Aerofit sells 3 types of treadmills belonging to 3 tiers:

Entry-level - **KP281 - $1,500**

Mid-level - **KP481 - $1,750**

Advanced - **KP781 - $2,500**

The purpose of this research is to help Aerofit identify the target audience for each type of treadmill offered by the company in order to better understand what type of treadmill would a new user prefer.

In [1]:

```
#importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
from collections import Counter
from scipy.stats import mode
```

In [2]:

```
#reading the dataset from the given URL
df = pd. read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/o
```

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

1.Preliminary Analysis

In [4]:

```
#getting basic info about the dataset
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
```

Initial info shows no null values

In [5]:

```
#generating descriptive statistics
df.describe()
```

Out[5]:

	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

Dataset shows the following:

Average age of consumers : 26yrs
Average education of consumers : 16yrs
Average usage of the treadmill per week : 3 times, 94 miles
Average annual income of consumers : $50500

2.Outlier Detection and Removal

'Income' attribute

In [6]:

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

Out[6]:

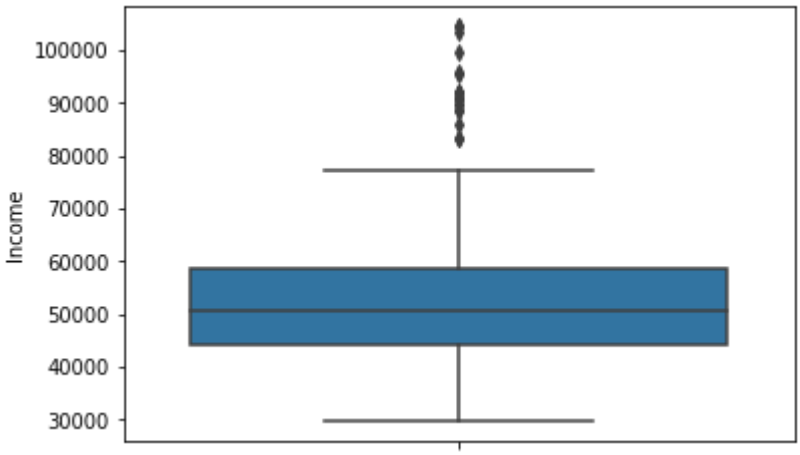
```
count       180.000000
mean      53719.577778
std       16506.684226
min       29562.000000
25%       44058.750000
50%       50596.500000
75%       58668.000000
max      104581.000000
Name: Income, dtype: float64
```

In [7]:

```
sns.boxplot(y = 'Income', data = df)
```

Out[7]:

```
<AxesSubplot:ylabel='Income'>
```



In [8]:

```
df['Income'].mean() - df['Income'].median()
```

Out[8]:

```
3123.0777777777766
```

most salary data is between 44k and 58k, but we notice big difference between mean and median due to some existing outliers

In [9]:

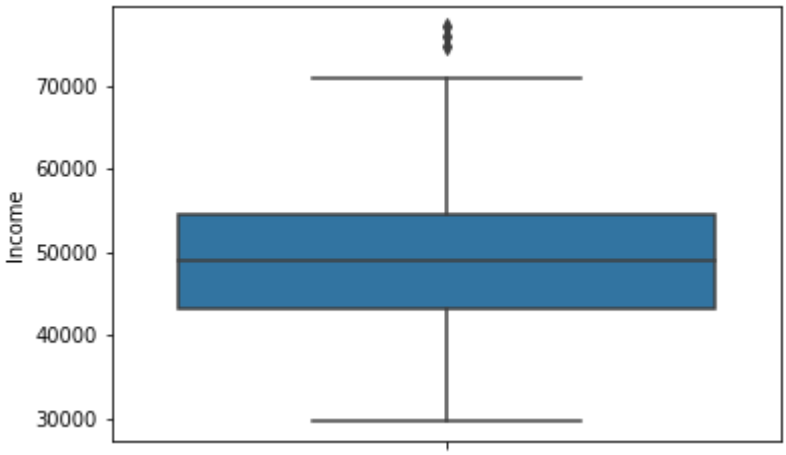
```
#removing the outliers
q1 = df['Income'].quantile(.25)
q3 = df['Income'].quantile(.75)
iqr = q3-q1
df = df[(df['Income'] > q1-1.5*iqr) & (df['Income'] < q3+1.5*iqr)]
```

In [10]:

```
sns.boxplot(y = 'Income', data = df)
```

Out[10]:

```
<AxesSubplot:ylabel='Income'>
```



In [11]:

```
df['Income'].mean() - df['Income'].median()
```

Out[11]:

```
228.18012422360334
```

outliers significantly reduced

Categorizing attributes

Categorizing AGE to a new column 'age_bins'

In [12]:

```
df['age_bins'] = pd.cut(x=df['Age'], bins=[0,20,30,40,50,70,100], labels = ['0-20','20-30',
```

In [13]:

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

Out[13]:

```
20-30     101
30-40      42
0-20       10
40-50       8
50-70       0
70-100      0
Name: age_bins, dtype: int64
```

Categorizing INCOME to a new column 'income_bins'

In [14]:

```
df['income_bins'] = pd.cut(x=df['Income'], bins=[29000,40000,50000,60000,70000,80000], labe
```

In [15]:

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

Out[15]:

```
50k-60k    55
40k-50k    51
29k-40k    32
60k-70k    19
70k-80k     4
Name: income_bins, dtype: int64
```

3.Detailed Analysis

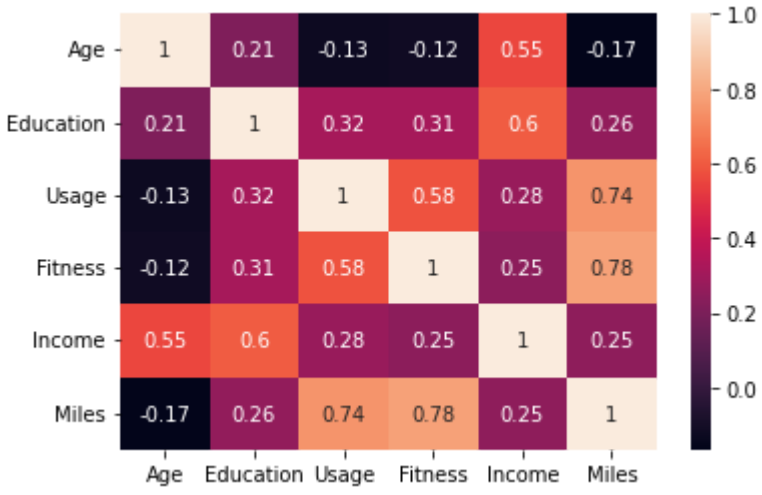
Checing correlation

In [16]:

```
sns.heatmap(df.corr(), annot = True)
```

Out[16]:

```
<AxesSubplot:>
```



> Usage, Miles and Fitness see an obvious correlation.
> **Education, Age and Income** seems to have a high correlation between each other which can be further explored..

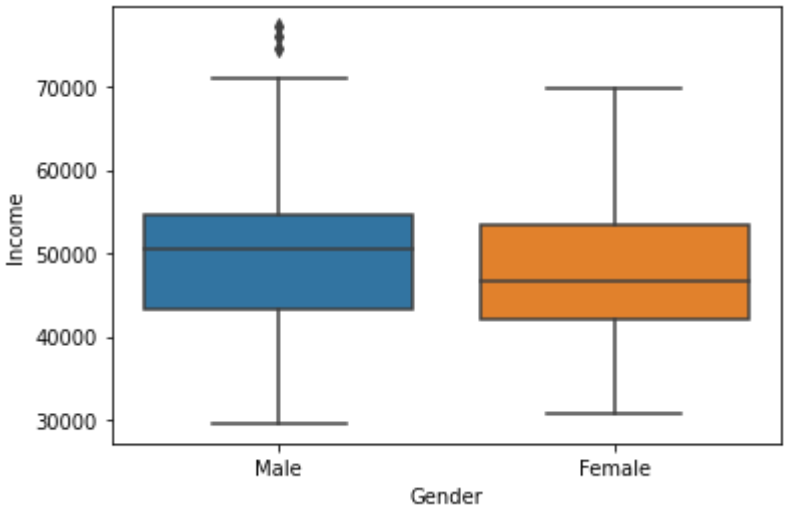
First checking how can *'Gender'* affect

In [17]:

```
sns.boxplot(x = 'Gender', y = 'Income', data = df)
```

Out[17]:

```
<AxesSubplot:xlabel='Gender', ylabel='Income'>
```



In [19]:

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

Out[19]:

```
Gender
Female    48056.356164
Male      50000.840909
Name: Income, dtype: float64
```

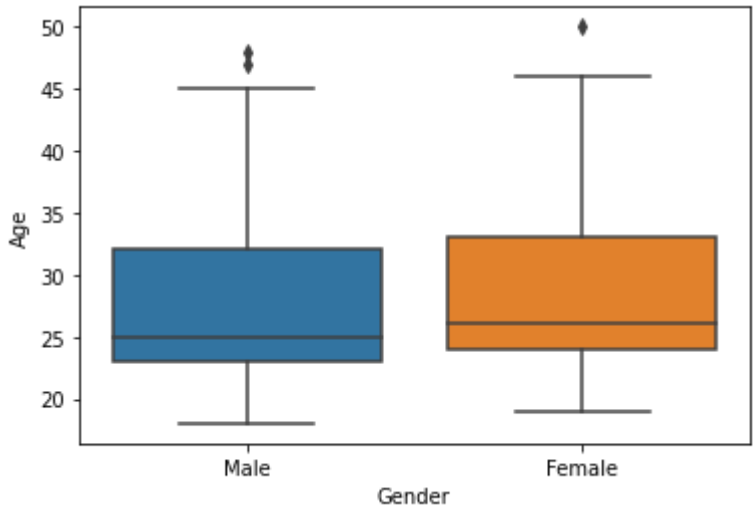
Males earn more than females, so would most likely spend more

In [20]:

```
sns.boxplot(x = 'Gender', y = 'Age', data = df)
```

Out[20]:

```
<AxesSubplot:xlabel='Gender', ylabel='Age'>
```



In [22]:

```
df.groupby('Gender')['Age'].mean()
```

Out[22]:

```
Gender
Female    28.493151
Male      27.875000
Name: Age, dtype: float64
```

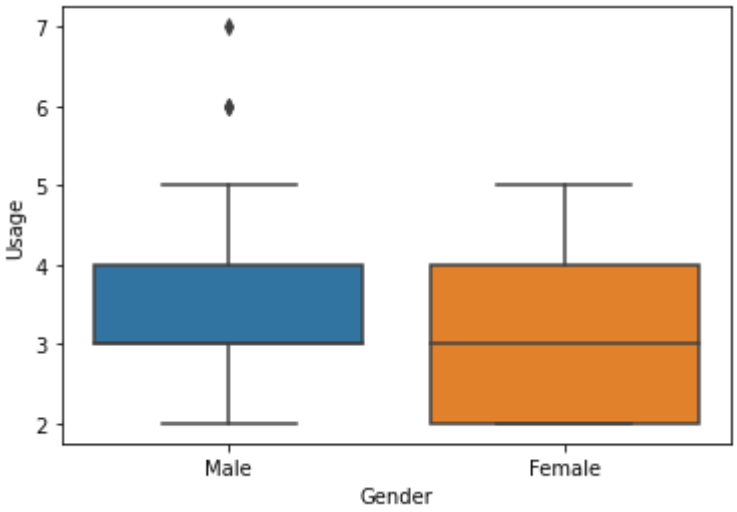
Males tend to purchase the equipment at a younger age while females tend to buy it at an older age.
This could indicate that when compared to males, females get more health concious as they age

In [23]:

```
sns.boxplot(x = 'Gender', y = 'Usage', data = df)
```

Out[23]:

```
<AxesSubplot:xlabel='Gender', ylabel='Usage'>
```



In [24]:

```
df.groupby('Gender')['Usage'].describe()
```

Out[24]:

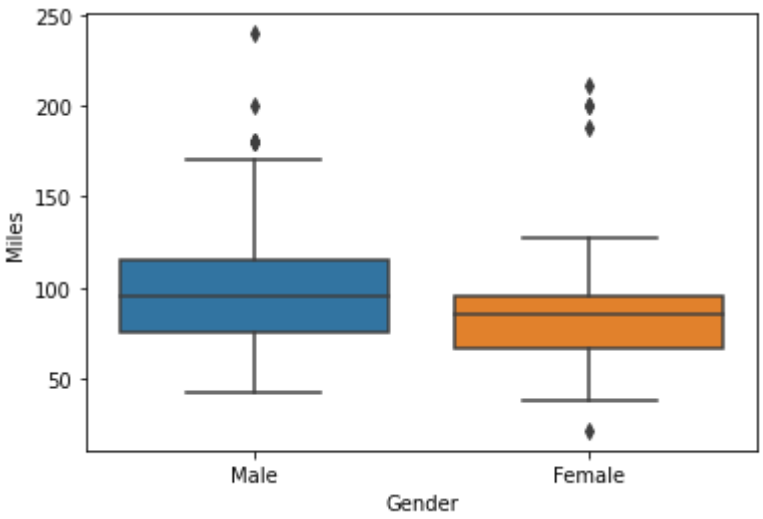
Gender	count	mean	std	min	25%	50%	75%	max
Female	73.0	3.095890	0.915369	2.0	2.0	3.0	4.0	5.0
Male	88.0	3.420455	0.955584	2.0	3.0	3.0	4.0	7.0

In [25]:

```
sns.boxplot(x = 'Gender', y = 'Miles', data = df)
```

Out[25]:

```
<AxesSubplot:xlabel='Gender', ylabel='Miles'>
```



In [26]:

```
df.groupby('Gender')['Miles'].describe()
```

Out[26]:

Gender	count	mean	std	min	25%	50%	75%	max
Female	73.0	84.671233	35.753809	21.0	66.0	85.0	95.00	212.0
Male	88.0	100.386364	40.755913	42.0	75.0	95.0	114.75	240.0

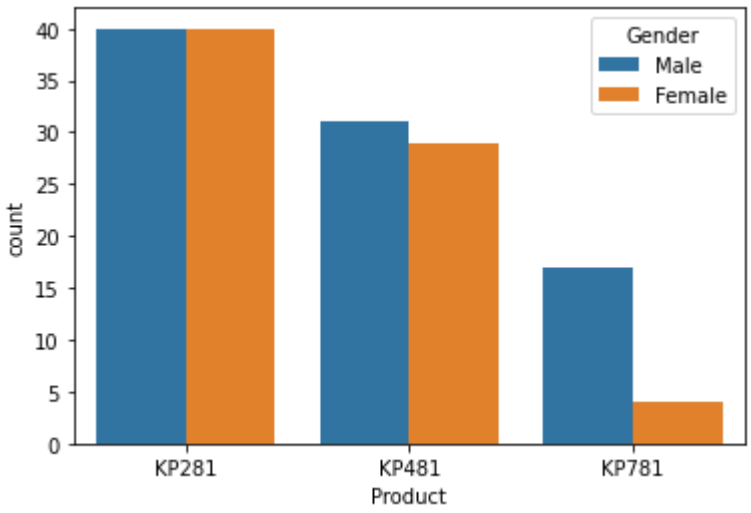
Males use the treadmill much frequently and run more miles than females

In [27]:

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

Out[27]:

```
<AxesSubplot:xlabel='Product', ylabel='count'>
```



In [28]:

```
df.groupby('Product')['Gender'].value_counts()
```

Out[28]:

```
Product  Gender
KP281    Female    40
         Male      40
KP481    Male      31
         Female    29
KP781    Male      17
         Female     4
Name: Gender, dtype: int64
```

Data shows that males give more importance to the quality and features even if it costs more,

while females seem not too interested in spending more on the equipment

In [29]:

```
df.groupby(['Gender','MaritalStatus']).sum()['Miles']
```

Out[29]:

```
Gender  MaritalStatus
Female  Partnered        3633
        Single           2548
Male    Partnered        4926
        Single           3908
Name: Miles, dtype: int64
```

Males and Females with partners seem to be more motivated to purchase the treadmill

Inference: ***Males earn more, they use the treadmill more while they are relatively young and with a partner, and most of them do not mind paying a premium for extra features and quality.***

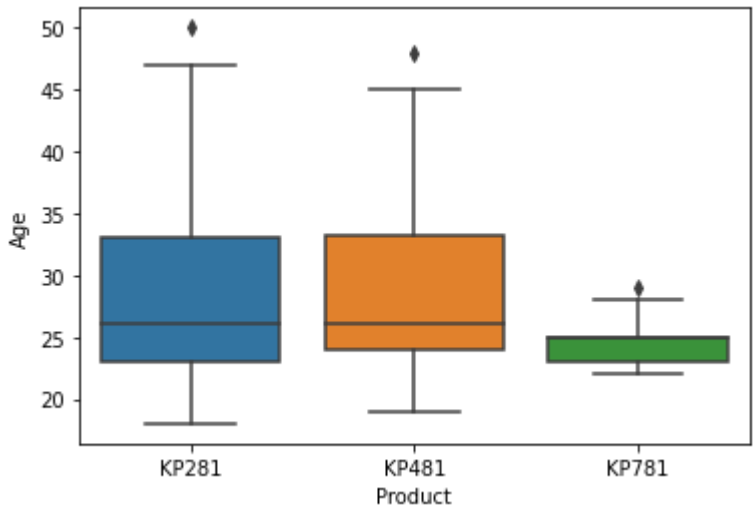
How can *Age and Education* affect?

In [30]:

```
sns.boxplot(x = 'Product', y = 'Age', data = df)
```

Out[30]:

```
<AxesSubplot:xlabel='Product', ylabel='Age'>
```



In [31]:

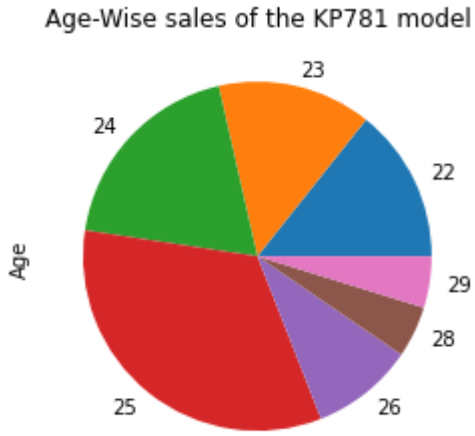
```
df.Product.value_counts()
```

Out[31]:

```
KP281    80
KP481    60
KP781    21
Name: Product, dtype: int64
```

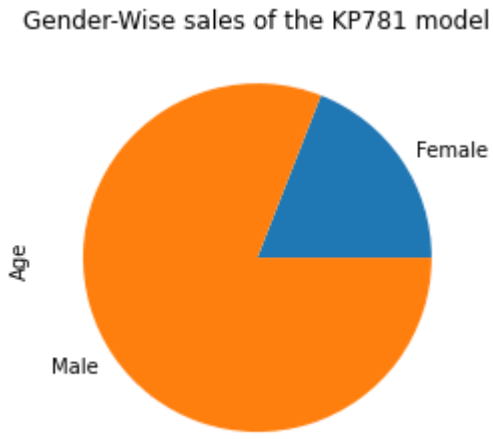
In [32]:

```
df[df.Product == 'KP781'].groupby('Age').Age.count().plot(kind='pie', title = 'Age-Wise sal
plt.show()
```



In [33]:

```
df[df.Product == 'KP781'].groupby('Gender').Age.count().plot(kind='pie', title = 'Gender-Wi
plt.show()
print(df[df.Product == 'KP781'].groupby('Gender').Age.count())
```




```
Gender
Female     4
Male      17
Name: Age, dtype: int64
```

The high-end model KP781 is only bought by consumers in their twenties, mostly 25, and

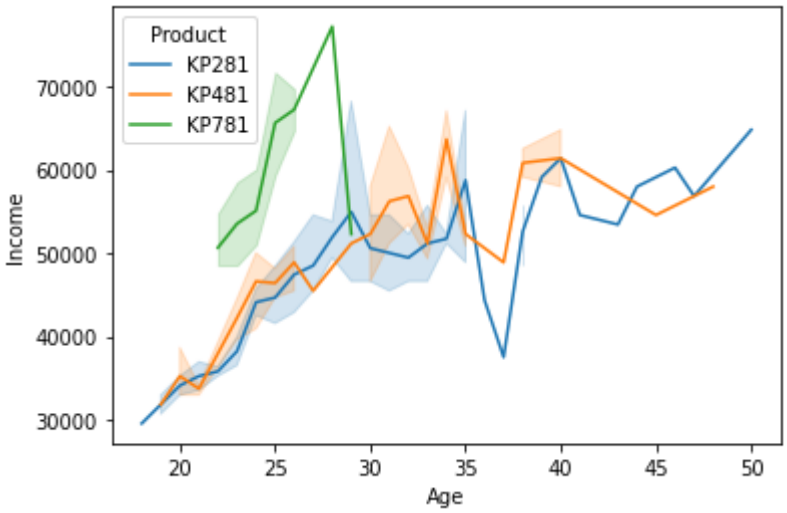
mostly males

In [34]:

```
sns.lineplot(x = 'Age', y = 'Income', data = df, hue = 'Product')
```

Out[34]:

```
<AxesSubplot:xlabel='Age', ylabel='Income'>
```



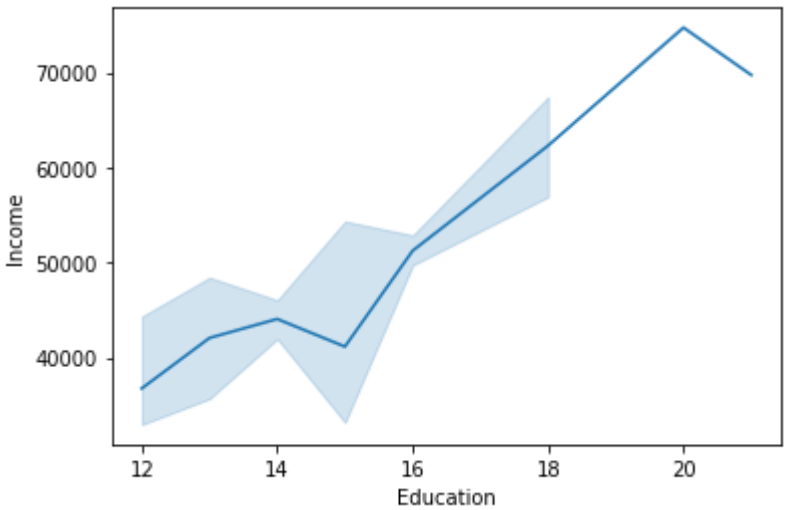
As mentioned before, people between 20 and 30 years of age buy the KP781 model, they are also among the highest earning consumers

In [35]:

```
sns.lineplot(x = 'Education', y = 'Income', data = df)
```

Out[35]:

```
<AxesSubplot:xlabel='Education', ylabel='Income'>
```



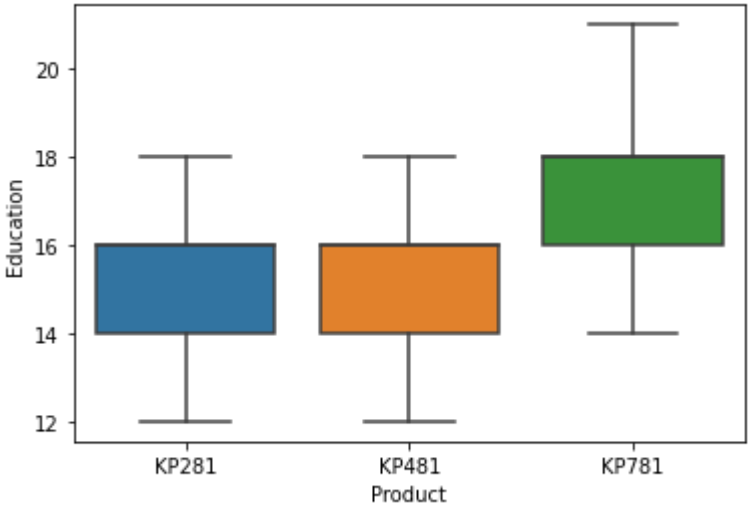
More years of education, more the income

In [36]:

```
sns.boxplot(x = 'Product', y = 'Education', data = df)
```

Out[36]:

```
<AxesSubplot:xlabel='Product', ylabel='Education'>
```



In [37]:

```
df[df.Product == 'KP781'].groupby('Education').Education.count()
```

Out[37]:

```
Education
14    1
16    9
18    9
20    1
21    1
Name: Education, dtype: int64
```

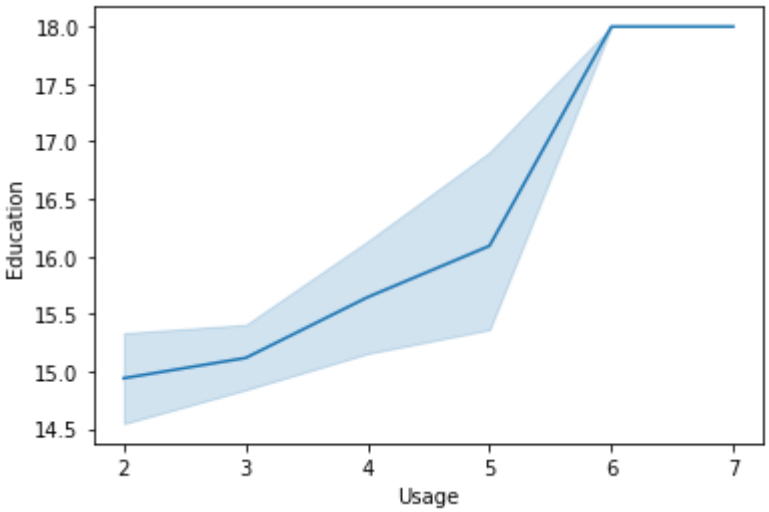
Consumers who typically have more than 16 years of education tend to purchase the high end model.
This could possibly be due to the fact that they earn more. Let's find that out!

In [38]:

```
sns.lineplot(x = 'Usage', y = 'Education', data = df)
```

Out[38]:

```
<AxesSubplot:xlabel='Usage', ylabel='Education'>
```

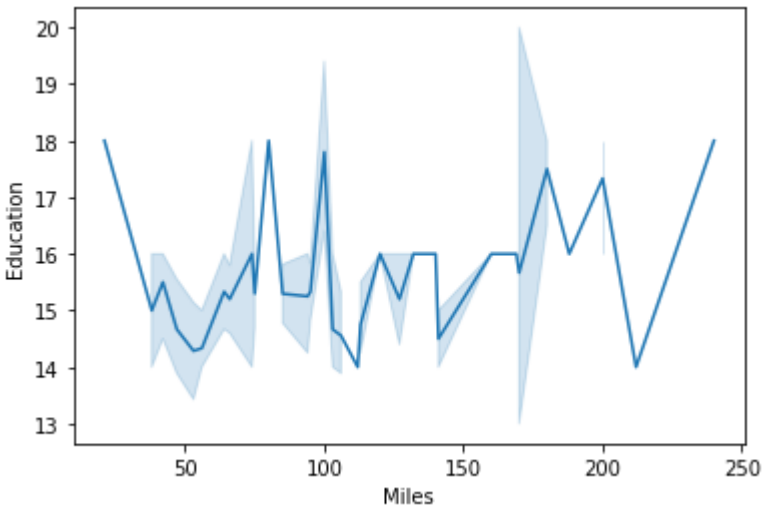


In [39]:

```
sns.lineplot(x = 'Miles', y = 'Education', data = df)
```

Out[39]:

```
<AxesSubplot:xlabel='Miles', ylabel='Education'>
```



> We can deduce that consumers with more education tend to use the treadmill more frequently in a week possibly because of being better aware about the benefits of consistent physical activity

Inference: ***Consumers between the age of 20 and 30 includes the most educated and most frequent treadmill users, and with the highest income they are the only category of consumers who are mostly males that purchases the higher end model 'KP781'***

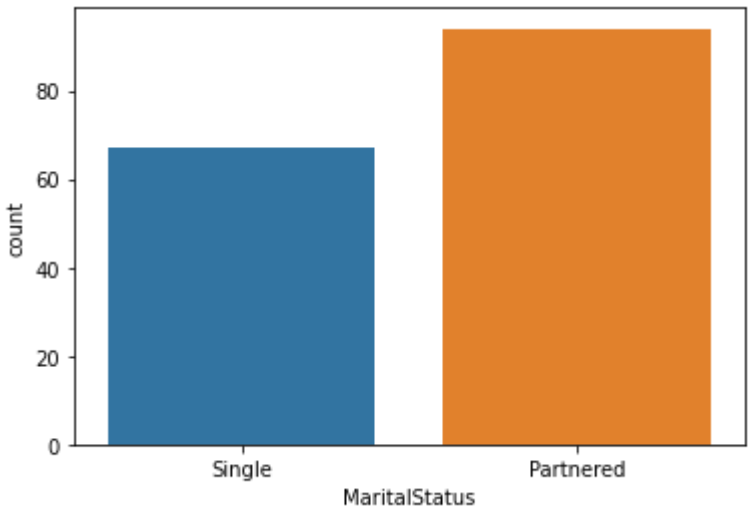
How can *Marital Status* affect?

In [40]:

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

Out[40]:

```
<AxesSubplot:xlabel='MaritalStatus', ylabel='count'>
```



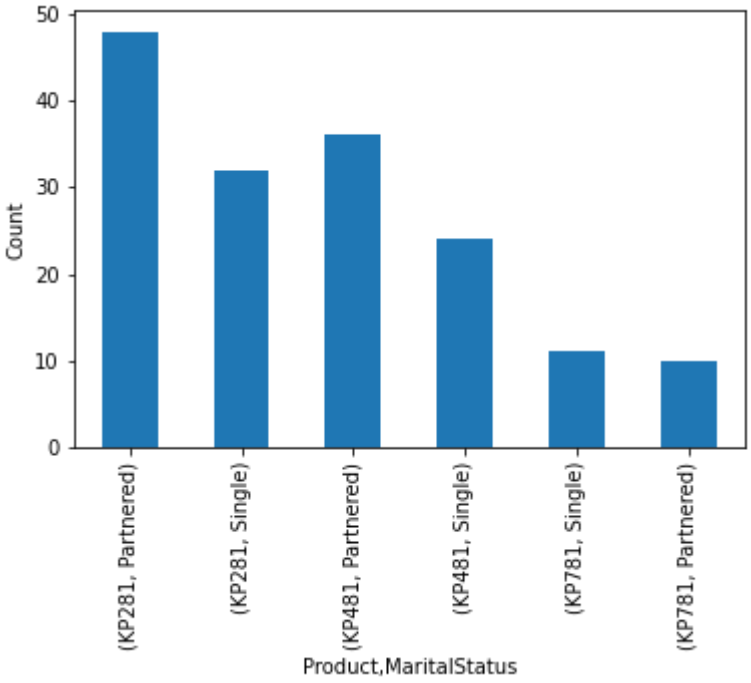
people with partners tend to purchase more

In [41]:

```
df.groupby('Product')['MaritalStatus'].value_counts().plot(kind = 'bar', ylabel = 'Count')
```

Out[41]:

```
<AxesSubplot:xlabel='Product,MaritalStatus', ylabel='Count'>
```



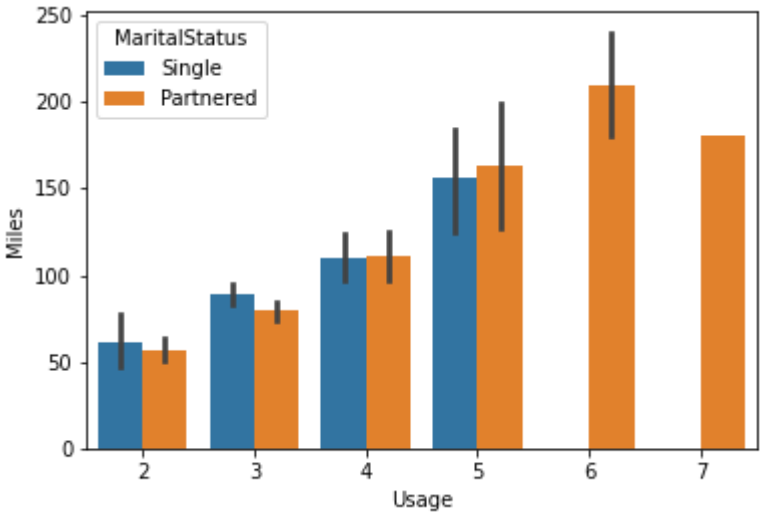
Most partnered people tend to purchase the cheaper version, while the advanced version is slightly more preferred by single consumers

In [42]:

```
sns.barplot(x = 'Usage', y = 'Miles', hue = 'MaritalStatus', data = df)
```

Out[42]:

```
<AxesSubplot:xlabel='Usage', ylabel='Miles'>
```



We can see increased usage patterns for partnered consumers

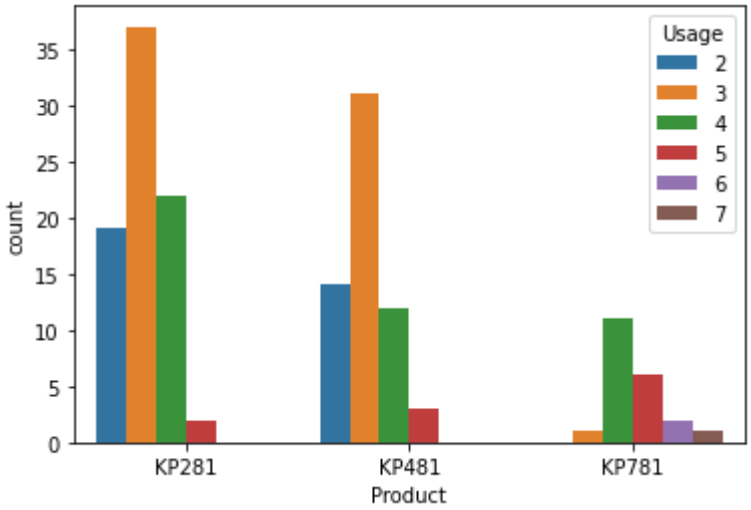
Inference: ***We can see that Partnered consumers are more likely to use the treadmill more frequently, but are not comfortable spending more for advanced features and hence prefer to buy the basic entry level and mid level versions.***

In [43]:

```
sns.countplot(x = 'Product', hue = 'Usage', data = df)
```

Out[43]:

```
<AxesSubplot:xlabel='Product', ylabel='count'>
```



Consumers purchasing the advanced level treadmill uses them more frequently

In [44]:

```
sns.pairplot(df)
```

Out[44]:

```
<seaborn.axisgrid.PairGrid at 0x28b97c95e40>
```

4.Probability

Marginal Probability

In [45]:

```
pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True, normalize = True)*100
```

Out[45]:

Product	KP281	KP481	KP781	All
Gender				
Female	24.844720	18.012422	2.484472	45.341615
Male	24.844720	19.254658	10.559006	54.658385
All	49.689441	37.267081	13.043478	100.000000

Out of 45% females and 54% males:

- 24% of males and females have purchased KP281
- 18-19% of males and females have purchased KP781
- 10% of males and just 2% of females have purchased KP781

In [46]:

```
pd.crosstab(index=df['income_bins'],columns=df['Product'],margins=True, normalize = True)*1
```

Out[46]:

Product	KP281	KP481	KP781	All
income_bins				
29k-40k	14.285714	5.590062	0.000000	19.875776
40k-50k	15.527950	13.043478	3.105590	31.677019
50k-60k	16.149068	14.285714	3.726708	34.161491
60k-70k	3.726708	4.347826	3.726708	11.801242
70k-80k	0.000000	0.000000	2.484472	2.484472
All	49.689441	37.267081	13.043478	100.000000

- Consumers in the income range 40k-60k have purchased the most amount of treadmills

In [47]:

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

Out[47]:

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	29.813665	22.360248	6.211180	58.385093
Single	19.875776	14.906832	6.832298	41.614907
All	49.689441	37.267081	13.043478	100.000000

- Partnered consumers have the highest amount of purchase and they mostly prefer the entry level model

Conditional Probability

Checking conditional probability of key indicators

Product vs Age

In [48]:

```
df.groupby(['Product'])['age_bins'].value_counts()/df.groupby('Product')['age_bins'].count(
```

Out[48]:

```
Product
KP281    20-30      61.250000
         30-40      23.750000
         0-20        7.500000
         40-50       7.500000
         50-70       0.000000
         70-100      0.000000
KP481    20-30      51.666667
         30-40      38.333333
         0-20        6.666667
         40-50       3.333333
         50-70       0.000000
         70-100      0.000000
KP781    20-30     100.000000
         0-20        0.000000
         30-40       0.000000
         40-50       0.000000
         50-70       0.000000
         70-100      0.000000
Name: age_bins, dtype: float64
```

- The probability of a purchase is very high if the consumer is between age 20-30, and sees a

steady decline after 40
- Advanced models are only purchased by consumers between age 20 and 30
- No purchases are made by consumers above age 50

Product vs Income

PRODUCT 1 : KP281

In [49]:

```
df[df.Product == 'KP281'].groupby(['Product'])['income_bins'].value_counts()/df.groupby('Pr
```

Out[49]:

```
Product
KP281    50k-60k    32.50
         40k-50k    31.25
         29k-40k    28.75
         60k-70k     7.50
         70k-80k     0.00
Name: income_bins, dtype: float64
```

32% of consumers between 50k-60k,31% of consumers between 40k-50k and 28% of consumers between 29k-40k have purchased KP281

- This seems to be considered as the best value product as probability of purchase is high for consumers with income between 29k and 60k
- However people with higher income seems to mostly avoid the low end model

PRODUCT 2 : KP481

In [50]:

```
df[df.Product == 'KP481'].groupby(['Product'])['income_bins'].value_counts()/df.groupby('Pr
```

Out[50]:

```
Product
KP481    50k-60k    38.333333
         40k-50k    35.000000
         29k-40k    15.000000
         60k-70k    11.666667
         70k-80k     0.000000
Name: income_bins, dtype: float64
```

38% of consumers between 50k-60k,35% of consumers between 40k-50k and 15% of consumers between 29k-40k have purchased KP481

- Probability of purchase is high for consumers with income between 40k and 60k

- This mid level product is preferred by mid tier consumers, but the top tier consumers do not seem to prefer this

PRODUCT 3 : KP781

In [51]:

```
df[df.Product == 'KP781'].groupby(['Product'])['income_bins'].value_counts()/df.groupby('Pr
```

Out[51]:

```
Product
KP781    50k-60k    28.571429
         60k-70k    28.571429
         40k-50k    23.809524
         70k-80k    19.047619
         29k-40k     0.000000
Name: income_bins, dtype: float64
```

28% of consumers between 50k-60k,28% of consumers between 60k-70k and 23% of consumers between 40k-50k have purchased KP781

- Probability of purchase of this advanced version is high for the consumers between the income range 50k to 70k
- Definetly not preferrend by low income consumers

Product vs Marital Status

In [52]:

```
df.groupby(['Product'])['MaritalStatus'].value_counts()/df.groupby('Product')['MaritalStatu
```

Out[52]:

```
Product  MaritalStatus
KP281    Partnered        60.000000
         Single           40.000000
KP481    Partnered        60.000000
         Single           40.000000
KP781    Single           52.380952
         Partnered        47.619048
Name: MaritalStatus, dtype: float64
```

- Clearly evident that the probability of a consumer is the highest when he/she is not single

Product vs Fitness Rating

In [53]:

```
df.groupby(['Product'])['Fitness'].value_counts()/df.groupby('Product')['Fitness'].count()*
```

Out[53]:

```
Product  Fitness
KP281    3          67.500000
         2          17.500000
         4          11.250000
         5           2.500000
         1           1.250000
KP481    3          65.000000
         2          20.000000
         4          13.333333
         1           1.666667
KP781    5          66.666667
         4          19.047619
         3          14.285714
Name: Fitness, dtype: float64
```

- People who have higher fitness rating and think they are adequately fit has a higher probabaility on purchasing the advanced model
- People who think they have an average level of fitness tend to start with the entry level or mid level model

5.Customer Profiling

Based on all the analysis done above, we can categorize the consumers to 3 key profiles:

1. **High income, young and educated consumers:**
   Highest chance of a purchase, especially the advanced model
2. **Partnered Consumers**
   Probability of a purchase shoots up as a person gets a partner
3. **Entry level consumers**
   Majority of the consumers fall in this category. Thier fitness rating is around 3 to 4 and they often prefer the entry level model

6.Recommendations

- Obtaining a fitness rating from a customer through a survey can help identify the model that they would most likely purchase. Higher the fitness rating the higher they might spend.

- Introducing a new model that sits between the mid level and advanced level model could attract a lot of consumers. The advanced model can be pitched to consumers between age 20-30, and that can be used as an anchor to sell this newly introduced model.

- Special offers or coupons can be provided to partnered consumers in order to upsell the mid level and advanced level models. A referal bonus could also boost sales among this category of consumers.

- Discounts for females on the advanced models.

- Entry level consumers can be given discounts to motivate them to upgrade to a newer model.