Cardio Good Fitness Project / Yeoman Yoon.

Import pandas, numpy, seaborn, etc for analysis

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
pd.set_option('display.float_format', lambda x: '%.5f' % x) # To supress numer
ical display in scientific notations
```

Store the data using pandas

In [2]:
```
data = pd.read_csv('CardioGoodFitness.csv')
```

Dataset:

- Product - the model no. of the treadmill
- Age - in no of years, of the customer
- Gender - of the customer
- Education - in no. of years, of the customer
- Marital Status - of the customer
- Usage - Avg. # times the customer wants to use the treadmill every week
- Fitness - Self rated fitness score of the customer (5 - very fit, 1 - very unfit)
- Income - of the customer
- Miles- expected to run

Quick overview of the data

In [3]:
```
data.isnull().sum() # data is not missing any values.
```

Out[3]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

In [4]:
```
data.head(10)
```

Out[4]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	TM195	18	Male	14	Single	3	4	29562	112
1	TM195	19	Male	15	Single	2	3	31836	75
2	TM195	19	Female	14	Partnered	4	3	30699	66
3	TM195	19	Male	12	Single	3	3	32973	85
4	TM195	20	Male	13	Partnered	4	2	35247	47
5	TM195	20	Female	14	Partnered	3	3	32973	66
6	TM195	21	Female	14	Partnered	3	3	35247	75
7	TM195	21	Male	13	Single	3	3	32973	85
8	TM195	21	Male	15	Single	5	4	35247	141
9	TM195	21	Female	15	Partnered	2	3	37521	85

In [5]:
```
data.describe()
```

Out[5]:

	Age	Education	Usage	Fitness	Income	Miles
count	180.00000	180.00000	180.00000	180.00000	180.00000	180.00000
mean	28.78889	15.57222	3.45556	3.31111	53719.57778	103.19444
std	6.94350	1.61705	1.08480	0.95887	16506.68423	51.86360
min	18.00000	12.00000	2.00000	1.00000	29562.00000	21.00000
25%	24.00000	14.00000	3.00000	3.00000	44058.75000	66.00000
50%	26.00000	16.00000	3.00000	3.00000	50596.50000	94.00000
75%	33.00000	16.00000	4.00000	4.00000	58668.00000	114.75000
max	50.00000	21.00000	7.00000	5.00000	104581.00000	360.00000

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
Product          180 non-null object
Age              180 non-null int64
Gender           180 non-null object
Education        180 non-null int64
MaritalStatus    180 non-null object
Usage            180 non-null int64
Fitness          180 non-null int64
Income           180 non-null int64
Miles            180 non-null int64
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

See how many products we have

In [7]:
```
data.Product.unique()
```

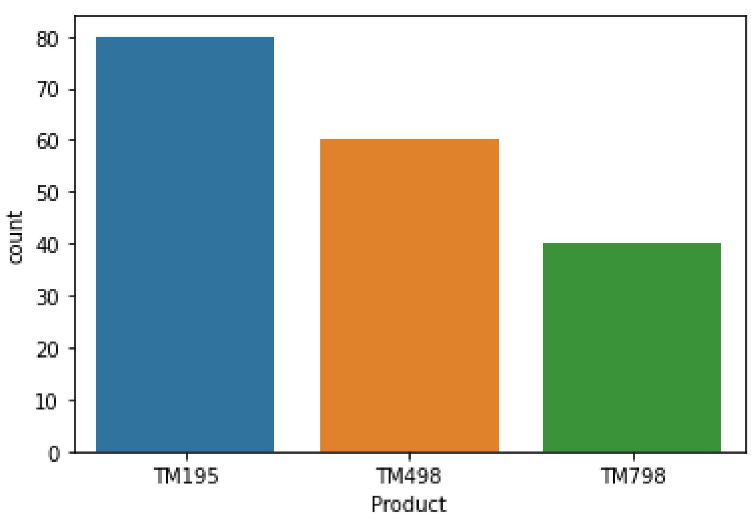
Out[7]:
```
array(['TM195', 'TM498', 'TM798'], dtype=object)
```

Observation:

Univariable analysis

In [8]:
```
sns.countplot(data['Product'])
```

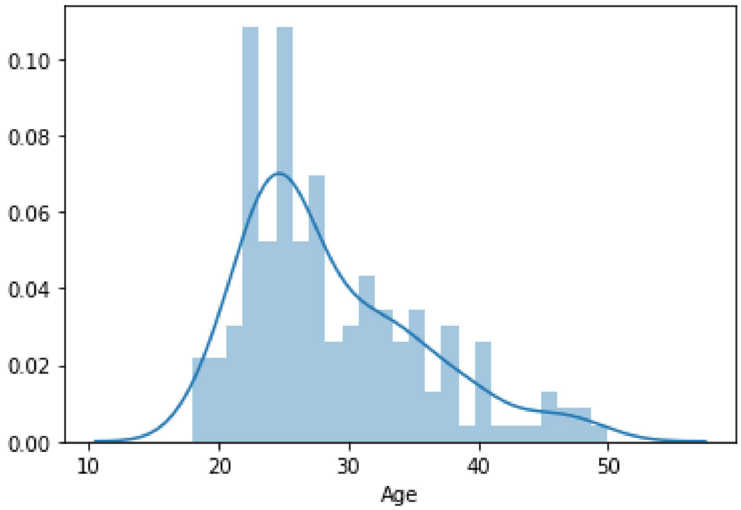
Out[8]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x275fb70c388>
```



Assuming there is no bias in the data, TM195 is the most popular product among consumers followed by TM498 and TM798.

In [9]:
```
sns.distplot(data['Age'], bins=25)
# plt.hist(data['Age'], bins =25)
```

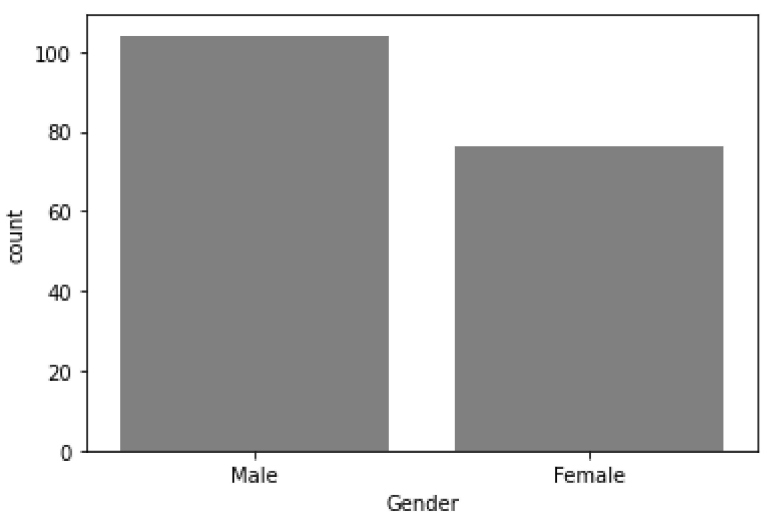
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x275ff917708>



As written in the data description, mean is 28.79 and median is 26. It is slightly right skewed.

In [10]:
```
sns.countplot(data['Gender'], color = 'grey') # removed colors since there is only two categories
```

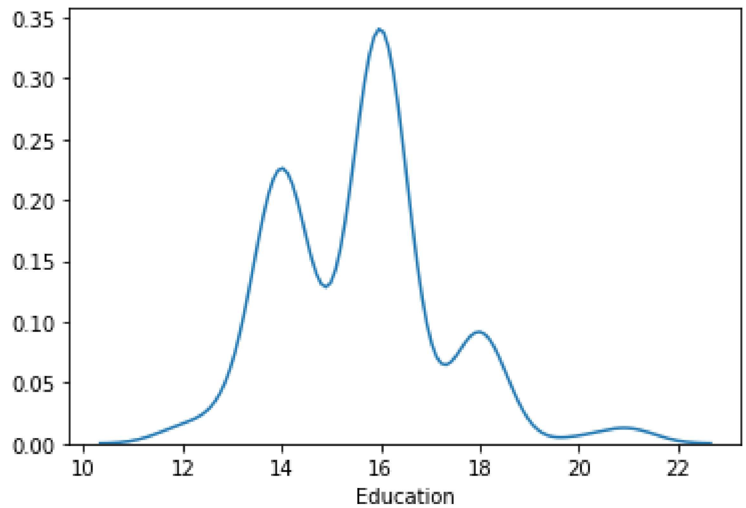
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x275ff9e3188>



Assuming the data is collected randomly, we have slighlty more male buyers.

In [11]:
```
sns.distplot(data['Education'], hist = False) # removed hist for visual purpose. The bars are sticking too high.
```

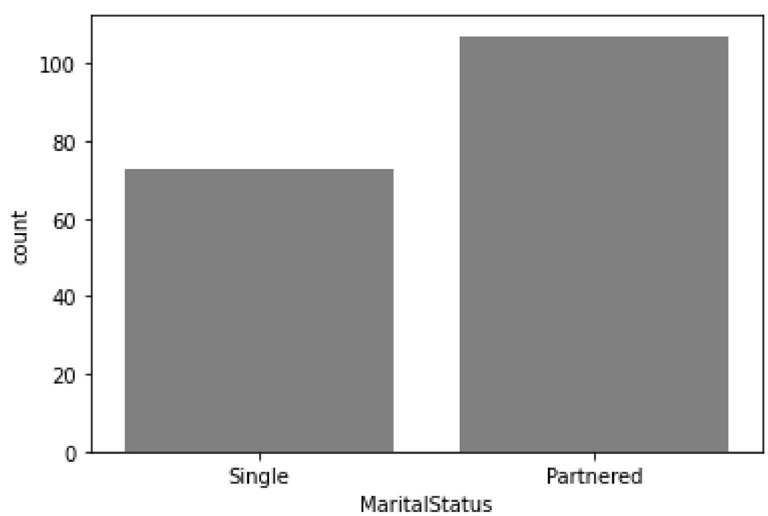
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x275ffa2d848>



Buyers have 15.57 years of education. Seems to have gap (camel shape) becuase many collages have curriculum of either 2 years or 4 years.

In [12]:
```
sns.countplot(data['MaritalStatus'], color = 'grey') # removed colors since there is only two categories
```

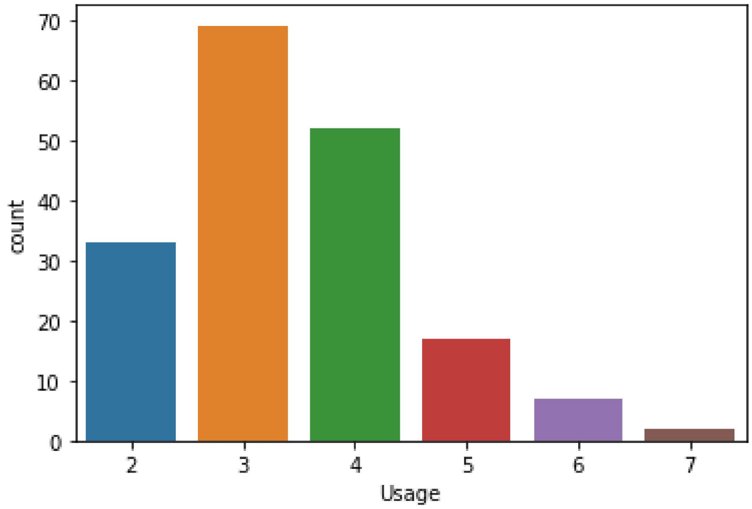
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x275ffabe208>



We have more partnered buyers. It is considered independent from gender with univariable analysis.

In [13]:
```
sns.countplot(data['Usage'])
```

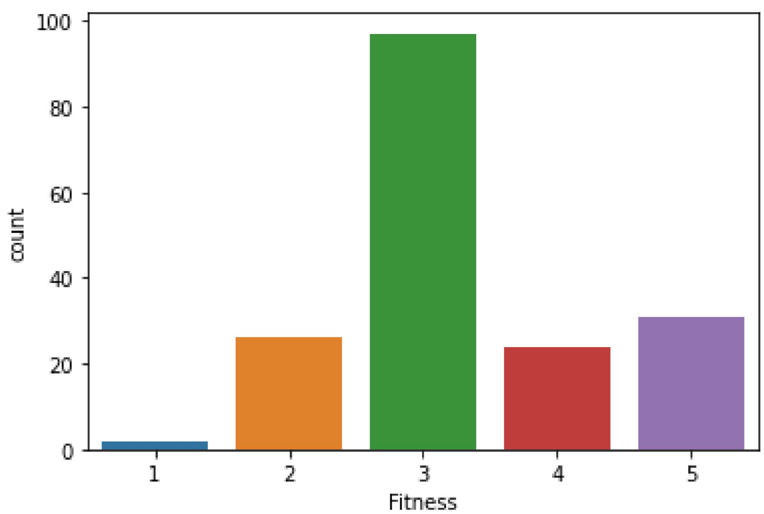
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x275ffb09608>



Overall, buyers use the treadmill 3.4 times per week.

In [14]:
```
sns.countplot(data['Fitness'])
```

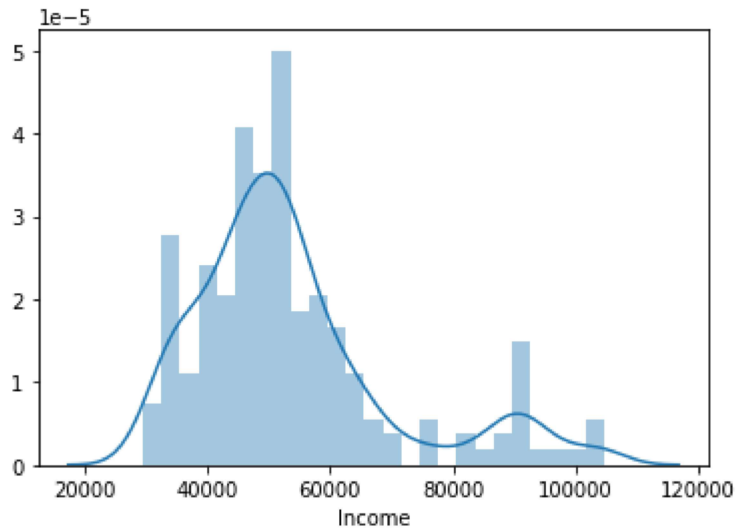
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x275ffb72908>



Buyers evaluate themselves to have 3.31/5.00 fitness level.

In [15]:
```
sns.distplot(data['Income'], bins=25)
```

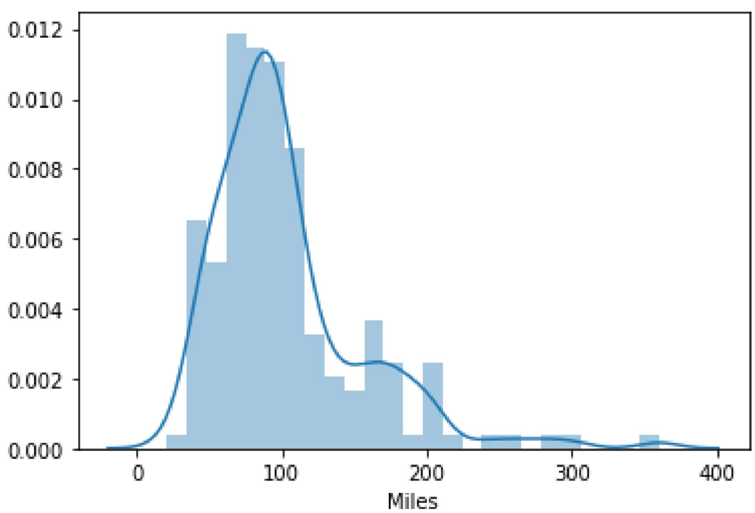
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x275ffbead48>



Buyers make about $53k per year in average.
It is important that graph is making somewhat of camel shape around $50,000 area and $90,000 area.
Later we can observe if the grouping happens.

In [16]:
```
sns.distplot(data['Miles'], bins = 25)
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x275ffca8088>



Buyers run around 103 Miles (per week). Some runs very heavily.

Conclusion from Univariable:

From univariable analysis, we could see the average buyers are 26 years old, have 15.73 years of education, make $ 53k per year, use treadmill 3.4 times per week, have 3.31/5.00 fitness level, and run 103 miles (per week).
This shows that the buyers are mostly young and healthy runners. Knowing this information, we should do analysis on product.

Before going into Multivariable Analysis, we can imagine how the correlation is going to be like. Usage and Running Miles must have high correlations, Age and Income are also expected to have high correlation.

Multivariable Analysis

Quick overview in correlations of data.

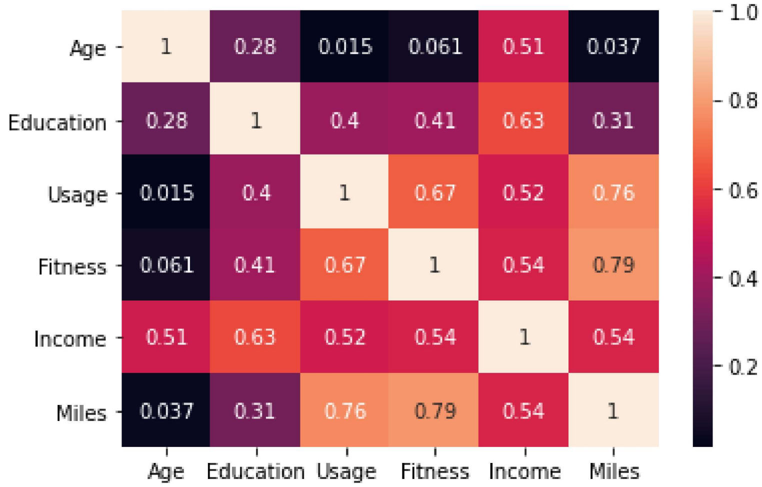
Lets observe the correlation of each data as well.

In [17]:
```
sns.pairplot(data)
```

Out[17]: <seaborn.axisgrid.PairGrid at 0x275ffd29b48>

In [18]:
```
sns.heatmap(data.corr(), annot=True)
```

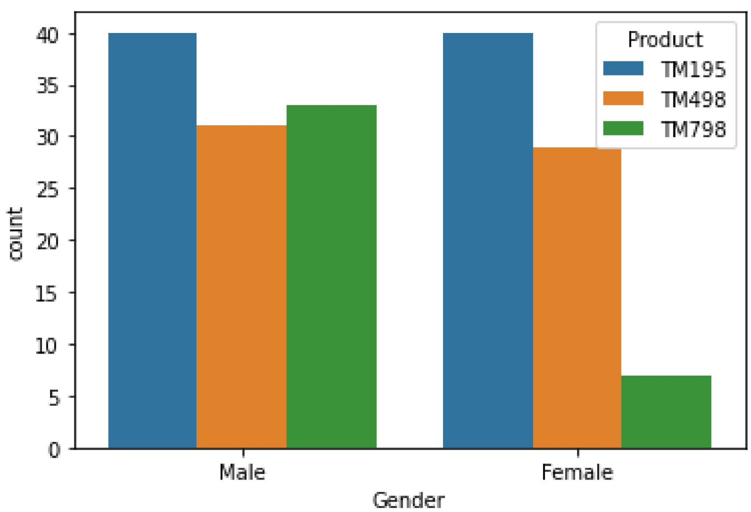
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x2759001d348>



Now, Lets see how the customer profile works for each products.

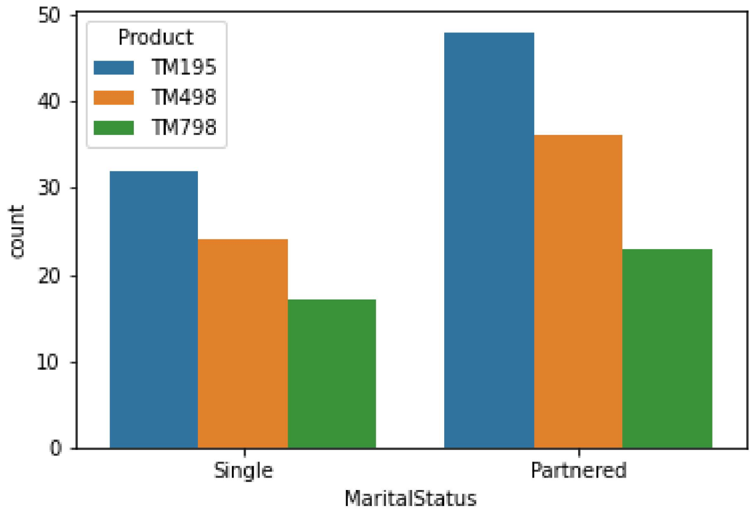
Categorical vs Categorical

In [19]:
```
# sns.countplot(data['Product'], hue= data['Gender']);
sns.countplot(data['Gender'], hue= data['Product']);
```



Females don't prefer buying TM798.

In [20]:
```
# sns.countplot(data['Product'], hue= data['MaritalStatus']);
sns.countplot(data['MaritalStatus'], hue= data['Product']);
```



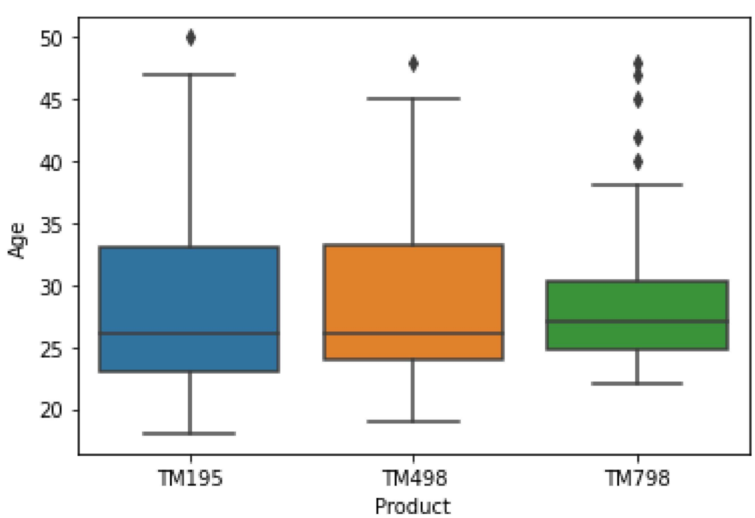
We have more partnered people buying the product generally. It seems very normal, because data is collected from people older than 18. The proportion of purchased product seems consistent over marrage.

Knowing that we only have little females buying TM798, it is likely most or singles customers who bought TM798 are male.

Categorical vs Numerical

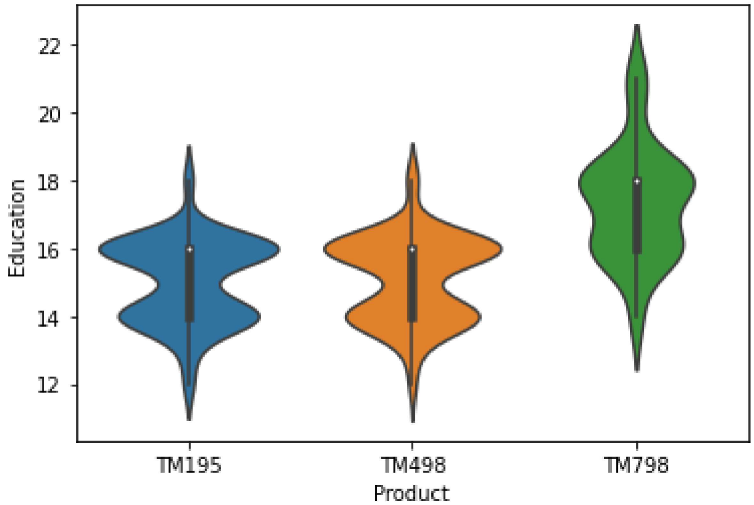
I chose box plot or violin plot because this data in particular looks hallow since we only have 180 data.

In [21]:
```
sns.boxplot( data['Product'], data['Age']); # I chose box plot over violin bec
ause it shows median more clear
```



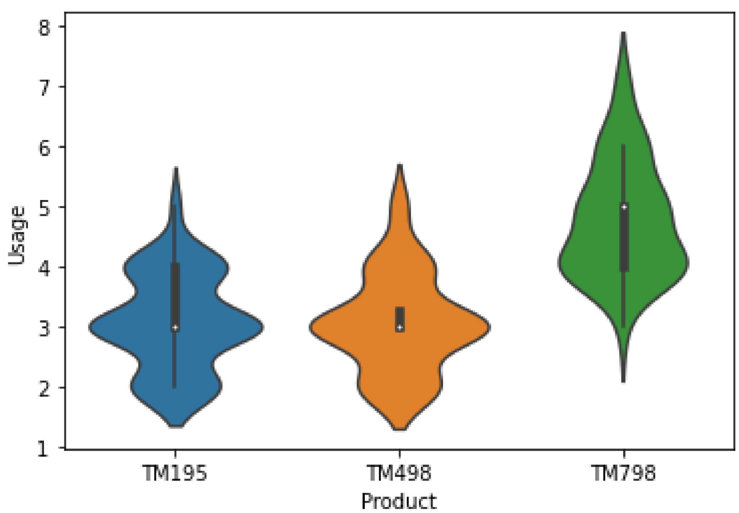
Most of Treadmill buyers are rather young. Although we have some TM798 buyers over age of 40, most of them are 25-30 years old.

In [22]:
```
sns.violinplot(data['Product'], data['Education']);
```



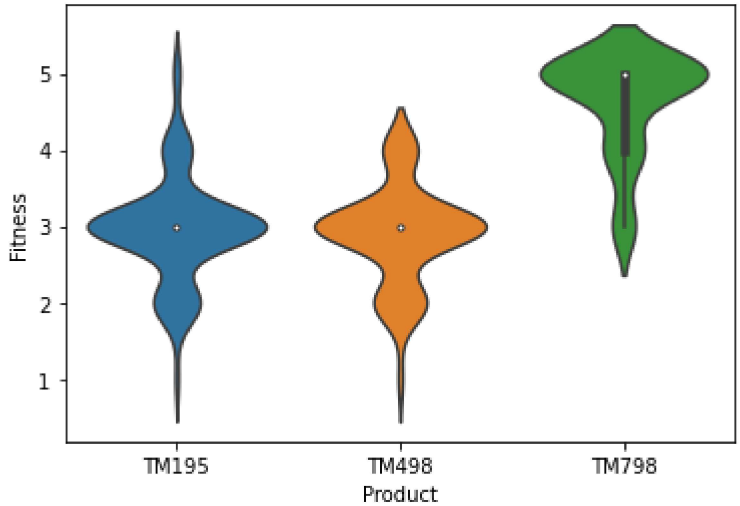
TM195, TM498 buyers have very similar education background. But, TM798 seems to have higher education.

In [23]:
```
sns.violinplot(data['Product'], data['Usage']);
```



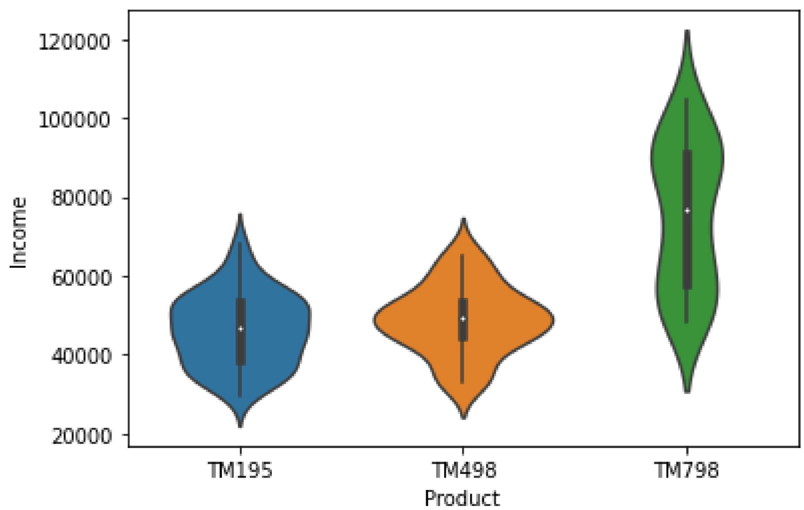
TM798 buyers use treadmill more often than TM195 and TM498 buyers.

In [24]:
```
sns.violinplot(data['Product'], data['Fitness']);
```



TM798 buyers seem to evaluate themselves to be more fit, and they are probably more fit as well. Ppeople who buys treadmill are probably already somewhat fit or are wanting to be more fit.)

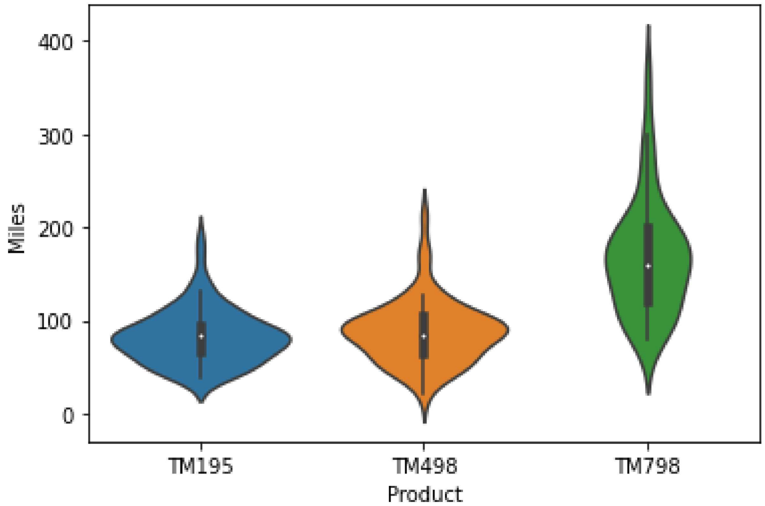
In [25]:
```
sns.violinplot(data['Product'], data['Income']);
```



Most, if not all, TM195 and TM498 buyers make less than 80k per year. Yet, TM798 buyers have pretty even distribution of income .

This can be an indicator that TM798 buyers are wealthy enough or they value exercise more.

In [26]:
```
sns.violinplot(data['Product'], data['Miles']);
```



TM798 buyers actually run more miles than TM195 and TM498 buyers.
Most people, if not all, who run more than 250 miles use TM798.

Conclusion and Recommendation

Main customers are relatively young. This should be interpreted in two ways:

1. **Since The running activity is more popular to younger people, it is important to let marketing teams target younger customers when advertising. Let them choose the right platform and use suitable models.**
2. **Even though running is popular for the younger generation, older people also desire to exercise. The R & D Team (Research and Development Team) should focus on creating a new product model that targets older people. (e.g. Stairs treadmill, bicycle) Older people are willing to spend more in buying products as long as the product quality supports it.**

It is observed that TM798 is more popular for intense runners and people who are wealthier. (Assume the product is more expensive as well.) When new technology gets developed, it is better to implement it on TM798 over other products because the price is not the attractive point of TM798. It is not as critical to raise the price for TM798 as long as it supports advanced technology.

In [ ]:
```
```