In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm

import warnings
warnings.filterwarnings("ignore")
```

Problem Statement

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.

Initial Data Exploration

In [2]:
```python
df = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/origir
```

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

Out[3]: (180, 9)

In [4]:
```python
df.head()
```

Out[4]:

	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

In [5]:
```python
df.tail()
```

Out[5]:

	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

In [6]:
```
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
```

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

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

There seems to be `Zero Null Count`

In [8]:
```
df.duplicated().sum()
```

Out[8]:
```
0
```

There are no duplicated records

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

Out[9]:

	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

Observations

Age:

1. Customers from 18 to 50 years of age use these Products.

2. Most of the Customers are of 24 to 33 years to old.

Education:

1. Customers that use these Products have 12 to 21 years of Education.
2. Most of the Customers had Education 12 to 16 years of Education.

Usage:

1. Customers try to use these Products 2 to 7 times a week.
2. Most of the Customers plan to use the Products either 3 or 4 times a week.

Fitness:

1. Customers using these Products have Fitness level 1-5, 5 being excellent and 1 being poor fitness.
2. Most of the Customers have 3-4 level of Fitness.

Income:

1. Customers using these Products have approx Income band of 30k to 105k.
2. Most of the Customers lie in the 44k to 59k Income band.

Miles

1. Customers using these Products expect to walk 21 to 360 Miles.
2. Most of the Customers expect to walk within 66 to 115 Miles.

In [10]:
```
df.describe(include = 'object')
```

Out[10]:

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

Observations

1. KP281 is the highest used product
2. Male Customers are more compared to Female
3. Partnered Customers are more compared to Single Customers

Non-Graphical Analysis: Value counts and unique attributes

In [11]:
```
cols_list = ['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
        'Fitness', 'Income', 'Miles']
```

In [12]:

```python
# this function is to bold python output
def bold_text(text):
    bold_start = '\033[1m'
    bold_end = '\033[0m'
    return bold_start + text + bold_end

def value_counts_new(d,column_name):
    d = d[column_name].value_counts().reset_index()
    d.columns = 'index',column_name
    dum = d.sort_values(by=[column_name,'index'],ascending = [False,True]).set_index('index')
    dum.index.name = None
    dum = pd.Series(dum[column_name],index =dum.index )

    return dum
```

In [13]:

```
for i in cols_list:
    print(bold_text(i.upper()+':'))
    print(f'Number of unique elements in {i} is:\n {df[i].nunique()}\n')
    print(f'Unique elements present in {i} column is:\n {np.sort(df[i].unique())}\n')
    print(f'Value Counts of {i} columns is:\n{value_counts_new(df,i)}\n\n\n')
```

PRODUCT:

```
Number of unique elements in Product is:
 3

Unique elements present in Product column is:
 ['KP281' 'KP481' 'KP781']

Value Counts of Product columns is:
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

AGE:

```
Number of unique elements in Age is:
 32

Unique elements present in Age column is:
 [18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
 42 43 44 45 46 47 48 50]

Value Counts of Age columns is:
25    25
23    18
24    12
26    12
28     9
33     8
35     8
21     7
22     7
27     7
30     7
38     7
29     6
31     6
34     6
20     5
40     5
19     4
32     4
37     2
45     2
47     2
48     2
18     1
36     1
39     1
41     1
42     1
43     1
44     1
46     1
50     1
Name: Age, dtype: int64
```

GENDER:

```
Number of unique elements in Gender is:
 2

Unique elements present in Gender column is:
 ['Female' 'Male']
```

```
Value Counts of Gender columns is:
Male      104
Female     76
Name: Gender, dtype: int64
```

EDUCATION:

```
Number of unique elements in Education is:
 8

Unique elements present in Education column is:
 [12 13 14 15 16 18 20 21]

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

MARITALSTATUS:

```
Number of unique elements in MaritalStatus is:
 2

Unique elements present in MaritalStatus column is:
 ['Partnered' 'Single']

Value Counts of MaritalStatus columns is:
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

USAGE:

```
Number of unique elements in Usage is:
 6

Unique elements present in Usage column is:
 [2 3 4 5 6 7]

Value Counts of Usage columns is:
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

FITNESS:

```
Number of unique elements in Fitness is:
 5

Unique elements present in Fitness column is:
 [1 2 3 4 5]
```

```
Value Counts of Fitness columns is:
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

INCOME:

```
Number of unique elements in Income is:
 62

Unique elements present in Income column is:
 [ 29562  30699  31836  32973  34110  35247  36384  37521  38658  39795
  40932  42069  43206  44343  45480  46617  47754  48556  48658  48891
  49801  50028  51165  52290  52291  52302  53439  53536  54576  54781
  55713  56850  57271  57987  58516  59124  60261  61006  61398  62251
  62535  64741  64809  65220  67083  68220  69721  70966  74701  75946
  77191  83416  85906  88396  89641  90886  92131  95508  95866  99601
 103336 104581]

Value Counts of Income columns is:
45480     14
52302      9
46617      8
53439      8
54576      8
          ..
85906      1
95508      1
95866      1
99601      1
103336     1
Name: Income, Length: 62, dtype: int64
```

MILES:

```
Number of unique elements in Miles is:
 37

Unique elements present in Miles column is:
 [ 21  38  42  47  53  56  64  66  74  75  80  85  94  95 100 103 106 112
 113 120 127 132 140 141 150 160 169 170 180 188 200 212 240 260 280 300
 360]

Value Counts of Miles columns is:
85     27
95     12
66     10
75     10
47      9
106     9
94      8
113     8
53      7
100     7
56      6
64      6
180     6
200     6
127     5
160     5
42      4
150     4
```

```
38      3
74      3
103     3
120     3
170     3
132     2
141     2
21      1
80      1
112     1
140     1
169     1
188     1
212     1
240     1
260     1
280     1
300     1
360     1
Name: Miles, dtype: int64
```

In [14]:
```
df['Product'].value_counts(normalize = True)
```

Out[14]:
```
Product
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: proportion, dtype: float64
```

Observations

Product:

1. Only Half of the Customers that use KP281 use KP781.
2. 4/9th, 3/9th, 2/9th are the number of records for KP281,KP481 and KP781 respectively.

Age:

1. 45% of Customers are early twenties

Education:

1. Most of the Customers had 16 years followed by 14 years of Education

Marital Status:

1. Most of the Customer that use these Products are Partnered

Usage:

1. Most of the Customer use the Product 3 to 4times a week

Fitness:

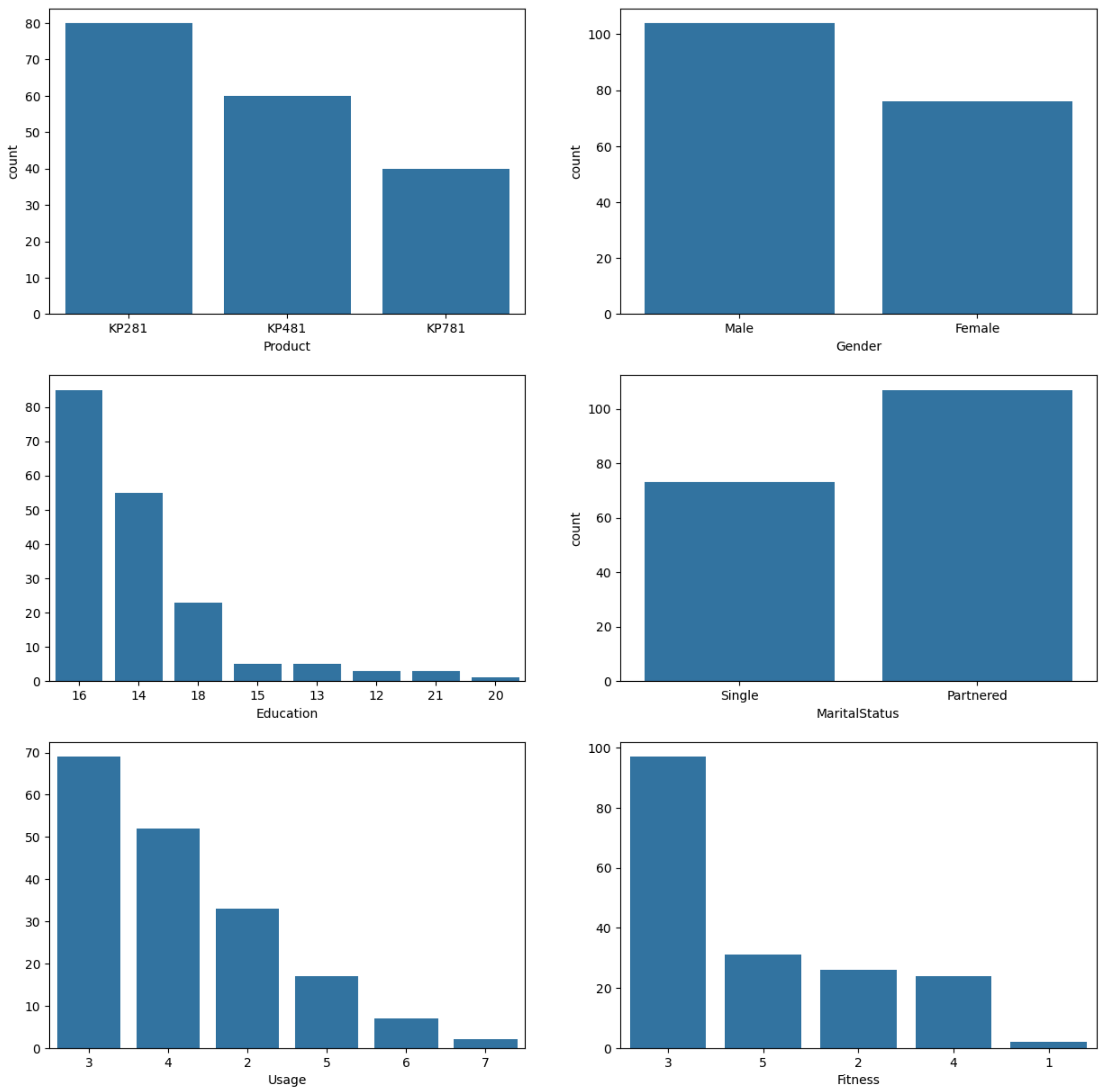
1. Most of the Customers are of average Fitness Level
2. 1/6th of the Customers in this dataset are in excellent shape

Visual Analysis - Univariate & Bivariate

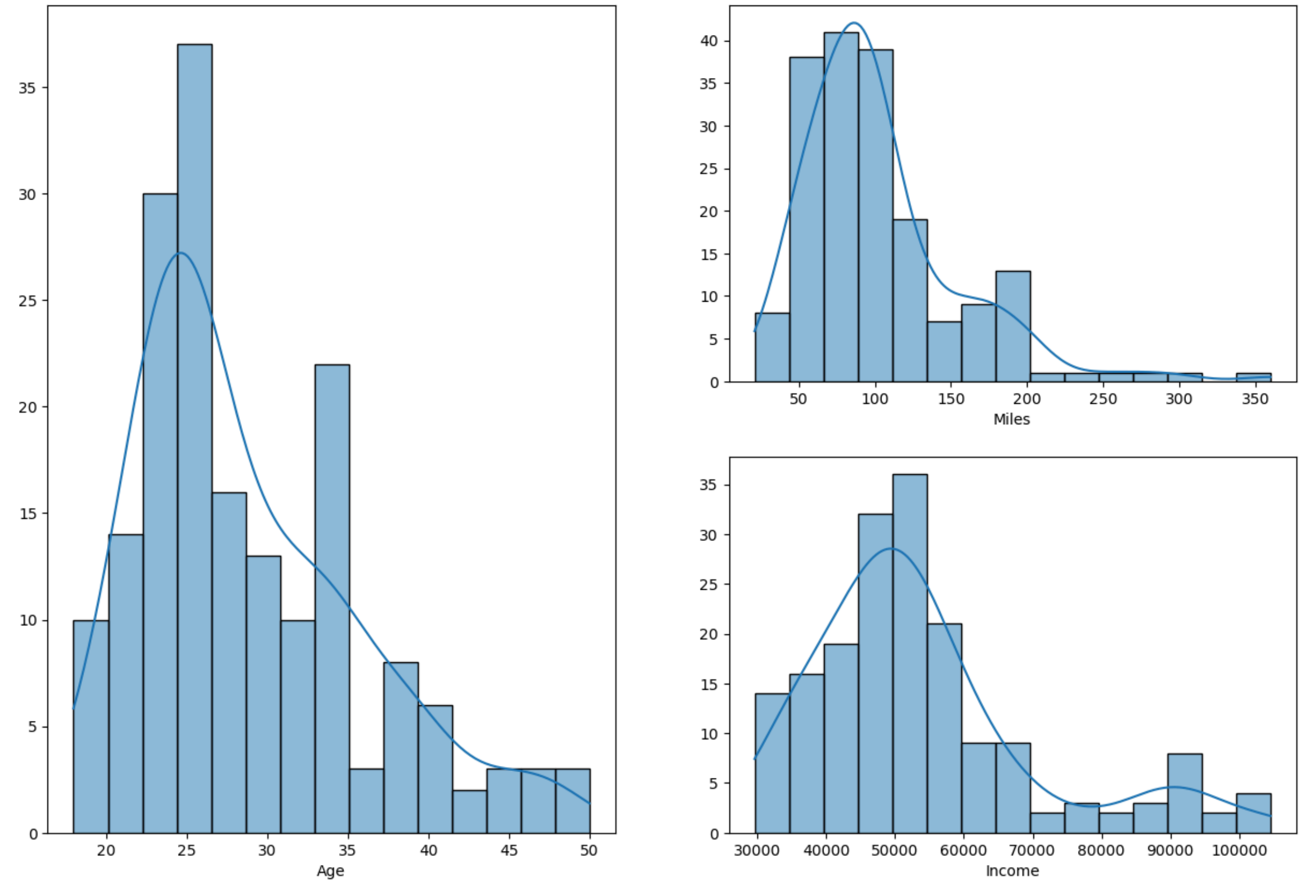
Univariate Analysis

In [15]:
```
plt.figure(figsize = (15,15))

plt.subplot(3,2,1)
sns.countplot(data = df,x = 'Product')

plt.subplot(3,2,2)
sns.countplot(data = df,x = 'Gender')

plt.subplot(3,2,3)
edu = df['Education'].value_counts()
sns.barplot(x = edu.index,y = edu,order = edu.index)
plt.xlabel('Education')
plt.ylabel('')

plt.subplot(3,2,4)
sns.countplot(data = df,x = 'MaritalStatus')

plt.subplot(3,2,5)
us = df['Usage'].value_counts()
sns.barplot(y = us,x = us.index, order = us.index)
plt.xlabel('Usage')
plt.ylabel('')

plt.subplot(3,2,6)
fit = df['Fitness'].value_counts()
sns.barplot(y = fit,x = fit.index, order = fit.index)
plt.xlabel('Fitness')
plt.ylabel('')

plt.suptitle("Count Plots of Categorical Variables")
plt.show()
```

Count Plots of Categorical Variables



In [16]:
```
plt.figure(figsize = (15,10))

plt.subplot(1,2,1)
sns.histplot(data = df, x= 'Age',kde = True,bins = 15)
# sns.lineplot(x = [24,24],y = [0,37],color = 'red',estimator=None,linewidth = 1.5)
plt.ylabel('')


plt.subplot(2,2,2)
sns.histplot(data = df, x= 'Miles',kde = True,bins = 15)
plt.ylabel('')

plt.subplot(2,2,4)
sns.histplot(data = df, x= 'Income',kde = True,bins = 15)
plt.ylabel('')

plt.show()
```



In [17]:
```
(df['Gender'] == 'Female').sum()/(df['Gender'] == 'Male').sum()
```

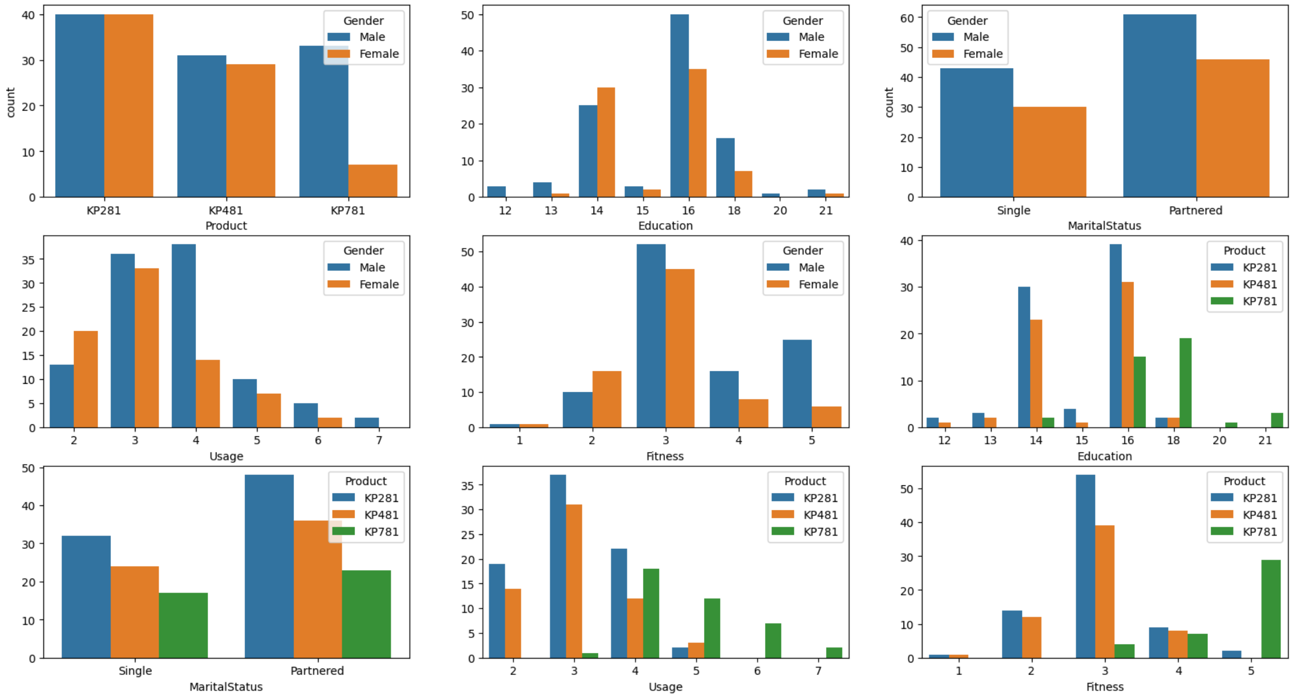
Out[17]:
```
0.7307692307692307
```

Observations

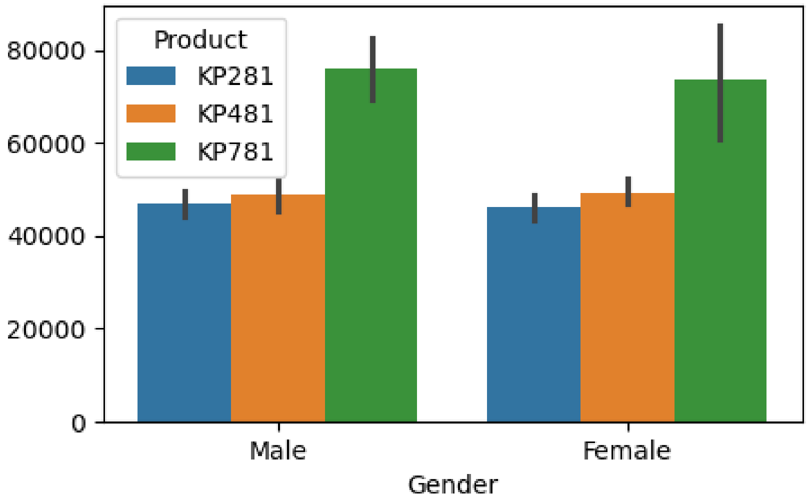
1. All the Numerical Variables are Postively Skewed
2. Female to Male ratio is around 73%
3. Most of the Customers that use the dataset had 16 years of Education
4. Most of the Customers are Partnered
5. Most of the Customers try to use the Products 3 or 4 times a week
6. Most of the Customers have an average level of fitness
7. Most used Product is KP281 followed by KP481 and by KP781

Bivariate Analysis

In [18]:
```
plt.figure(figsize = (20,18))

plt.subplot(5,3,1)
sns.countplot(data = df,x = 'Product',hue = 'Gender')


plt.subplot(5,3,2)
edu = df['Education'].value_counts()
sns.countplot(df,x = 'Education',hue = 'Gender' )
plt.xlabel('Education')
plt.ylabel('')

plt.subplot(5,3,3)
sns.countplot(data = df,x = 'MaritalStatus',hue = 'Gender')

plt.subplot(5,3,4)
us = df['Usage'].value_counts()
sns.countplot(data = df,x = 'Usage',hue = 'Gender')
plt.xlabel('Usage')
plt.ylabel('')

plt.subplot(5,3,5)
fit = df['Fitness'].value_counts()
sns.countplot(data = df,x = 'Fitness',hue = 'Gender')
plt.xlabel('Fitness')
plt.ylabel('')


plt.subplot(5,3,6)
edu = df['Education'].value_counts()
sns.countplot(data = df,x = 'Education',hue = 'Product')
plt.xlabel('Education')
plt.ylabel('')

plt.subplot(5,3,7)
sns.countplot(data = df,x = 'MaritalStatus',hue = 'Product',)
plt.ylabel('')

plt.subplot(5,3,8)
usage = df['Usage'].value_counts()
sns.countplot(data = df,x = 'Usage',hue = 'Product')
plt.xlabel('Usage')
plt.ylabel('')


plt.subplot(5,3,9)
fit = df['Fitness'].value_counts()
sns.countplot(data = df,x = 'Fitness',hue = 'Product')
plt.xlabel('Fitness')
plt.ylabel('')


# plt.subplot(5,3,10)
# sns.barplot(df,y = 'Income',x = 'Gender',hue = 'Product')
# plt.ylabel('')

plt.show()
```



In [19]:
```
plt.figure(figsize = (5,3))

sns.barplot(df,y = 'Income',x = 'Gender',hue = 'Product')
plt.ylabel('')

plt.show()
```



Observations

Product:

1. Product KP281 is used by equal number of Males and Females
2. Product KP481 is slightly more used by Males.
3. Product KP781 is mostly used by Males.

Fitness:

1. Most of the Customers who have excellent level of fitness use KP781 Product
2. Most of the Customers who have an average level of fitness use KP281 Product

Usagee:

1. Customers who try to use the product more than 4 times a week prerfer KP781 Product
2. Customers who use the product for at most 4 times prefer KP281 product
3. Males tend to use the Product for 3 to 4 times a week
4. Females tend to use the Product for 2 to 3 times a week

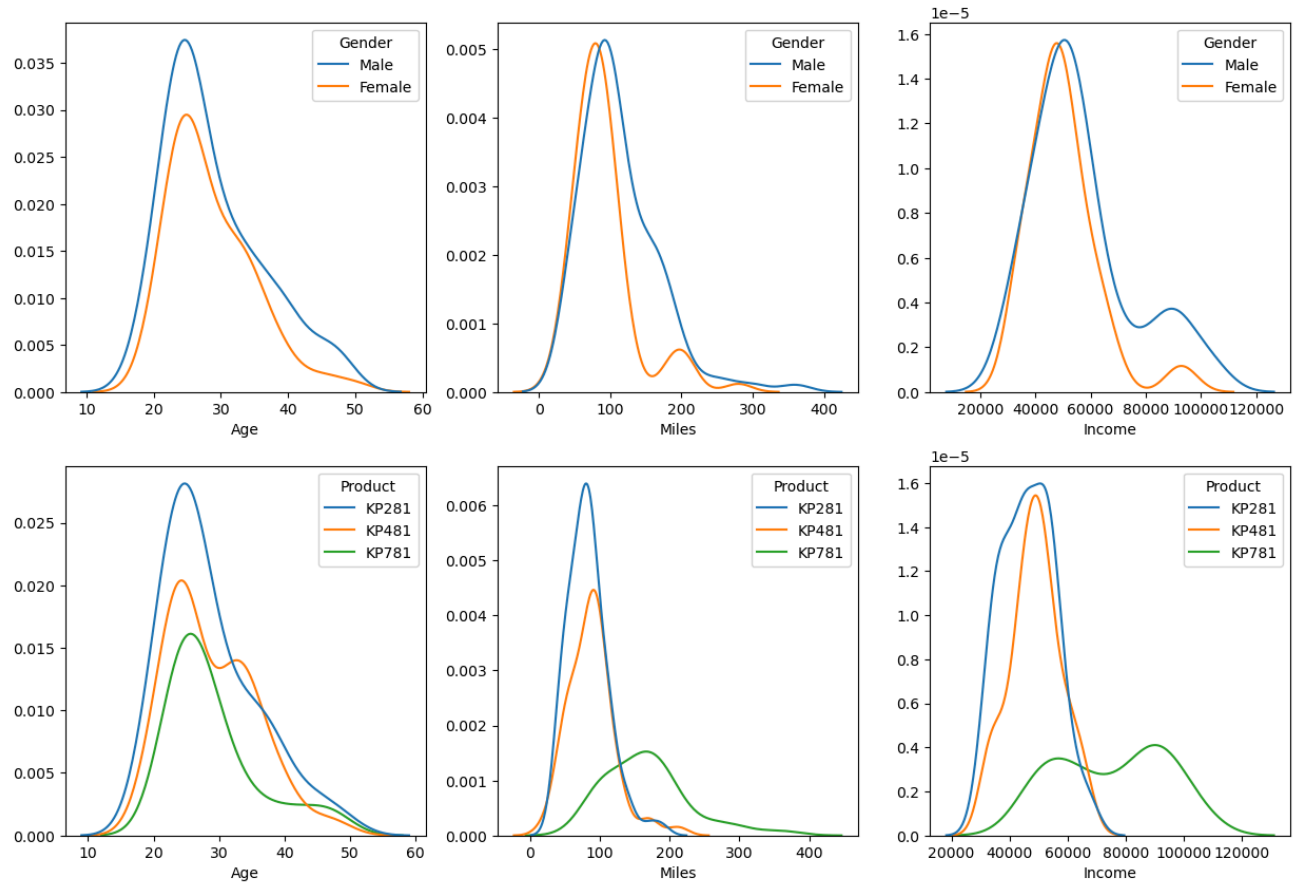
Education:

1. Most of the Customers who have had education for more thatn 16 years prefer the KP781 Product
2. Customers having at most 16 years of education prefer the KP281 Product followed by KP481.

Income:

1. Most of the Customers who have high income prefer to use KP781

In [20]:
```
plt.figure(figsize = (15,10))

plt.subplot(2,3,1)
sns.kdeplot(data = df, x= 'Age',hue = 'Gender')
# sns.lineplot(x = [24,24],y = [0,37],color = 'red',estimator=None,linewidth = 1.5)
plt.ylabel('')


plt.subplot(2,3,2)
sns.kdeplot(data = df, x= 'Miles',hue = 'Gender')
plt.ylabel('')

plt.subplot(2,3,3)
sns.kdeplot(data = df, x= 'Income',hue = 'Gender')
plt.ylabel('')


plt.subplot(2,3,4)
sns.kdeplot(data = df, x= 'Age',hue = 'Product')
# sns.lineplot(x = [24,24],y = [0,37],color = 'red',estimator=None,linewidth = 1.5)
plt.ylabel('')


plt.subplot(2,3,5)
sns.kdeplot(data = df, x= 'Miles',hue = 'Product')
plt.ylabel('')

plt.subplot(2,3,6)
sns.kdeplot(data = df, x= 'Income',hue = 'Product')
plt.ylabel('')


plt.show()
```

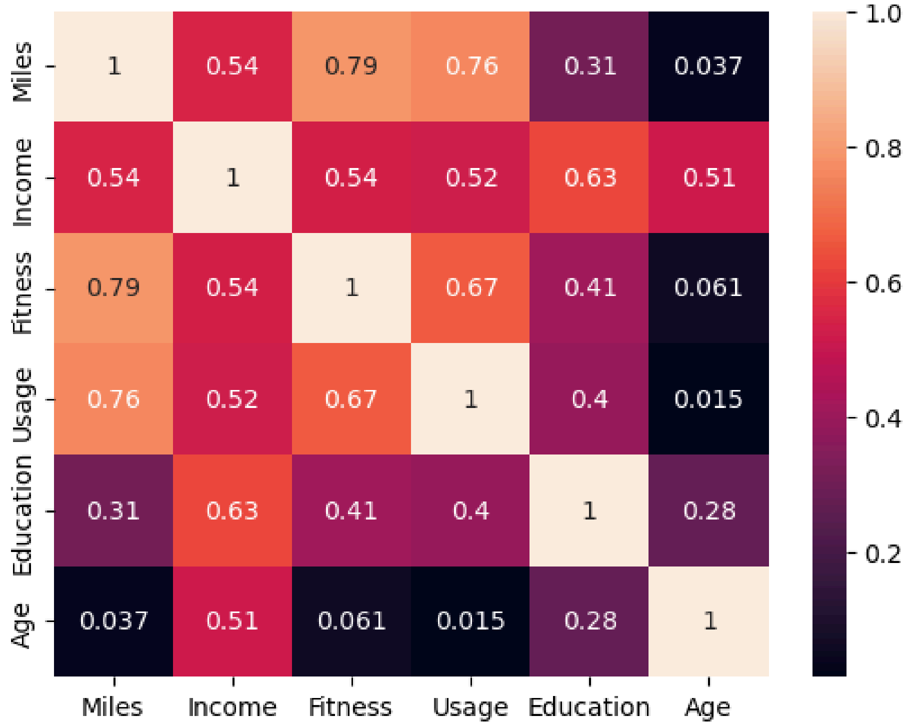


Pair Plot

In [21]:
```
sns.pairplot(df)
plt.show()
```

Correlation

In [22]:
```
sns.heatmap(df[['Miles','Income','Fitness','Usage','Education','Age']].corr(),annot= True)
plt.show()
```



Observations

1. Miles and Fitness and Usage has high correlation

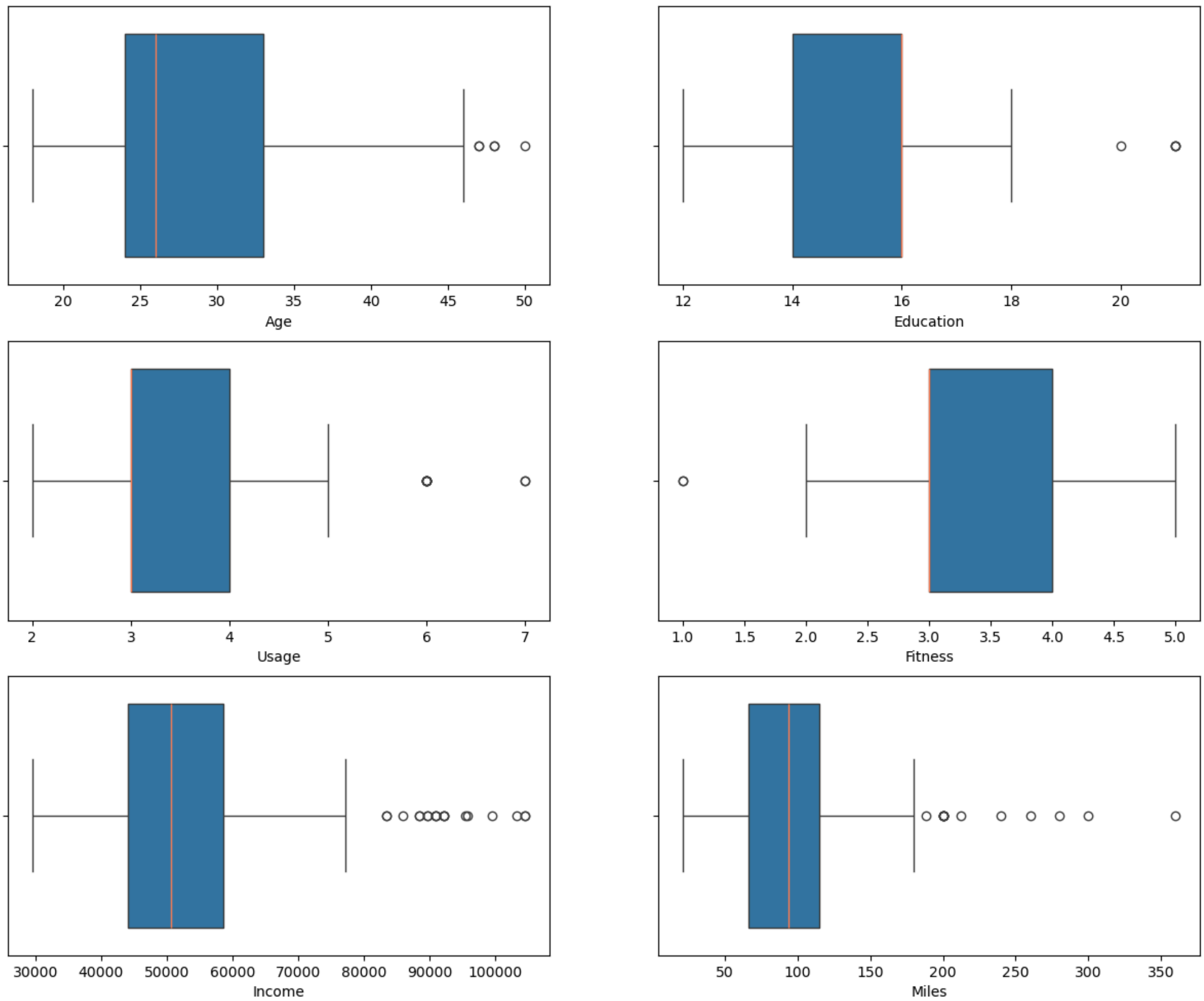
Missing Value & Outlier Detection

In [23]:
```
df.isna().sum()
```

Out[23]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

There is no `Null` count.

In [24]:
```
plt.figure(figsize = (15,12))

plt.subplot(3,2,1)
sns.boxplot(df,x = 'Age', medianprops={"color": "coral"})

plt.subplot(3,2,2)
sns.boxplot(df,x = 'Education',  medianprops={"color": "coral"})

plt.subplot(3,2,3)
sns.boxplot(df,x = 'Usage',  medianprops={"color": "coral"})

plt.subplot(3,2,4)
sns.boxplot(df,x = 'Fitness',  medianprops={"color": "coral"})

plt.subplot(3,2,5)
sns.boxplot(df,x = 'Income',  medianprops={"color": "coral"})

plt.subplot(3,2,6)
sns.boxplot(df,x = 'Miles',  medianprops={"color": "coral"})



plt.show()
```



Observations

1. We are able to see a lot of Outliers of Income and Miles, Other Columns have less Outliers, We won't remove

Distributing Income, Age and Miles to bins

In [25]:
```
df.Age.sort_values().unique()
```

Out[25]:
```
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, 42, 43, 44, 45, 46, 47, 48, 50],
      dtype=int64)
```

In [26]:
```
df['income_class'] = pd.cut(
                            df['Income'],\
                            bins = [20000,35000,50000,65000,80000,120000],\
                            labels = ['low','below avg','avg','above avg','high']
                           )

df['age_class'] = pd.cut(df['Age'], bins = [16,20,25,30,35,40,45,51],\
                        labels = ['late teens','early 20s','late 20s','early 30s','late 30s','e

df['miles_class'] = pd.cut(df['Miles'],bins = [1,40,80,120,160,200,500])
```

Marginal Probability

In [27]:
```
#function to calcuate Marginal Probability
def print_marginal_probability(df,i):
    dum = round((df[i].value_counts(normalize = True).sort_index()* 100),2).reset_index()
    print(bold_text(i.upper()+':'))
    for j in range(len(dum)):
        print(f'Marginal Probabilty for {dum.loc[j,i]} value in {i} column is {dum.iloc[j,1]}%'
    print()
```

In [28]:

```
col_list = ['Gender','Education','MaritalStatus','Usage','Fitness','income_class','age_class','
for i in col_list:
    print_marginal_probability(df,i)
```

```
GENDER:
Marginal Probabilty for Female value in Gender column is 42.22%
Marginal Probabilty for Male value in Gender column is 57.78%

EDUCATION:
Marginal Probabilty for 12 value in Education column is 1.67%
Marginal Probabilty for 13 value in Education column is 2.78%
Marginal Probabilty for 14 value in Education column is 30.56%
Marginal Probabilty for 15 value in Education column is 2.78%
Marginal Probabilty for 16 value in Education column is 47.22%
Marginal Probabilty for 18 value in Education column is 12.78%
Marginal Probabilty for 20 value in Education column is 0.56%
Marginal Probabilty for 21 value in Education column is 1.67%

MARITALSTATUS:
Marginal Probabilty for Partnered value in MaritalStatus column is 59.44%
Marginal Probabilty for Single value in MaritalStatus column is 40.56%

USAGE:
Marginal Probabilty for 2 value in Usage column is 18.33%
Marginal Probabilty for 3 value in Usage column is 38.33%
Marginal Probabilty for 4 value in Usage column is 28.89%
Marginal Probabilty for 5 value in Usage column is 9.44%
Marginal Probabilty for 6 value in Usage column is 3.89%
Marginal Probabilty for 7 value in Usage column is 1.11%

FITNESS:
Marginal Probabilty for 1 value in Fitness column is 1.11%
Marginal Probabilty for 2 value in Fitness column is 14.44%
Marginal Probabilty for 3 value in Fitness column is 53.89%
Marginal Probabilty for 4 value in Fitness column is 13.33%
Marginal Probabilty for 5 value in Fitness column is 17.22%

INCOME_CLASS:
Marginal Probabilty for low value in income_class column is 7.78%
Marginal Probabilty for below avg value in income_class column is 38.33%
Marginal Probabilty for avg value in income_class column is 38.33%
Marginal Probabilty for above avg value in income_class column is 5.0%
Marginal Probabilty for high value in income_class column is 10.56%

AGE_CLASS:
Marginal Probabilty for late teens value in age_class column is 5.56%
Marginal Probabilty for early 20s value in age_class column is 38.33%
Marginal Probabilty for late 20s value in age_class column is 22.78%
Marginal Probabilty for early 30s value in age_class column is 17.78%
Marginal Probabilty for late 30s value in age_class column is 8.89%
Marginal Probabilty for early 40s value in age_class column is 3.33%
Marginal Probabilty for late 40s value in age_class column is 3.33%

MILES_CLASS:
Marginal Probabilty for (1, 40] value in miles_class column is 2.22%
Marginal Probabilty for (40, 80] value in miles_class column is 31.11%
Marginal Probabilty for (80, 120] value in miles_class column is 43.33%
Marginal Probabilty for (120, 160] value in miles_class column is 10.56%
Marginal Probabilty for (160, 200] value in miles_class column is 9.44%
Marginal Probabilty for (200, 500] value in miles_class column is 3.33%
```

Conditional Probability

In [29]:
```
i = 'Gender'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'index')*100),2).res
dum.columns.name = None
rows = dum.shape[0]
for row in range(len(dum)):
    print('Probability of using KP281, given the customer is a',dum.loc[row,i],'is:',f'{dum.loc
    print('Probability of using KP481, given the customer is a',dum.loc[row,i],'is:',f'{dum.loc
    print('Probability of using KP781, given the customer is a',dum.loc[row,i],'is:',f'{dum.loc
    print()
```

```
Probability of using KP281, given the customer is a Female is: 52.63%
Probability of using KP481, given the customer is a Female is: 38.16%
Probability of using KP781, given the customer is a Female is: 9.21%

Probability of using KP281, given the customer is a Male is: 38.46%
Probability of using KP481, given the customer is a Male is: 29.81%
Probability of using KP781, given the customer is a Male is: 31.73%
```

In [30]:
```
i = 'Education'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'columns')*100),2).r
dum.columns.name = None
dum
```

Out[30]:

	Education	KP281	KP481	KP781
0	12	2.50	1.67	0.0
1	13	3.75	3.33	0.0
2	14	37.50	38.33	5.0
3	15	5.00	1.67	0.0
4	16	48.75	51.67	37.5
5	18	2.50	3.33	47.5
6	20	0.00	0.00	2.5
7	21	0.00	0.00	7.5

In [31]:
```
def encode_edu(x):
    if x == 12:
        return 'Higher Secondary'
    elif x> 12 and x<=16:
        return 'Bachelors'
    elif x>16 and x<= 18:
        return 'Masters'
    else:
        return 'Doctorate'
```

In [32]:
```
df['Education_Level'] = df['Education'].apply(encode_edu)
```

In [33]:
```
i = 'Education_Level'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'index')*100),2).res
dum.columns.name = None
rows = dum.shape[0]
for row in range(len(dum)):
    print(f'Probability of using KP281, given the customer\'s highest education Level is {dum.]
    print(f'Probability of using KP481, given the customer\'s highest education Level is {dum.]
    print(f'Probability of using KP781, given the customer\'s highest education Level is {dum.]
    print()
```

```
Probability of using KP281, given the customer's highest education Level is Bachelors is: 50.
67%
Probability of using KP481, given the customer's highest education Level is Bachelors is: 38.
0%
Probability of using KP781, given the customer's highest education Level is Bachelors is: 11.
33%

Probability of using KP281, given the customer's highest education Level is Doctorate is: 0.
0%
Probability of using KP481, given the customer's highest education Level is Doctorate is: 0.
0%
Probability of using KP781, given the customer's highest education Level is Doctorate is: 10
0.0%

Probability of using KP281, given the customer's highest education Level is Higher Secondary
is: 66.67%
Probability of using KP481, given the customer's highest education Level is Higher Secondary
is: 33.33%
Probability of using KP781, given the customer's highest education Level is Higher Secondary
is: 0.0%

Probability of using KP281, given the customer's highest education Level is Masters is: 8.7%
Probability of using KP481, given the customer's highest education Level is Masters is: 8.7%
Probability of using KP781, given the customer's highest education Level is Masters is: 82.6
1%
```

In [34]:
```
i = 'Education'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'index')*100),2).res
dum.columns.name = None
rows = dum.shape[0]
for row in range(len(dum)):
    print(f'Probability of using KP281, given the customer had {dum.loc[row,i]} years of Educat
    print(f'Probability of using KP481, given the customer had {dum.loc[row,i]} years of Educat
    print(f'Probability of using KP781, given the customer had {dum.loc[row,i]} years of Educat
    print()
```

```
Probability of using KP281, given the customer had 12 years of Education is: 66.67%
Probability of using KP481, given the customer had 12 years of Education is: 33.33%
Probability of using KP781, given the customer had 12 years of Education is: 0.0%

Probability of using KP281, given the customer had 13 years of Education is: 60.0%
Probability of using KP481, given the customer had 13 years of Education is: 40.0%
Probability of using KP781, given the customer had 13 years of Education is: 0.0%

Probability of using KP281, given the customer had 14 years of Education is: 54.55%
Probability of using KP481, given the customer had 14 years of Education is: 41.82%
Probability of using KP781, given the customer had 14 years of Education is: 3.64%

Probability of using KP281, given the customer had 15 years of Education is: 80.0%
Probability of using KP481, given the customer had 15 years of Education is: 20.0%
Probability of using KP781, given the customer had 15 years of Education is: 0.0%

Probability of using KP281, given the customer had 16 years of Education is: 45.88%
Probability of using KP481, given the customer had 16 years of Education is: 36.47%
Probability of using KP781, given the customer had 16 years of Education is: 17.65%

Probability of using KP281, given the customer had 18 years of Education is: 8.7%
Probability of using KP481, given the customer had 18 years of Education is: 8.7%
Probability of using KP781, given the customer had 18 years of Education is: 82.61%

Probability of using KP281, given the customer had 20 years of Education is: 0.0%
Probability of using KP481, given the customer had 20 years of Education is: 0.0%
Probability of using KP781, given the customer had 20 years of Education is: 100.0%

Probability of using KP281, given the customer had 21 years of Education is: 0.0%
Probability of using KP481, given the customer had 21 years of Education is: 0.0%
Probability of using KP781, given the customer had 21 years of Education is: 100.0%
```

In [35]:
```
i = 'MaritalStatus'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'index')*100),2).res
dum.columns.name = None
rows = dum.shape[0]
for row in range(len(dum)):
    print(f'Probability of using KP281, given the customer is {dum.loc[row,i]} is:',f'{dum.loc[
    print(f'Probability of using KP481, given the customer is {dum.loc[row,i]} is:',f'{dum.loc[
    print(f'Probability of using KP781, given the customer is {dum.loc[row,i]} is:',f'{dum.loc[
    print()
```

```
Probability of using KP281, given the customer is Partnered is: 44.86%
Probability of using KP481, given the customer is Partnered is: 33.64%
Probability of using KP781, given the customer is Partnered is: 21.5%

Probability of using KP281, given the customer is Single is: 43.84%
Probability of using KP481, given the customer is Single is: 32.88%
Probability of using KP781, given the customer is Single is: 23.29%
```

In [36]:
```
i = 'Usage'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'index')*100),2).res
dum.columns.name = None
rows = dum.shape[0]
for row in range(len(dum)):
    print(f'Probability of using KP281, given the customer uses the Product {dum.loc[row,i]} ti
    print(f'Probability of using KP481, given the customer uses the Product {dum.loc[row,i]} ti
    print(f'Probability of using KP781, given the customer uses the Product {dum.loc[row,i]} ti
    print()
```

```
Probability of using KP281, given the customer uses the Product 2 times a week is: 57.58%
Probability of using KP481, given the customer uses the Product 2 times a week is: 42.42%
Probability of using KP781, given the customer uses the Product 2 times a week is: 0.0%

Probability of using KP281, given the customer uses the Product 3 times a week is: 53.62%
Probability of using KP481, given the customer uses the Product 3 times a week is: 44.93%
Probability of using KP781, given the customer uses the Product 3 times a week is: 1.45%

Probability of using KP281, given the customer uses the Product 4 times a week is: 42.31%
Probability of using KP481, given the customer uses the Product 4 times a week is: 23.08%
Probability of using KP781, given the customer uses the Product 4 times a week is: 34.62%

Probability of using KP281, given the customer uses the Product 5 times a week is: 11.76%
Probability of using KP481, given the customer uses the Product 5 times a week is: 17.65%
Probability of using KP781, given the customer uses the Product 5 times a week is: 70.59%

Probability of using KP281, given the customer uses the Product 6 times a week is: 0.0%
Probability of using KP481, given the customer uses the Product 6 times a week is: 0.0%
Probability of using KP781, given the customer uses the Product 6 times a week is: 100.0%

Probability of using KP281, given the customer uses the Product 7 times a week is: 0.0%
Probability of using KP481, given the customer uses the Product 7 times a week is: 0.0%
Probability of using KP781, given the customer uses the Product 7 times a week is: 100.0%
```

In [37]:
```
i = 'Fitness'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'index')*100),2).res
dum.columns.name = None
rows = dum.shape[0]
for row in range(len(dum)):
    print(f'Probability of using KP281, given the customer has {dum.loc[row,i]} level of Fitnes
    print(f'Probability of using KP481, given the customer has {dum.loc[row,i]} level of Fitnes
    print(f'Probability of using KP781, given the customer has {dum.loc[row,i]} level of Fitnes
    print()
```

```
Probability of using KP281, given the customer has 1 level of Fitness is: 50.0%
Probability of using KP481, given the customer has 1 level of Fitness is: 50.0%
Probability of using KP781, given the customer has 1 level of Fitness is: 0.0%

Probability of using KP281, given the customer has 2 level of Fitness is: 53.85%
Probability of using KP481, given the customer has 2 level of Fitness is: 46.15%
Probability of using KP781, given the customer has 2 level of Fitness is: 0.0%

Probability of using KP281, given the customer has 3 level of Fitness is: 55.67%
Probability of using KP481, given the customer has 3 level of Fitness is: 40.21%
Probability of using KP781, given the customer has 3 level of Fitness is: 4.12%

Probability of using KP281, given the customer has 4 level of Fitness is: 37.5%
Probability of using KP481, given the customer has 4 level of Fitness is: 33.33%
Probability of using KP781, given the customer has 4 level of Fitness is: 29.17%

Probability of using KP281, given the customer has 5 level of Fitness is: 6.45%
Probability of using KP481, given the customer has 5 level of Fitness is: 0.0%
Probability of using KP781, given the customer has 5 level of Fitness is: 93.55%
```

In [38]:
```
i = 'income_class'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'index')*100),2).res
dum.columns.name = None
rows = dum.shape[0]
for row in range(len(dum)):
    print(f'Probability of using KP281, given the customer belongs to {dum.loc[row,i]} income c
    print(f'Probability of using KP481, given the customer belongs to {dum.loc[row,i]} income c
    print(f'Probability of using KP781, given the customer belongs to {dum.loc[row,i]} income c
    print()
```

```
Probability of using KP281, given the customer belongs to low income class: 57.14%
Probability of using KP481, given the customer belongs to low income class: 42.86%
Probability of using KP781, given the customer belongs to low income class: 0.0%

Probability of using KP281, given the customer belongs to below avg income class: 57.97%
Probability of using KP481, given the customer belongs to below avg income class: 34.78%
Probability of using KP781, given the customer belongs to below avg income class: 7.25%

Probability of using KP281, given the customer belongs to avg income class: 43.48%
Probability of using KP481, given the customer belongs to avg income class: 40.58%
Probability of using KP781, given the customer belongs to avg income class: 15.94%

Probability of using KP281, given the customer belongs to above avg income class: 22.22%
Probability of using KP481, given the customer belongs to above avg income class: 22.22%
Probability of using KP781, given the customer belongs to above avg income class: 55.56%

Probability of using KP281, given the customer belongs to high income class: 0.0%
Probability of using KP481, given the customer belongs to high income class: 0.0%
Probability of using KP781, given the customer belongs to high income class: 100.0%
```

In [39]:
```python
i = 'age_class'
dum = round((pd.crosstab(index = df[i],columns = df['Product'],normalize = 'index')*100),2).res
dum.columns.name = None
rows = dum.shape[0]
for row in range(len(dum)):
    print(f'Probability of using KP281, given the customer belongs to {dum.loc[row,i]} age clas
    print(f'Probability of using KP481, given the customer belongs to {dum.loc[row,i]} age clas
    print(f'Probability of using KP781, given the customer belongs to {dum.loc[row,i]} age clas
    print()
```

```
Probability of using KP281, given the customer belongs to late teens age class is: 60.0%
Probability of using KP481, given the customer belongs to late teens age class is: 40.0%
Probability of using KP781, given the customer belongs to late teens age class is: 0.0%

Probability of using KP281, given the customer belongs to early 20s age class is: 40.58%
Probability of using KP481, given the customer belongs to early 20s age class is: 34.78%
Probability of using KP781, given the customer belongs to early 20s age class is: 24.64%

Probability of using KP281, given the customer belongs to late 20s age class is: 51.22%
Probability of using KP481, given the customer belongs to late 20s age class is: 17.07%
Probability of using KP781, given the customer belongs to late 20s age class is: 31.71%

Probability of using KP281, given the customer belongs to early 30s age class is: 34.38%
Probability of using KP481, given the customer belongs to early 30s age class is: 53.12%
Probability of using KP781, given the customer belongs to early 30s age class is: 12.5%

Probability of using KP281, given the customer belongs to late 30s age class is: 50.0%
Probability of using KP481, given the customer belongs to late 30s age class is: 37.5%
Probability of using KP781, given the customer belongs to late 30s age class is: 12.5%

Probability of using KP281, given the customer belongs to early 40s age class is: 50.0%
Probability of using KP481, given the customer belongs to early 40s age class is: 16.67%
Probability of using KP781, given the customer belongs to early 40s age class is: 33.33%

Probability of using KP281, given the customer belongs to late 40s age class is: 50.0%
Probability of using KP481, given the customer belongs to late 40s age class is: 16.67%
Probability of using KP781, given the customer belongs to late 40s age class is: 33.33%
```

Customer Profiling

Using Probabilites below Customer Profiling was done.

- **K281**
  - Gender => Female
  - Education => Higher Secondary
  - Usage => 2 to 3 times a week
  - Fitness => 1 to 2 Level of Fitness
  - Income => low to below avg income class
  - Age => all age levels, slightly more inclined towards late teens

- **K481**
  - Gender => Male and Female
  - Education => Bachelors
  - Usage => 2 to 3 times a week
  - Fitness => 1 to 3 Level of Fitness
  - Income => low, avg income class
  - Age => late teens to 30s

- **K781**
  - Gender => Male
  - Education => Doctorate
  - Usage => 5 to 7 times a week
  - Fitness => 5th Level of Fitness
  - Income => High Income Class
  - Age => 40s

Business Insights

Product:

1. Only Half of the Customers that use KP281 use KP781.
2. 4/9th, 3/9th, 2/9th are the number of records for KP281, KP481 and KP781 respectively.
3. Product KP281 is used by equal number of Males and Females
4. Product KP481 is slightly more used by Males.
5. Product KP781 is mostly used by Males.

Gender:

1. Most of the Customers are Males, Female to Male ratio is around 73%.

Age:

1. Customers from 18 to 50 years of age use these Products.
2. Maximum Customers are of 24 to 33 years to old.
3. 45% of Customers are early twenties.

Education:

1. Customers using these Products have 12 to 21 years of Education.
2. Most of the Customers had Education 12 to 16 years of Education.
3. Highest number of Customers had 16 years followed by 14 years of Education.
4. Most of the Customers who have had education for more thatn 16 years prefer the KP781 Product
5. Customers having less than 16 years of education prefer the KP281 Product followed by KP481.

Marital Status:

1. Most of the Customers are Partnered

Usage:

1. Customers to use these Products 2 to 7 times a week.
2. Most of the Customers plan to use the Products either 3 or 4 times a week.
3. Customers who use the product more than 4 times a week prerfer KP781 treadmill.
4. Customers who use the product for less than 4 times prefer KP281 treadmill.
5. Males tend to use the Product for 3 to 4 times a week
6. Females tend to use the Product for 2 to 3 times a week

Fitness:

1. Customers using these Products have Fitness level 1-5, 5 being excellent and 1 being poor fitness.
2. Most of the Customers have 3-4 level of Fitness.
3. 1/6th of the Customers in this dataset are in excellent shape.
4. Most of the Customers who have excellent level of fitness use KP781 Product
5. Most of the Customers who have an average level of fitness use KP281 Product

Income:

1. Customers using these Products have approx Income band of 30k to 105k.
2. Most of the Customers lie in the 44k to 59k Income band.
3. Most of the Customers who have high income prefer to use KP781

Miles

1. Customers using these Products expect to walk 21 to 360 Miles.
2. Most of the Customers expect to walk within 66 to 115 Miles.

General Observations:

1. All the Numerical Variables are Postively Skewed

Recommendations

- As KP281 is popular among average fitness levels and is a budget-priced product, we should focus more on affordability and simplicity when marketing it. This product can be targeted at individuals or families.
- Since KP281 is used for shorter distances and less than 4 times a week, its ease of use and compact design should be highlighted.
- The target audience for KP481 should be male and female customers who are more conscious about their fitness level, as this product is popular among customers having above-average fitness.
- As KP781 is preferred by customers having excellent fitness and high income, while marketing, we should consider highlighting the new technological/advanced features and high-quality aspect of this product.
- This can be targeted at gyms, state-of-the-art fitness centers, athletic clubs, etc.
- Since KP781 is used for higher distances, its durability, comfort, and high quality should be highlighted.

In [ ]:
```

```