f5eut74lz

February 28, 2023

1 Cardio Good Fitness Case Study

1.1 Introduction

- The market research team at AdRight is assigned the task to identify the profile of the typical customer for each treadmill product offered by CardioGood Fitness.
- The market research team decides to investigate whether there are differences across the product lines with respect to customer characteristics.
- The team decides to collect data on individuals who purchased a treadmill at a CardioGood-Fitness retail store during the prior three months.
- The data are stored in the CardioGoodFitness.csv file

1.2 The team identifies the following customer variables to study:

- product purchased, TM195, TM498, or TM798
- gender;
- age, in years;
- education, in years;
- relationship status, single or partnered;
- annual household income ;
- average number of times the customer plans to use the treadmill each week;
- average number of miles the customer expects to walk/run each week;
- self-rated fitness on an 1-to-5 scale, where 1 is poor shape and 5 is excellent shape.

1.3 Descriptive Statistics

1.3.1 Definition

- Descriptive statistics are used to describe the basic features of the data in a study. They provide simple summaries about the sample and the measures. Together with simple graphics analysis, they form the basis of virtually every quantitative analysis of data.
- Descriptive statistics are typically distinguished from inferential statistics. With descriptive statistics we are simply describing what is or what the data shows. With inferential statistics, we are trying to reach conclusions that extend beyond the immediate data alone.
  - For instance, we use inferential statistics to try to infer from the sample data what the population might think. Or, we use inferential statistics to make judgments of the probability that an observed difference between groups is a dependable one or one that

might have happened by chance in this study. Thus, we use inferential statistics to make inferences from our data to more general conditions; we use descriptive statistics simply to describe what's going on in our data.

- Descriptive statistics provide a powerful summary that may enable comparisons across people or other units.
- Univariate Analysis :- It involves the examination of the single feature
- The distribution is a summary of the frequency of individual values or ranges of values for a variable. The simplest distribution would list every value of a variable and the number of persons who had each value.
  - For instance, a typical way to describe the distribution of college students is by year in college, listing the number or percent of students at each of the four years. Or, we describe gender by listing the number or percent of males and females. In these cases, the variable has few enough values that we can list each one and summarize how many sample cases had the value. But what do we do for a variable like income or GPA? With these variables there can be a large number of possible values, with relatively few people having each one. In this case, we group the raw scores into categories according to ranges of values. For instance, we might look at GPA according to the letter grade ranges. Or, we might group income into four or five ranges of income values.
- The central tendency The central tendency of a distribution is an estimate of the "center" of a distribution of values. There are three major types of estimates of central tendency:
  - Mean
  - Median
  - Mode
- The dispersion, it refers to the spread of the values around the central tendency. There are two common measures of dispersion, the range and the standard deviation.
  - The range is simply the highest value minus the lowest value. In our example distribution, the high value is 36 and the low is 15, so the range is 36 - 15 = 21.
  - The Standard Deviation is a more accurate and detailed estimate of dispersion because an outlier can greatly exaggerate the range (as was true in this example where the single outlier value of 36 stands apart from the rest of the values. The Standard Deviation shows the relation that set of scores has to the mean of the sample. the formula for the standard deviation:
  - the square root of the sum of the squared deviations from the mean divided by the number of scores minus one.

1.4 Importing libraries

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

1.5 Loading the Dataset

[3]:
```
db = pd.read_csv('C:/Users/suman/Desktop/DS learn/Project/Study/
↪CardioGoodFitness/CardioGoodFitness.csv')
```

[4]:
```
db.head()
```

[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]:
```
db.describe().T
```

[5]:

	count	mean	std	min	25%	50%	\
Age	180.0	28.788889	6.943498	18.0	24.00	26.0	
Education	180.0	15.572222	1.617055	12.0	14.00	16.0	
Usage	180.0	3.455556	1.084797	2.0	3.00	3.0	
Fitness	180.0	3.311111	0.958869	1.0	3.00	3.0	
Income	180.0	53719.577778	16506.684226	29562.0	44058.75	50596.5	
Miles	180.0	103.194444	51.863605	21.0	66.00	94.0	

	75%	max
Age	33.00	50.0
Education	16.00	21.0
Usage	4.00	7.0
Fitness	4.00	5.0
Income	58668.00	104581.0
Miles	114.75	360.0

[6]:
```
db.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
```

[7]:
```
db.shape
```

[7]:
```
(180, 9)
```

[8]:
```
db.isna().any()
```

[8]:
```
Product          False
Age              False
Gender           False
Education        False
MaritalStatus    False
Usage            False
Fitness          False
Income           False
Miles            False
dtype: bool
```

1.5.1 No Missing values in the dataset.

[9]:
```
db.describe(include='all')
```

[9]:

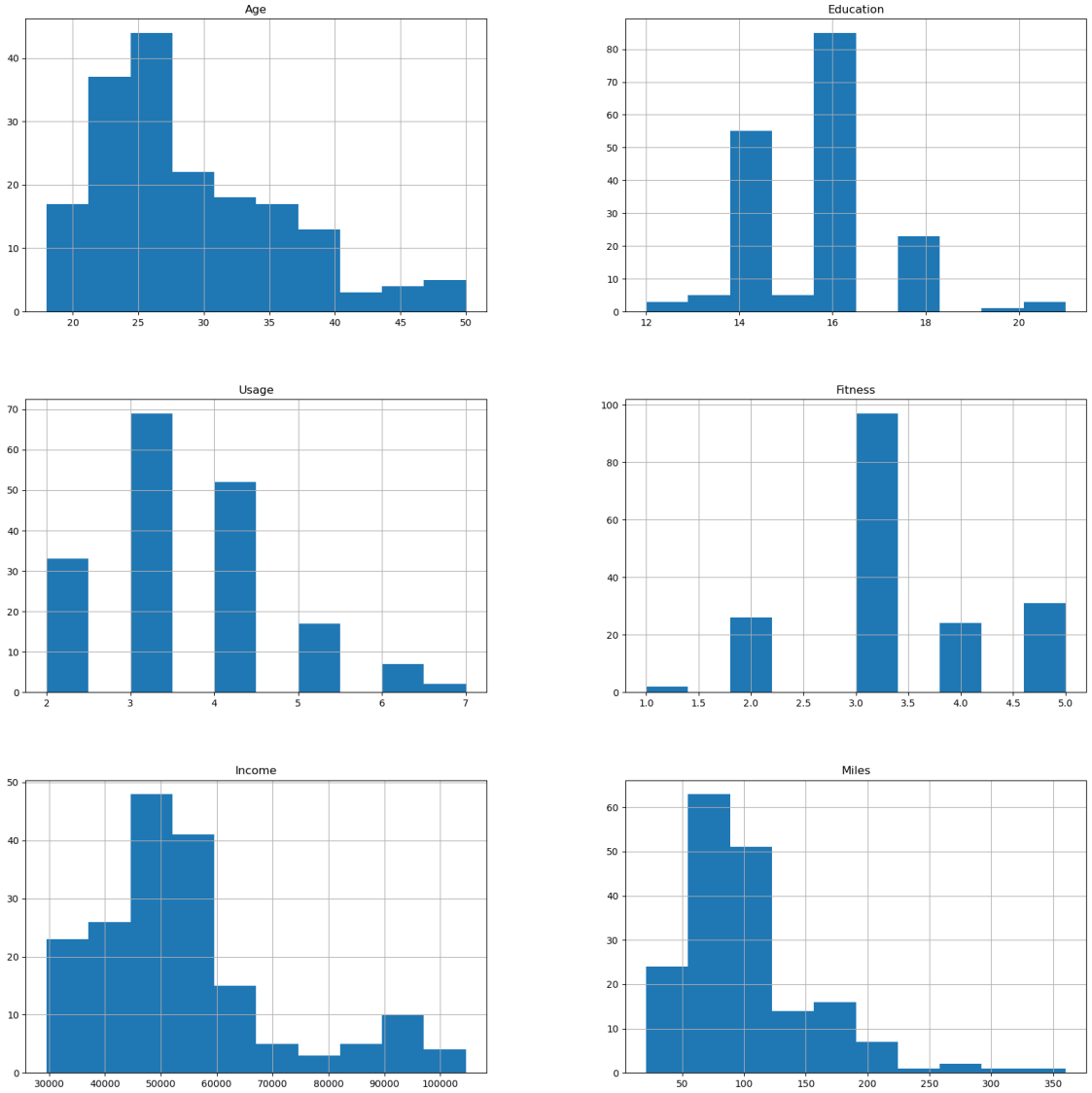
	Product	Age	Gender	Education	MaritalStatus	Usage	\
count	180	180.000000	180	180.000000	180	180.000000	
unique	3	NaN	2	NaN	2	NaN	
top	TM195	NaN	Male	NaN	Partnered	NaN	
freq	80	NaN	104	NaN	107	NaN	
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	

	Fitness	Income	Miles
count	180.000000	180.000000	180.000000
unique	NaN	NaN	NaN
top	NaN	NaN	NaN
freq	NaN	NaN	NaN
mean	3.311111	53719.577778	103.194444
std	0.958869	16506.684226	51.863605
min	1.000000	29562.000000	21.000000
25%	3.000000	44058.750000	66.000000
50%	3.000000	50596.500000	94.000000

```
75%        4.000000    58668.000000   114.750000
max        5.000000   104581.000000   360.000000
```

[10]:
```
db.hist(figsize=(20,20))
```

[10]:
```
array([[<AxesSubplot:title={'center':'Age'}>,
        <AxesSubplot:title={'center':'Education'}>],
       [<AxesSubplot:title={'center':'Usage'}>,
        <AxesSubplot:title={'center':'Fitness'}>],
       [<AxesSubplot:title={'center':'Income'}>,
        <AxesSubplot:title={'center':'Miles'}>]], dtype=object)
```



1.6 Boxplots

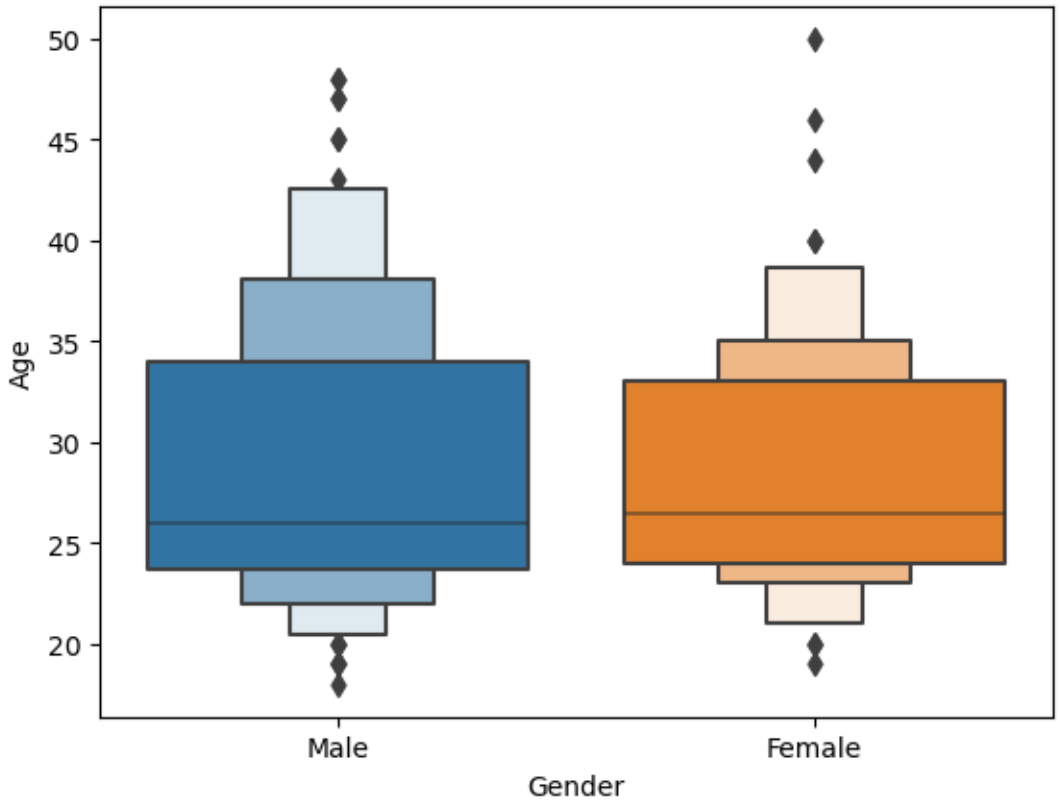
Which is the most popular model by gender?

[12]:
```
pd.crosstab(db['Product'],db['Gender'] )
```

[12]:
```
Gender   Female  Male
Product
TM195        40    40
TM498        29    31
TM798         7    33
```

[13]:
```
sns.boxenplot(x='Gender',y='Age',data=db)
```

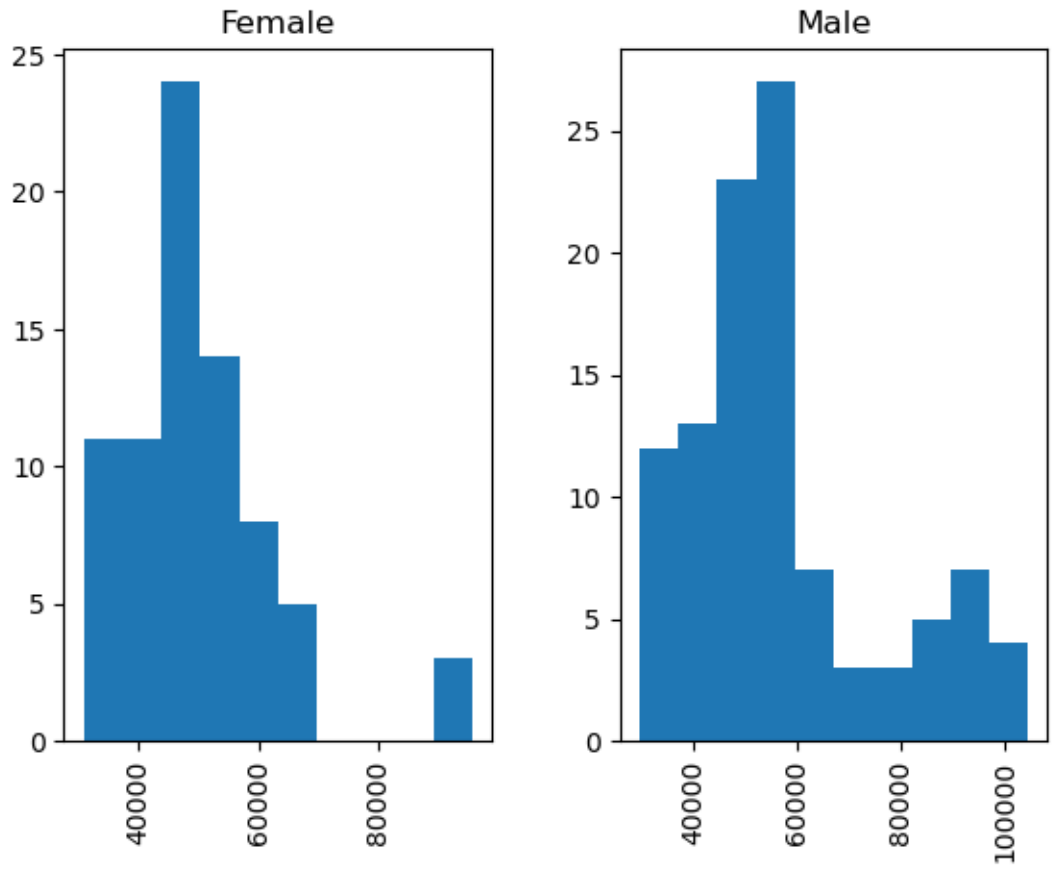
[13]:
```
<AxesSubplot:xlabel='Gender', ylabel='Age'>
```



1.6.1 Seperated data by Gender.

[14]:
```
db.hist(by='Gender',column = 'Income')
```

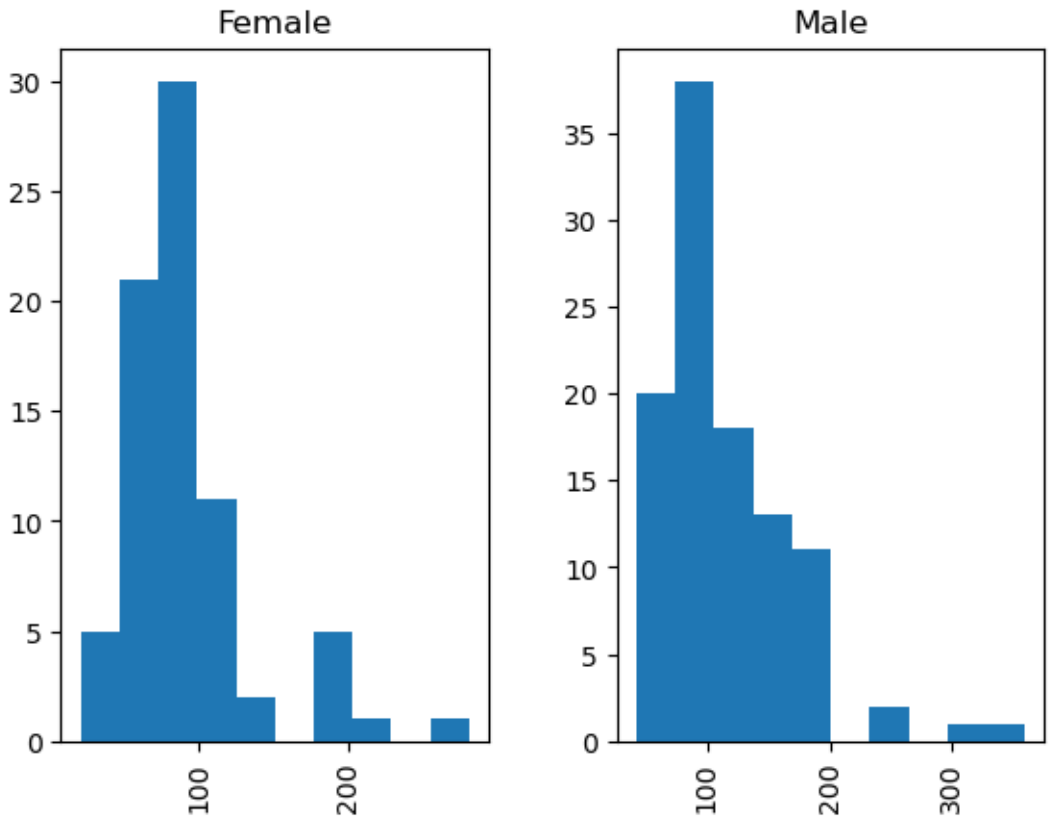
[14]:
```
array([<AxesSubplot:title={'center':'Female'}>,
       <AxesSubplot:title={'center':'Male'}>], dtype=object)
```



1.6.2 Gender v/s Miles

[22]:
```
db.hist(by='Gender',column = 'Miles')
```

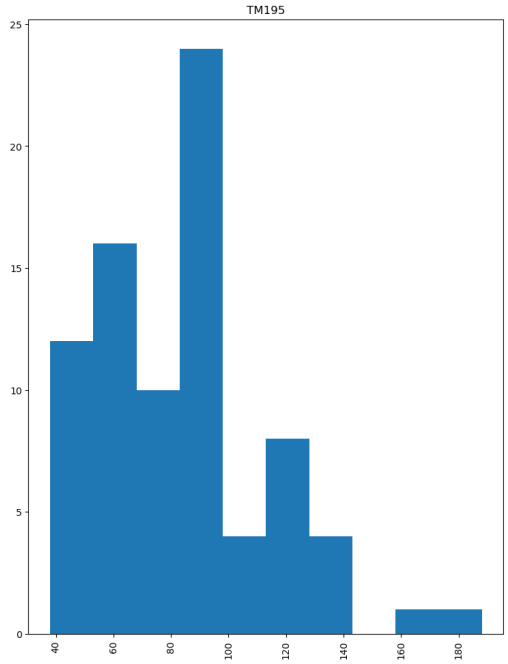
[22]:
```
array([<AxesSubplot:title={'center':'Female'}>,
       <AxesSubplot:title={'center':'Male'}>], dtype=object)
```

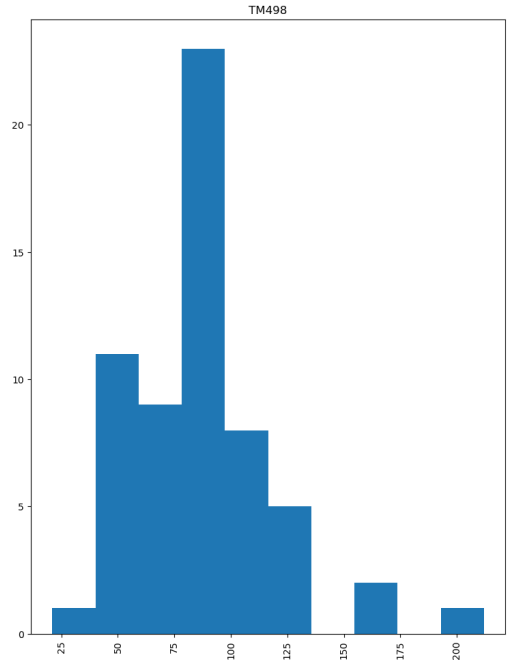


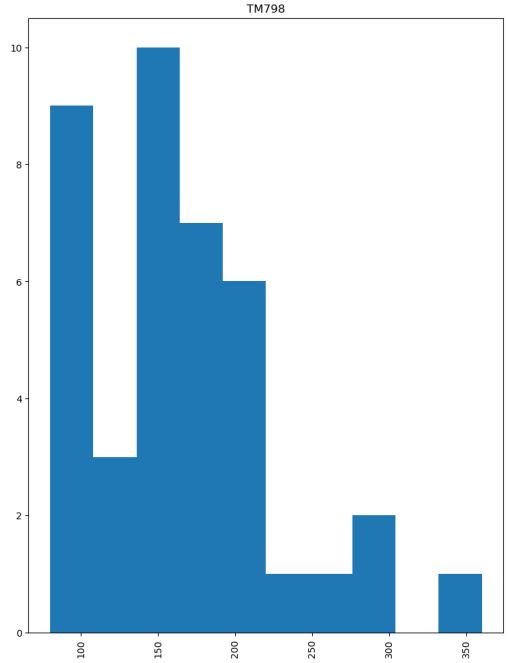
1.6.3 Product v/s miles

[23]:
```
db.hist(by='Product',column = 'Miles', figsize=(20,30))
```

[23]:
```
array([[<AxesSubplot:title={'center':'TM195'}>,
        <AxesSubplot:title={'center':'TM498'}>],
       [<AxesSubplot:title={'center':'TM798'}>, <AxesSubplot:>]],
      dtype=object)
```


TM195
25
20
15
10
5
0
40
60
80
100
120
140
160
180


TM498
20
15
10
5
0
25
50
75
100
125
150
175
200


TM798
10
8
6
4
2
0
100
150
200
250
300
350

1.6.4 Average of age

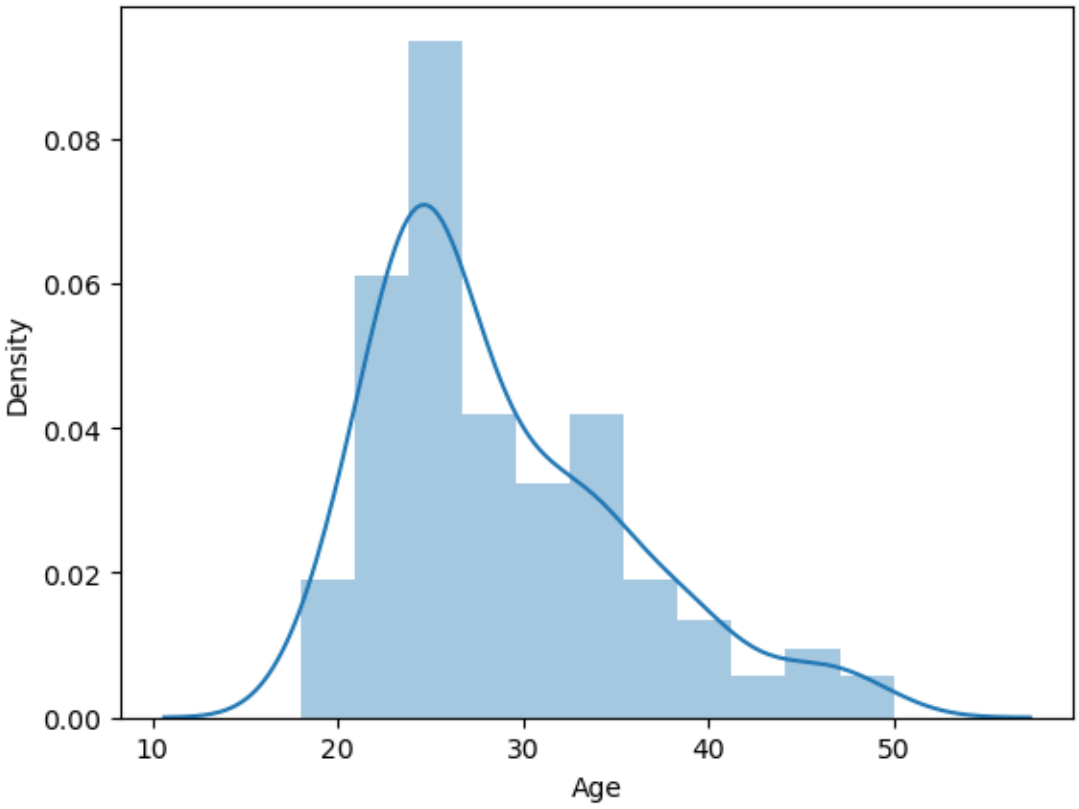
[15]:
```
db['Age'].mean()
```

[15]:
```
28.788888888888888
```

[16]:
```
sns.distplot(db['Age'])
```

```
C:\Users\suman\anaconda\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[16]:
```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```

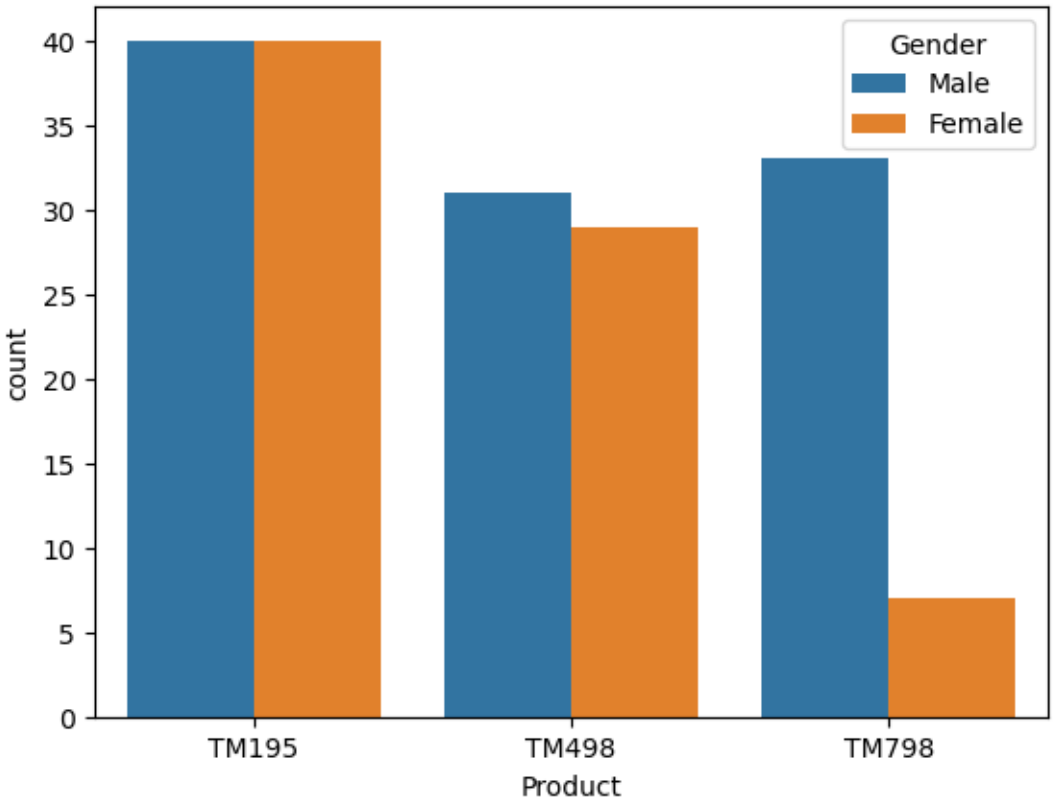


1.7 Count Plot

- x= number of each product
- hue= seperated by Gender

[18]:
```
sns.countplot(x='Product',hue='Gender',data=db)
```

[18]:
```
<AxesSubplot:xlabel='Product', ylabel='count'>
```

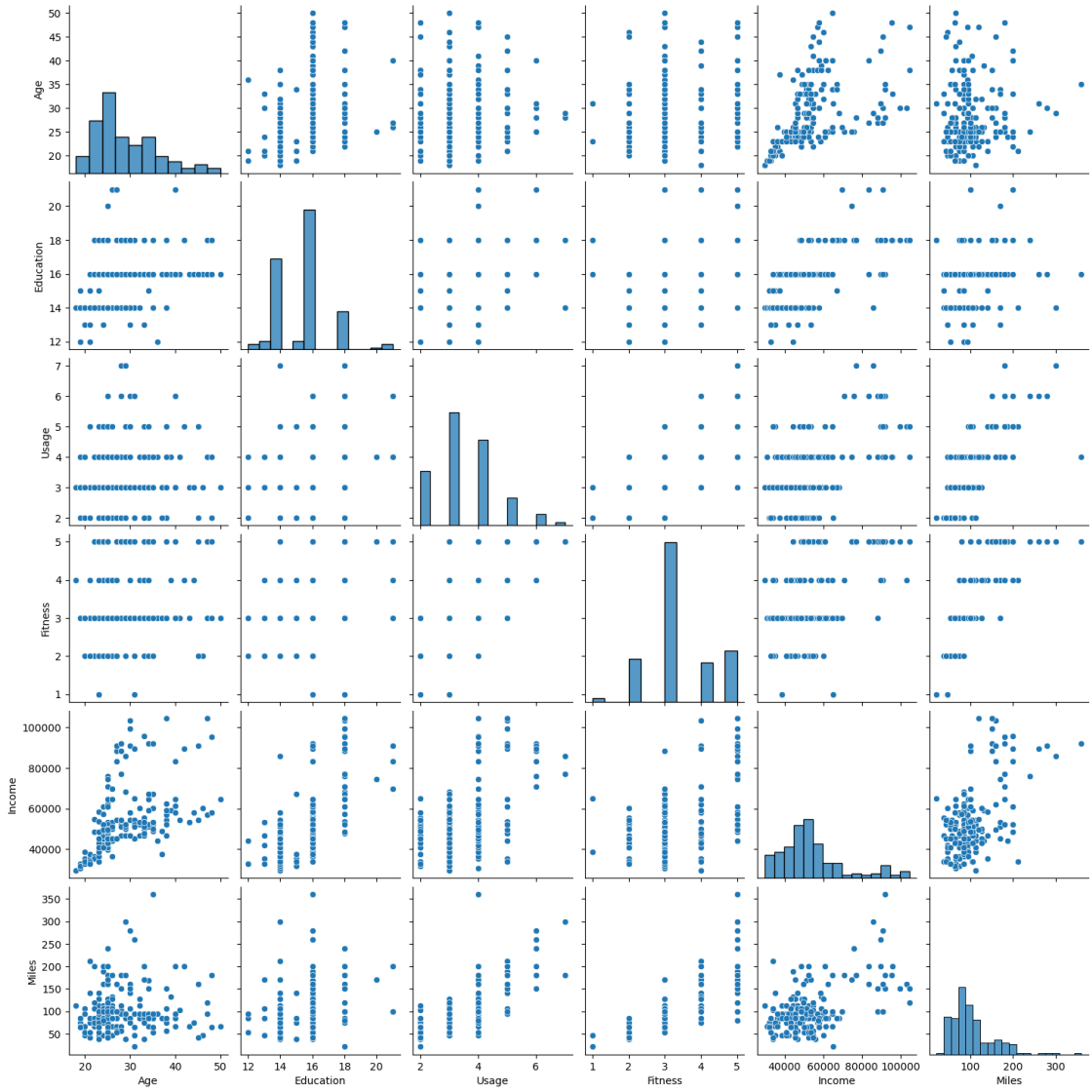


1.8 Pairplot

Quick overview of the data

[19]:
```
sns.pairplot(db)
```

[19]:
```
<seaborn.axisgrid.PairGrid at 0x2241cfea970>
```



1.9 Corelation Heat Map

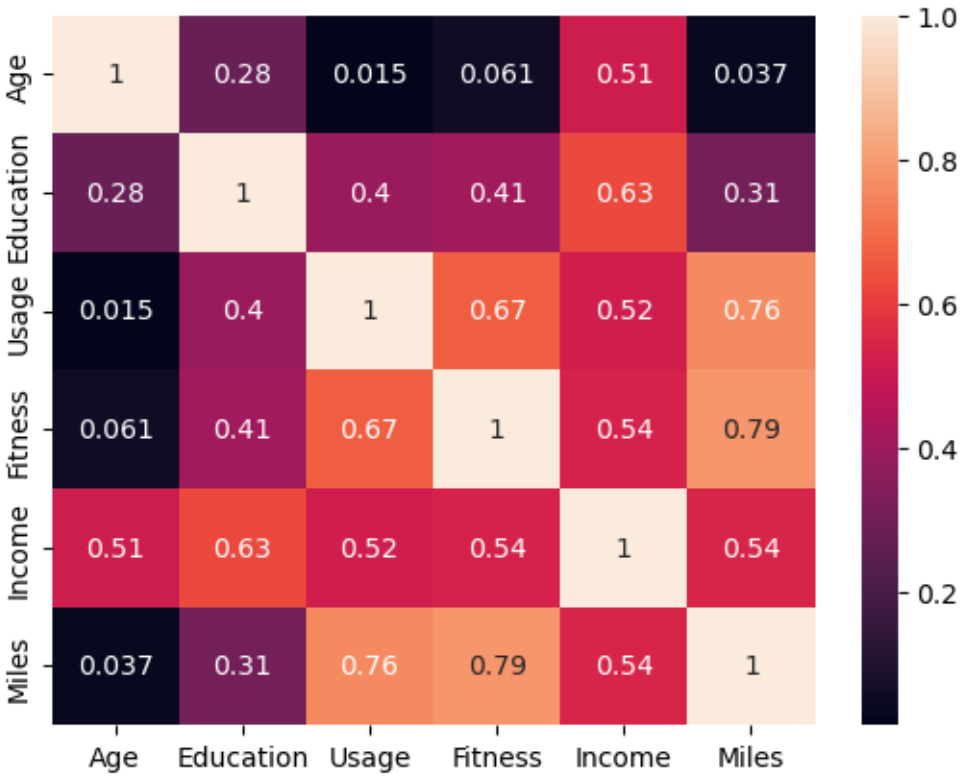
[20]:
```
corr=db.corr()
corr
```

[20]:

	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

[21]:
```
sns.heatmap(corr,annot=True)
```

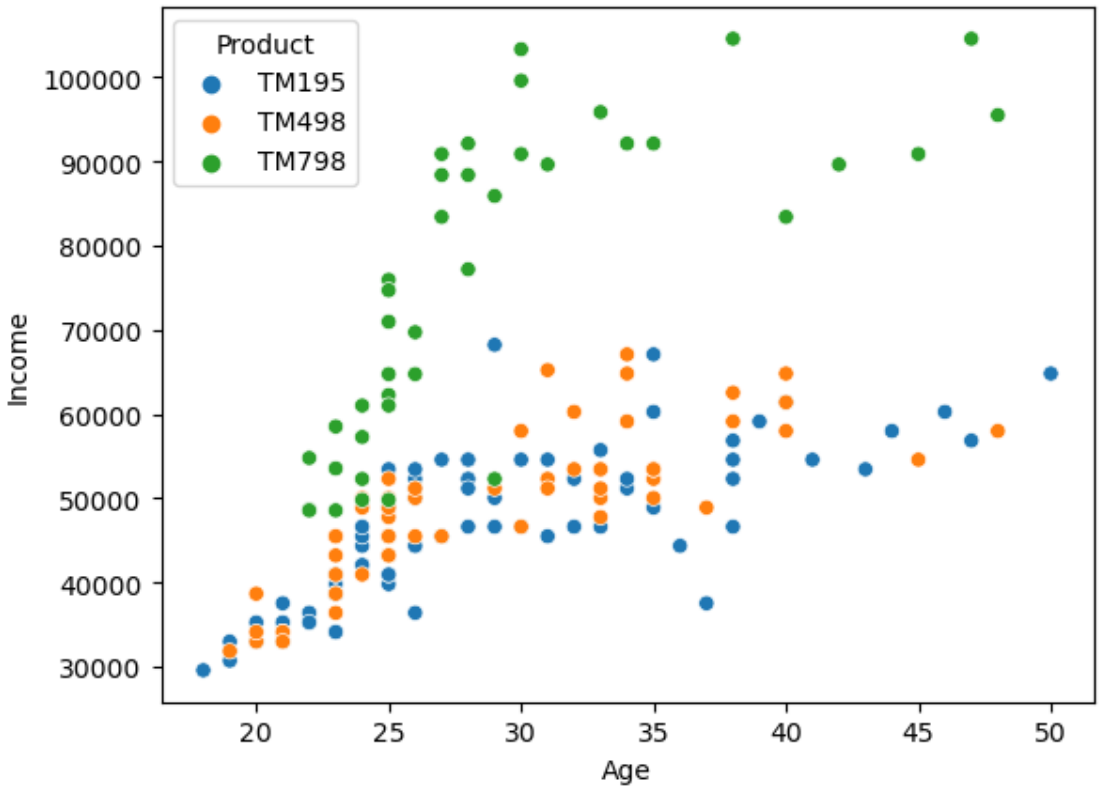
[21]: <AxesSubplot:>



1.10 How do income and age affect the decision of which model is bought?

We can infer that TM798 is the more expensive model

[24]:
```
sns.scatterplot(x='Age', y='Income',data=db, hue = 'Product')
plt.show()
```


Product
TM195
TM498
TM798
Income
Age
100000
90000
80000
70000
60000
50000
40000
30000
20
25
30
35
40
45
50