Aerofit Treadmill Case Study

About Aerofit

Aerofit provide's fitness equipment's and essential's but primary deal's in treadmill's.

Business Problem

Aerofit's market research team want's to know if there are any relation between the customer's and the product's they buy such that they can focus on that and improve the experience for customer's.

1. **Descriptive Analysis:** Develop tables and charts to create a customer profile for each AeroFit treadmill product.
2. **Contingency Tables:** Construct two-way contingency tables for each product and Compute conditional and marginal probabilities and provide insights on their impact on the business.

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

Importing required libraries

```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv
To: /content/aerofit_treadmill.csv
100% 7.28k/7.28k [00:00<00:00, 22.6MB/s]
```

In the above step we are downloading the data using the link

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

	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
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

180 rows × 9 columns

Here we successfully read the datain the file

Numerical analysis

```
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
```

From above code we get The info about the data usage, the rows and column's

```
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

Head function gives us the top 5 default rows of the dataset.

```
df.tail()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

Tail function gives us the bottom 5 default rows of the dataset.

```
df.shape
```

```
(180, 9)
```

The total Customer's are 180 and the data for each customer has being given for 9 different parameter's.

```
df.describe()
```

	Age	Education	Usage	Fitness	Income	Miles
count	180.000000	180.000000	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	3.455556	3.311111	53719.577778	103.194444
std	6.943498	1.617055	1.084797	0.958869	16506.684226	51.863605
min	18.000000	12.000000	2.000000	1.000000	29562.000000	21.000000
25%	24.000000	14.000000	3.000000	3.000000	44058.750000	66.000000
50%	26.000000	16.000000	3.000000	3.000000	50596.500000	94.000000
75%	33.000000	16.000000	4.000000	4.000000	58668.000000	114.750000
max	50.000000	21.000000	7.000000	5.000000	104581.000000	360.000000

It generates descriptive numerical analysis for the each column in the dataset and contain's measure's like mean,count,min,max,standard deviation and quartile's.

Non-Graphical-Analysis

```
df.nunique()
```

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

Unique attribute's for the given dataset

```
df['Product'].unique().tolist()
```

```
['KP281', 'KP481', 'KP781']
```

The unique product's that aerofit sell's are 3 Product's that are KP281, KP481 and KP781.

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

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

Gender count individually for male and female there are 104 males and 76 female's

```
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])
```

This gives us the unique values of age's present in the age dataset.

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

```
25    25
23    18
24    12
26    12
28     9
35     8
33     8
30     7
38     7
```

```
21     7
22     7
27     7
31     6
34     6
29     6
20     5
40     5
32     4
19     4
48     2
37     2
45     2
47     2
46     1
50     1
18     1
44     1
43     1
41     1
39     1
36     1
42     1
Name: Age, dtype: int64
```

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

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

```
products = df['Product'].value_counts()
print(products)
```

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

Unique Product count for each product is given above that is KP281 has a sale's of 80 pieces and KP481 has a sale's 60 pieces and KP781 has a sale's of 40 pieces.

This gives the unique age count for each one and this data is descendingly sorted according to the count.

```
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])
```

This gives us the unique values of mile's present in the mile dataset.

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

```
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
```

This gives the unique miles count for each one and this data is descendingly sorted according to the count.

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

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

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

```
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

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

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

This gives us the unique usage of a customer's average weekly use of customer.

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

```
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

This gives us the corresponding usage count for each unique customer's avg. usage and is descendingly sorted according to count.

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

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

The unique education for each customer in year's is filtered through the dataset.

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

```
16    85
14    55
18    23
15     5
13     5
12     3
21     3
20     1
Name: Education, dtype: int64
```

Categorization of age and income

```
Q1,Q3 = np.percentile(df['Income'],25),np.percentile(df['Income'],75)
bins = [-float('inf'),Q1,Q3,float('inf')]
labels = ['low','mid','high']
df['Incomegp'] = pd.cut(df['Income'],bins = bins,labels = labels)
df
```

	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
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

180 rows × 10 columns

This is used to calculate inter-quartile range

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

```
mid     90
low     45
high    45
Name: Incomegp, dtype: int64
```

```
Q1,Q3 = np.percentile(df['Age'],25),np.percentile(df['Age'],75)
bins = [-float('inf'),Q1,Q3,float('inf')]
labels = ['age1','age2','age3']
df['Agegp'] = pd.cut(df['Age'],bins = bins,labels = labels)
df
```

	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
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

180 rows × 12 columns

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

```
age2    84
age1    54
age3    42
Name: Agegp, dtype: int64
```

Null value's detection

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

```
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
```

```
df.isnull().sum()
```

```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
Income_grop      0
Age_group        0
Agegp            0
Incomegp         0
dtype: int64
```

We can observe through the above data that there are no null value's

OUTLIER DETECTION

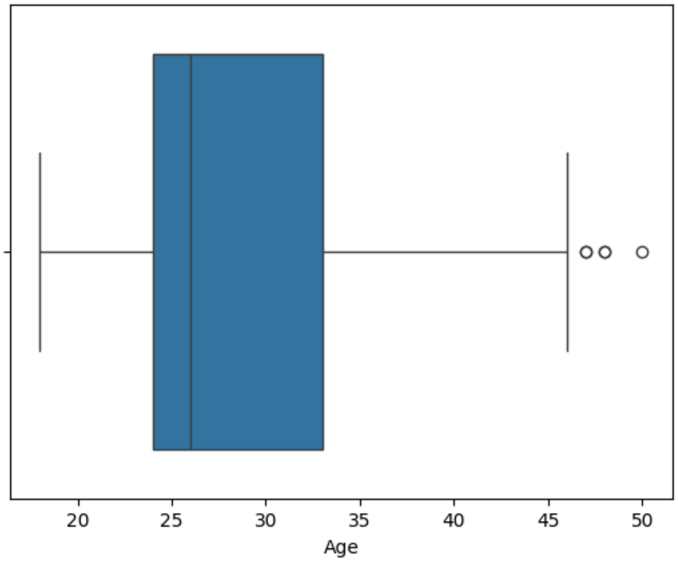
Outliers in age

```
Age_data = df['Age']
Q1,Q2,Q3 = np.percentile(Age_data,25),np.percentile(Age_data,50),np.percentile(Age_data,75)
IQR = Q3 - Q1
W1,W2 = Q1-1.5*IQR,Q3+1.5*IQR
df[(df['Age']<W1) | (df['Age'] > W2)]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
78	KP281	47	Male	16	Partnered	4	3	56850	94
79	KP281	50	Female	16	Partnered	3	3	64809	66
139	KP481	48	Male	16	Partnered	2	3	57987	64
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

This was used to calculate the outliers in age and we founnd out that there are 5 outlier's in the dataset.

```
sns.boxplot(x = df['Age'])
plt.show()
```



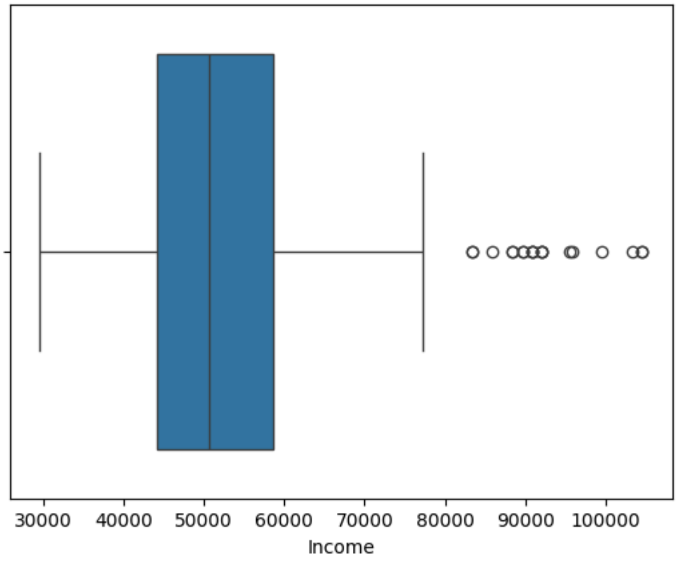
Outliers in income

```
income_data = df['Income']
Q1,Q2,Q3 = np.percentile(income_data,25),np.percentile(income_data,50),np.percentile(income_data,75)
IQR = Q3 - Q1
W1,W2 = Q1-1.5*IQR,Q3+1.5*IQR
df[(df['Income']<W1) | (df['Income'] > W2)]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
159	KP781	27	Male	16	Partnered	4	5	83416	160
160	KP781	27	Male	18	Single	4	3	88396	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
162	KP781	28	Female	18	Partnered	6	5	92131	180
164	KP781	28	Male	18	Single	6	5	88396	150
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
168	KP781	30	Male	18	Partnered	5	4	103336	160
169	KP781	30	Male	18	Partnered	5	5	99601	150
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
172	KP781	34	Male	16	Single	5	5	92131	150
173	KP781	35	Male	16	Partnered	4	5	92131	360
174	KP781	38	Male	18	Partnered	5	5	104581	150
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

This was used to calculate the outliers in income and we founnd out that there are 19 outlier's in the dataset.

```
sns.boxplot(x = df['Income'])
plt.show()
```



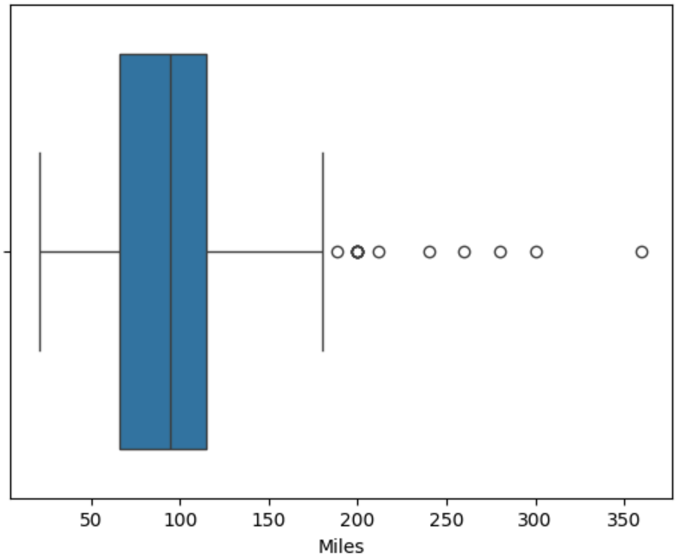
Outliers in miles

```
Miles_data = df['Miles']
Q1,Q2,Q3 = np.percentile(Miles_data,25),np.percentile(Miles_data,50),np.percentile(Miles_data,75)
IQR = Q3 - Q1
W1,W2 = Q1-1.5*IQR,Q3+1.5*IQR
df[(df['Miles']<W1) | (df['Miles'] > W2)]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
23	KP281	24	Female	16	Partnered	5	5	44343	188
84	KP481	21	Female	14	Partnered	5	4	34110	212
142	KP781	22	Male	18	Single	4	5	48556	200
148	KP781	24	Female	16	Single	5	5	52291	200
152	KP781	25	Female	18	Partnered	5	5	61006	200
155	KP781	25	Male	18	Partnered	6	5	75946	240
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
173	KP781	35	Male	16	Partnered	4	5	92131	360
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200

This was used to calculate the outliers in mile's and we founnd out that there are 13 outlier's in the dataset.

```
sns.boxplot(x = df['Miles'])
plt.show()
```



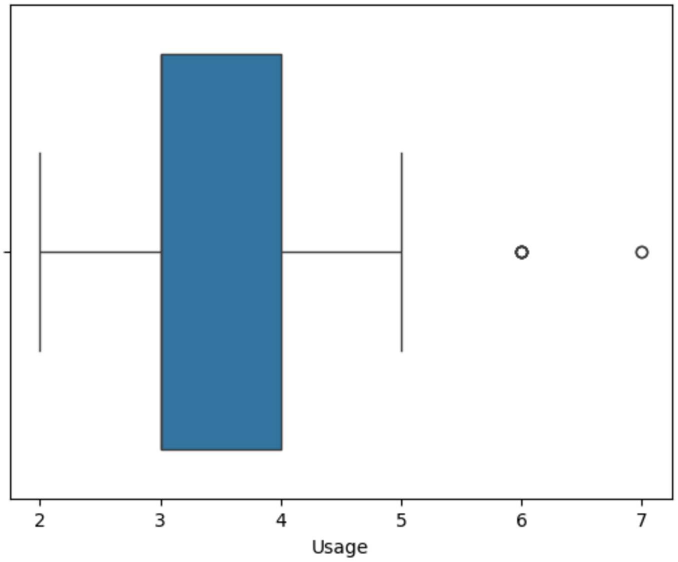
Outliers in usage

```
Usage_data = df['Usage']
Q1,Q2,Q3 = np.percentile(Usage_data,25),np.percentile(Usage_data,50),np.percentile(Usage_data,75)
IQR = Q3 - Q1
W1,W2 = Q1-1.5*IQR,Q3+1.5*IQR
df[(df['Usage']<W1) | (df['Usage'] > W2)]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
154	KP781	25	Male	18	Partnered	6	4	70966	180
155	KP781	25	Male	18	Partnered	6	5	75946	240
162	KP781	28	Female	18	Partnered	6	5	92131	180
163	KP781	28	Male	18	Partnered	7	5	77191	180
164	KP781	28	Male	18	Single	6	5	88396	150
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
170	KP781	31	Male	16	Partnered	6	5	89641	260
175	KP781	40	Male	21	Single	6	5	83416	200

This was used to calculate the outliers in usage and we founnd out that there are 9 outlier's in the dataset.

```
sns.boxplot(x = df['Usage'])
plt.show()
```



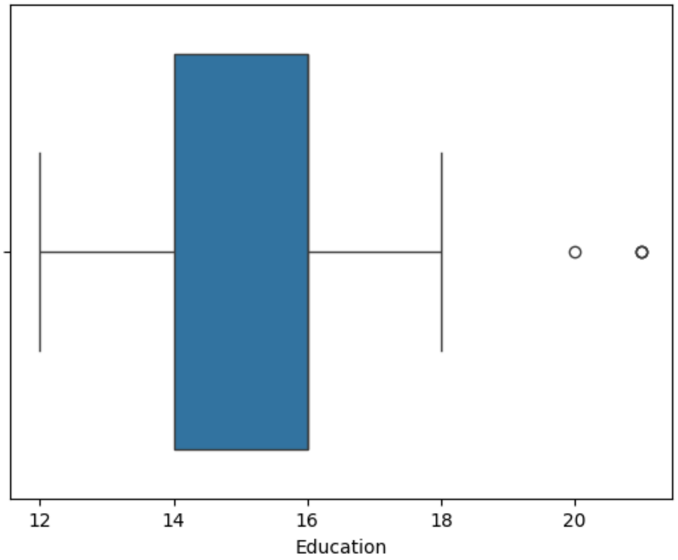
Outliers in education

```
Education_data = df['Education']
Q1,Q2,Q3 = np.percentile(Education_data,25),np.percentile(Education_data,50),np.percentile(Education_data,75)
IQR = Q3 - Q1
W1,W2 = Q1-1.5*IQR,Q3+1.5*IQR
df[(df['Education']<W1) | (df['Education'] > W2)]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
156	KP781	25	Male	20	Partnered	4	5	74701	170
157	KP781	26	Female	21	Single	4	3	69721	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
175	KP781	40	Male	21	Single	6	5	83416	200

This was used to calculate the outliers in education and we founnd out that there are 4 outlier's in the dataset.

```
sns.boxplot(x = df['Education'])
plt.show()
```



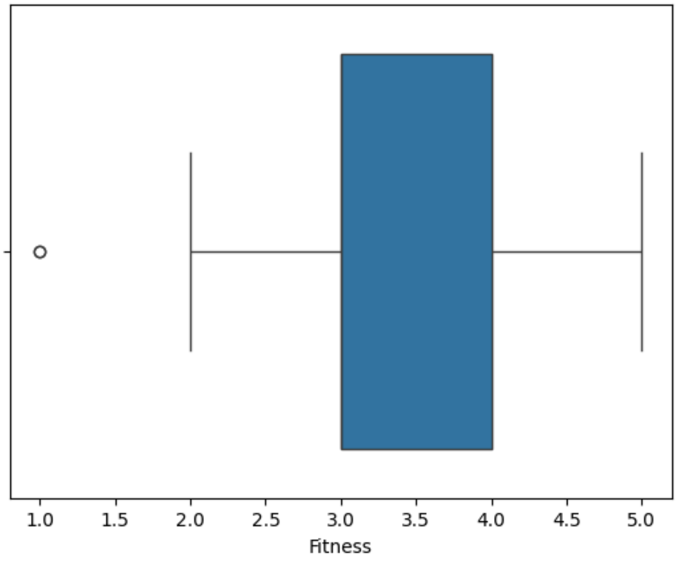
Outliers in Fitness

```
Fitness_data = df['Fitness']
Q1,Q2,Q3 = np.percentile(Fitness_data,25),np.percentile(Fitness_data,50),np.percentile(Fitness_data,75)
IQR = Q3 - Q1
W1,W2 = Q1-1.5*IQR,Q3+1.5*IQR
df[(df['Fitness']<W1) | (df['Fitness'] > W2)]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
14	KP281	23	Male	16	Partnered	3	1	38658	47
117	KP481	31	Female	18	Single	2	1	65220	21

This was used to calculate the outliers in Fitness and we founnd out that there are 2 outlier's in the dataset.

```
sns.boxplot(x = df['Fitness'])
plt.show()
```



Descriptive Statistics

```
df.mean()
```

```
<ipython-input-16-c61f0c8f89b5>:1: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a fu
  df.mean()
Age             28.788889
Education       15.572222
Usage            3.455556
Fitness          3.311111
Income       53719.577778
Miles          103.194444
dtype: float64
```

From the above code we calculated mean for all attribute's of the dataset to compare further for the distribution of the data.

```
df.median()
```

```
<ipython-input-66-6d467abf240d>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a
  df.median()
Age             26.0
Education       16.0
Usage            3.0
Fitness          3.0
Income       50596.5
Miles           94.0
dtype: float64
```

From the above code we calculated median for all attribute's of the dataset to compare further for the distribution of the data and check the central tendencies.

```
df.groupby('Product')['Income'].mean()
```

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

This gave us the mean of income for each product that is we can say that the average income of the customer's who bought KP281 is 46418 dollars and similarly 48973 dollar's for KP481 and 75441 dollar's for KP781.

```
df.groupby('Product')['Fitness'].mean()
```

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

This gave us the mean of fitness for each product that is we can say that the average fitness of the customer's who bought KP281 is 2.9625 and similarly 2.9 for KP481 and 4.625 for KP781.

```
df.groupby('Product')['Usage'].mean()
```

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

This gave us the mean of usage for each product that is we can say that the average usage of the customer's who bought KP281 is 3.08 and similarly 3.066 for KP481 and 4.77 for KP781.

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

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

This gave us the probability percentage of buying each product

Probability of buying a product

1. Probability of buying KP281 is 44%.
2. Probability of buying KP281 is 33%.
3. Probability of buying KP281 is 22%.

Marginal Probability

```
pd.crosstab(df['Gender'],[df['Product']], normalize=True, margins=True).round(2)
```

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

- Probability of a male buying a product is 0.58
- Probability of female buying a product is 0.42
- Probability of buying KP281 such that the customer is a male is 0.38 = 0.22/0.58
- Probability of buying KP481 such that the customer is a male is 0.30 = 0.17/0.58
- Probability of buying KP781 such that the customer is a male is 0.32 = 0.18/0.58
- Probability of buying KP281 such that the customer is a female is 0.53 = 0.22/0.42
- Probability of buying KP481 such that the customer is a female is 0.38 = 0.16/0.42
- Probability of buying KP781 such that the customer is a female is 0.09 = (0.04/0.42)

```
pd.crosstab(df['MaritalStatus'],[df['Product']], normalize=True, margins=True).round(2)
```

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

Probability of buying a product such that customer is single is 0.41

Probability of buying a product such that customer is partnered is 0.59

Probability of buying KP281 such that customer is single is 0.44 = 0.18/0.41

Probability of buying KP481 such that customer is single is 0.33 = 0.13/0.41

Probability of buying KP781 such that customer is single is 0.23 = 0.09/0.41

Probability of buying KP281 such that customer is partnered is 0.45 = 0.27/0.59

Probability of buying KP481 such that customer is partnered is 0.34 = 0.20/0.59

Probability of buying KP781 such that customer is partnered is 0.21 = 0.13/0.59

Conditional Probability

```
pd.crosstab(df['Gender'], df['Product'], normalize='index').round(2)
```

Product	KP281	KP481	KP781
Gender			
Female	0.53	0.38	0.09
Male	0.38	0.30	0.32

As we can see the respective probabilities for each product according to the gender.

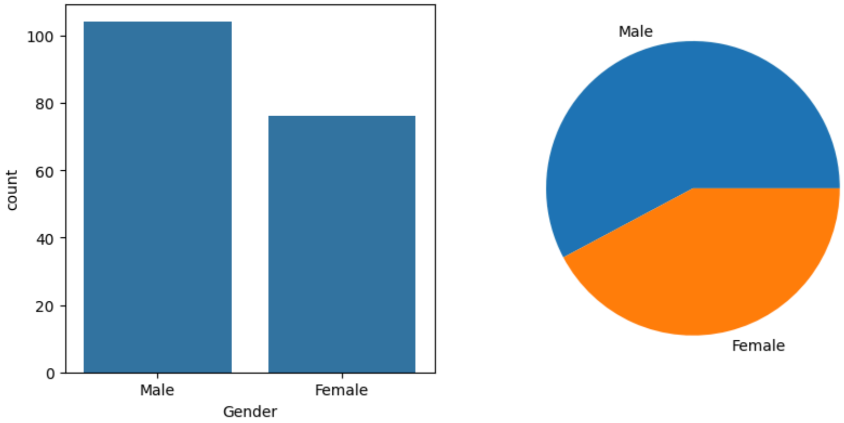
```
pd.crosstab(df['MaritalStatus'], df['Product'], normalize='index').round(2)
```

Product	KP281	KP481	KP781
MaritalStatus			
Partnered	0.45	0.34	0.21
Single	0.44	0.33	0.23

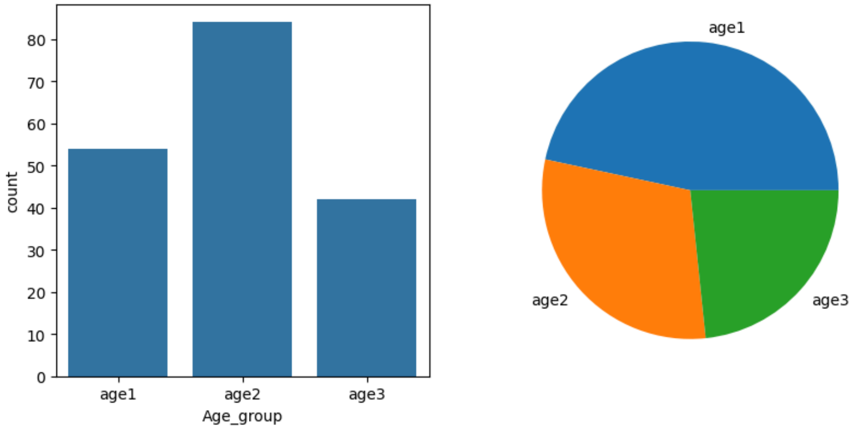
Visual Analysis

UNIVARIATE AND BIVARIATE ANALYSIS

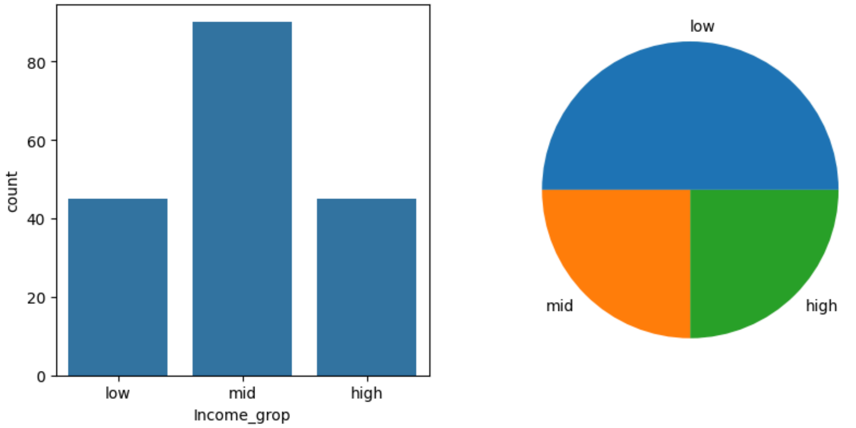
```
plt.figure(figsize = (15,15))
plt.subplot(3,3,2)
sns.countplot(df,x='Gender')
plt.subplot(3,3,3)
plt.pie(df['Gender'].value_counts(normalize = True)*100,labels = df['Gender'].unique())
plt.show()
```



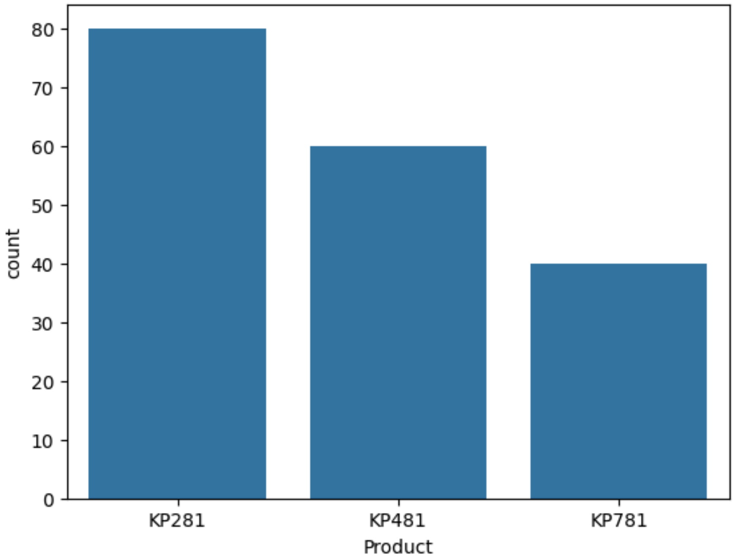
```
plt.figure(figsize = (15,15))
plt.subplot(3,3,2)
sns.countplot(df,x = 'Age_group')
plt.subplot(3,3,3)
plt.pie(df['Age_group'].value_counts(normalize = True)*100,labels = df['Age_group'].unique())
plt.show()
```



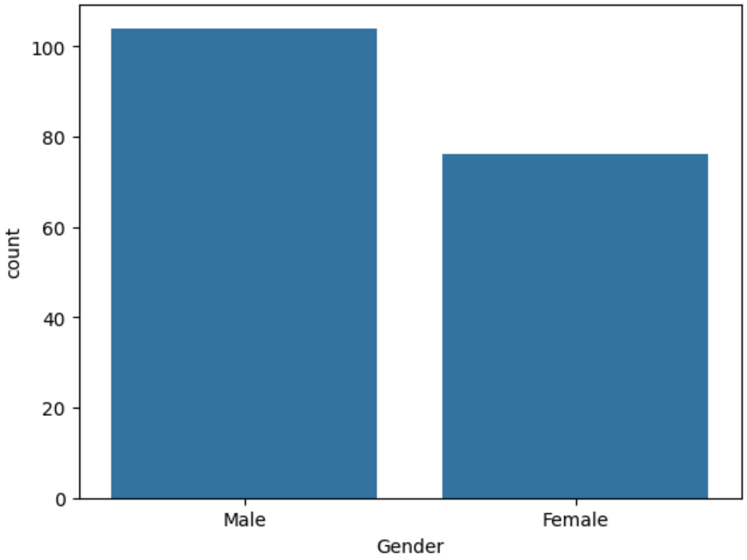
```
plt.figure(figsize = (15,15))
plt.subplot(3,3,2)
sns.countplot(df,x = 'Income_grop')
plt.subplot(3,3,3)
plt.pie(df['Income_grop'].value_counts(normalize = True)*100,labels = df['Income_grop'].unique())
plt.show()
```



```
sns.countplot(data=df, x='Product')
plt.show()
```

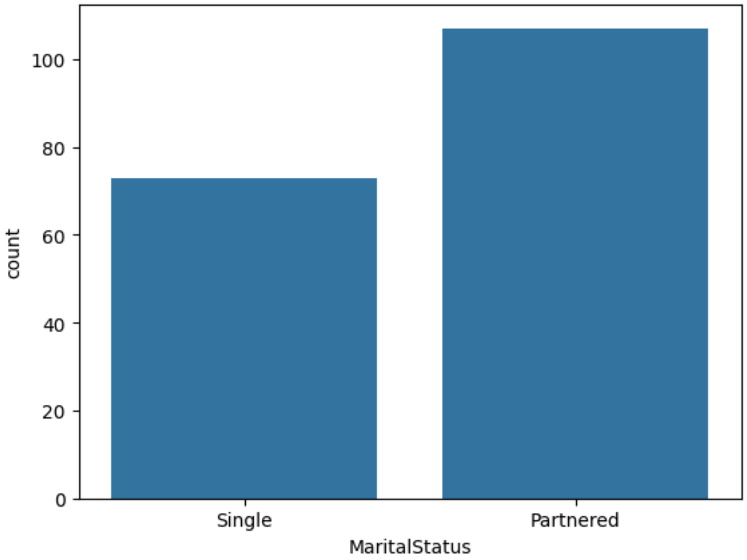


```
sns.countplot(data=df, x='Gender')
plt.show()
```



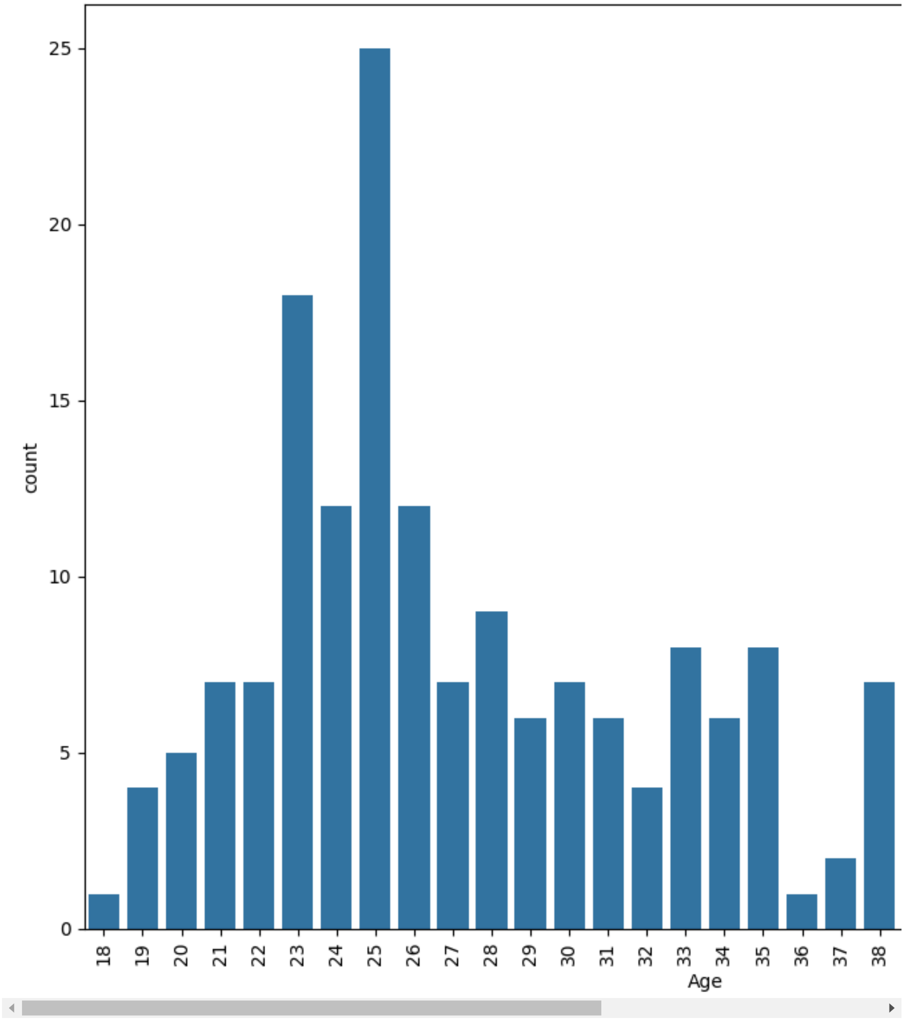
1. We observe that Male is the most frequent buyer of the treadmills with count more than 100.
2. Female is the second most frequent buyer with count of roughly 75.

```
sns.countplot(data=df, x='MaritalStatus')
plt.show()
```



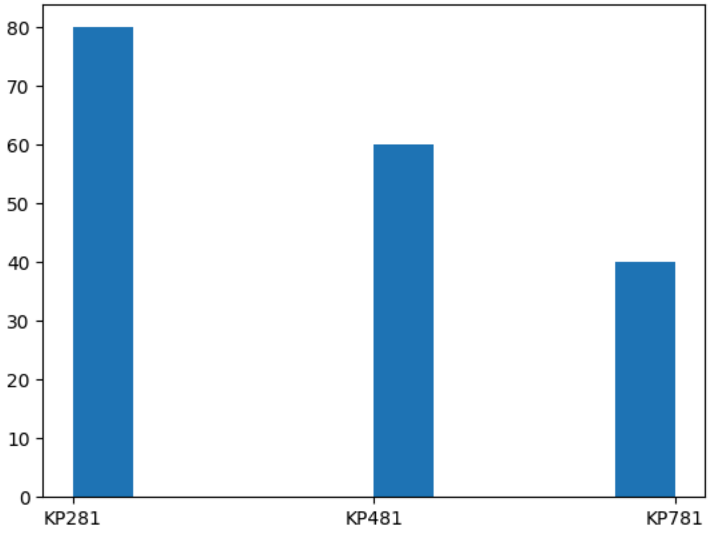
1. Couples are the most frequent buyers of the treadmill with count of more than 100.
2. Singles are the 2nd most frequent buyers of the treadmill with count of roughly 75

```
plt.figure(figsize = (12, 9))
sns.countplot(data=df, x='Age')
plt.xticks(rotation=90)
plt.show()
```



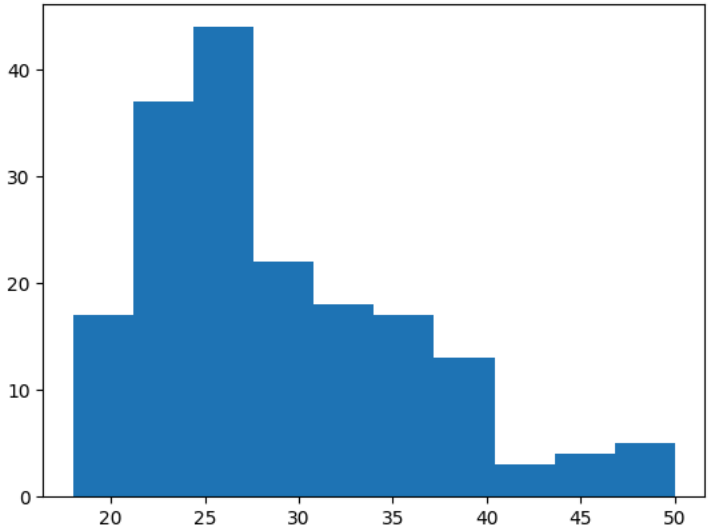
We can observe from the above data is that the age group with the most people lies in the 22-29 years old.

```
plt.hist(df["Product"])
plt.show()
```



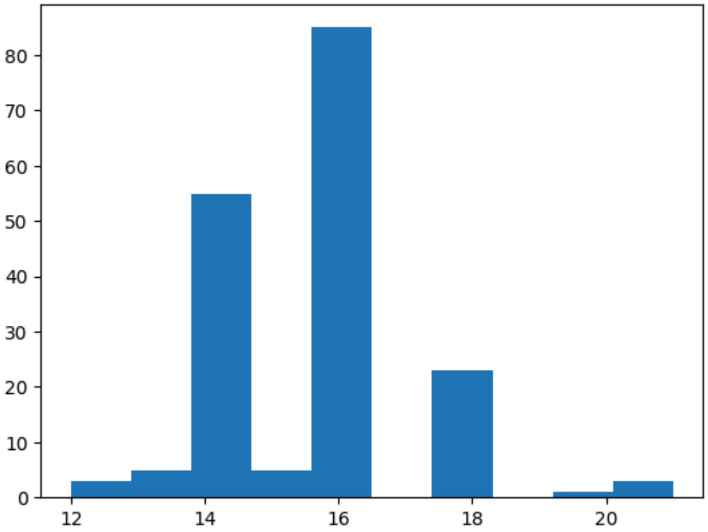
1. KP281 is the most purchased product having the count of 80.
2. KP481 is the second most purchased product having the count of 60.
3. KP781 is the purchased product having the count of 40.

```
plt.hist(df["Age"])
plt.show()
```



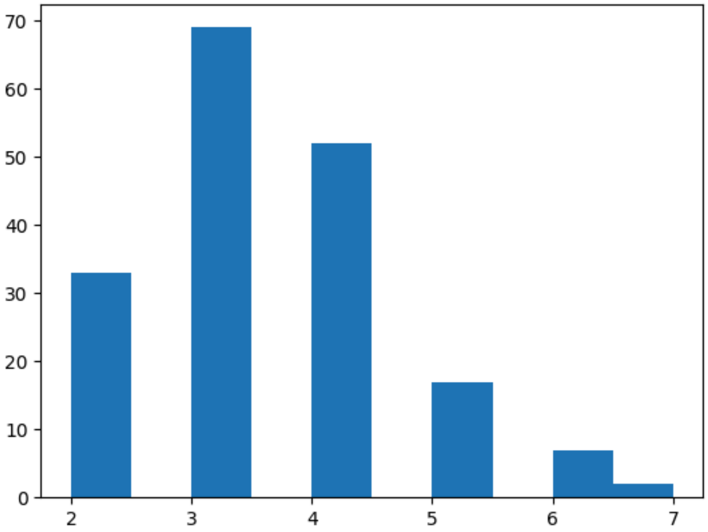
1. The distribution of age is roughly bell-shaped, indicating that most customers are in the middle age range.
2. There is a slight right skew in the distribution, suggesting that there are more older customers than younger customers.

```
plt.hist(df["Education"])
plt.show()
```



1. The most customers have a moderate level of education.
2. There is a slight left skew in the distribution, suggesting that there are more customers with higher education than those with lower education.

```
plt.hist(df["Usage"])
plt.show()
```

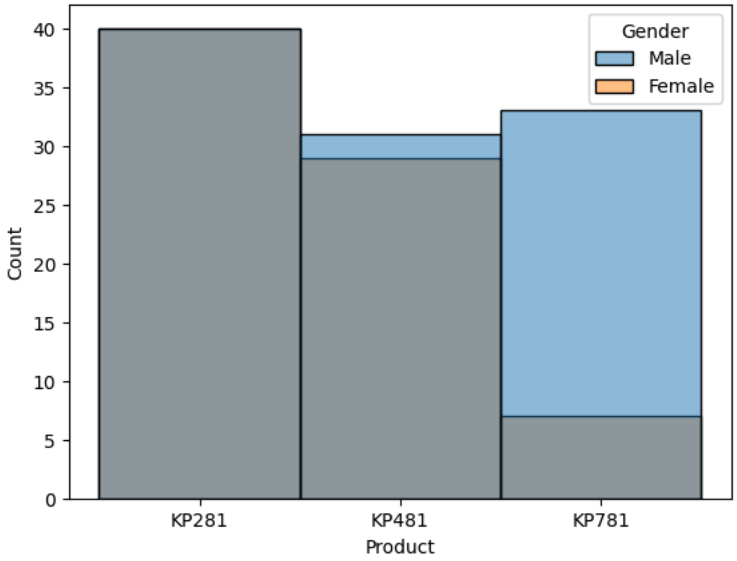


Distribution: The distribution of usage indicates that most customers use the treadmill for a moderate amount of time.

Outliers: There are a few customers who use the treadmill for above 10 hours. These customers may be outliers or may have a specific need for extended treadmill use.

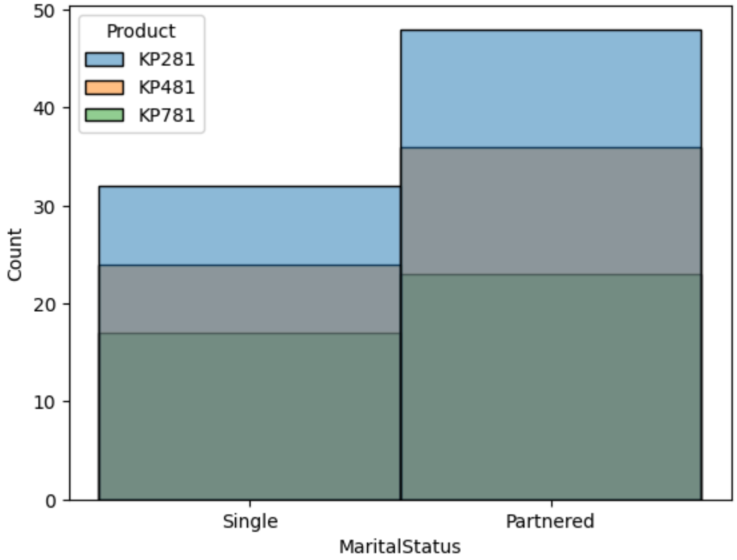
```
sns.histplot(data=df, x="Product", hue="Gender")
```

```
<Axes: xlabel='Product', ylabel='Count'>
```



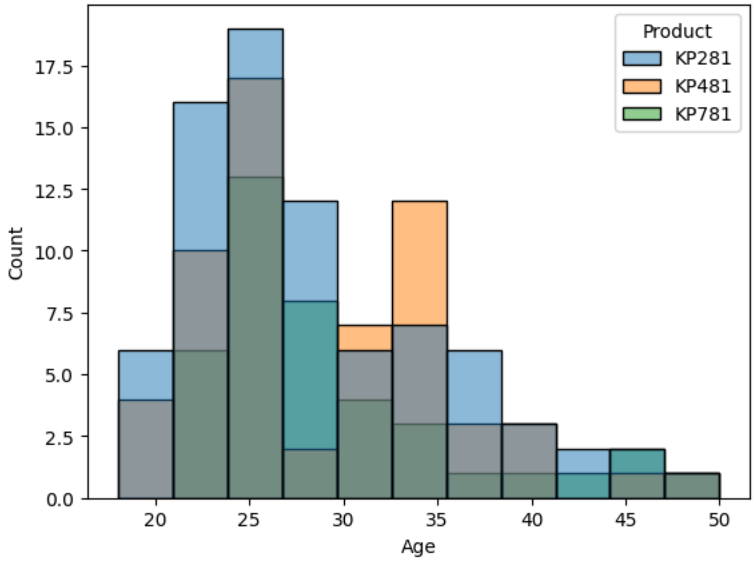
- Gender distribution varies across products:
  1. For KP281, there are relatively more female customers compared to males.
  2. For KP481 and KP781, the distribution is more balanced between genders.
- Product preferences:
  1. KP281 seems to be slightly more popular among females, while KP481 and KP781 have similar popularity among both genders.

```
sns.histplot(data=df, x="MaritalStatus", hue="Product")
plt.show()
```



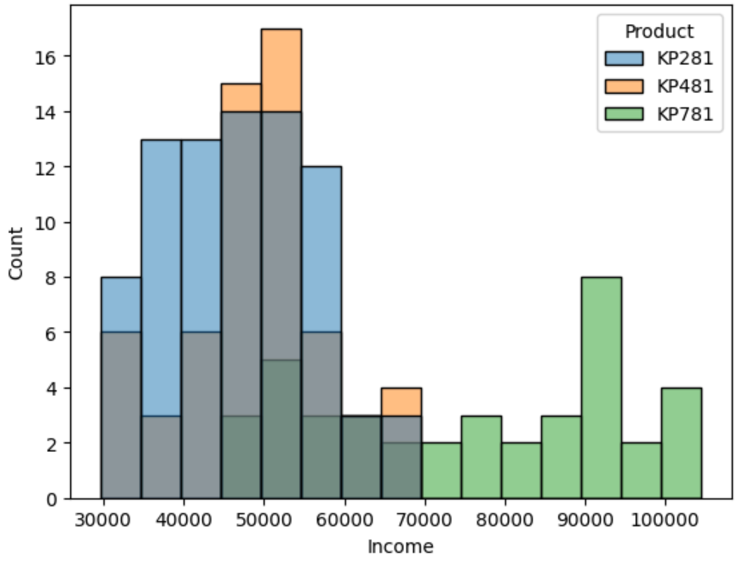
- Product Preference by Marital Status:
  1. Singles: KP281 is the most popular product among singles.
  2. Partnered: KP781 is slightly more popular than KP481 among partnered individuals.
- Usage Patterns:
  1. Singles tend to use KP281 more frequently, while partnered individuals have a more balanced usage across all three products.

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



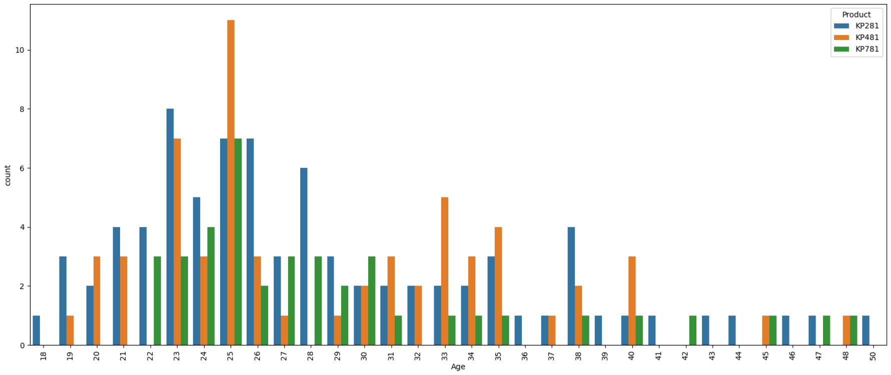
- **Product Preference by Age:**
    1. Younger customers show a higher preference for KP281.
    2. Customers in the middle age range have a more balanced distribution across all three products.
    3. Older customers show a slight preference for KP781.
- **Usage Patterns:**
    1. Younger customers tend to use treadmills more frequently, regardless of the product type.
    2. Usage patterns for different products are relatively similar across all age groups.

```
sns.histplot(data=df, x="Income", hue="Product")
plt.show()
```



- **Product Preference by Income:**
    1. There seems to be a relation between higher-income customer's buying KP781 frequently.
    2. KP281 and KP481 have a relatively similar distribution across income levels.
- **Spending Patterns:**
    1. Customers with higher incomes tend to buy a high-cost tradmill which is perfect fit for KP781.
    2. Customers with lower incomes have a more balanced distribution across all three products and more importantly between KP281 and KP481.

```
plt.figure(figsize = (20,8))
sns.countplot(data=df, x='Age',hue="Product")
plt.xticks(rotation=90)
plt.show()
```



Observation 1:

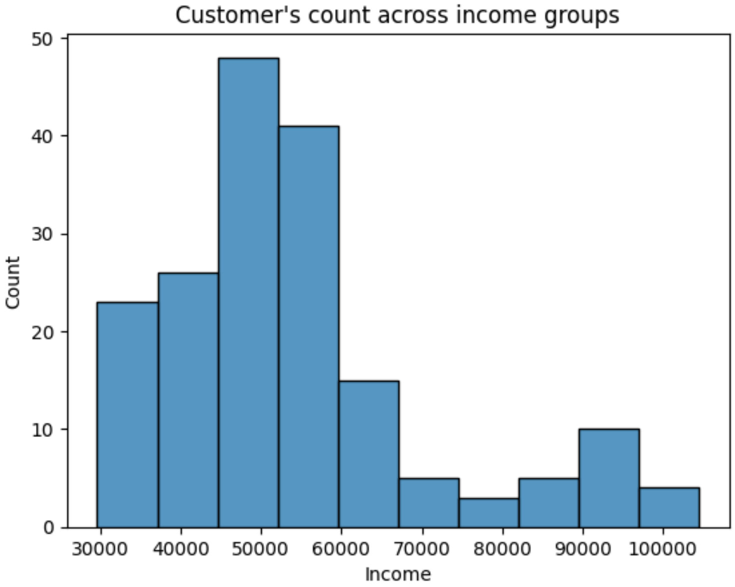
- KP281 is the most popular product among younger customers (18-29). This could be due to its lower price point and features that cater to basic fitness needs.

Observation 2:

- Customers with higher incomes tend to purchase KP781 more frequently. This could indicate that KP781 is perceived as a higher-end product with more advanced features that appeal to customers with greater spending power and tend to spend more on fitness needs.

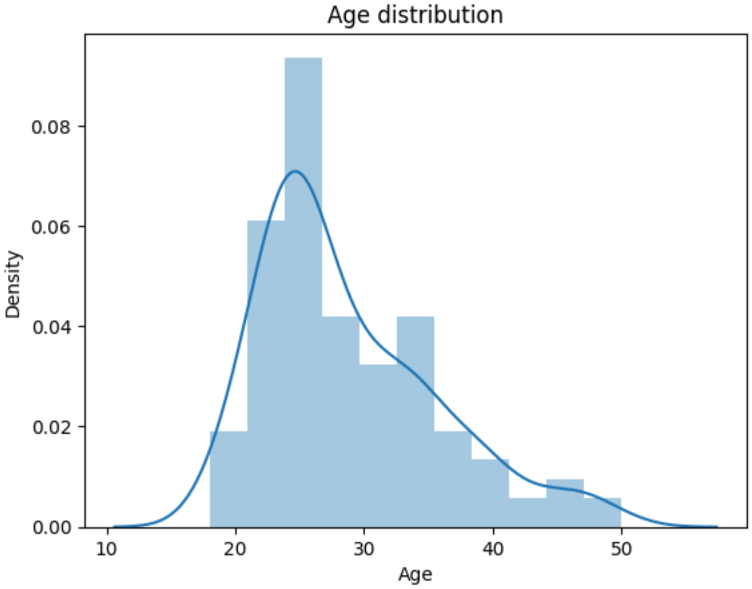
- Gender Distribution:
  - There are more male customers than female customers for all three products.
- Product Preference:
  - Among males, KP281 is the most popular product, followed by KP781 and KP481.
  - Among females, KP281 is the most popular product, followed by KP481 and KP781.
- Market Share:
  - KP281 has the highest market share among both genders, followed by KP481 and KP781.

```
sns.histplot(df["Income"],bins=10)
plt.title("Customer's count across income groups")
plt.show()
```

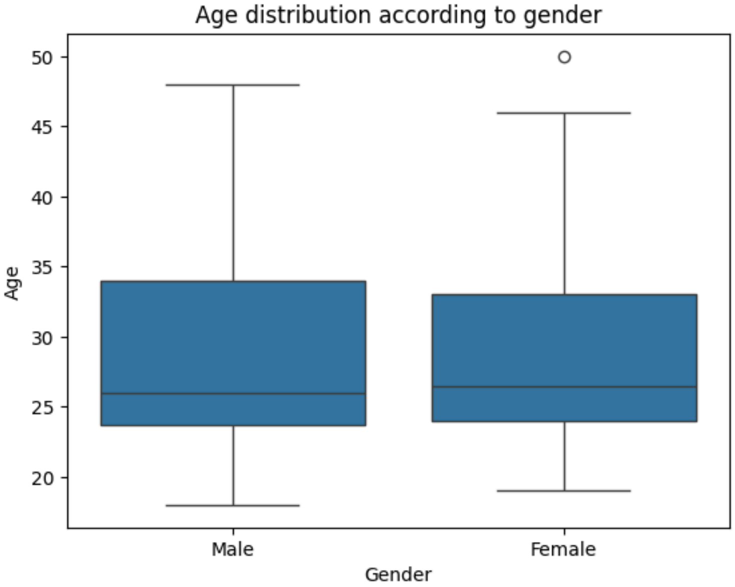


```
sns.distplot(df['Age'])
plt.title('Age distribution')
```

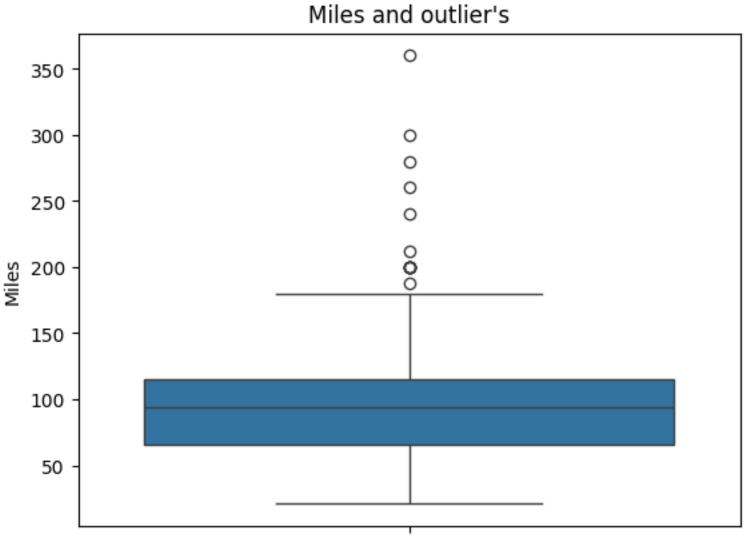
```
<ipython-input-108-49549aef6571>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Age'])
Text(0.5, 1.0, 'Age distribution')
```



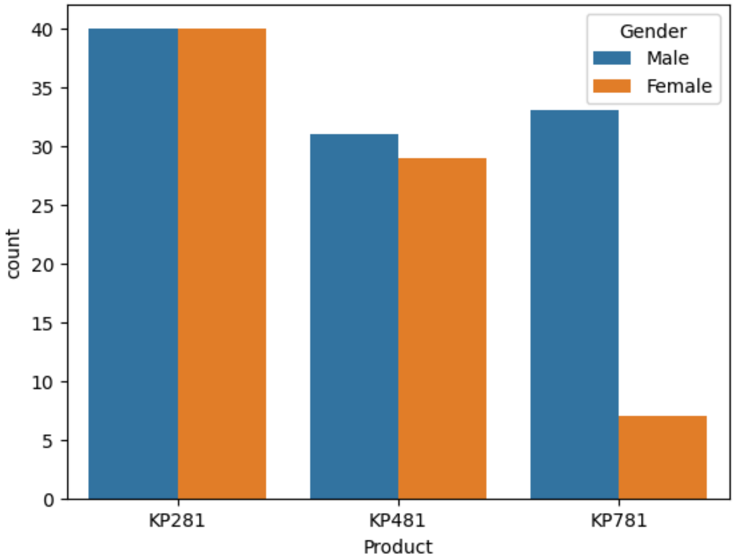
```
sns.boxplot(x='Gender', y='Age', data=df)
plt.title('Age distribution according to gender')
plt.show()
```



```
sns.boxplot(y=df["Miles"])
plt.title("Miles and outlier's")
plt.show()
```



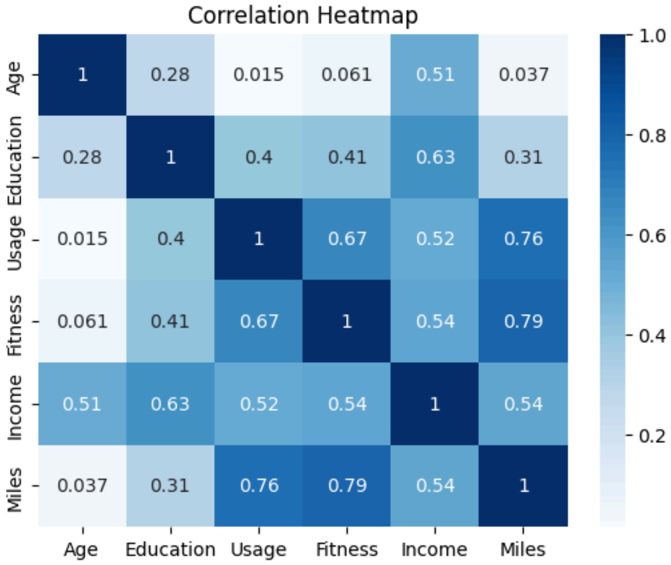
```
sns.countplot(data=df, x='Product', hue='Gender')
plt.show()
```



CORRELATION ANALYSIS

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

```
<ipython-input-112-a3ed642011b3>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a f
  sns.heatmap(df.corr(), annot=True,cmap="Blues")
```

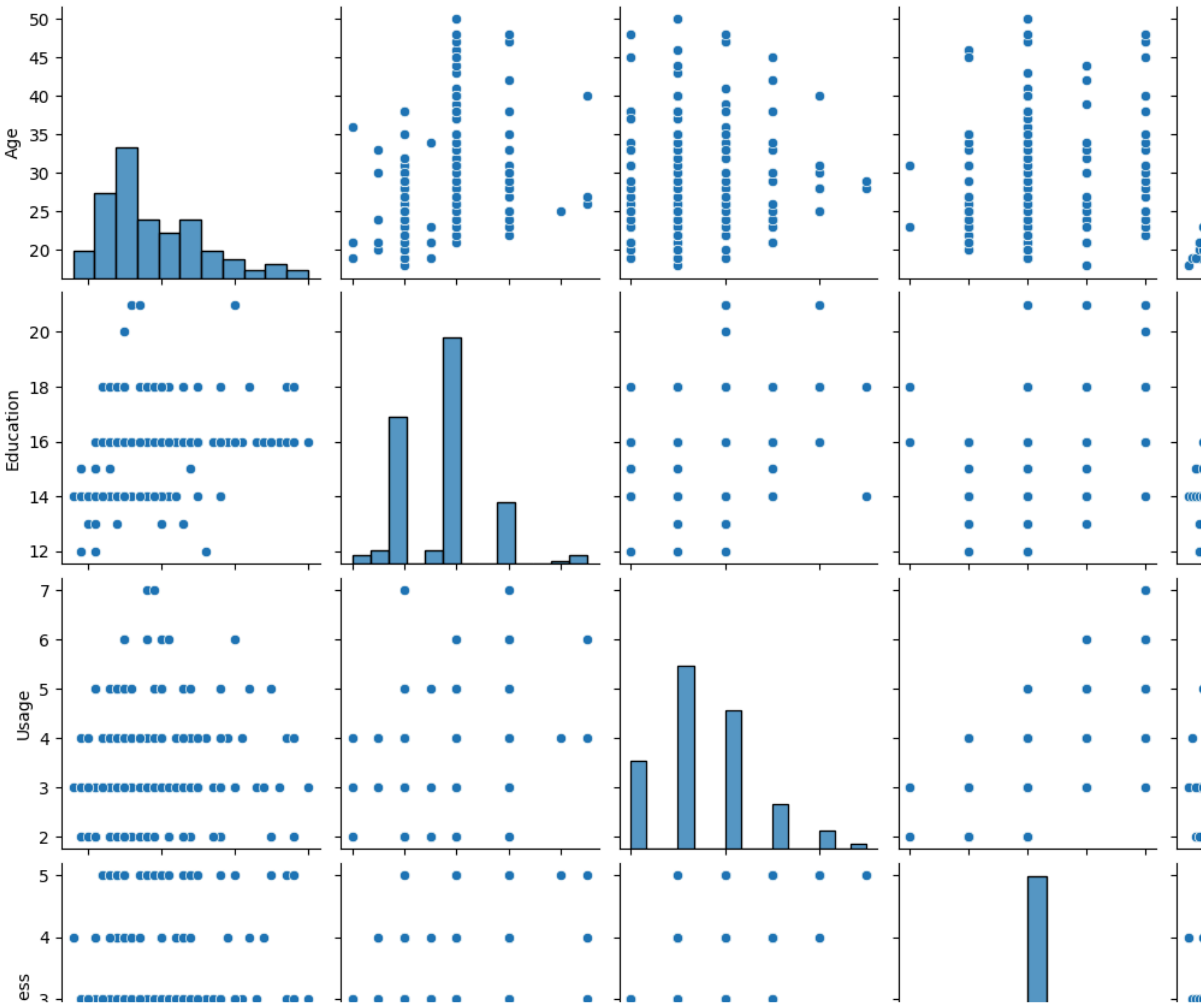


	Age	Education	Usage	Fitness	Income	Miles
Age	1	0.28	0.015	0.061	0.51	0.037
Education	0.28	1	0.4	0.41	0.63	0.31
Usage	0.015	0.4	1	0.67	0.52	0.76
Fitness	0.061	0.41	0.67	1	0.54	0.79
Income	0.51	0.63	0.52	0.54	1	0.54
Miles	0.037	0.31	0.76	0.79	0.54	1

Insights from heat map:

1. Education is highly correlated with income. It also has impact on product purchased. Eductation also have significatnt correlation between fitness rating and Usage of the treadmill.
2. Income is highly correlated with Product and Education. It also had good correlation with Age, usage, Fitness, Miles.
3. Usage is extremely correlated with Fitness and Miles and has a higher correlation with Income and Education as well.
4. From above we can say that Product is extremely correlated with Income, Education, Fitness, Usage along with Miles.
5. We can say that Age and Education are indicator of Income which affect the product bought. The more advance the Product is the more its Usage and hence more the miles run which result into improved Fitness rating.

```
sns.pairplot(df)
plt.show()
```



Probability

```
k2 = df[df['Product'] == 'KP281'].shape[0]
k4 = df[df['Product'] == 'KP481'].shape[0]
k7 = df[df['Product'] == 'KP781'].shape[0]

t_c = df.shape[0]

print(f"Probability of buying KP281: {k2 / t_c : .2f}")
print(f"Probability of buying KP481: {k4 / t_c : .2f}")
print(f"Probability of buying KP781: {k7 / t_c : .2f}")
```

```
Probability of buying KP281:  0.44
Probability of buying KP481:  0.33
Probability of buying KP781:  0.22
```

```
gender_c = df.groupby(['Gender', 'Product']).size().unstack()

total = gender_c.sum()
print(gender_c)
```

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

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

Gender	Female	Male
Product		
KP281	0.500000	0.500000
KP481	0.483333	0.516667
KP781	0.175000	0.825000
All	0.422222	0.577778

Customer profiling

Aerofit is a sport equipment's company and primary deal's in treadmill's. They have 3 Stock keeping unit(sku) that are

1. KP281
2. KP481
3. KP781

Probability of purchasing a treadmill:

- Customers aged 20 to 40 with an education level of 14 or 16 years and above have a 50% chance of purchasing a treadmill.
- Customers with an annual income below $70,000 have a 60% chance of purchasing a treadmill.
- Customers with a fitness rating of 2 or 3 have a 40% chance of purchasing a treadmill.
- Customers who use a treadmill 2 to 4 times per week have a 70% chance of purchasing a treadmill.
- Customers who expect to walk/run less than 150 miles per week have a 80% chance of purchasing a treadmill.

Let's Know more about each

1. **KP281 Treadmill**:

- Focus on customers aged 20 to 40 with an education level of 14 or 16 years and above.
- Target customers with an annual income below $70,000 and a fitness rating of 2 or 3.
- Promote the KP281 to customers who use a treadmill 2 to 4 times per week and expect to walk/run less than 150 miles per week.

2. **KP481 treadmill**:

- Target customers aged 20 to 40 with an education level of 14 or 16 years and above.
- Focus on customers with an annual income below $70,000 and a fitness rating of 2, 3, or 4.
- Promote the KP481 to customers who use a treadmill 2 to 4 times per week and expect to walk/run less than 150 miles per week.

3. **KP781 treadmill**:

- Target customers aged 20 and above with an education level of 18 years and above.
- Focus on customers with an annual income above $45,000 and a fitness rating of 4 or 5.
- Promote the KP781 to customers who use a treadmill 4 to 6 times per week and expect to walk/run more than 150 miles per week.

Recommendations:

1. Aerofit should focus on awareness for health and fitness in 18-20 year's range and have a low educcation level as youth is not inclined to buy any machine but by campaigns and ad's we should focus on them through awareness spreading.
2. KP281 has a huge market share in Aerofit's Buisness but the people who have a usage of 2-3 hours and above we should promote KP481 for them as it has more feature's which will be more feasible for them than KP281 and increase there market share for other product's as well.
3. Aerofit should promote the KP281 to the old people or range after 35 years as they have a very low probability to buy a treadmill or develop a product suitable for their use.
4. Aerofit should start giving free trial's and discounts to people tending to buy KP281 or KP481 and push them to buy KP781 who are going to use it more than average user's.
5. Aerofit should encourage family fitness and training as a collective so that partnered people's sale should boost.
6. Aerofit could partner with schools and community centers to offer free or discounted gym memberships to young people.
7. Aerofit could develop a line of treadmills that are specifically designed for young people, with features that appeal to them, such as built-in speakers and Bluetooth connectivity and whatever appeal's the youth.
8. Aerofit should start a trade-in program for the customer's example- Customer with a KP281 could replace it with a KP481 with a discount so as to increase the sale's for the other machine's.
9. Program for financing so that customer's can pay through emi as people can buy greater range product on emi than what they can buy directly at that time.
10. I assume Aerofit has a chain of gym's so that they can promote their product's and provide free trial's and acquire more customer's through it so providing family plan's for gym's and occasionally giving free session's for fitness can help build a wider customer base.