In [35]:
```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

In [4]:
```
df=pd.read_csv("https://raw.githubusercontent.com/Mukund94/Datasets/main/CardioGoodFitness-1.csv",sep=','
    )
```

In [5]:
```
df.head()
```

Out[5]:

	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

In [6]:
```
df.shape
```

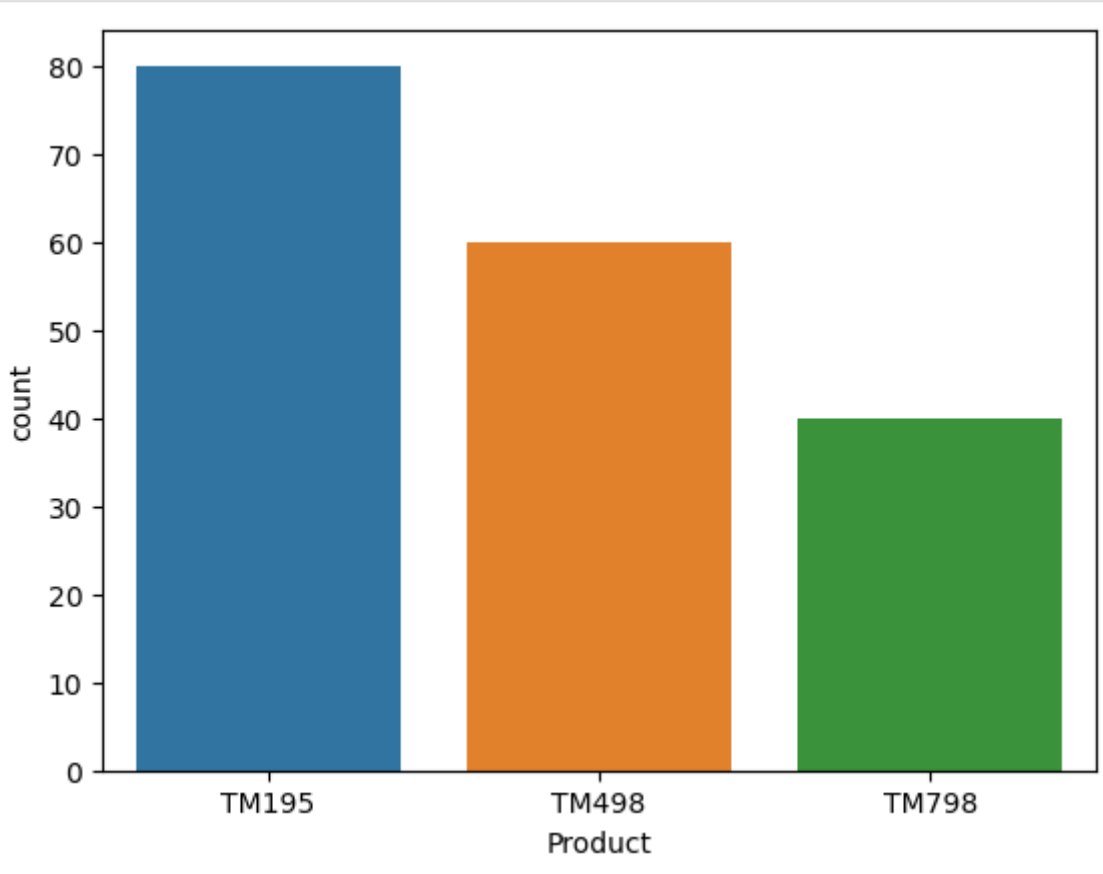
Out[6]:
```
(180, 9)
```

In [7]:
```
df.describe(include='all')
```

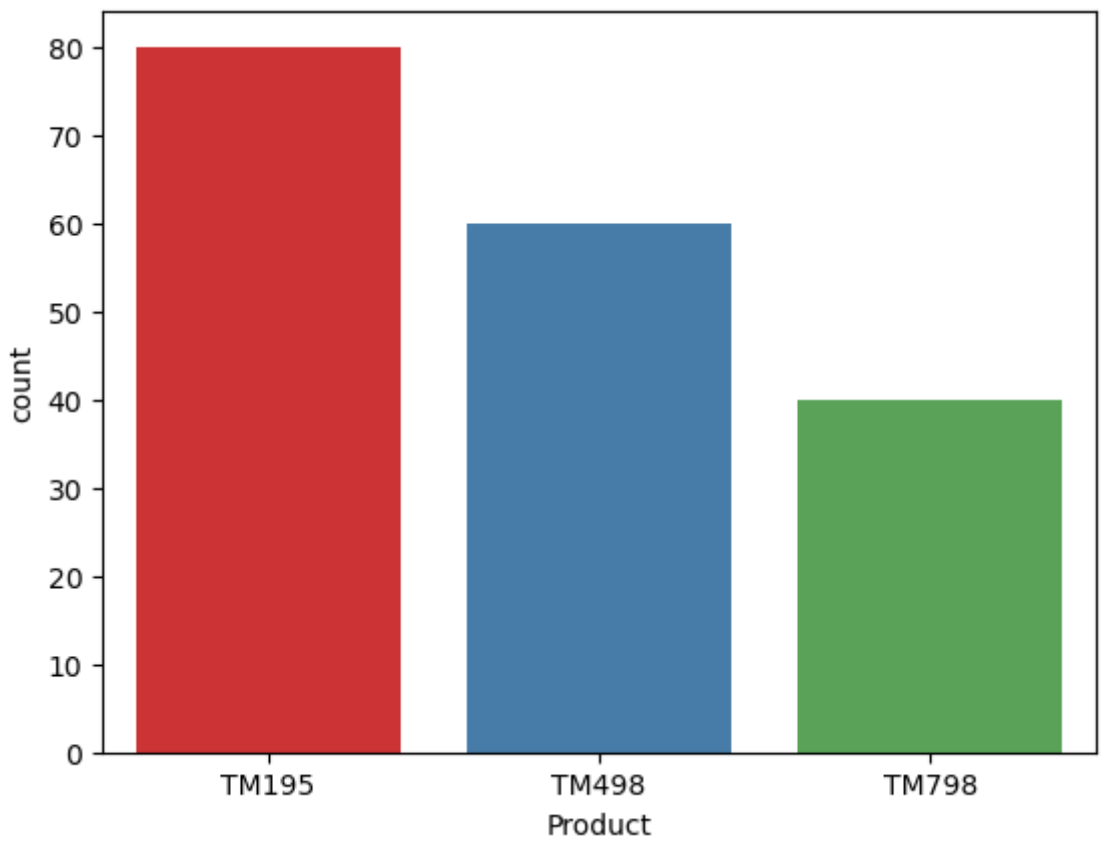
Out[7]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	TM195	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

In [9]:
```
sns.countplot(x='Product',data=df)
plt.show()
```



In [29]:
```
sns.countplot(x='Product',data=df,palette="Set1")
plt.show()
```



In [10]:
```
#Number of unique products?
df.Product.nunique()
```

Out[10]:
```
3
```

In [11]:
```
#Name of unique products?
df.Product.unique()
```

Out[11]:
```
array(['TM195', 'TM498', 'TM798'], dtype=object)
```

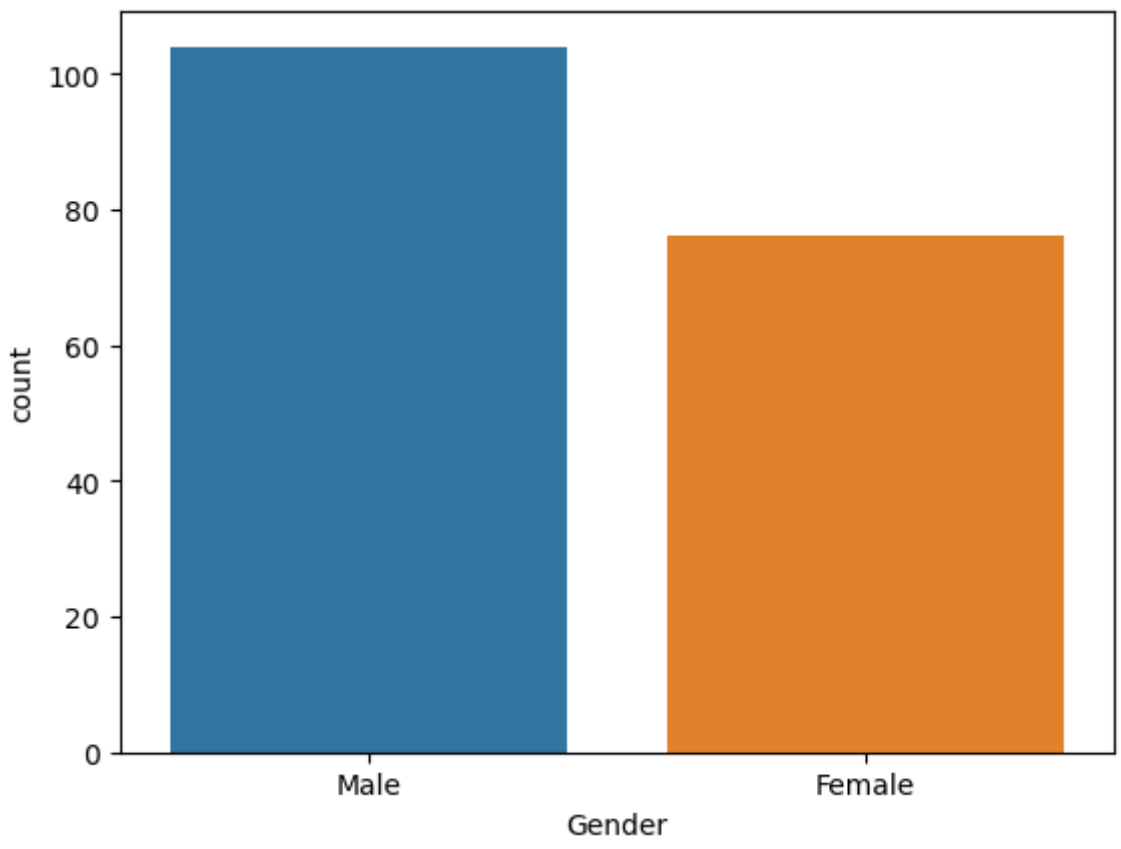
In [14]:
```
#Frequency of products?
df.Product.value_counts()
```

Out[14]:
```
TM195    80
TM498    60
TM798    40
Name: Product, dtype: int64
```

In [19]:
```
#How many products are there?
df.Product.shape[0]
```

Out[19]:
```
180
```

In [39]:
```
sns.countplot(x='Gender',data=df)
plt.show()
```



In [40]:
```
#Number of unique Gender?
df.Gender.nunique()
```

Out[40]:
```
2
```

In [42]:
```
#Genders?
df.Gender.unique()
```

Out[42]:
```
array(['Male', 'Female'], dtype=object)
```

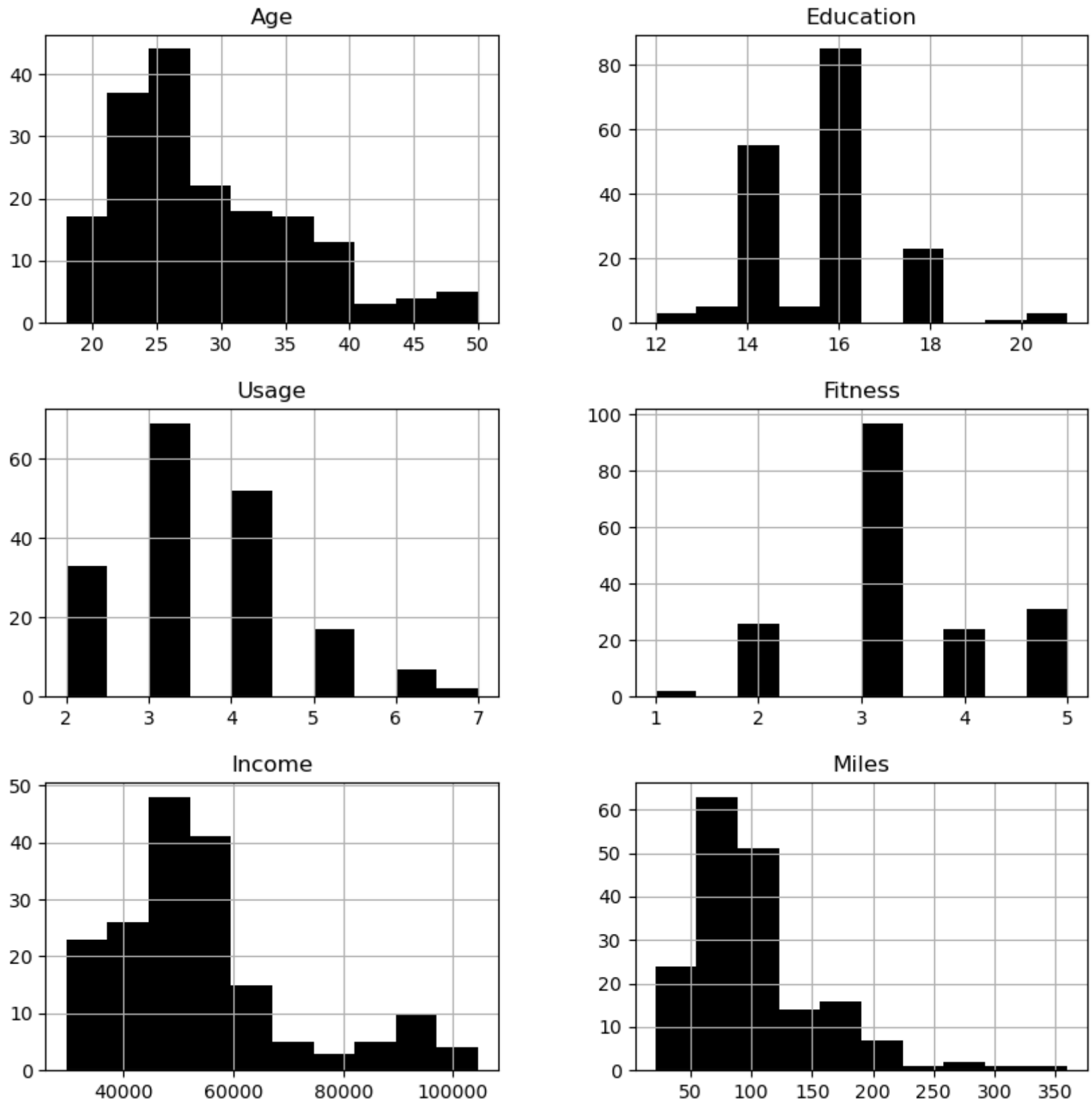
In [44]:
```
#Genders counts?
df.Gender.value_counts()
```

Out[44]:
```
Male      104
Female     76
Name: Gender, dtype: int64
```

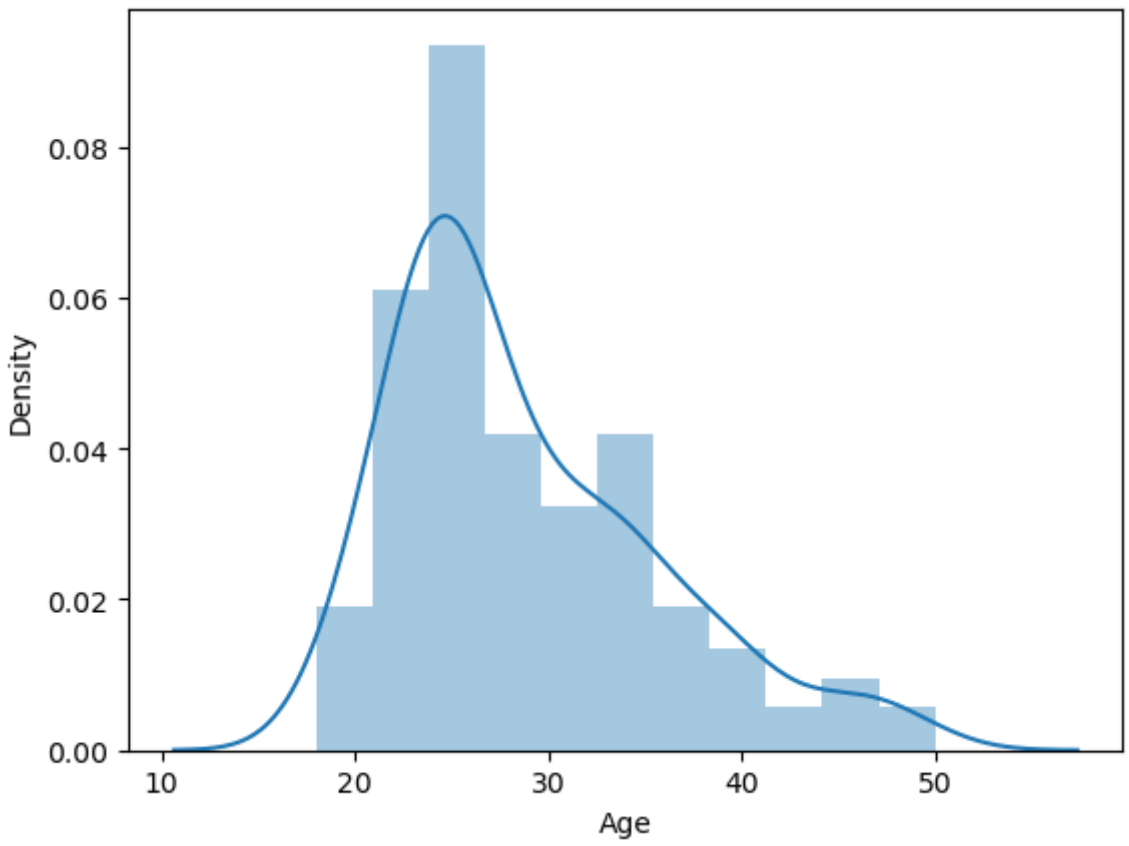
In [45]:
```
df.Gender.shape[0]
```

Out[45]:
```
180
```

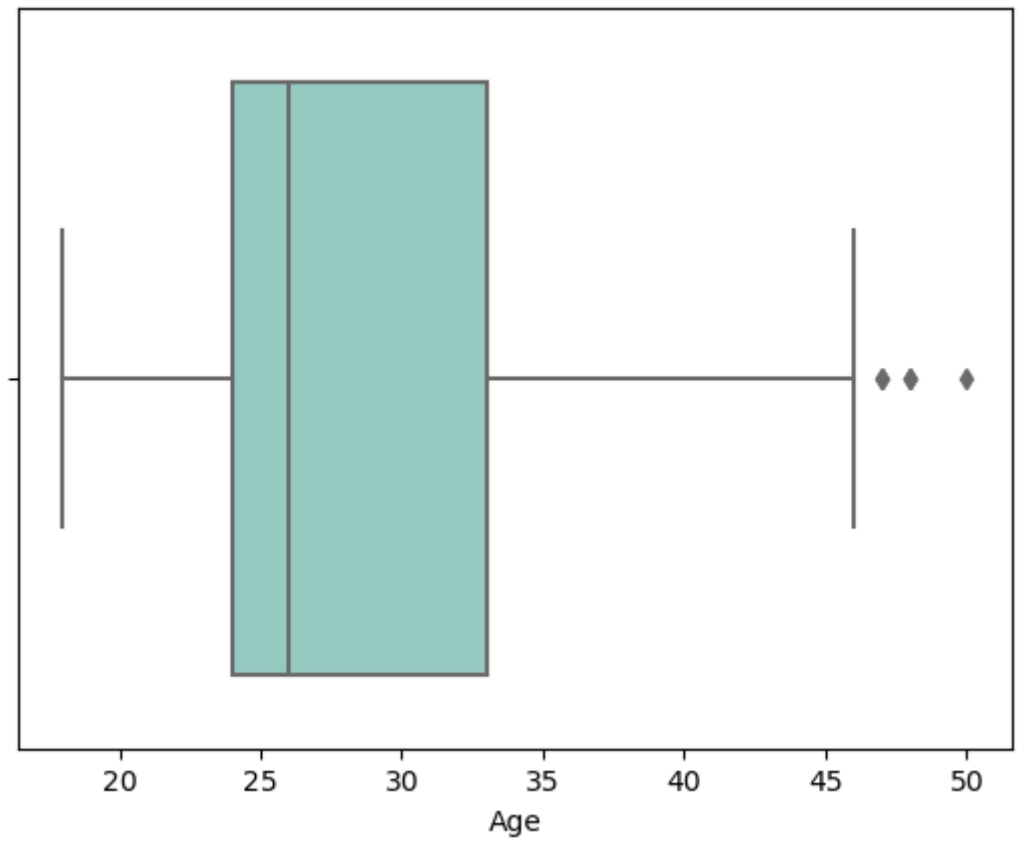
In [34]:
```
df.hist(figsize=(10,10),color='k')
plt.show()
```



In [49]:
```
sns.distplot(df["Age"])
plt.show()
```



In [52]:
```
sns.boxplot(x=df["Age"],data=df,palette="Set3")
plt.show()
```



In [54]:
```
#Outlier calculation
Q3=33
Q1=24
IQR=Q3-Q1
IQR
```

Out[54]: 9

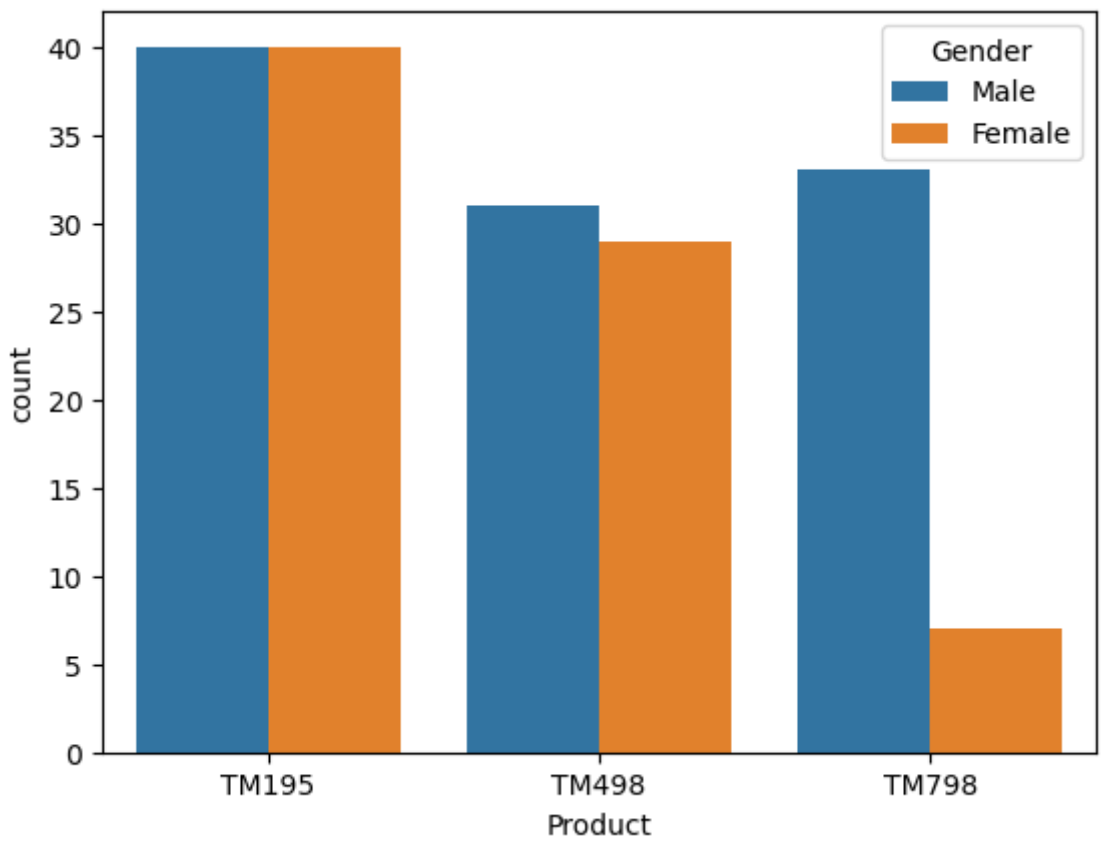
In [57]:
```
#Lower Outlier
Q1-1.5*IQR
```

Out[57]: 10.5

In [58]:
```
#Upper Outlier
Q3+1.5*IQR
```

Out[58]: 46.5

In [62]:
```
sns.countplot(x="Product",hue="Gender",data=df)
plt.show()
```

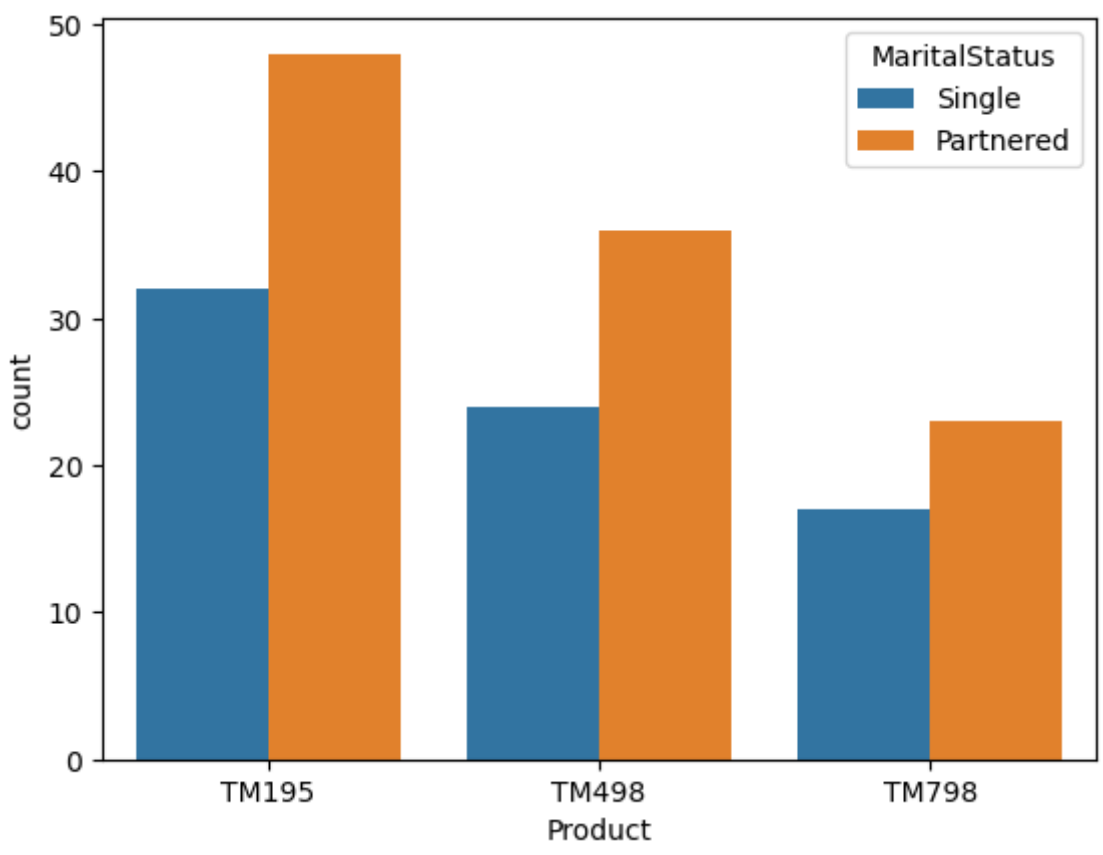


In [66]:
```
# Cross Tabs of Product and Gender
pd.crosstab(df['Product'],df['Gender'])
```

Out[66]:

Product	Female	Male
TM195	40	40
TM498	29	31
TM798	7	33

In [69]:
```
sns.countplot(x="Product",hue="MaritalStatus",data=df)
plt.show()
```



In [68]:
```
# Cross Tabs of Product and Marital Status
pd.crosstab(df['Product'],df['MaritalStatus'])
```

Out[68]:

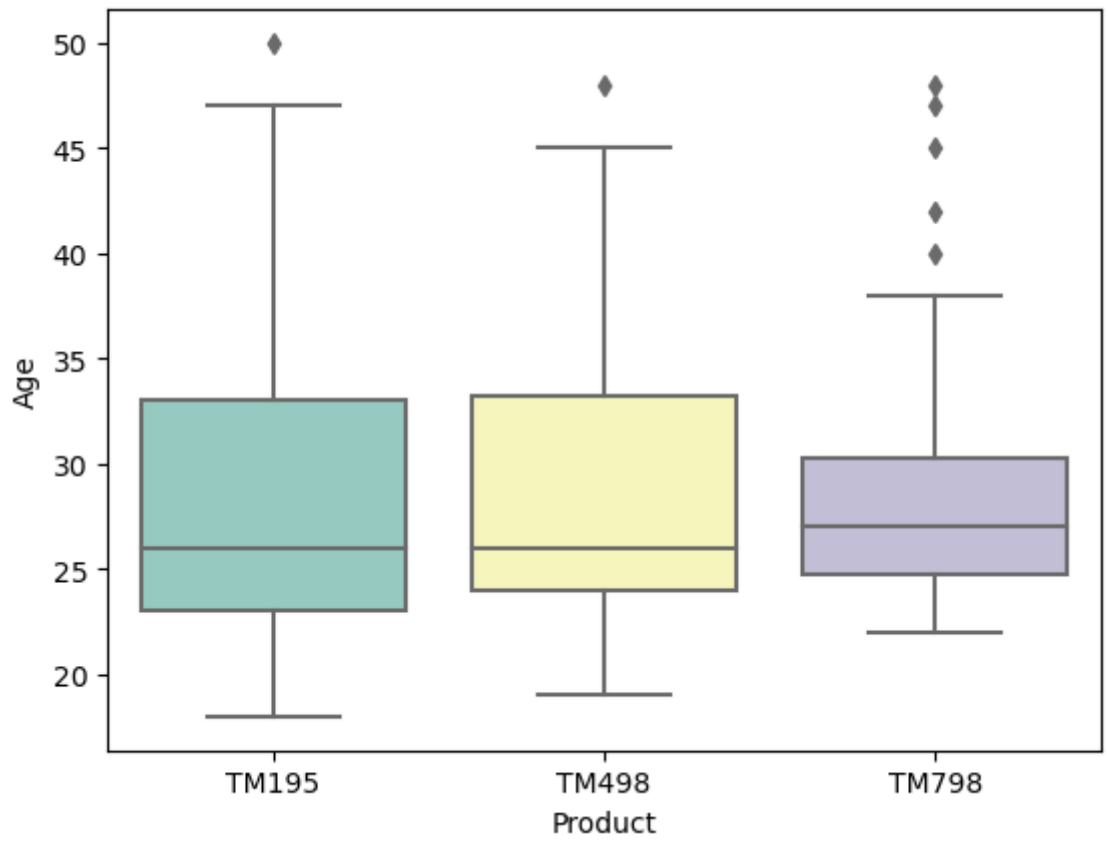
Product	Partnered	Single
TM195	48	32
TM498	36	24
TM798	23	17

In [67]:
```
df.head()
```

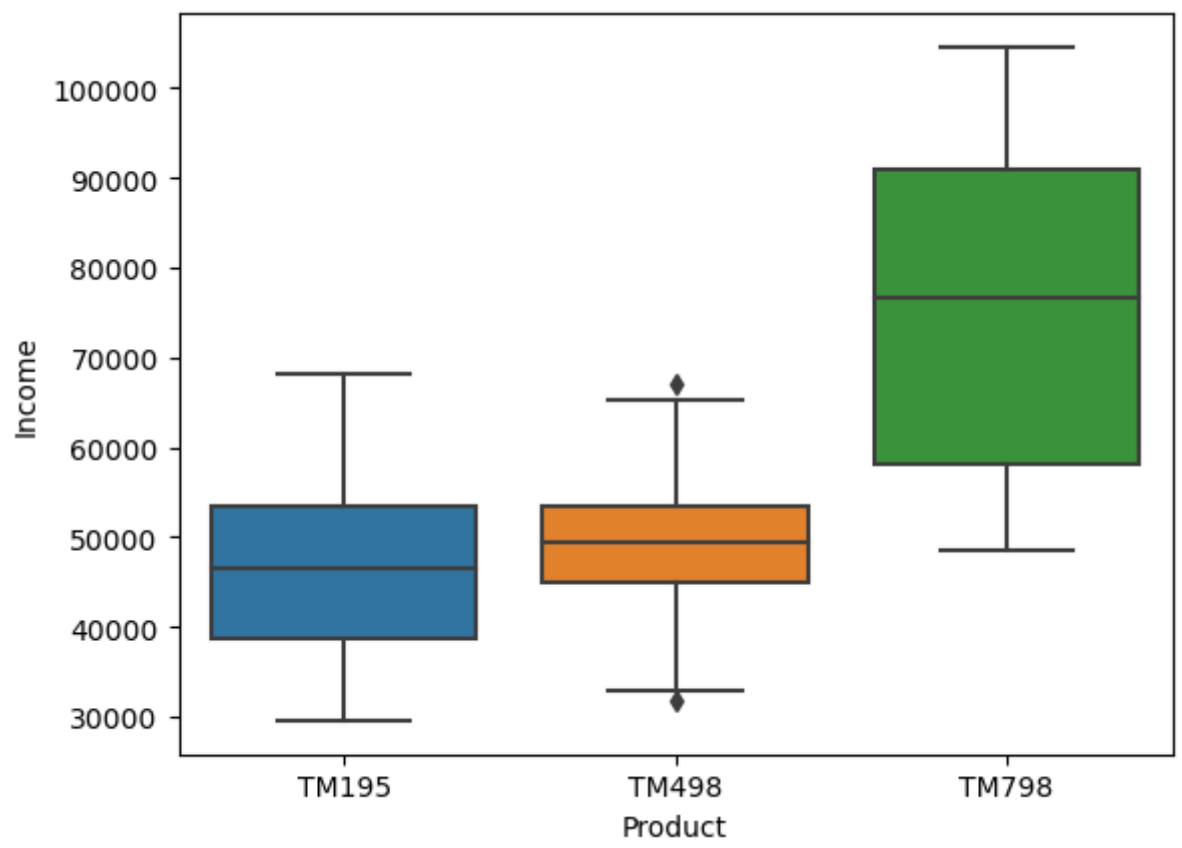
Out[67]:

	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

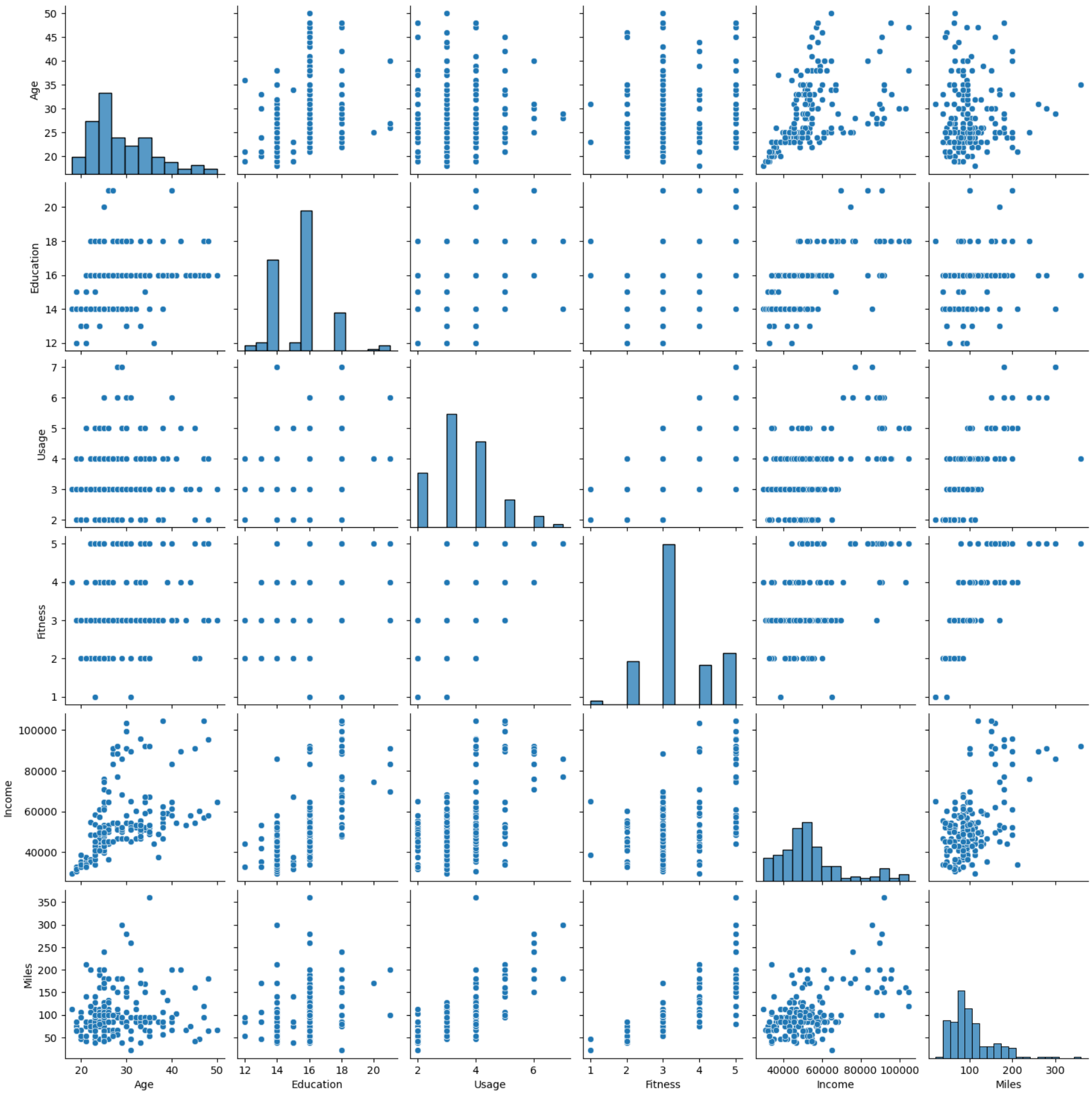
In [73]:
```
sns.boxplot(x="Product",y="Age",data=df,palette="Set3")
plt.show()
```



In [77]:
```
sns.boxplot(x="Product",y="Income",data=df)
plt.show()
```



In [81]:
```
sns.pairplot(data=df)
plt.show()
```



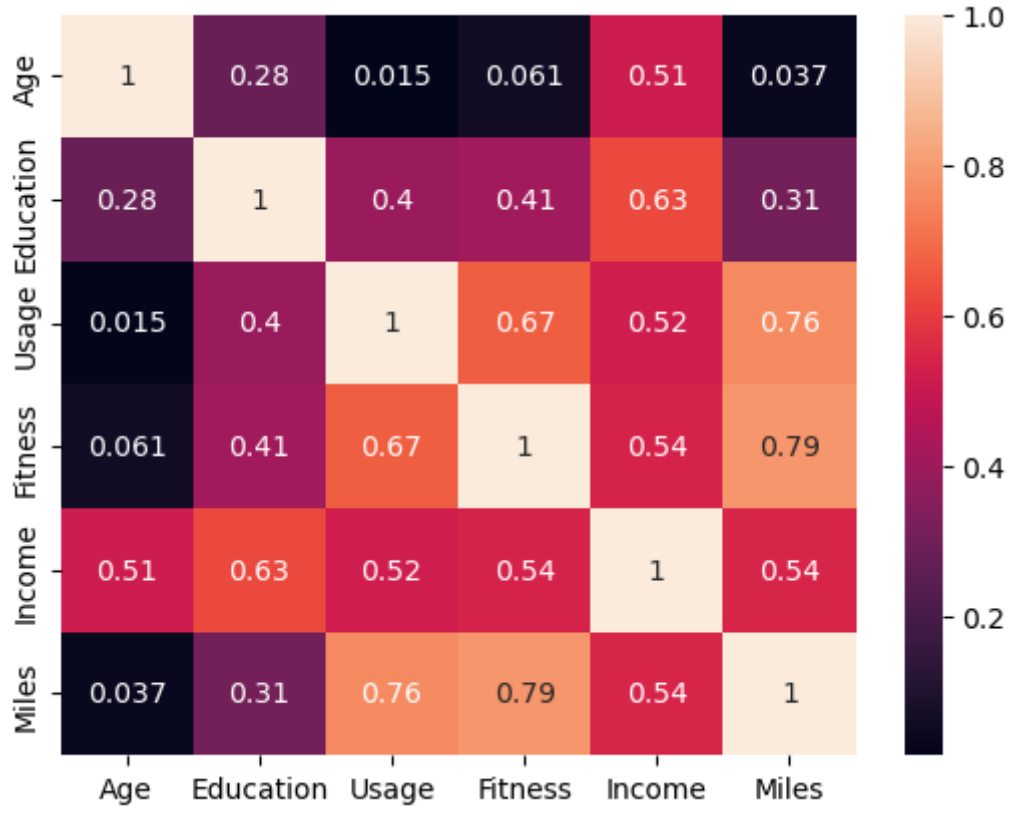
In [82]:
```
correlation=df.corr()
correlation
```

Out[82]:

	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 [85]:
```
sns.heatmap(correlation,annot=True,)
```

Out[85]:
```
<Axes: >
```


Age
Education
Usage
Fitness
Income
Miles
1
0.28
0.015
0.061
0.51
0.037
0.28
1
0.4
0.41
0.63
0.31
0.015
0.4
1
0.67
0.52
0.76
0.061
0.41
0.67
1
0.54
0.79
0.51
0.63
0.52
0.54
1
0.54
0.037
0.31
0.76
0.79
0.54
1
1.0
0.8
0.6
0.4
0.2

In [ ]: