# Business Case: Aerofit - Descriptive Statistics & Probability

Task

- Performing descriptive analytics to create a customer profile for each AeroFit treadmill product by developing appropriate tables and charts.
- For each AeroFit treadmill product, constructing two-way contingency tables and computing all conditional and marginal probabilities along with their insights/impact on the business.

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

In [2]:
```
df=pd.read_csv(r'C:\Users\sudhanshu tomar\Desktop\datasets\aerofit_treadmill
```

In [3]:
```
df.head()
```

Out[3]:

| | 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 [4]:
```
df.describe()
```

Out[4]:

| | 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 |

In [5]:
```
1 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 [6]:
```
1 df['Product'].value_counts()
```

Out[6]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

In [7]:
```
1 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
```

No null value or missing value is detected

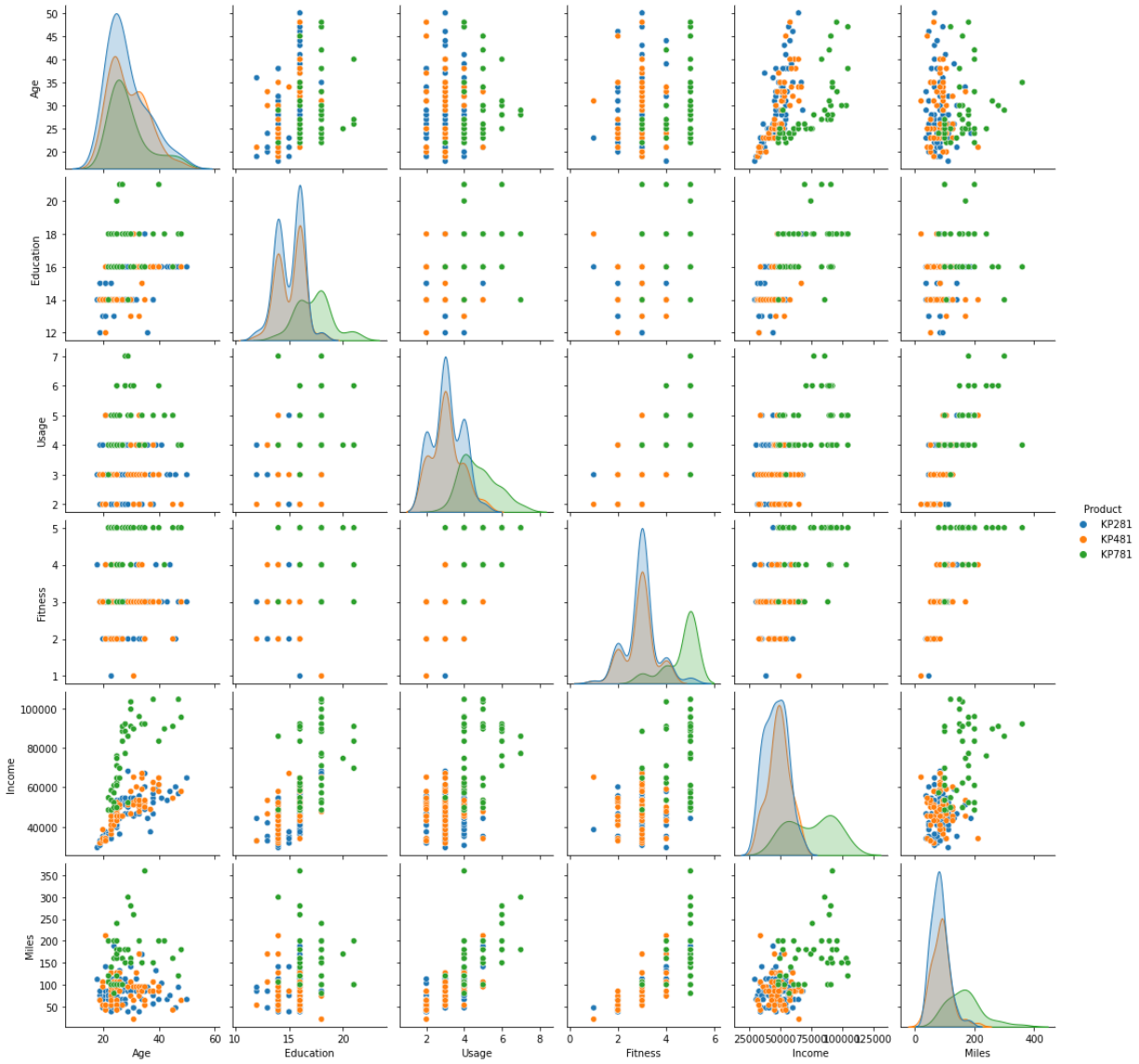
In [8]:
```
1 df['Age'].unique()
```

Out[8]:
```
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],
      dtype=int64)
```

Treadmill is bought by the people who are in age range 18 to 42

In [9]:
```
1 sns.pairplot(data=df,hue='Product')
```

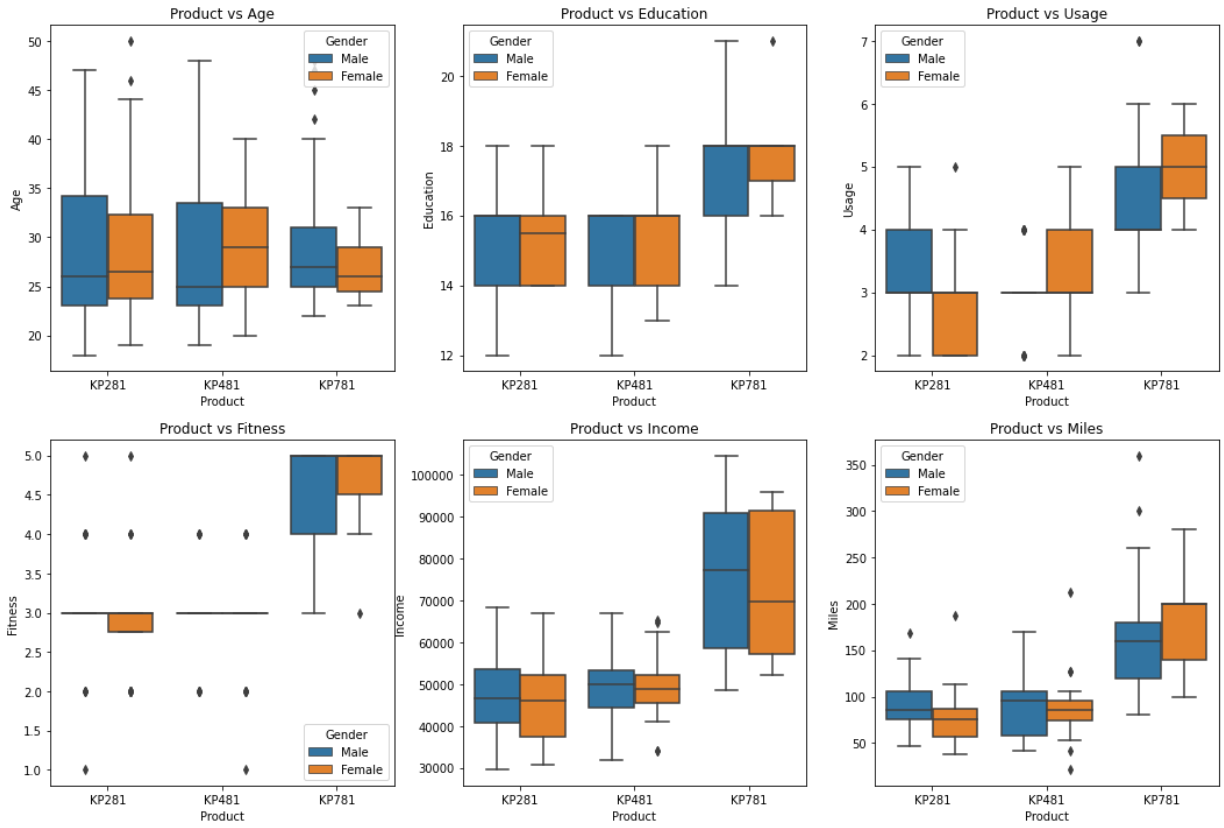
Out[9]: <seaborn.axisgrid.PairGrid at 0x16e697e5390>




# Boxplot for various attributes with the model

In [10]:

```
a = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(18, 12))
count = 0
for i in range(2):
    for j in range(3):
        sns.boxplot(data=df, x='Product', y=a[count],hue='Gender' , ax=axs[i
        count += 1
```




Observations:

1) Product vs Age

- Both male and female customers purchasing product KP281 are having same Age median value.
- Female customer using Product KP481 have higher median age than that of male customer using same product.
- Customers whose age lies between 25-30, are more likely to buy KP781 product.

2) Product vs Education

- Irrespective of Gender customers whose Education is greater than 16, have more chances to purchase the KP781 product.
- While the customers with Education less than 16 have equal chances of purchasing KP281 or KP481.

3) Product vs Usage

- Both male and female customers who are planning to use the treadmill greater than 4 times a week, are more likely to purchase the KP781 product.
- While the other customers are likely to purchasing KP281 or KP481.

4) Product vs Fitness

- The more the customer is fit (fitness >= 4), higher the chances of the customer to purchase the KP781 product.

5) Product vs Income

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

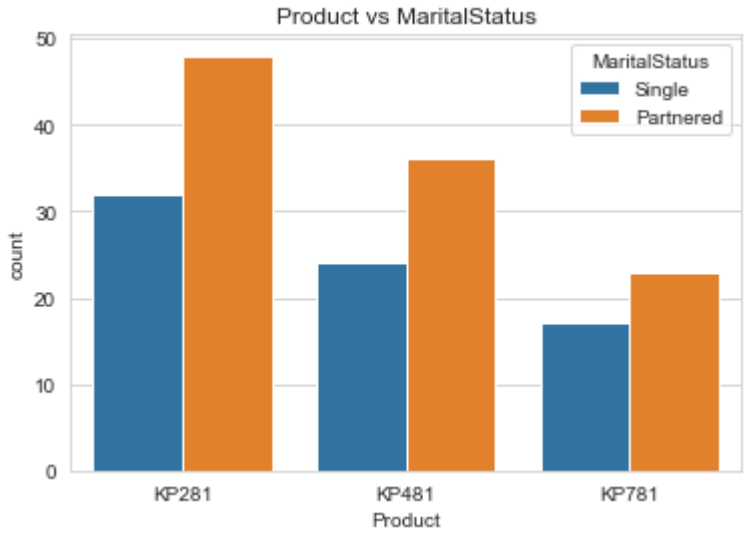
6) Product vs Miles

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

In [11]:
```
sns.set_style(style='whitegrid')
sns.countplot(df.Product,hue=df.MaritalStatus).set(title='Product vs Marital
```

```
C:\Users\sudhanshu tomar\AppData\Local\Programs\Python\Python310\lib\site-packa
ges\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be `
data`, and passing other arguments without an explicit keyword will result in a
n error or misinterpretation.
  warnings.warn(
```

Out[11]:
```
[Text(0.5, 1.0, 'Product vs MaritalStatus')]
```



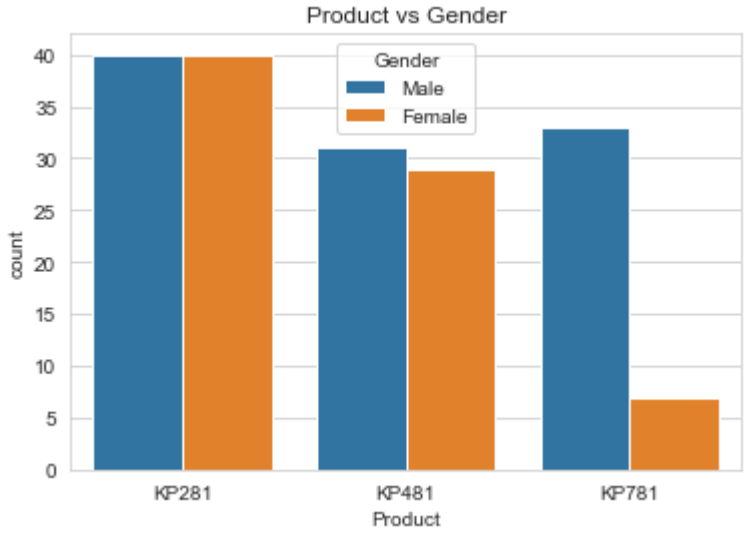
Product vs MaritalStatus

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

In [12]:
```
1 sns.set_style(style='whitegrid')
2 sns.countplot(df.Product,hue=df.Gender).set(title='Product vs Gender')
```

```
C:\Users\sudhanshu tomar\AppData\Local\Programs\Python\Python310\lib\site-packa
ges\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be `
data`, and passing other arguments without an explicit keyword will result in a
n error or misinterpretation.
  warnings.warn(
```

Out[12]:
```
[Text(0.5, 1.0, 'Product vs Gender')]
```

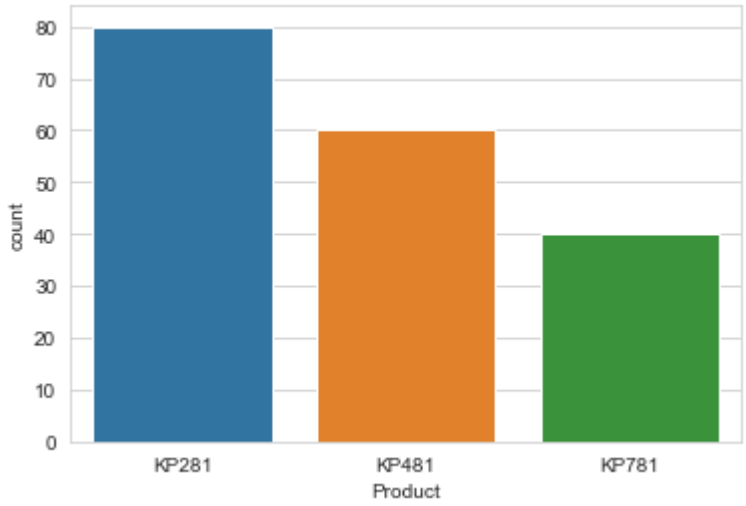



Product vs Gender

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

In [13]:
```
1 sns.countplot(x=df['Product'])
```

Out[13]:
```
<AxesSubplot:xlabel='Product', ylabel='count'>
```




It is observed that Product KP281 is more frequently bought than other two.

# Computing Marginal & Conditional Probabilities

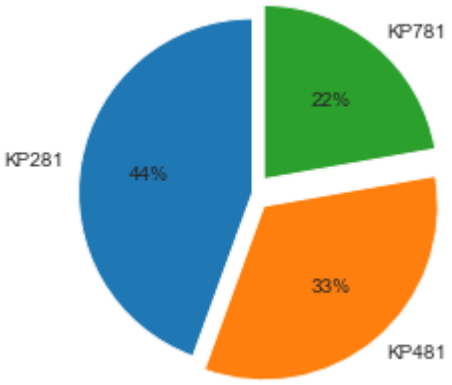
Marginal Probability

In [14]:
```
1 round(df['Product'].value_counts()/180,2)
```

Out[14]:
```
KP281    0.44
KP481    0.33
KP781    0.22
Name: Product, dtype: float64
```

In [15]:
```
1 plt.pie(df.Product.value_counts(), startangle=90, explode=(0,0.1,0.1), label
```

Out[15]:
```
([<matplotlib.patches.Wedge at 0x16e70bca3e0>,
  <matplotlib.patches.Wedge at 0x16e70bcab00>,
  <matplotlib.patches.Wedge at 0x16e70bcb220>],
 [Text(-1.0832885303005317, 0.19101298416420232, 'KP281'),
  Text(0.7713451794386843, -0.9192532916213595, 'KP481'),
  Text(0.7713450503386199, 0.9192533999491719, 'KP781')],
 [Text(-0.5908846528911991, 0.10418890045320126, '44%'),
  Text(0.4499513546725658, -0.5362310867791263, '33%'),
  Text(0.44995127936419493, 0.5362311499703502, '22%')])
```



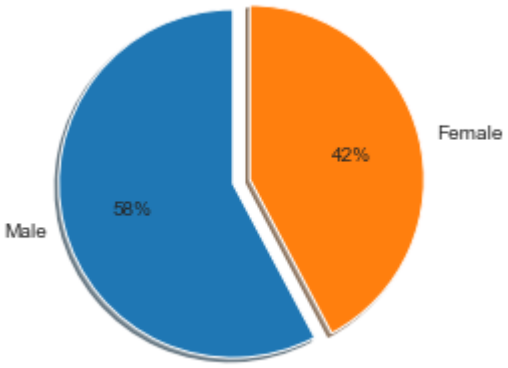

Product KP281 is more popular.

In [16]:
```
1 round(df['Gender'].value_counts()/180,2)
```

Out[16]:
```
Male      0.58
Female    0.42
Name: Gender, dtype: float64
```

In [17]:
```
1 plt.pie(df.Gender.value_counts(),shadow=True, startangle=90, explode=(0,0.1)
```

Out[17]:
```
([<matplotlib.patches.Wedge at 0x16e70c162f0>,
  <matplotlib.patches.Wedge at 0x16e70c16c80>],
 [Text(-1.0673252778639226, -0.26611416954514144, 'Male'),
  Text(1.1643548213983763, 0.2903064757912845, 'Female')],
 [Text(-0.5821774242894123, -0.14515318338825894, '58%'),
  Text(0.6792069791490527, 0.1693454442115826, '42%')])
```



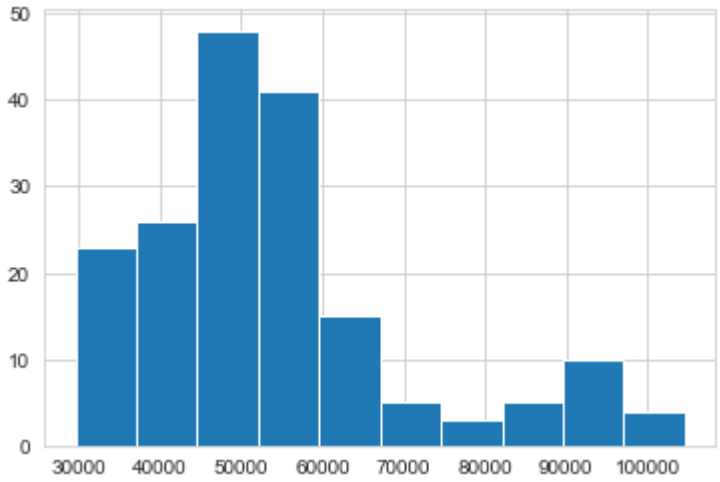

More male customer are present than female customer

In [18]:
```
1 df['Income'].unique()
```

Out[18]:
```
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], dtype=int64)
```

In [19]:
```
1 plt.hist(df['Income'],bins=10)
```

Out[19]:
```
(array([23., 26., 48., 41., 15.,  5.,  3.,  5., 10.,  4.]),
 array([ 29562. ,  37063.9,  44565.8,  52067.7,  59569.6,  67071.5,
         74573.4,  82075.3,  89577.2,  97079.1, 104581. ]),
 <BarContainer object of 10 artists>)
```




Most of the treadmills are bought by the people who have income in range 45000 to 60000.

# Conditional Probabilities

Probability of each product given gender

In [20]:
```
1 df1 = pd.crosstab(index=df['Gender'], columns=[df['Product']])
```

In [21]:
```
df1
```

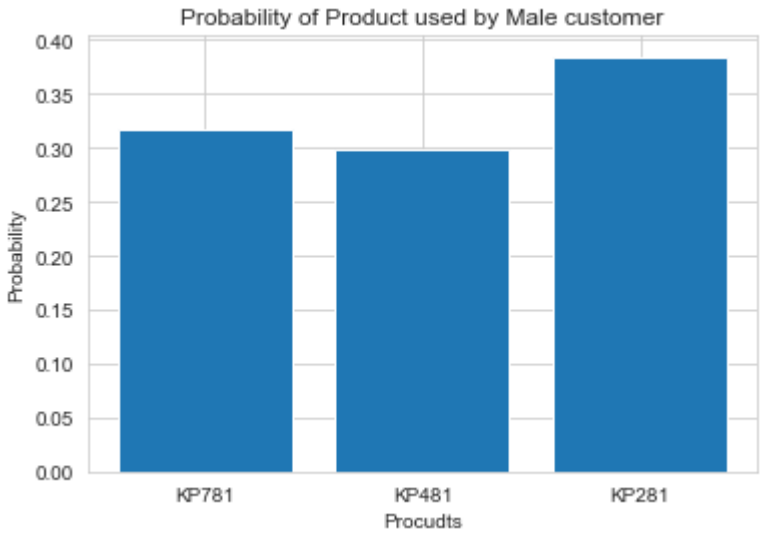
Out[21]:

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

In [22]:
```
p_781m = df1['KP781']['Male'] / df1.loc['Male'].sum()
p_481m= df1['KP481']['Male'] / df1.loc['Male'].sum()
p_281m = df1['KP281']['Male'] / df1.loc['Male'].sum()
```

In [23]:
```
plt.bar(['KP781','KP481','KP281'],[p_781m,p_481m,p_281m])
plt.title('Probability of Product used by Male customer')
plt.xlabel('Procudts')
plt.ylabel('Probability')
```

Out[23]:
```
Text(0, 0.5, 'Probability')
```

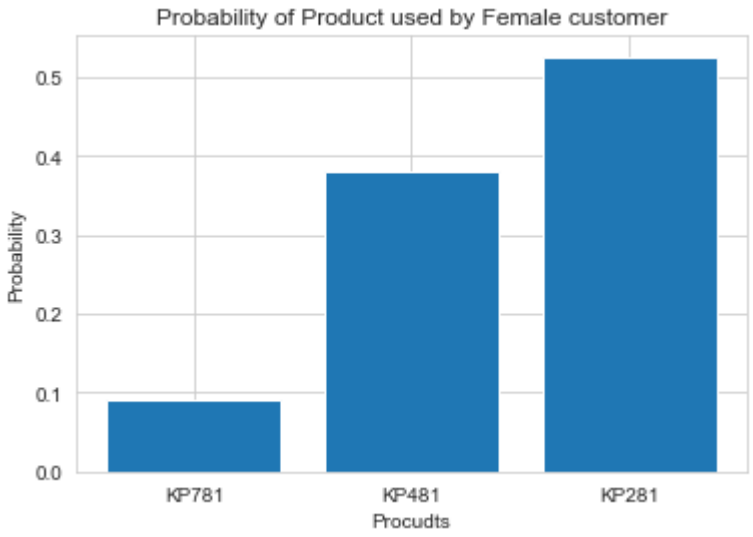



Probability that a male customer buy Product KP281 is higher than he buys other two product

In [24]:
```
p_781f = df1['KP781']['Female'] / df1.loc['Female'].sum()
p_481f= df1['KP481']['Female'] / df1.loc['Female'].sum()
p_281f = df1['KP281']['Female'] / df1.loc['Female'].sum()
```

In [25]:
```
plt.bar(['KP781','KP481','KP281'],[p_781f,p_481f,p_281f])
plt.title('Probability of Product used by Female customer')
plt.xlabel('Procudts')
plt.ylabel('Probability')
```

Out[25]: Text(0, 0.5, 'Probability')




It is clearly seen that female customer tend to buy product KP281 with greater probability then product KP481 and probability of buying product KP781 is least.

In [26]:
```
df2 = pd.crosstab(index=df['MaritalStatus'], columns=[df['Product']])
```

In [27]:
```
df2
```

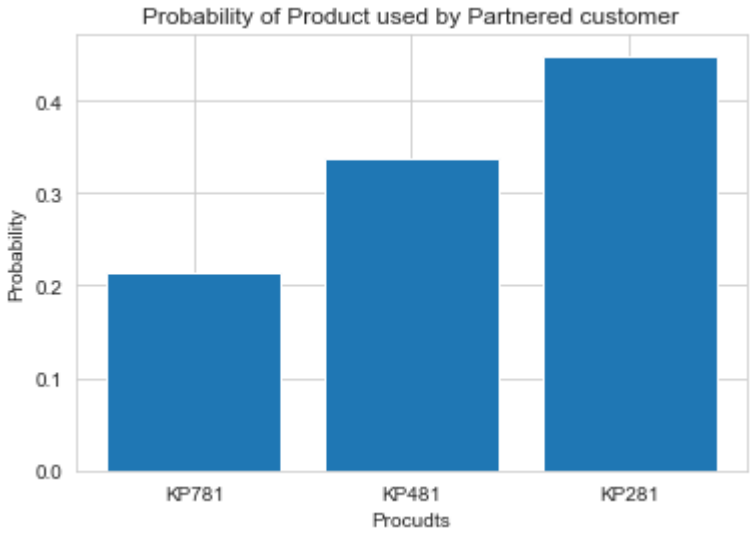
Out[27]:

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

In [28]:
```
p_781p = df2['KP781']['Partnered'] / df2.loc['Partnered'].sum()
p_481p= df2['KP481']['Partnered'] / df2.loc['Partnered'].sum()
p_281p = df2['KP281']['Partnered'] / df2.loc['Partnered'].sum()
```

In [29]:
```
plt.bar(['KP781','KP481','KP281'],[p_781p,p_481p,p_281p])
plt.title('Probability of Product used by Partnered customer')
plt.xlabel('Procudts')
plt.ylabel('Probability')
```

Out[29]: Text(0, 0.5, 'Probability')

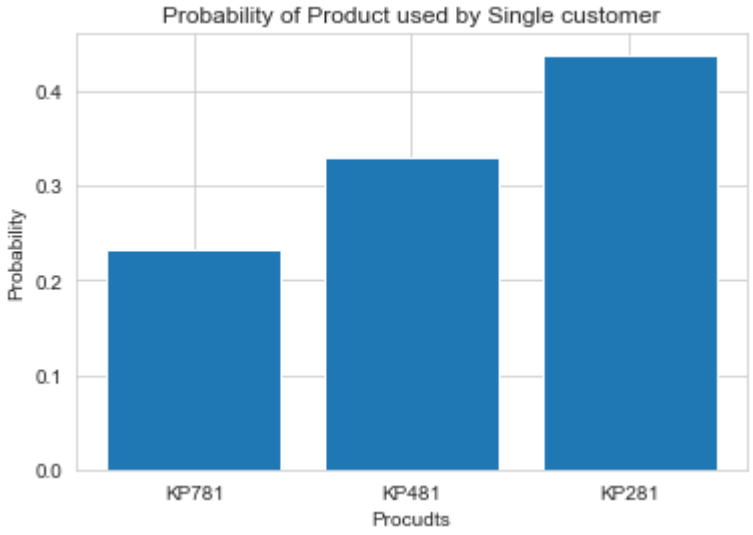



Partnered customer more likely to buy product KP281 than other two

In [30]:
```
p_781p = df2['KP781']['Single'] / df2.loc['Single'].sum()
p_481p= df2['KP481']['Single'] / df2.loc['Single'].sum()
p_281p = df2['KP281']['Single'] / df2.loc['Single'].sum()
```

In [31]:
```
plt.bar(['KP781','KP481','KP281'],[p_781p,p_481p,p_281p])
plt.title('Probability of Product used by Single customer')
plt.xlabel('Procudts')
plt.ylabel('Probability')
```

Out[31]: Text(0, 0.5, 'Probability')




Single customer more likely to buy product KP281 than other two

# Correlation

In [32]:
```
df.corr()
```

Out[32]:

| | 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 |

In [33]:
```
plt.figure(figsize = (12, 8))
sns.heatmap(df.corr(), cmap= "YlGnBu", annot=True)
```

Out[33]: <AxesSubplot:>


- Miles and Fitness and Miles and Usage are highly correlated, which means if a customer's fitness level is high they use more treadmills.
- Income and education show a strong correlation. High-income and highly educated people prefer high-end models (TM798), as mentioned during Bivariant analysis of Categorical variables.

# Conclusion (Important Observations):

- Model KP281 is the best-selling product, 44% of all treadmill sales go to model KP281.

- The majority of treadmill customers fall within the 45,000 - 60,000 income bracket.
- 88% of treadmills are purchased by customers aged 20 to 35.
- Miles and Fitness & Miles and Usage are highly correlated, which means if a customer's fitness level is high they use more treadmills.
- Both male and female customers who are planning to use the treadmill greater than 4 times a week, are more likely to purchase the KP781 product. While the other customers are likely to purchasing KP281 or KP481.
- KP781 is the only model purchased by a customer who has more than 16 years of education and an income of over 60,000.
- With Fitness level 4 and 5, the customers tend to use high-end models and the average number of miles is above 120 per week.
- Pobability of a female customer buying a treadmill is high for model KP281 than KP781.

# Recommendations:

- KP281 & KP481 are popular with customers earning below 60,000 and can be offered by these companies as affordable models.
- KP781 should be marketed as a Premium Model and marketing it to high income groups and educational over 16 years market segments could result in more sales.
- Aerofit should target the age group between 20 to 35 as majority of treadmills are bought in this age range.
- The KP781 is a premium model, so it is ideally suited for sporty people who have a high average weekly number of miles.
- Product KP281 should be targeted with the female and partnered people as it is more popular among female and partnered people.

In [ ]:
```
1
```