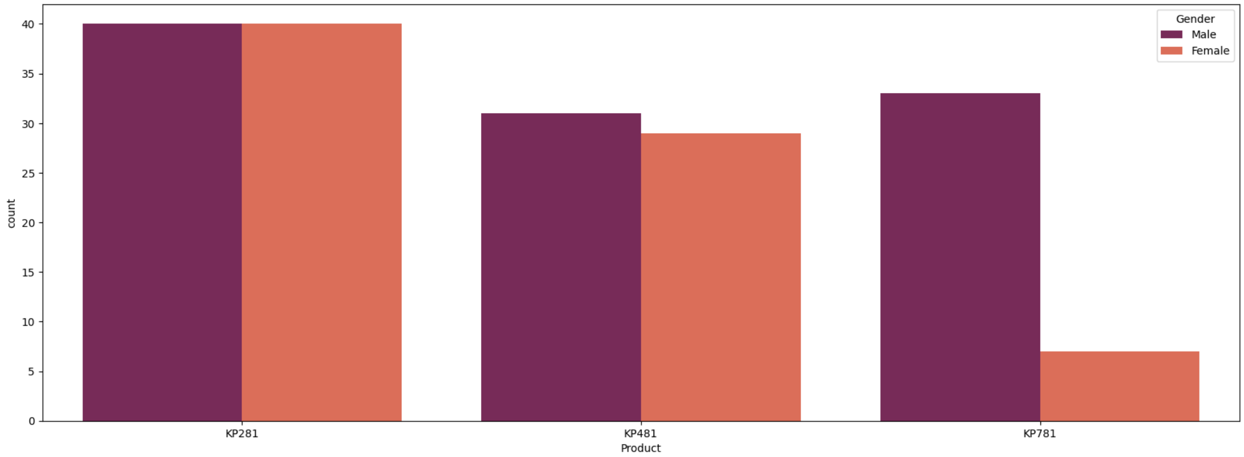
	Miles
23	188
84	212
142	200
148	200
152	200
155	240
166	300
167	280
170	260
171	200
173	360
175	200

Observations

1. Most of the distance travelled by the customer on the treadmill is roughly between 75-120 Miles.
2. Very few people have the travelled more than roughly 200 Miles (Outliers).
3. Difference between 25th and 75th percentile is 48.75 miles (running/walking).

Bivariate Analysis

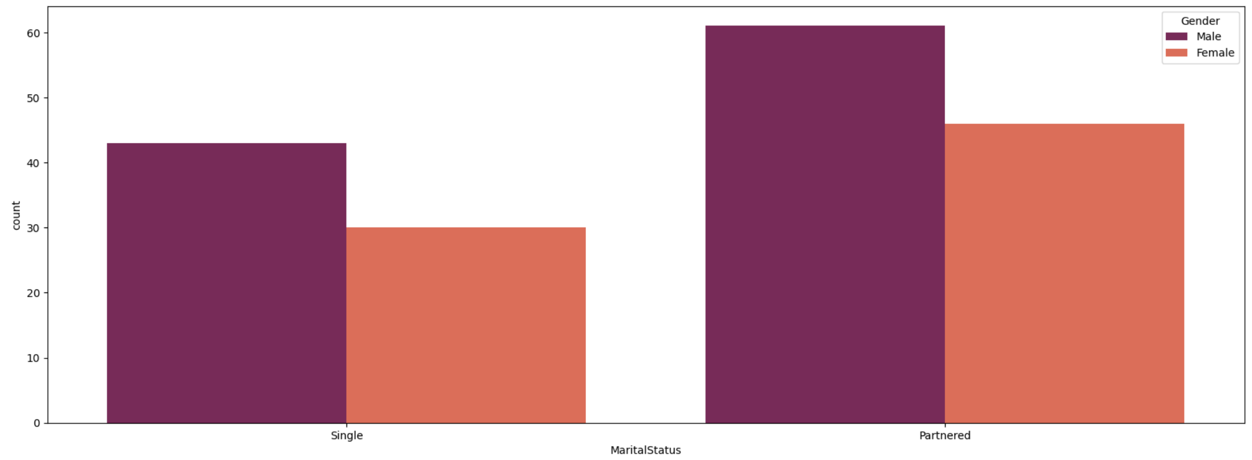
```
# Gender & Product
sns.countplot(data=df, x='Product', hue='Gender',palette='rocket')
plt.show()
```



Observations

1. Most common preference for both gender is KP281.
2. Ratio of Male/Female customers is huge in KP781.
3. Distribution of Male & Female is roughly same for KP481.
4. Males have bought more KP781 compare to KP481.

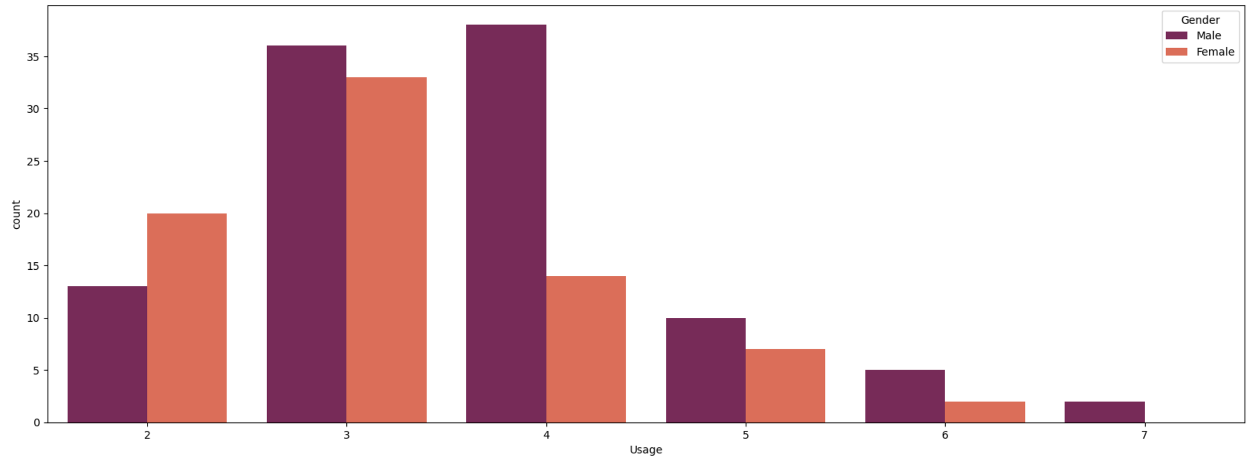
```
# Gender & MaritalStatus
sns.countplot(data=df, x='MaritalStatus', hue='Gender',palette='rocket')
plt.show()
```



Observations

1. Irrespective of Martial Status, Men are the most frequent buyer of the treadmill.
2. Partnered female are more frequent buyers compare to Single females.

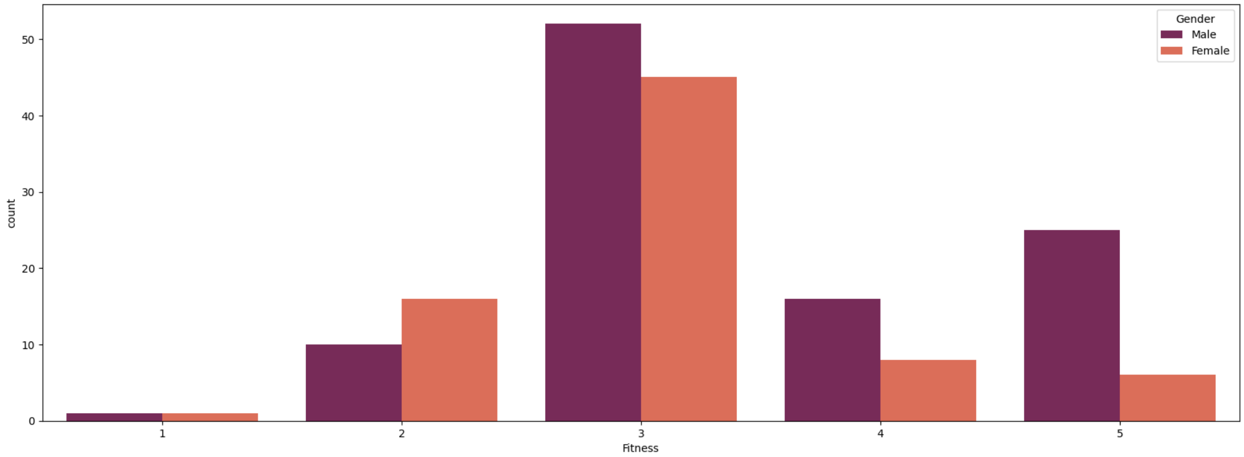
```
# Gender & Usage
sns.countplot(data=df, x='Usage', hue='Gender',palette='rocket')
plt.show()
```



Observations

1. Majority of males seems to use treadmill 4 times per week.
2. Followed by males using 3 times per week. Most of the women seems to use treadmills 3 times per week.
3. Very few males use treadmills 7 times per week, while no female seems to using 7 times per week.

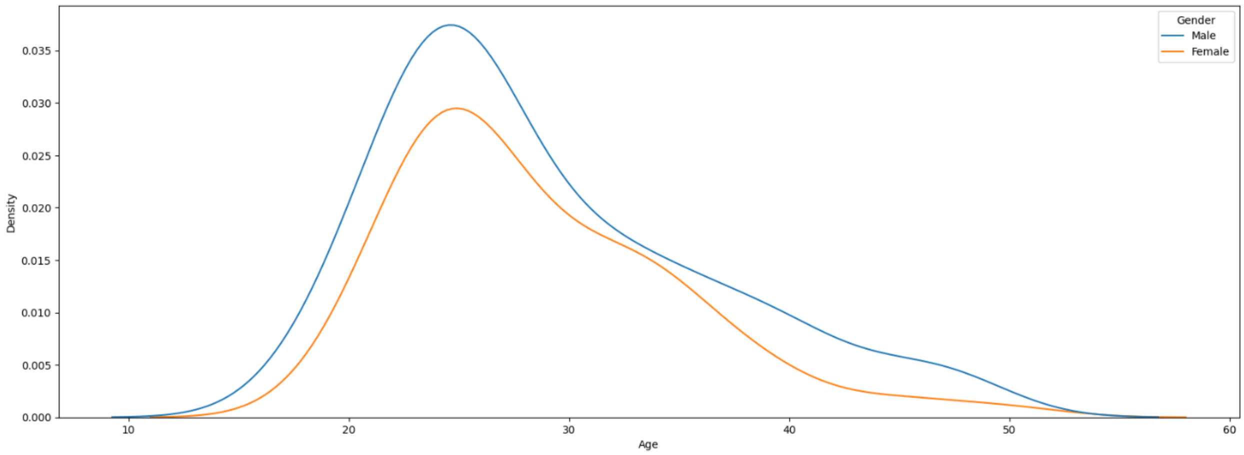
```
# Gender & Fitness
sns.countplot(data=df, x='Fitness', hue='Gender',palette='rocket')
plt.show()
```



Observations

1. Both genders are moderately fit (Fitness scale 3).
2. There are more men who have self-rating of 5 compare to women.
3. Distribution of self-rating 1 for both genders is roughly same.
4. There are more women who have self-rating of 2 compare to men.

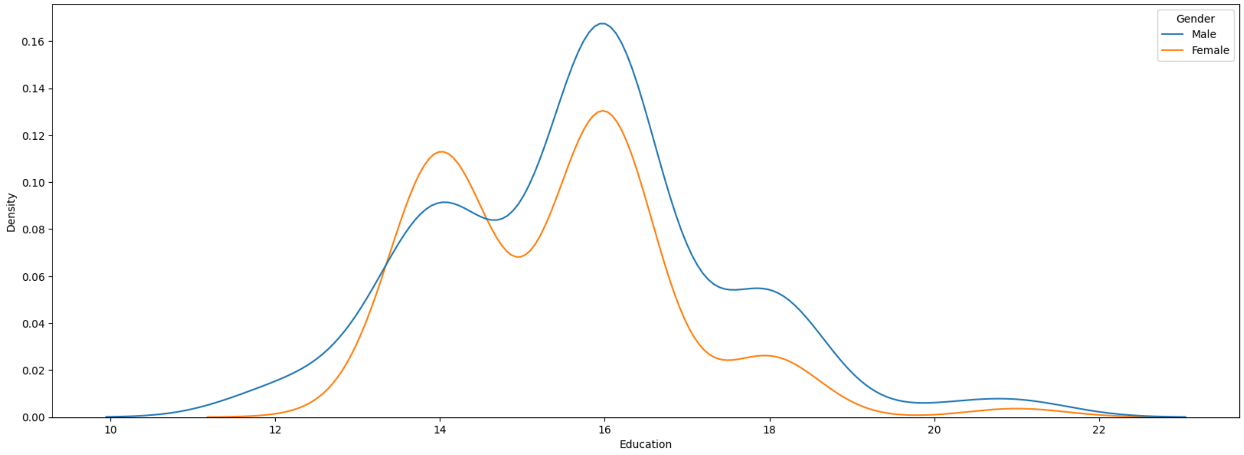
```
# Gender & Age
sns.kdeplot(data=df, x='Age', hue='Gender')
plt.show()
```



Observations

1. Most of the customers are in the age range of 20-40.
2. Most of the Men are treadmill buyers.

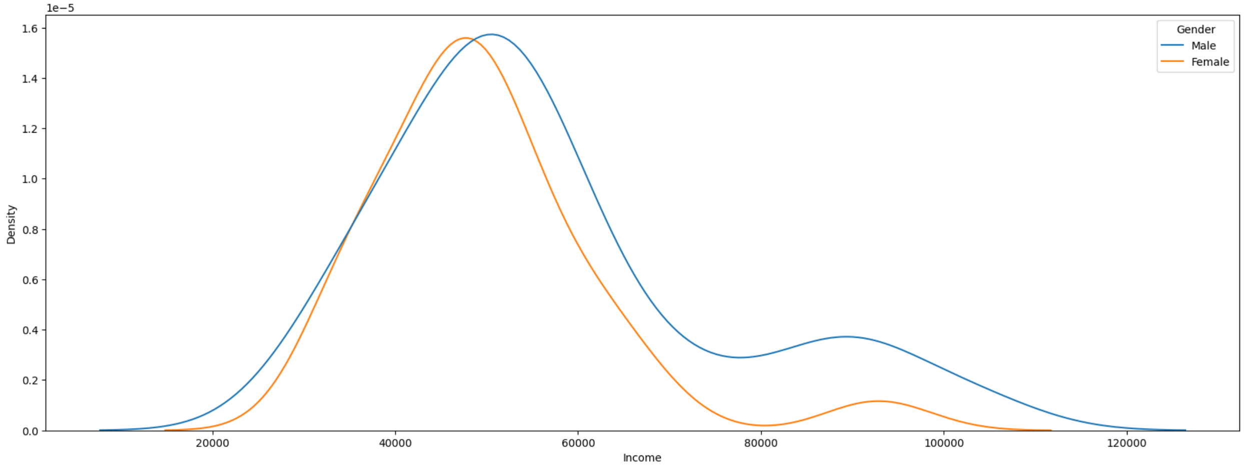
```
# Gender & Education
sns.kdeplot(data=df, x='Education', hue='Gender')
plt.show()
```



Observations

1. Both genders have roughly same education years.
2. As the dataset contains majority of Male customers, Education count of male are high.
3. There are some female who has 13-15 years of education.

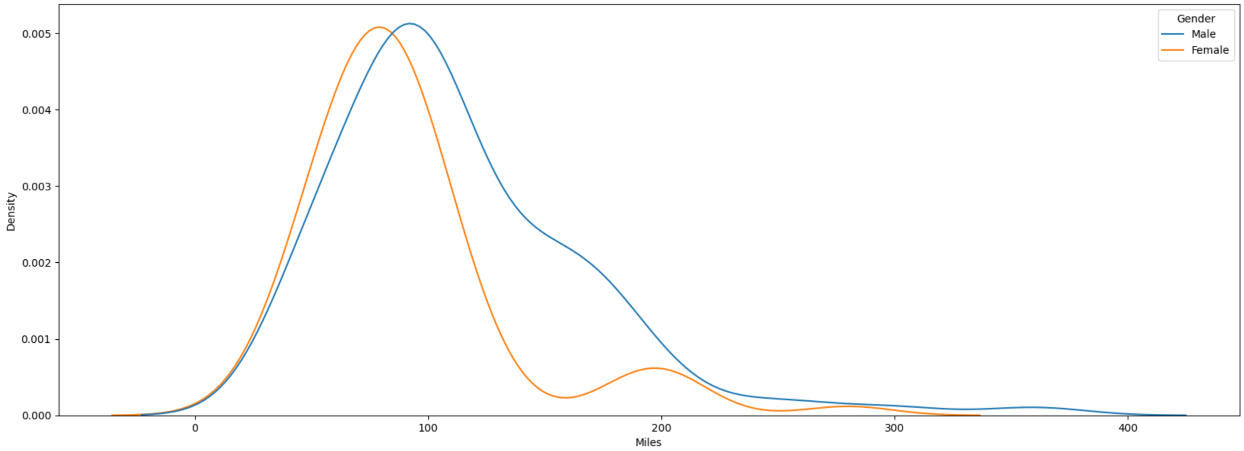
```
# Gender & Income
sns.kdeplot(data=df, x='Income', hue='Gender')
plt.show()
```



Observations

1. Peek Income of both genders are roughly same i.e., between 40K - 70K.
2. Majority of the males have annual salary more than 75K, while few female have the same.
3. Distribution of salary less 30K is same in both genders.

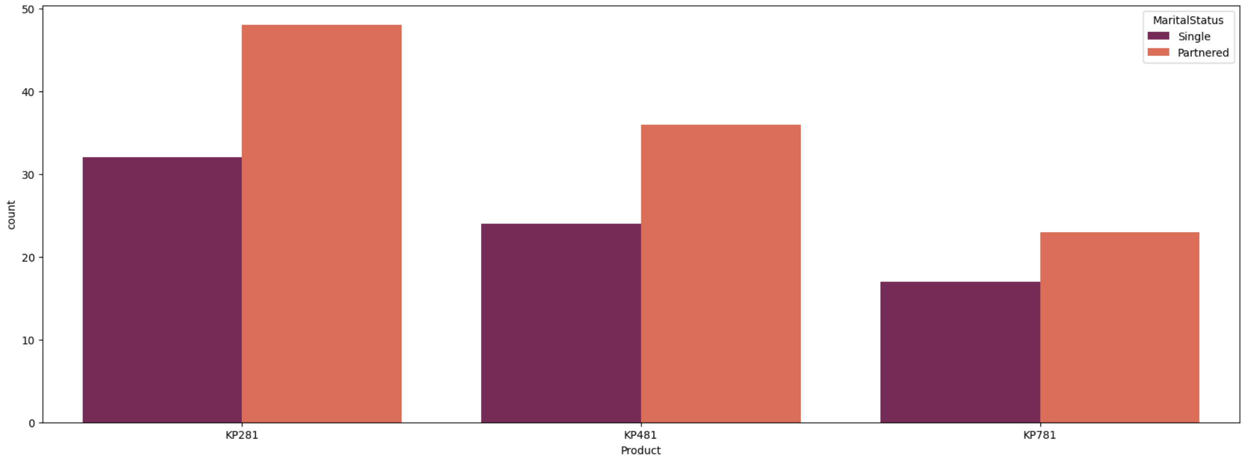
```
# Gender & Miles
sns.kdeplot(data=df, x='Miles', hue='Gender')
plt.show()
```



Observations

1. Both gender have peak miles **roughly** between 80-100 miles.
2. Very few males walk/run on treadmill for more than **roughly** 320 miles, while few females walk/run for 320 miles.
3. Distribution starts to deviate onwards 150 miles as the gap starts to increase and the gap stops at 200 miles.

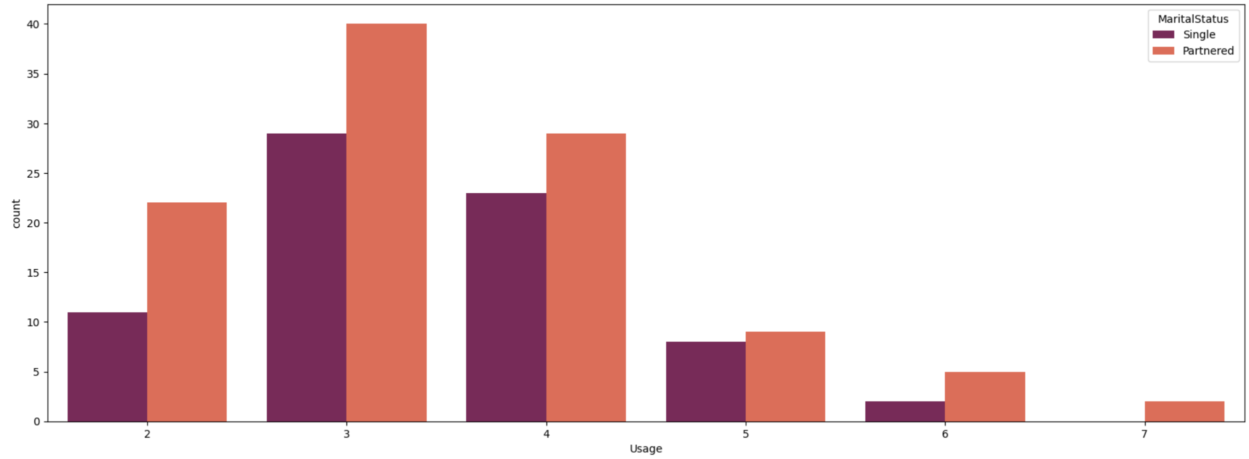
```
# Preferred product of Couples and Singles
sns.countplot(data=df, x="Product", hue='MaritalStatus',palette='rocket')
plt.show()
```



Observations

1. Most preferred product for married couples is KP281. It is also the preferred prdouct of Singles as well.
2. Followed by KP481 & KP781 respectively.
3. Maybe it's because KP281 is cheaper than KP481 and KP781.

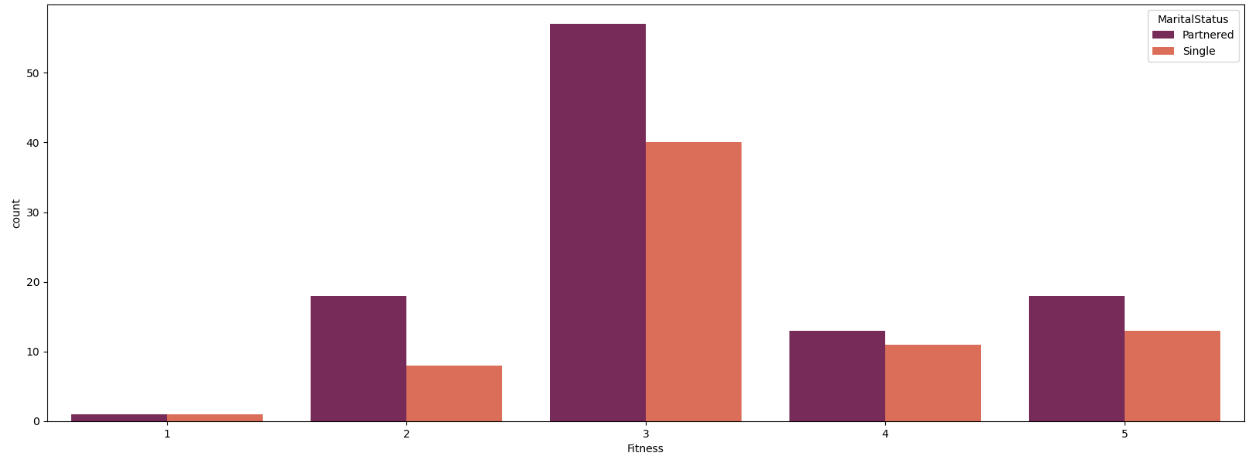
```
#  Usage of Couples and Singles
sns.countplot(data=df, x="Usage", hue='MaritalStatus',palette='rocket')
plt.show()
```



Observations

1. Overall usage of married couples is more compare to Singles.
2. Irrespective of Martial Status, usage is 3 times per week. Followedby 4 times per week.
3. Partnered status have small sample of doing workout 7 times per week, while no single people have more than 6 times per week.

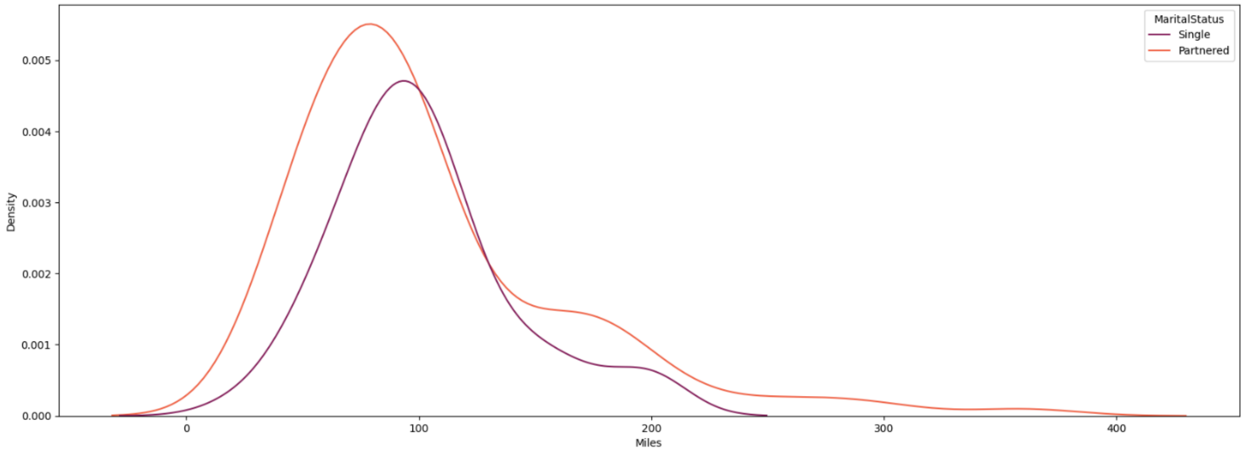
```
# Fitness of Couples and Singles
sns.countplot(data=df, x="Fitness", hue='MaritalStatus',palette='rocket')
plt.show()
```



Observations

1. Most of the customers are moderately fit irrespective of their Martial Status.
2. Followed by Fitness rating of 5 where majority of customers are married.
3. Fitness rating of 1 is same for Single and Married people. Same thing can be seen in fitness vs gender graph.

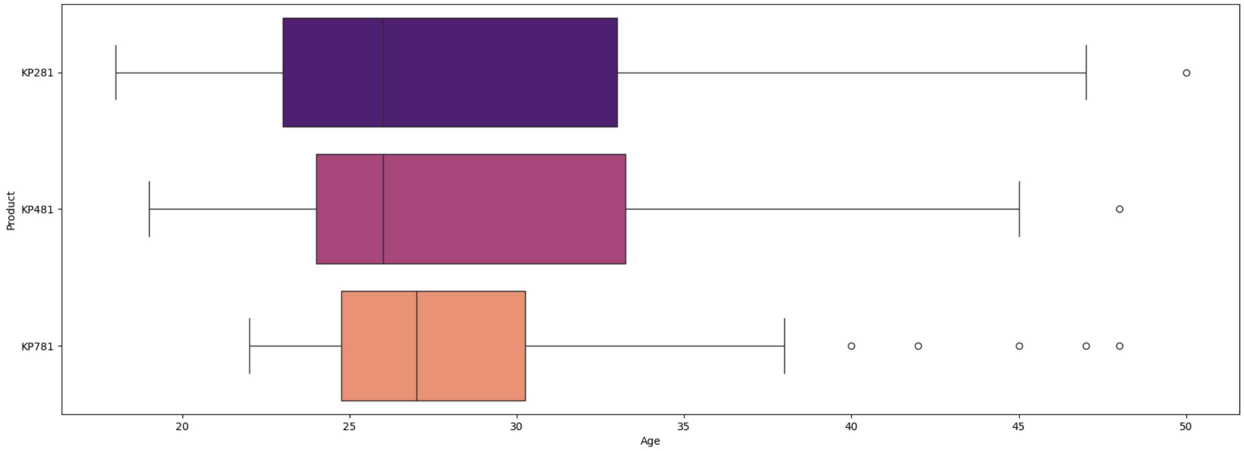
```
# Miles (walk/run) of Couples and Singles
sns.kdeplot(data=df, x="Miles", hue='MaritalStatus',palette='rocket')
plt.show()
```



Observations

1. Partnered status prefers to workout more on treadmills.
2. Distribution of Single & Partnered tends overlap at 100-150 miles.
3. Parterned tends to workout even after 200 miles, where 200 is the upper limit of singles.

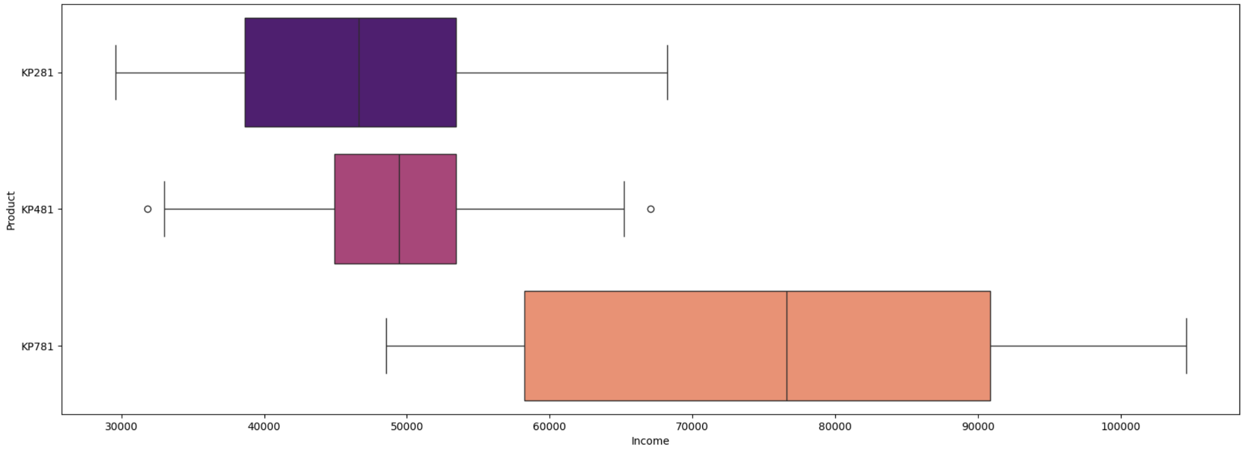
```
# Ages on different Products
sns.boxplot(x=df['Age'], y=df['Product'],palette='magma')
plt.show()
```



Observations

1. Age distribution in KP281 is maximum, followed by KP481, and KP781 respectively.
2. There is only one outlier in KP281, followed by one and five outliers in KP481 and KP781 respectively.
3. Buyer of KP281 is in the range of roughly 22 - 33 years.
4. Buyer of KP481 is in the range of roughly 24 - 34 years.
5. Buyer of KP781 is in the range of roughly 25 - 30 years.

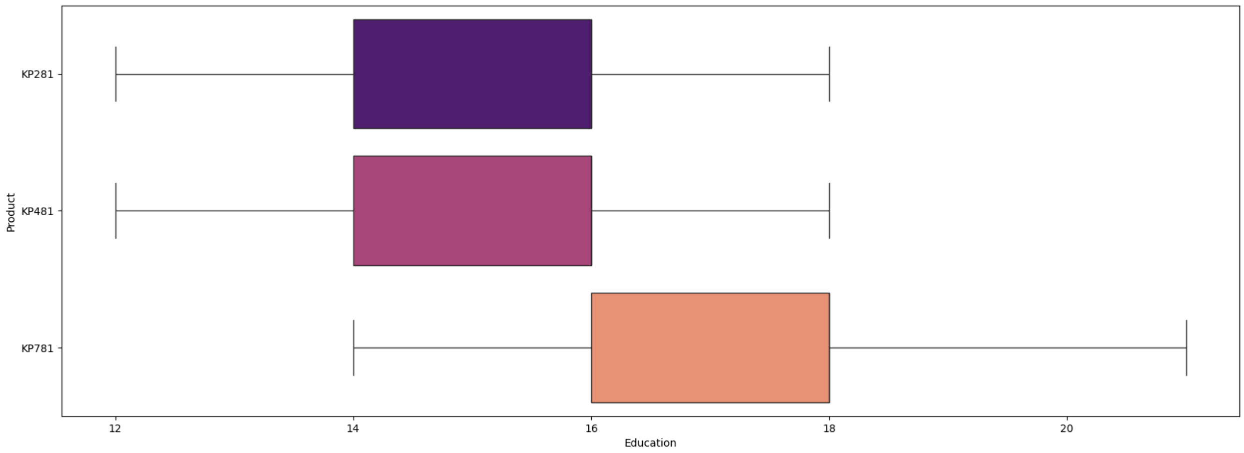
```
# Income on different Products
sns.boxplot(x=df['Income'], y=df['Product'],palette='magma')
plt.show()
```



Observations

1. Income distribution in KP781 is maximum, followed by KP281, and KP481 respectively.
2. There is no outlier in KP781 and KP281, while KP481 has two outliers at extreme ends i.e., Lower IQR and Upper IQR.
3. Income distribution of KP281 buyer is roughly between 39K - 53K dollars.
4. Income distribution of KP481 buyer is roughly between 45K - 53K dollars.
5. Income distribution of KP781 buyer is roughly between 59K - 92K dollars.

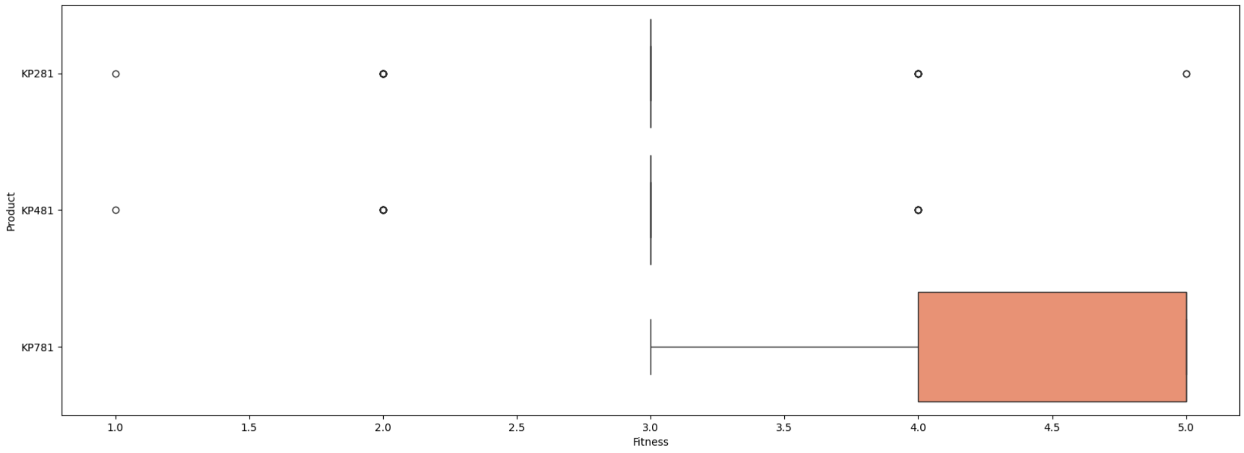
```
# Education on different Products
sns.boxplot(x=df['Education'], y=df['Product'],palette='magma')
plt.show()
```



Observations

1. Distributions of Education for all three models is same.
2. Range of distribution for KP281 and KP481 is exactly same i.e., 14 - 16 years.
3. Range of distribution for KP781 is between 16 - 18 years of education.
4. There is no outlier for all three models w.r.t Education.
5. People with more education years tend to buy KP781 as correlation between Education & Income is high (0.63).

```
# Fitness on different Products
sns.boxplot(x=df['Fitness'], y=df['Product'],palette='magma')
plt.show()
```

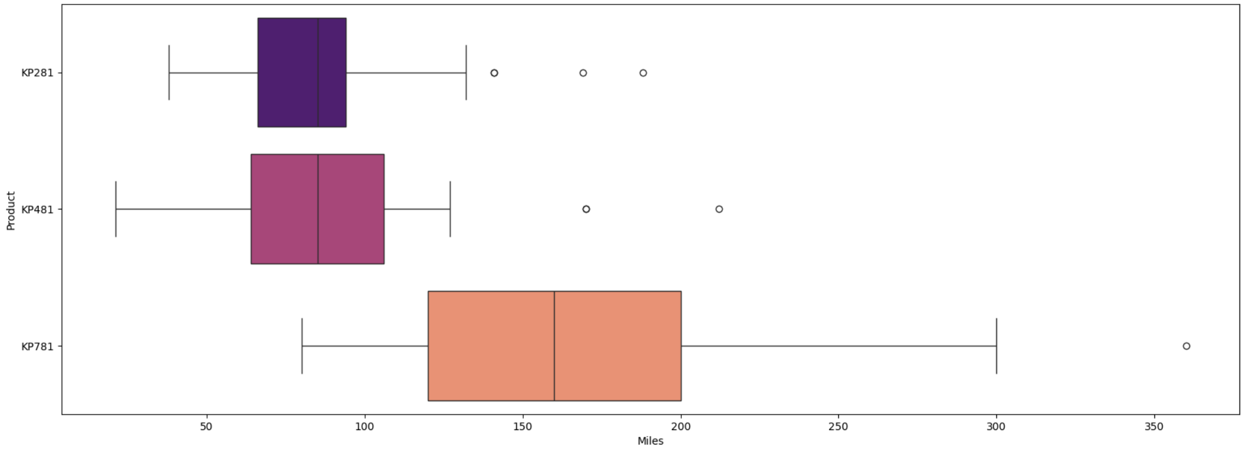


Observations

1. Fitness Distribution of KP781 is maximum, while for KP281 and KP481 distribution is same.

2. There's no outlier in fitness rating w.r.t to KP781, while there's three and four outliers in KP481 and KP281, respectively.
3. Fitness Median of KP281 and KP481 is same.

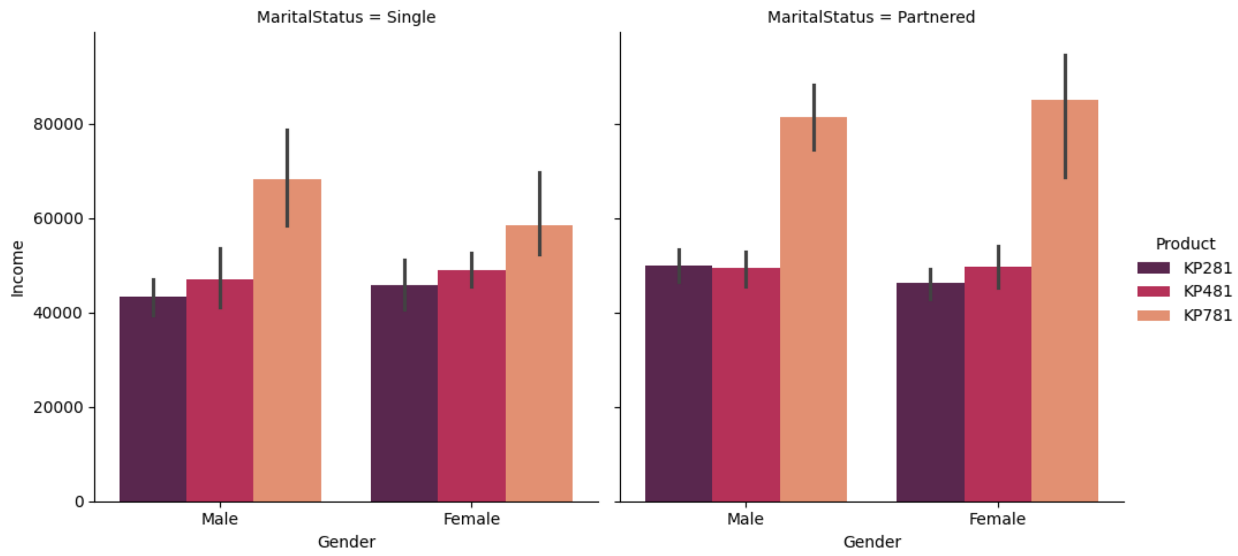
```
# Miles on different Products
sns.boxplot(x=df['Miles'], y=df['Product'],palette='magma')
plt.show()
```



Observations

1. Distribution of Miles is maximum in KP781, followed by KP481 and KP281, respectively.
2. People tend to workout more on KP781, range is roughly between 125 Miles - 200 Miles.
3. Range of Miles on KP481 is roughly between 75 - 100 Miles.
4. Range of Miles on KP281 is roughly between 75 - 80 Miles.
5. There is only one outlier in KP781 w.r.t Miles, followed by two and three in KP481 and KP281 respectively.

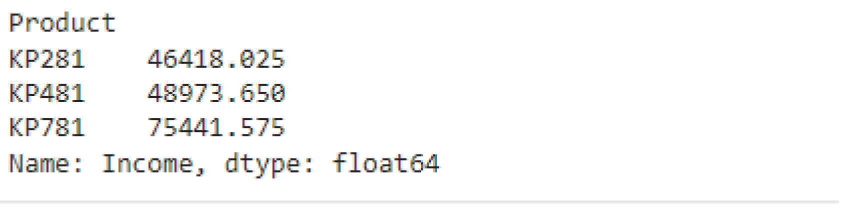
```
# Income by gender by product and by marital status
sns.catplot(x='Gender',y='Income', hue='Product', col='MaritalStatus',
data=df, kind='bar', palette='rocket')
plt.show()
```



Observations

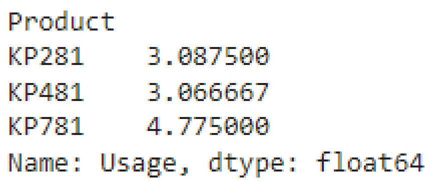
1. Partnered Female bought KP781 treadmill compared to Partnered Male.
2. Single Female customers bought KP281 treadmill slightly more compared to Single Male customers.
3. Partnered Male customers bought KP281 treadmill slightly more than Single Male customers.
4. There are more single Males buying treadmill than single Females.
5. Single Male customers bought KP781 treadmill compared to single Female.
6. Distribution of KP481 in Single & Partered, Male & Female is same.
7. Partnered customers are more than Single customers.

```
#Average Income of customer buying each model
df.groupby('Product')['Income'].mean()
```



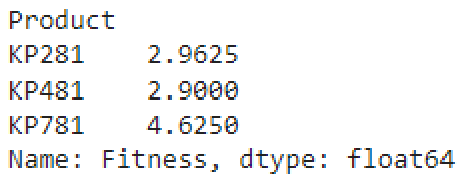
```
Product
KP281    46418.025
KP481    48973.650
KP781    75441.575
Name: Income, dtype: float64
```

```
#Average Usage of customer buying each model
df.groupby('Product')['Usage'].mean()
```



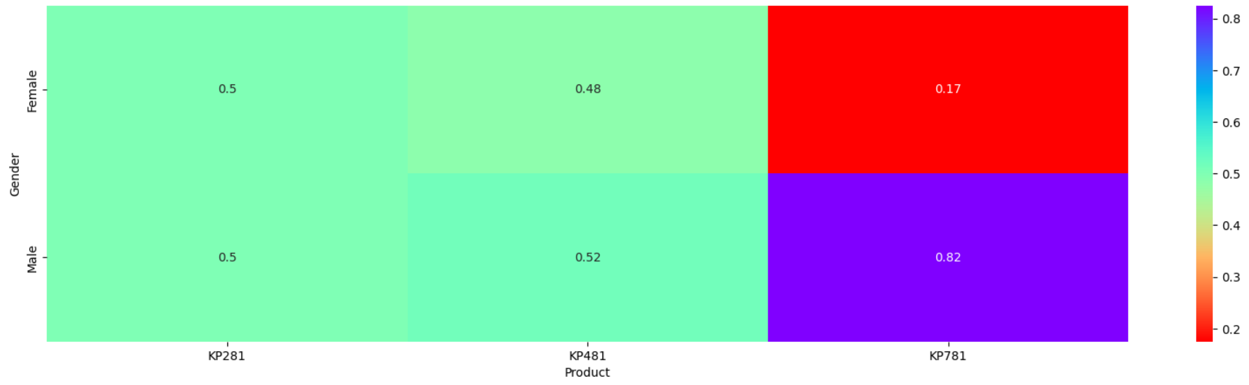
```
Product
KP281    3.087500
KP481    3.066667
KP781    4.775000
Name: Usage, dtype: float64
```

```
#Average Fitness of customer buying each model
df.groupby('Product')['Fitness'].mean()
```



```
Product
KP281    2.9625
KP481    2.9000
KP781    4.6250
Name: Fitness, dtype: float64
```

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['Gender'], df['Product'], normalize='columns'),
annot=True, cmap='rainbow_r')
plt.show()
```



Conditional Probability, P(Gender | Product)

1. Probability that customer is Male given that he bought KP281, P(Customer=Male | Producty=KP281) = 0.50.

2. Probability that customer is Female given that she bought KP281, P(Customer=Female | Product=KP281) = 0.50.

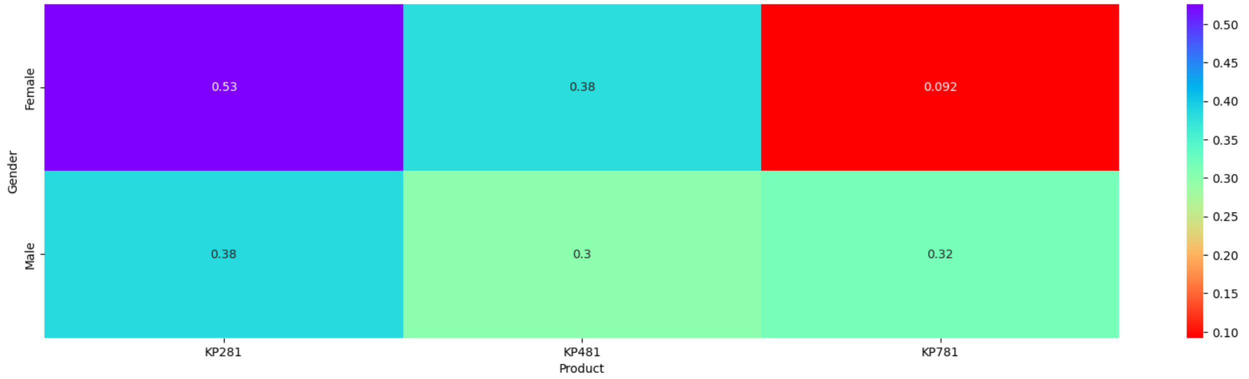
3. Probability that customer is Male given that he bought KP481, P(Customer=Male | Product=KP481) = 0.52.

4. Probability that customer is Female given that she bought KP481, P(Customer=Female | Product=KP481) = 0.48.

5. Probability that customer is Male given that he bought KP781, P(Customer=Male | Product=781) = 0.82.

6. Probability that customer is Female given that he bought KP781, P(Customer=Female | Product=KP781) = 0.17.

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['Gender'], df['Product'], normalize='index'),
annot=True, cmap='rainbow_r')
plt.show()
```



Conditional Probability, P(Product | Gender)

1. Probability of buying KP281 given that the customer is male, P(Product=KP281 | Customer=Male) = 0.38.

2. Probability of buying KP481 given that the customer is male, P(Product=KP481 | Customer=Male) = 0.3.

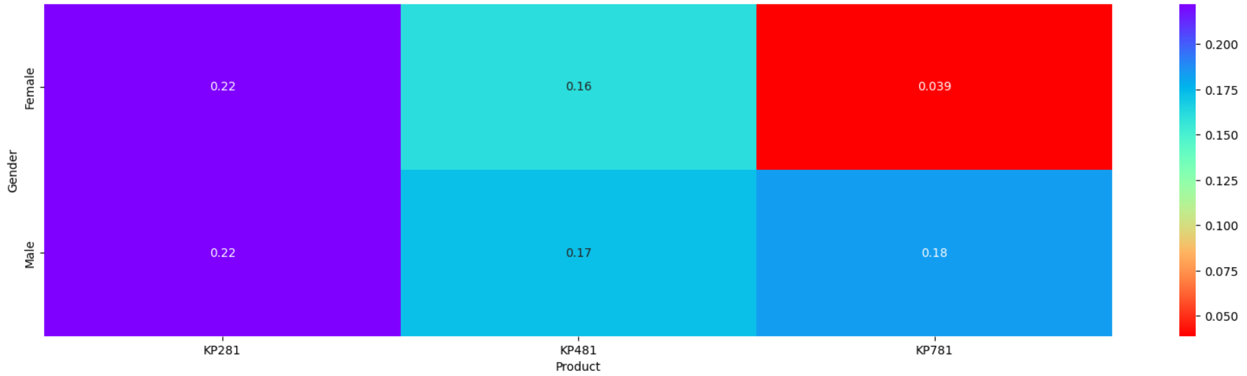
3. Probability of buying KP781 given that the customer is male, P(Product=KP781 | Customer=Male) = 0.32.

4. Probability of buying KP281 given that the customer is female, P(Product=KP281 | Customer=Female) = 0.53.

5. Probability of buying KP481 given that the customer is female, P(Product=KP481 | Customer=Female) = 0.38.

6. Probability of buying KP781 given that the customer is female, P(Product=KP781 | Customer=Female) = 0.092.

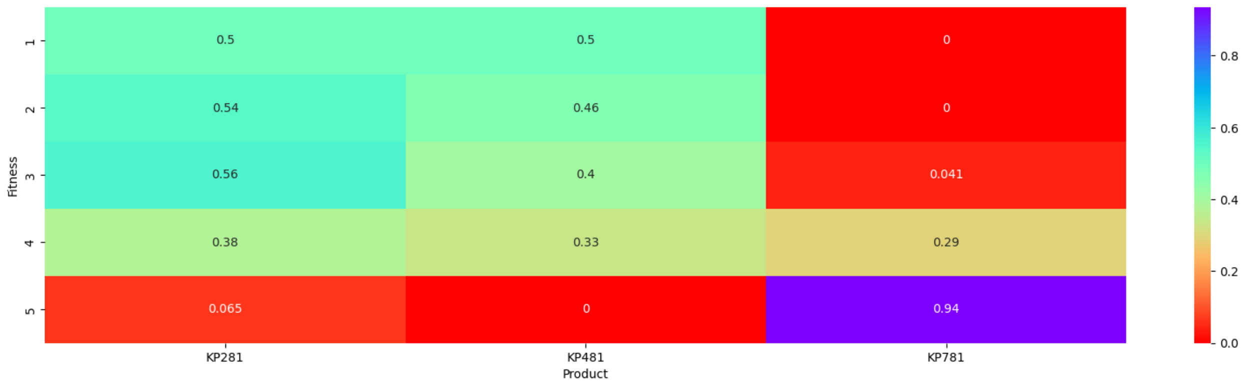
```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['Gender'], df['Product'], normalize=True),
annot=True, cmap='rainbow_r')
plt.show()
```



Joint Probability, P(Product Intersection Gender)

1. Probability that customer buys KP281 and gender is Male, P(KP281 Intersection Male) = 0.22.
2. Probability that customer buys KP481 and gender is Male, P(KP481 Intersection Male) = 0.17.
3. Probability that customer buys KP781 and gender is Male, P(KP781 Intersection Male) = 0.18.
4. Probability that customer buys KP281 and gender is Female, P(KP281 Intersection Female) = 0.22.
5. Probability that customer buys KP481 and gender is Female, P(KP481 Intersection Female) = 0.16.
6. Probability that customer buys KP781 and gender is Female, P(KP781 Intersection Female) = 0.039.

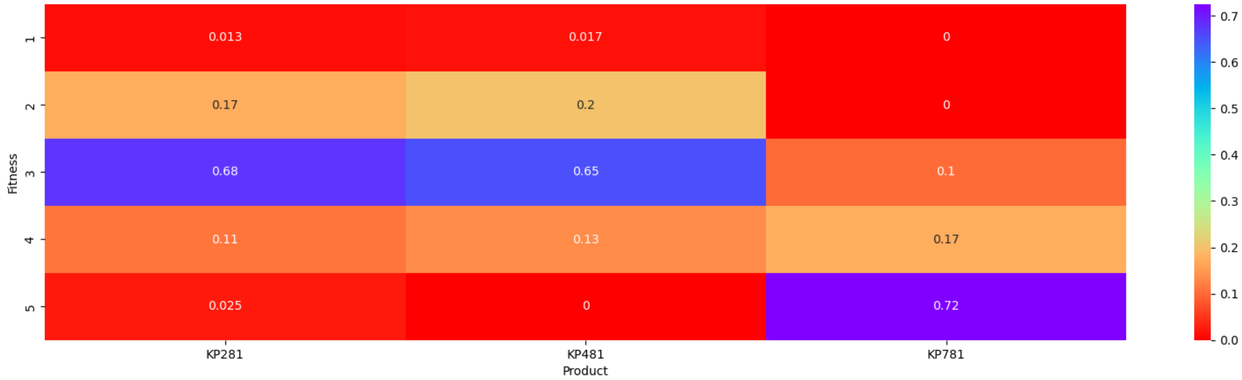
```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['Fitness'], df['Product'], normalize='index'),
annot=True, cmap='rainbow_r')
plt.show()
```



Conditional Probability P(Product | Fitness)

1. Probability of customer buying KP281 given fitness rating is 5, P(Product=KP281 | Fitness=5) = 0.06.
2. Probability of customer buying KP481 given fitness rating is 5, P(Product=KP481 | Fitness=5) = 0.0 (impossible event).
3. Probability of customer buying KP781 given fitness rating is 5, P(Product=KP781 | Fitness=5) = 0.94.
4. Probability of customer buying KP281 given fitness rating is 4, P(Product=KP281 | Fitness=4) = 0.38.
5. Probability of customer buying KP481 given fitness rating is 4, P(Product=KP481 | Fitness=4) = 0.33.
6. Probability of customer buying KP781 given fitness rating is 4, P(Product=KP781 | Fitness=4) = 0.29.
7. Probability of customer buying KP281 given fitness rating is 3, P(Product=KP281 | Fitness=3) = 0.56.
8. Probability of customer buying KP481 given fitness rating is 3, P(Product=KP481 | Fitness=3) = 0.4.
9. Probability of customer buying KP781 given fitness rating is 3, P(Product=KP781 | Fitness=3) = 0.04.
10. Probability of customer buying KP281 given fitness rating is 2, P(Product=KP281 | Fitness=2) = 0.54.
11. Probability of customer buying KP481 given fitness rating is 2, P(Product=KP481 | Fitness=2) = 0.46.
12. Probability of customer buying KP781 given fitness rating is 2, P(Product=KP781 | Fitness=2) = 0.0 (impossible event).
13. Probability of customer buying KP281 given fitness rating is 1, P(Product=KP281 | Fitness=1) = 0.5.
14. Probability of customer buying KP481 given fitness rating is 1, P(Product=KP481 | Fitness=1) = 0.5.
15. Probability of customer buying KP781 given fitness rating is 1, P(Product=KP781 | Fitness=1) = 0.0 (impossible event).

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['Fitness'], df['Product'],
normalize='columns'), annot=True, cmap='rainbow_r')
plt.show()
```

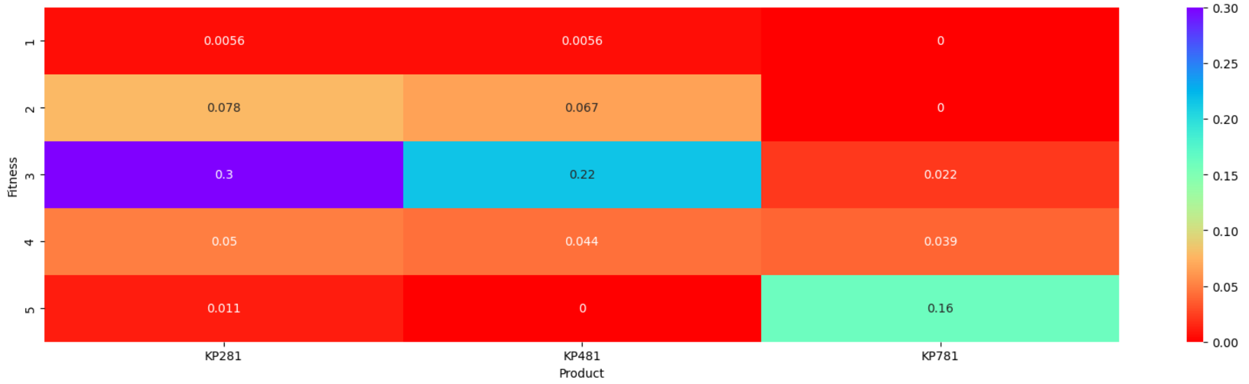


Conditional Probability P(Fitness | Product)

1. Probability of customer fitness rating is 5 given that they purchased KP281, P(Fitness=5 | Product=KP281) = 0.025.
2. Probability of customer fitness rating is 4 given that they purchased KP281, P(Fitness=4 | Product=KP281) = 0.11.
3. Probability of customer fitness rating is 3 given that they purchased KP281, P(Fitness=3 | Product=KP281) = 0.68.
4. Probability of customer fitness rating is 2 given that they purchased KP281, P(Fitness=2 | Product=KP281) = 0.17.
5. Probability of customer fitness rating is 1 given that they purchased KP281, P(Fitness=1 | Product=KP281) = 0.013.
6. Probability of customer fitness rating is 5 given that they purchased KP481, P(Fitness=5 | Product=KP481) = 0.0 (impossible event).
7. Probability of customer fitness rating is 4 given that they purchased KP481, P(Fitness=4 | Product=KP481) = 0.13.
8. Probability of customer fitness rating is 3 given that they purchased KP481, P(Fitness=3 | Product=KP481) = 0.65.
9. Probability of customer fitness rating is 2 given that they purchased KP481, P(Fitness=2 | Product=KP481) = 0.2.
10. Probability of customer fitness rating is 1 given that they purchased KP481, P(Fitness=1 | Product=KP481) = 0.017.
11. Probability of customer fitness rating is 5 given that they purchased KP781, P(Fitness=5 | Product=KP781) = 0.72.
12. Probability of customer fitness rating is 4 given that they purchased KP781, P(Fitness=4 | Product=KP781) = 0.17.

13. Probability of customer fitness rating is 3 given that they purchased KP781, P(Fitness=3 | Product=KP781) = 0.1.
14. Probability of customer fitness rating is 2 given that they purchased KP781, P(Fitness=2 | Product=KP781) = 0.0 (impossible event).
15. Probability of customer fitness rating is 1 given that they purchased KP781, P(Fitness=1 | Product=KP781) = 0.0 (impossible event).

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['Fitness'], df['Product'], normalize=True),
annot=True, cmap='rainbow_r')
plt.show()
```

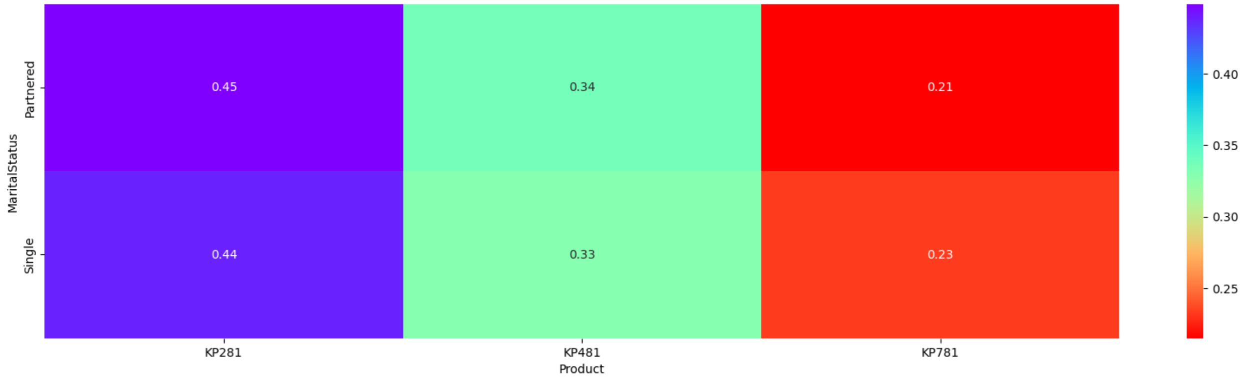


Joint Probability P(Product Intersection Fitness)

1. Probability of buying KP281 and their fitness rating is 5, P(KP281 Intersection Fitness=5) = 0.0011.
2. Probability of buying KP281 and their fitness rating is 4, P(KP281 Intersection Fitness=4) = 0.05.
3. Probability of buying KP281 and their fitness rating is 3, P(KP281 Intersection Fitness=3) = 0.3.
4. Probability of buying KP281 and their fitness rating is 2, P(KP281 Intersection Fitness=2) = 0.078.
5. Probability of buying KP281 and their fitness rating is 1, P(KP281 Intersection Fitness=1) = 0.0056.
6. Probability of buying KP481 and their fitness rating is 5, P(KP481 Intersection Fitness=5) = 0.0 (impossible event).
7. Probability of buying KP481 and their fitness rating is 4, P(KP481 Intersection Fitness=4) = 0.0044.
8. Probability of buying KP481 and their fitness rating is 3, P(KP481 Intersection Fitness=3) = 0.22.

9. Probability of buying KP481 and their fitness rating is 2, P(KP481 Intersection Fitness=2) = 0.067.
10. Probability of buying KP481 and their fitness rating is 1, P(KP481 Intersection Fitness=1) = 0.0056.
11. Probability of buying KP781 and their fitness rating is 5, P(KP781 Intersection Fitness=5) = 0.16.
12. Probability of buying KP781 and their fitness rating is 4, P(KP781 Intersection Fitness=4) = 0.039.
13. Probability of buying KP781 and their fitness rating is 3, P(KP781 Intersection Fitness=3) = 0.022.
14. Probability of buying KP781 and their fitness rating is 2, P(KP781 Intersection Fitness=2) = 0.0 (impossible event).
15. Probability of buying KP781 and their fitness rating is 1, P(KP781 Intersection Fitness=1) = 0.0 (impossible event).

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'],
normalize='index'), annot=True, cmap='rainbow_r')
plt.show()
```



Conditional Probability, P(Product | MaritalStatus)

1. Probability of buying KP281 given that the marital status is single, P(Product=KP281 | MaritalStatus=Single) = 0.44.

2. Probability of buying KP481 given that the marital status is single, P(Product=KP481 | MaritalStatus=Single) = 0.33.

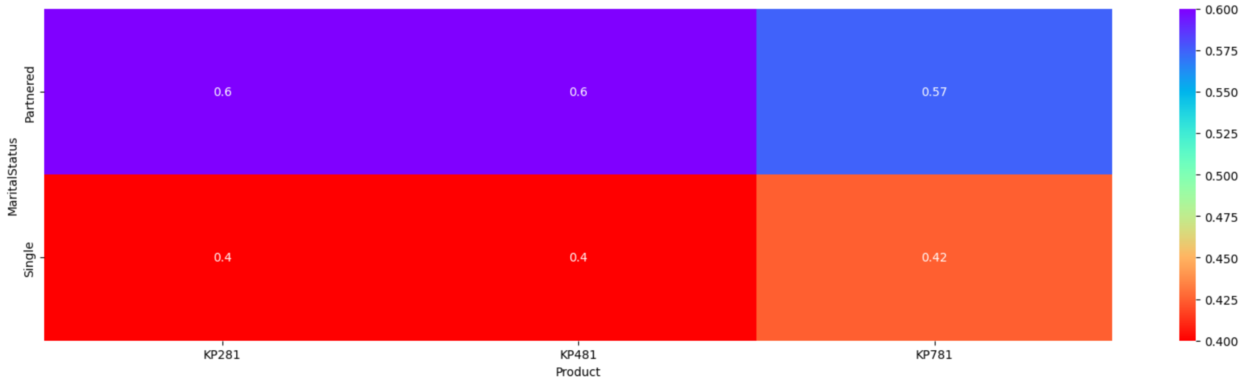
3. Probability of buying KP781 given that the marital status is single, P(Product=781 | MaritalStatus=Single) = 0.23.

4. Probability of buying KP281 given that the marital status is partnered, P(Product=KP281 | MaritalStatus=Single) = 0.45.

5. Probability of buying KP481 given that the cmarital status is partnered, P(Product=KP481 | MaritalStatus=Single) = 0.34.

6. Probability of buying KP781 given that the marital status is partnered, P(Product=KP781 | MaritalStatus=Single) = 0.21.

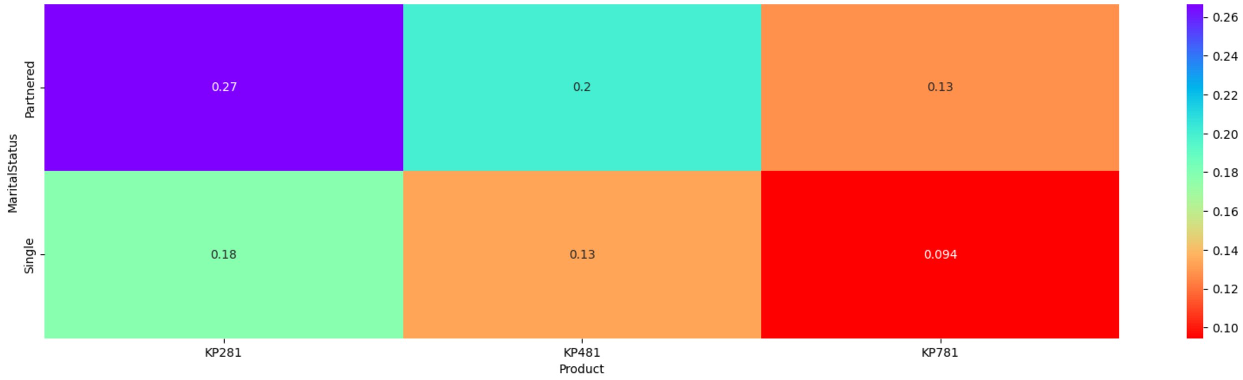
```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'],
normalize='columns'), annot=True, cmap='rainbow_r')
plt.show()
```



Conditional Probability P(MaritalStatus | Product)

1. Probability of Marital Status being Single given that KP281 is purchased, P(MaritalStatus=Single | Product=KP281) = 0.40.
2. Probability of Marital Status being Parterned given that KP281 is purchased, P(MaritalStatus=Parterned | Product=KP281) = 0.60.
3. Probability of Marital Status being Single given that KP481 is purchased, P(MaritalStatus=Single | Product=KP481) = 0.4.
4. Probability of Marital Status being Partnered given that KP481 is purchased, P(MaritalStatus=Partnered | Product=KP481) = 0.6.
5. Probability of Marital Status being Single given that KP781 is purchased, P(MaritalStatus=Single | Product=KP781) = 0.42.
6. Probability of Marital Status being Partnered given that KP781 is purchased, P(MaritalStatus=Partnered | Product=KP781) = 0.57.

```
from matplotlib import rcParams
rcParams['figure.figsize'] = 20,5

sns.heatmap(pd.crosstab(df['MaritalStatus'], df['Product'],
normalize=True), annot=True, cmap='rainbow_r')
plt.show()
```



Joint Probability, P(Product Intersection MartialStatus)

1. Probability of customer buying KP281 and their MartialStatus is Single, P(KP281 Intersection Single) = 0.18.
2. Probability of customer buying KP481 and their MartialStatus is Single, P(KP481 Intersection Single) = 0.13.
3. Probability of customer buying KP781 and their MartialStatus is Single, P(KP781 Intersection Single) = 0.094.
4. Probability of customer buying KP281 and their MartialStatus is Partnered, P(KP281 Intersection Partnered) = 0.27.
5. Probability of customer buying KP481 and their MartialStatus is Partnered, P(KP481 Intersection Partnered) = 0.2.
6. Probability of customer buying K7281 and their MartialStatus is Partnered, P(KP781 Intersection Partnered) = 0.13.

```
df['Product'].value_counts(normalize=True)
```

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

Marginal Probability P(Product)

1. Probability of buying KP281 treadmill, P(Product=KP281) = 0.44.
2. Probability of buying KP481 treadmill, P(Product=KP481) = 0.33.
3. Probability of buying KP781 treadmill, P(Product=KP781) = 0.22.

```
df['Gender'].value_counts(normalize=True)
```

```
Male      0.577778
Female    0.422222
Name: Gender, dtype: float64
```

Marginal Probability P(Gender)

1. Probability of customer gender is Male, P(Gender=Male) = 0.58.
2. Probability of customer gender is Female, P(Gender=Female) = 0.42.

```
df['MaritalStatus'].value_counts(normalize=True)
```

```
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
```

Marginal Probability P(MaritalStatus)

1. Probability of customer's MaritalStatus is Partnered, P(MaritalStatus=Partnered) = 0.60.
2. Probability of customer's MaritalStatus is Single, P(MaritalStatus=Single) = 0.40.

```
df['Usage'].value_counts(normalize=True)
```

```
3    0.383333
4    0.288889
2    0.183333
5    0.094444
6    0.038889
7    0.011111
Name: Usage, dtype: float64
```

Marginal Probability P(Usage)

1. Probability of customer having usage 3 times per week is P(Usage=3) = 0.38.
2. Probability of customer having usage 4 times per week is P(Usage=4) = 0.29.
3. Probability of customer having usage 2 times per week is P(Usage=2) = 0.18.
4. Probability of customer having usage 5 times per week is P(Usage=5) = 0.09.
5. Probability of customer having usage 6 times per week is P(Usage=6) = 0.03.
6. Probability of customer having usage 7 times per week is P(Usage=7) = 0.01.

```
df['Fitness'].value_counts(normalize=True)
```

```
3   0.538889
5   0.172222
2   0.144444
4   0.133333
1   0.011111
Name: Fitness, dtype: float64
```

Marginal Probability P(Fitness)

1. Probability of customer having fitness rating of 3 is P(Fitness=3) = 0.53.
2. Probability of customer having fitness rating of 5 is P(Fitness=5) = 0.17.
3. Probability of customer having fitness rating of 2 is P(Fitness=2) = 0.14.
4. Probability of customer having fitness rating of 4 is P(Fitness=4) = 0.13.
5. Probability of customer having fitness rating of 1 is P(Fitness=1) = 0.01.

Business Insights (based on Non-Graphical and Visual Analysis):

1. The top three purchased treadmill models are KP281, KP481, and KP781, in that order.
2. There is a higher proportion of male buyers compared to female buyers.
3. More customers are in a partnered marital status compared to single.
4. The average age of customers is 28, with a range between 18 to 50 years and a median of 26 years.
5. The average education level of customers is 15.5 years, with a range between 2 to 21 years and a median of 16 years.
6. On average, customers plan to use the treadmill three times per week, with a range between 2 to 7 times per week and a median of three times per week.
7. The average self-fitness rating of customers is 3, with a range between 1 to 5 and a median of 3.

8. Customers' average annual income is 53.7K dollars, with a range between 29.5K dollars to 104K dollars and a median income of 50.5K dollars.
9. The average distance traveled by customers on the treadmill is 103 miles, with a range between 21 to 360 miles and a median of 94 miles.
10. There is a moderately strong relationship between education and income.
11. The relationship between fitness and distance traveled on the treadmill is strong.
12. Similarly, there is a strong relationship between usage frequency and distance traveled on the treadmill.
13. The age difference between the 25th and 75th percentile is nine years, indicating a relatively narrow age spread among customers.
14. The education years difference between the 25th and 75th percentile is two years, suggesting a moderate spread in education levels among customers.
15. Most customers use the treadmill 3-4 times per week, with very few using it 6-7 times per week.
16. The majority of customers rate themselves as moderately fit.
17. The mean income for KP281 buyers is 46.4K dollars, for KP481 buyers is 48.9K dollars, and for KP781 buyers is 75.4K dollars.
18. KP281 and KP481 have the same mean usage of 3, while KP781 has a mean usage of 4.
19. The mean fitness rating for KP281 and KP481 buyers is 3, while for KP781 buyers, it is 4.6.
20. KP781 is the most preferred treadmill among male customers, while females show the least preference for it.
21. Overall, male customers tend to use treadmills more frequently than females.
22. The income distribution between both genders is roughly similar.
23. Males tend to have a higher fitness level compared to females.
24. The distance traveled on the treadmill is roughly the same for both genders, but men tend to cover longer distances, with some going beyond 320 miles.
25. Partnered customers tend to have a higher fitness level compared to singles.

Customer Profiling- Categorization of Users

KP281

1. KP281 stands as the top-selling treadmill model, contributing to 44.44% of total sales.
2. The average income of KP281 buyers is 46.4K dollars.
3. Customers using KP281 have an average planned usage of three times per week.
4. KP281 customers demonstrate an average fitness rating of 3 (rounded).
5. Both genders equally favor the KP281 model as their preferred treadmill choice.
6. The age range of KP281 buyers falls approximately between 22 to 33 years.

7. The income range of KP281 treadmill customers typically lies between 39K dollars to 53K dollars.
8. The education level of KP281 buyers ranges from 14 to 16 years.
9. On the KP281 treadmill, customers cover an approximate distance of 75 to 80 miles.
10. The median/mean fitness rating for KP281 users remains at 3.
11. Single female customers slightly outnumber single male customers in KP281 purchases, while partnered male customers bought KP281 slightly more than single male customers.

KP481

1. KP481 is the second highest-selling treadmill model, accounting for 33.33% of sales.
2. Customers purchasing KP481 have an average income of 49K dollars.
3. The average planned usage of KP481 customers is three times per week.
4. KP481 customers have an average fitness rating of 3.
5. The KP481 model is slightly more popular among male buyers.
6. Couples are more likely to buy the KP481 model than single customers.
7. The age range of KP481 treadmill customers is typically between 24-34 years.
8. The income range of KP481 customers is approximately 45K dollars - 53K dollars.
9. The educational background of KP481 buyers is similar to that of KP281, spanning 14 - 16 years of education.
10. KP481 customers typically cover a distance of about 75 - 100 miles on the treadmill, making it the second most frequently used model in terms of distance.
11. The median/mean fitness rating for KP481 customers is 3, similar to KP281.
12. The purchase probabilities show no significant gender-based or marital status-based differences for KP481 buyers.

KP781

1. KP781 is a less commonly purchased treadmill model due to its higher price.
2. The average income of KP781 buyers is 75.4K dollars.
3. KP781 customers use the treadmill an average of four times per week.
4. The average fitness rating of KP781 buyers is 4.
5. KP781 is predominantly preferred by males, while fewer females buy this model.
6. The KP781 treadmill is not popular among both single and partnered customers.
7. The age range of KP781 buyers is approximately between 25-30 years, and it has seen relatively fewer purchases, possibly due to its higher cost.
8. The income range for KP781 buyers is roughly between 59K dollars to 92K dollars, showing a wider range compared to KP281 & KP481 models.
9. KP781 buyers typically have an education level ranging from 16 to 18 years, and there is a high correlation between education and income, potentially influencing the purchase decision.
10. KP781's fitness range is between 4 - 5, making it more attractive to people who are already fit and seeking additional features.

11. Partnered females are more likely to buy KP781 treadmills compared to partnered males, whereas single male customers show a higher preference for KP781 compared to single females.

Recommendations

1. Promote KP281 and KP481 treadmills as budget-friendly options, especially targeting customers with annual incomes in the range of 39K - 53K Dollars.
2. Market KP781 treadmill as a premium product with advanced features, targeting professionals and athletes.
3. Enhance the marketing strategy for KP781 by associating it with renowned athletes as an example of Virat Kohli who will inspire many as he is fittest person, leveraging their achievements for better outreach.
4. Run special marketing campaigns on Women's Day and Mother's Day to encourage more women to adopt an exercise routine, highlighting the benefits of using our treadmills.
5. Conduct research to expand the customer base beyond 50 years of age. Offer basic treadmill models (KP281/KP481) as suitable options for beginners in this age group.
6. Encourage existing customers to upgrade their treadmills to high-end models as their usage increases over time, leading to increased revenue for the business.

Conclusions

We explore the Aerofit dataset and did usual data analysis steps like checking the structure & characteristics of the dataset. Detected Outliers using boxplot, “describe” method by checking the difference between mean and median. Went through I marital status, age have any effect on the product purchased using countplot, histplots, boxplots. Represented the marginal probability like - what percent of customers have purchased KP281, KP481, or KP781 in a table. Checked correlation among different factors using heat maps or pair plots. Worked on some Conditional, Joint and Marginal probability. And finally based on insights created Customer Profiling - Categorization of users. By understanding these patterns and trends, businesses can make informed decisions and implement strategies to optimize their operations and drive growth. Here are the key takeaways from the analysis:

Key Takeaways

- Need to promote KP781 more.
- Need to showcase, advertisement about advance technology for updated product.
- Need to encourage existing customers who are showing more interest.
- Target festive season and give some discount so, people can attract more.