```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('//aerofit_treadmill.csv')
df
```

	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
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

180 rows × 9 columns

Next steps: Generate code with df View recommended plots

```
df.head()
```

	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

```
df.describe()
```

	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

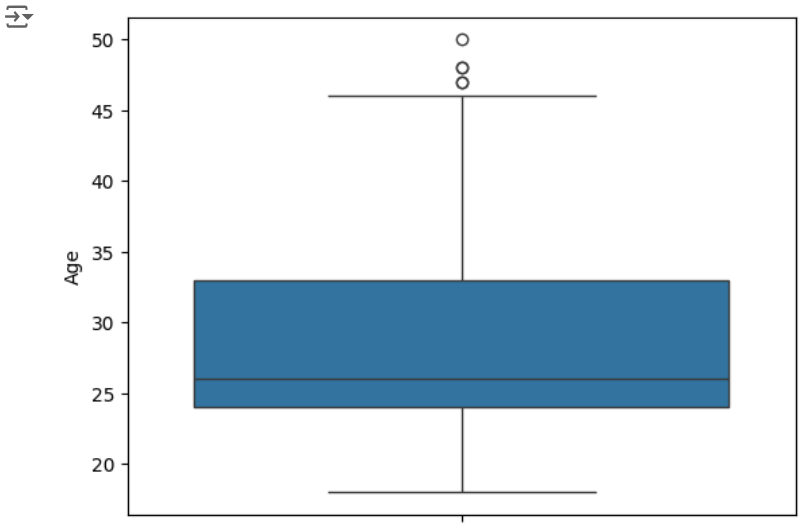
```
df.shape
```

```
(180, 9)
```

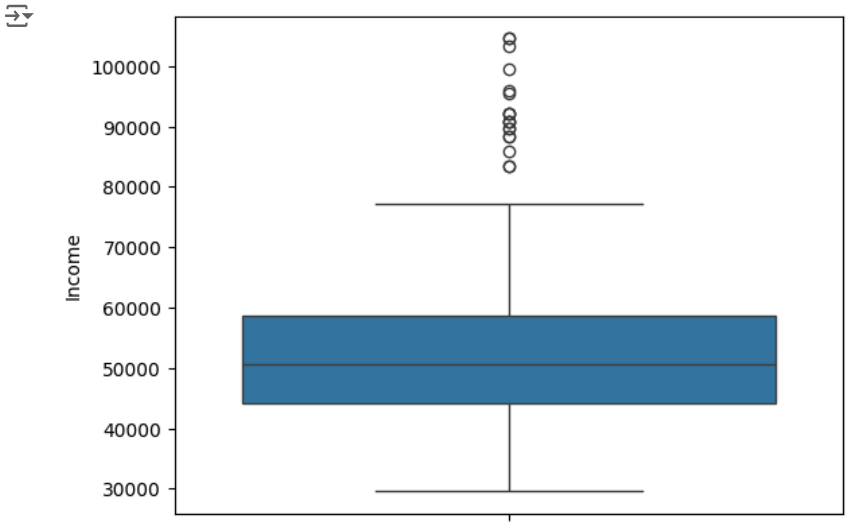
```
df.isnull().sum()
```

```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

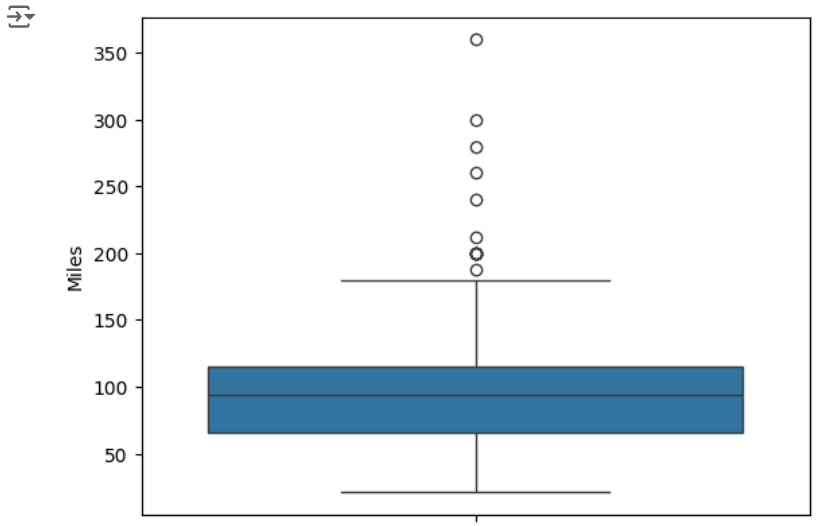
```
#Find the outliers for every continuous variable in the dataset

a=sns.boxplot(df['Age'])
```




```
b=sns.boxplot(df['Income'])
```

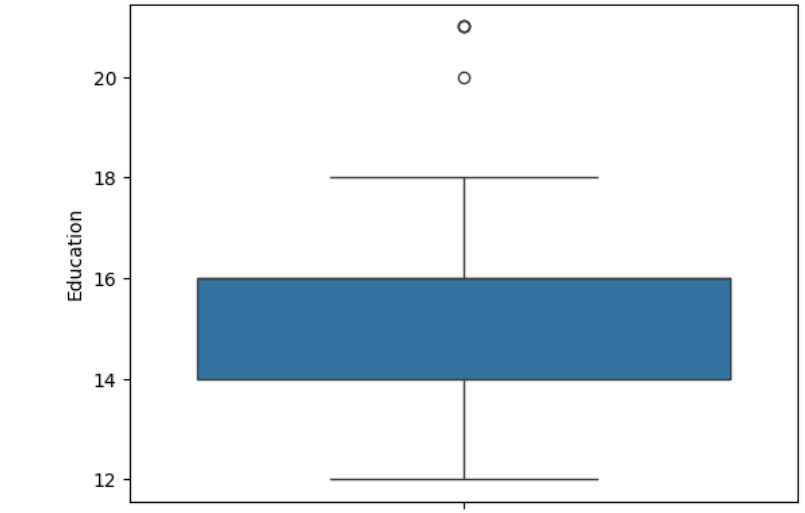



```
c=sns.boxplot(df['Miles'])
```



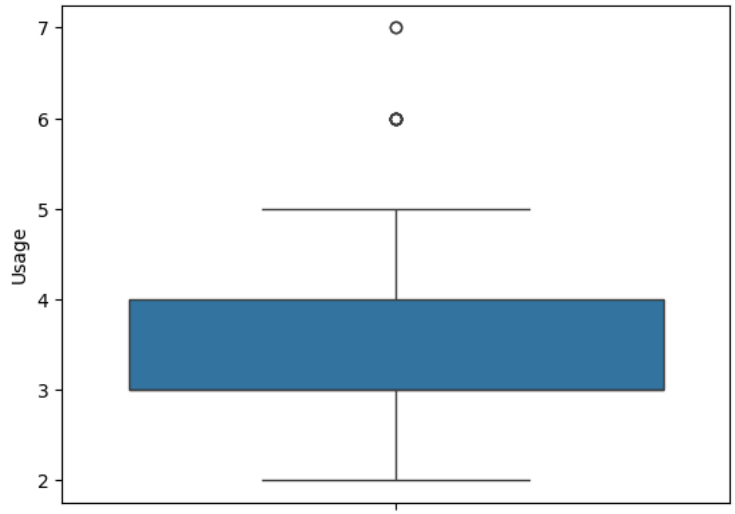

```
sns.boxplot(df['Education'])
```

```
<Axes: ylabel='Education'>
```



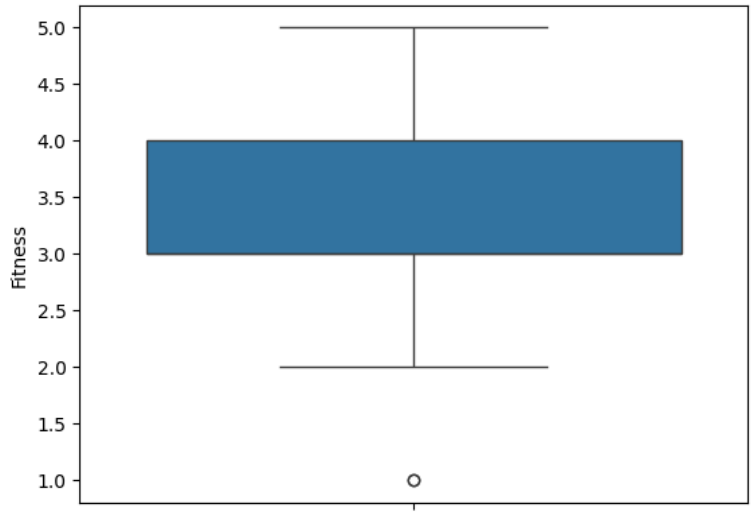

```
sns.boxplot(df['Usage'])
```

```
<Axes: ylabel='Usage'>
```




```
sns.boxplot(df['Fitness'])
```

```
<Axes: ylabel='Fitness'>
```



```
#Remove/clip the data between the 5 percentile and 95 percentile

remove_age= np.clip(df['Age'], np.percentile(df['Age'], 5), np.percentile(df['Age'], 95))
remove_income = np.clip(df['Income'], np.percentile(df['Income'], 5), np.percentile(df['Income'], 95))
remove_miles = np.clip(df['Miles'], np.percentile(df['Miles'], 5), np.percentile(df['Miles'], 95))
remove_fitness = np.clip(df['Fitness'], np.percentile(df['Fitness'], 5), np.percentile(df['Fitness'], 95))
remove_usage = np.clip(df['Usage'], np.percentile(df['Usage'], 5), np.percentile(df['Usage'], 95))
remove_education = np.clip(df['Education'], np.percentile(df['Education'], 5), np.percentile(df['Education'], 95))

df.head()
```

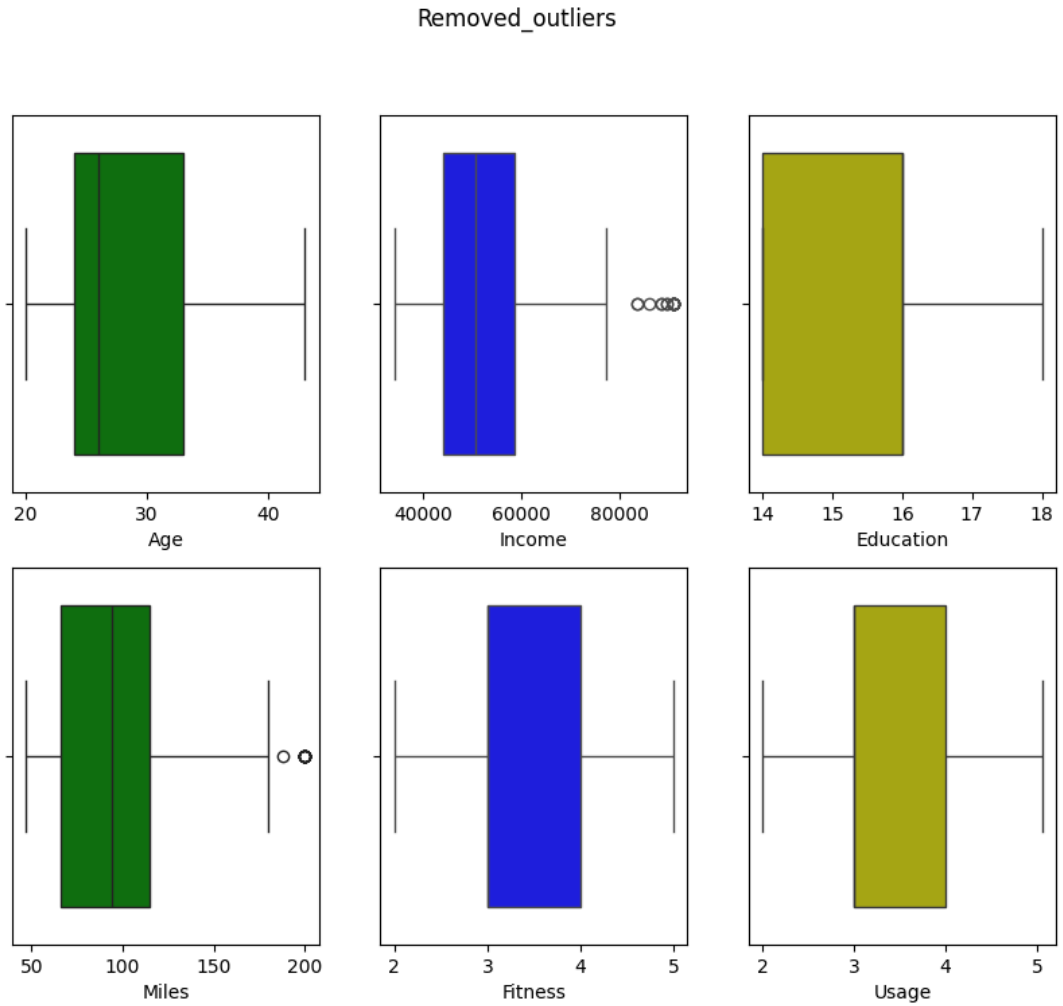
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	remove_age	remove_income	remove_miles	remove_fitness	remove_usage	remove_education
0	KP281	18	Male	14	Single	3	4	29562	112	20.0	34053.15	112	4	3.0	14
1	KP281	19	Male	15	Single	2	3	31836	75	20.0	34053.15	75	3	2.0	15
2	KP281	19	Female	14	Partnered	4	3	30699	66	20.0	34053.15	66	3	4.0	14
3	KP281	19	Male	12	Single	3	3	32973	85	20.0	34053.15	85	3	3.0	14
4	KP281	20	Male	13	Partnered	4	2	35247	47	20.0	35247.00	47	2	4.0	14

```
df['remove_age'] = np.clip(df['Age'], np.percentile(df['Age'], 5), np.percentile(df['Age'], 95))
df['remove_income'] = np.clip(df['Income'], np.percentile(df['Income'], 5), np.percentile(df['Income'], 95))
df['remove_miles']= np.clip(df['Miles'], np.percentile(df['Miles'], 5), np.percentile(df['Miles'], 95))
df['remove_fitness'] =  np.clip(df['Fitness'], np.percentile(df['Fitness'], 5), np.percentile(df['Fitness'], 95))
df['remove_usage'] = np.clip(df['Usage'], np.percentile(df['Usage'], 5), np.percentile(df['Usage'], 95))
df['remove_education'] = np.clip(df['Education'], np.percentile(df['Education'], 5), np.percentile(df['Education'], 95))

df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	remove_age	remove_income	remove_miles	remove_fitness	remove_usage	remove_education
0	KP281	18	Male	14	Single	3	4	29562	112	20.0	34053.15	112	4	3.0	14
1	KP281	19	Male	15	Single	2	3	31836	75	20.0	34053.15	75	3	2.0	15
2	KP281	19	Female	14	Partnered	4	3	30699	66	20.0	34053.15	66	3	4.0	14
3	KP281	19	Male	12	Single	3	3	32973	85	20.0	34053.15	85	3	3.0	14
4	KP281	20	Male	13	Partnered	4	2	35247	47	20.0	35247.00	47	2	4.0	14

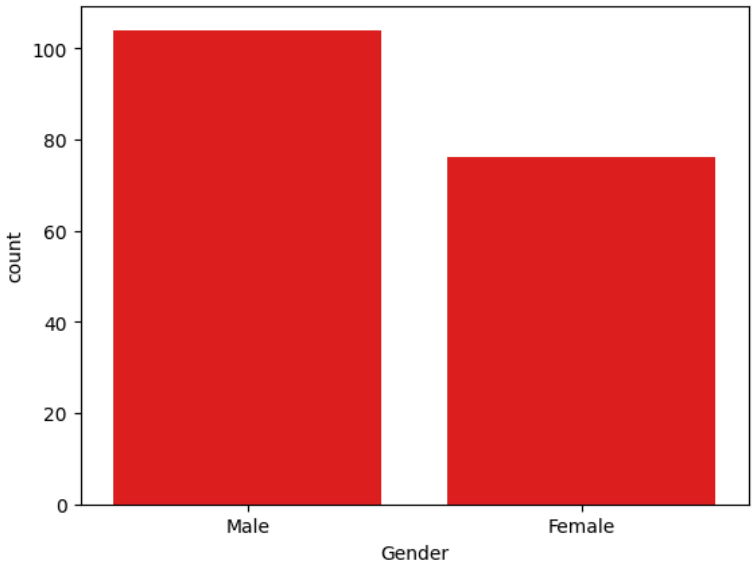
```
#printing result by using subplot
fig,ax=plt.subplots(2,3,figsize=(10,8))
sns.boxplot(data=df,x=remove_age,color='g',ax=ax[0,0])
sns.boxplot(data=df,x=remove_income,color='b',ax=ax[0,1])
sns.boxplot(data=df,x=remove_education,color='y',ax=ax[0,2])
sns.boxplot(data=df,x=remove_miles,color='g',ax=ax[1,0])
sns.boxplot(data=df,x=remove_fitness,color='b',ax=ax[1,1])
sns.boxplot(data=df,x=remove_usage,color='y',ax=ax[1,2])
fig.suptitle('Removed_outliers')
```

```
Text(0.5, 0.98, 'Removed_outliers')
```



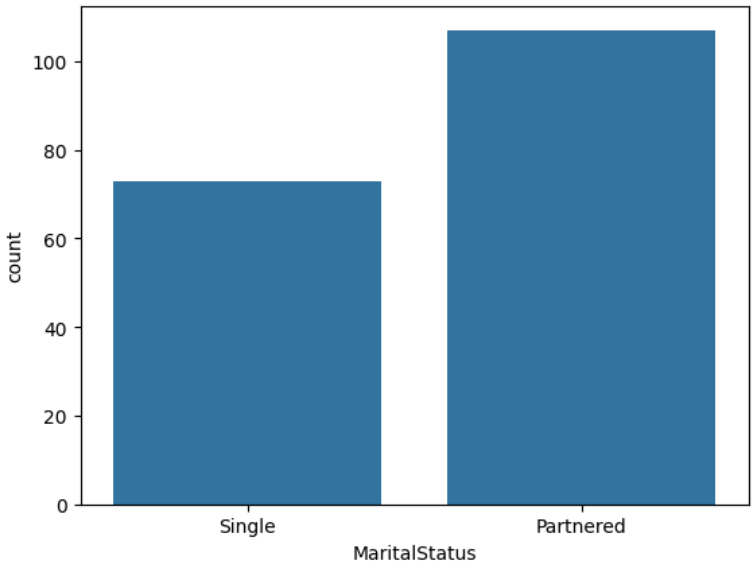

```
#Find if there is any relationship between the categorical variables and the output variable in the data.

sns.countplot(x='Gender',data=df,color='r')
```

```
<Axes: xlabel='Gender', ylabel='count'>
```



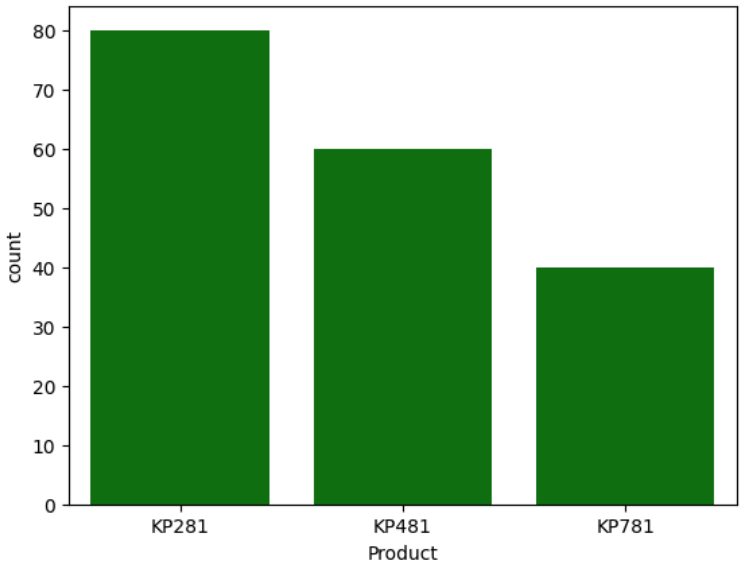

```
sns.countplot(x='MaritalStatus',data=df)
```

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




```
sns.countplot(x='Product',data=df,color='g')
```

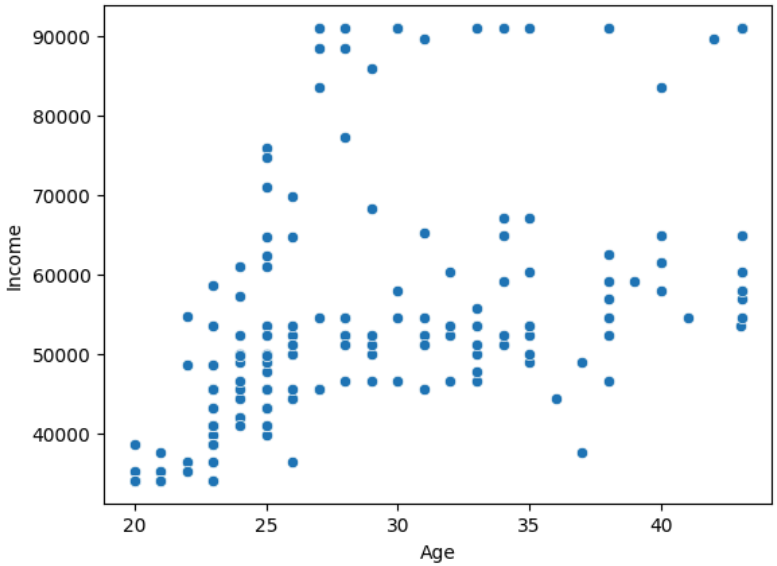
```
<Axes: xlabel='Product', ylabel='count'>
```



Insights : Male prefer the product more over females. Partnered people prefer the product over single. KP281 is sold the most.

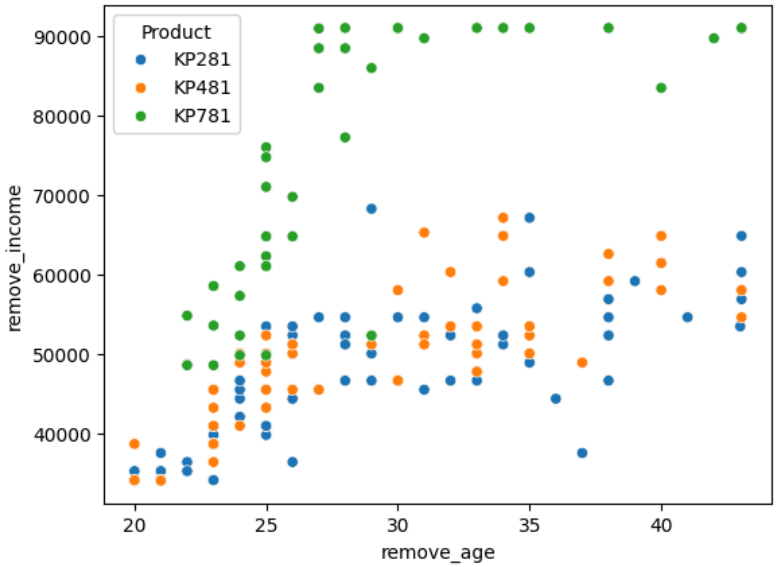
```
#Find if there is any relationship between the continuous variables and the output variable in the data.
sns.scatterplot(x='remove_age',y='remove_income',data=df)
```

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



```
sns.scatterplot(x='remove_age',y='remove_income',data=df,hue='Product')
```

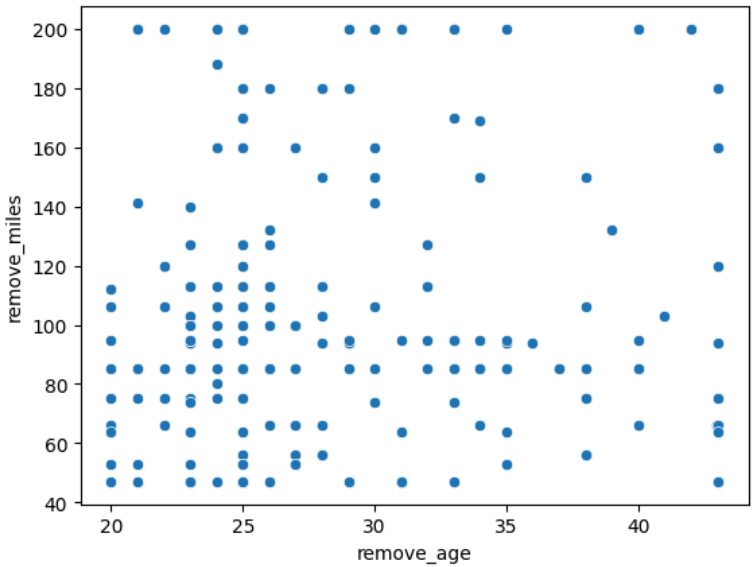
```
<Axes: xlabel='remove_age', ylabel='remove_income'>
```



**Insights : People with higher income are preferring to buy, the costliest product, and is focusing on quality. **

```
sns.scatterplot(x='remove_age',y='remove_miles',data=df)
```

```
<Axes: xlabel='remove_age', ylabel='remove_miles'>
```



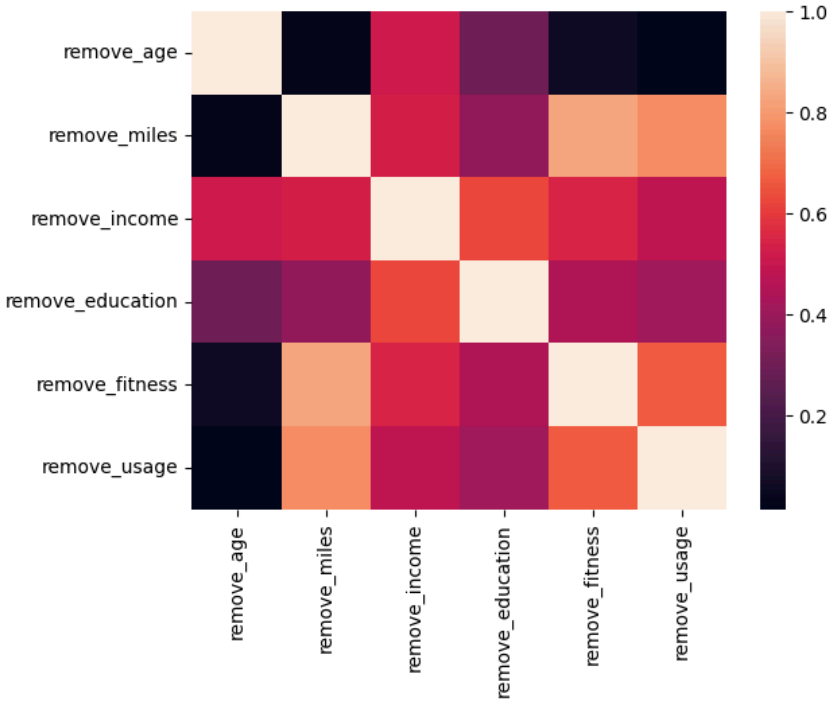

```
df1=df[['remove_age','remove_miles','remove_income','remove_education','remove_fitness','remove_usage']]
```

Double-click (or enter) to edit

Check the correlation among different factors

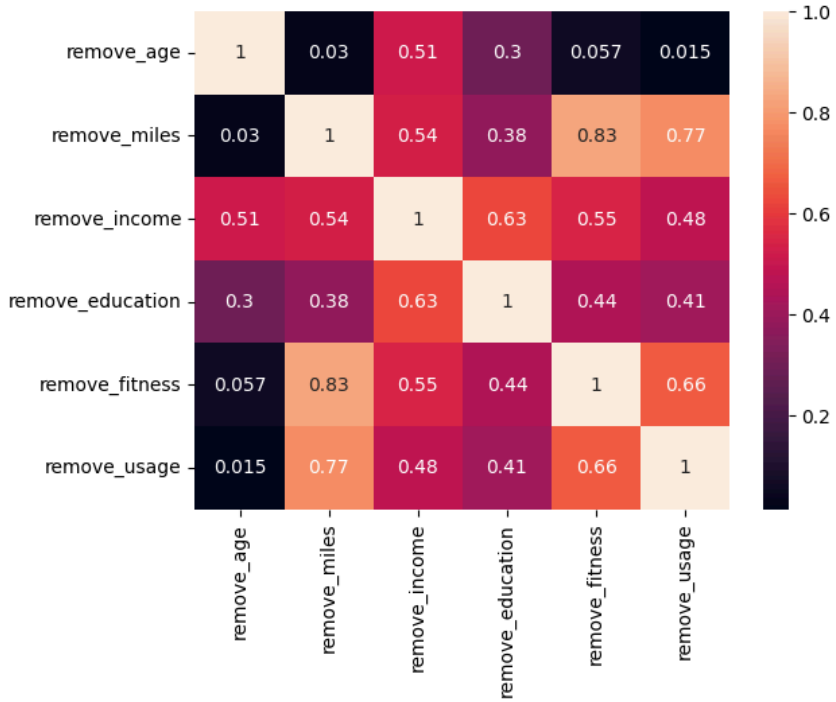
```
sns.heatmap(df1.corr())
```

```
<Axes: >
```



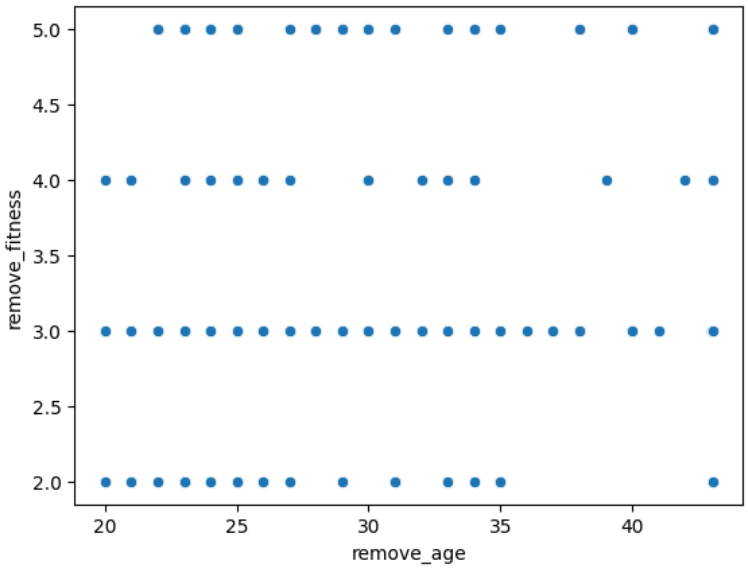

```
sns.heatmap(df1.corr(),annot=True)
```

```
<Axes: >
```




```
sns.scatterplot(x='remove_age',y='remove_fitness',data=df)
```

```
<Axes: xlabel='remove_age', ylabel='remove_fitness'>
```



Probability

Insight : People of age between 25-30 are more confident about their fitness

```
#Find the marginal probability (what percent of customers have purchased (KP281, KP481, or KP781)

df['Product'].value_counts(normalize=True)*100
```

```
Product
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: proportion, dtype: float64
```

```
df['Gender'].value_counts(normalize=True)*100
```

```
Gender
Male      57.777778
Female    42.222222
Name: proportion, dtype: float64
```

```
df['MaritalStatus'].value_counts(normalize=True)*100
```

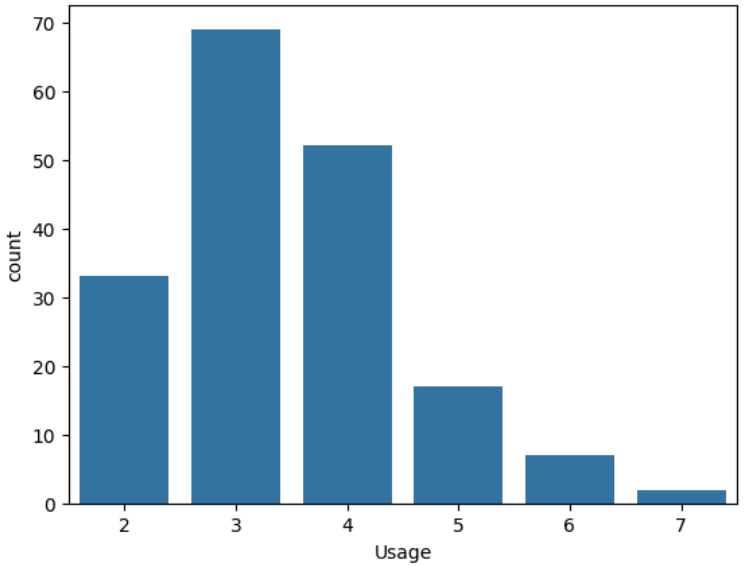
```
MaritalStatus
Partnered    59.444444
Single       40.555556
Name: proportion, dtype: float64
```

```
df['Usage'].value_counts(normalize=True)*100
```

```
Usage
3    38.333333
4    28.888889
2    18.333333
5     9.444444
6     3.888889
7     1.111111
Name: proportion, dtype: float64
```

```
sns.countplot(x = df['Usage'], data = df)
```

```
<Axes: xlabel='Usage', ylabel='count'>
```




```
sns.countplot(x = df['Age'], data = df,color='r')
plt.xticks(rotation=90)
```

```
(([0,
  1,
  2,
  3,
  4,
  5,
  6,
  7,
  8,
  9,
  10,
  11,
  12,
  13,
  14,
  15,
  16,
  17,
  18,
  19,
  20,
  21,
  22,
  23,
  24,
  25,
  26,
  27,
  28,
  29,
  30,
  31],
 [Text(0, 0, '18'),
  Text(1, 0, '19'),
  Text(2, 0, '20'),
  Text(3, 0, '21'),
  Text(4, 0, '22'),
  Text(5, 0, '23'),
  Text(6, 0, '24'),
  Text(7, 0, '25'),
  Text(8, 0, '26'),
  Text(9, 0, '27'),
  Text(10, 0, '28'),
  Text(11, 0, '29'),
  Text(12, 0, '30'),
  Text(13, 0, '31'),
  Text(14, 0, '32'),
  Text(15, 0, '33'),
  Text(16, 0, '34'),
  Text(17, 0, '35'),
  Text(18, 0, '36'),
  Text(19, 0, '37'),
  Text(20, 0, '38'),
  Text(21, 0, '39'),
  Text(22, 0, '40'),
  Text(23, 0, '41'),
  Text(24, 0, '42'),
```

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-1a6fe1782b2f> in <cell line: 1>()
----> 1 sns.pairplot(df)

NameError: name 'sns' is not defined
```

```
bins = [-1, 20, 25, 30, 35, 40, 50]
labels = ['<20', '20-25','25-30','30-35','35-40', '40+']

df['Age_bins'] = pd.cut(df['remove_age'], bins = bins, labels = labels)
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	remove_age	remove_income	remove_mile
0	KP281	18	Male	14	Single	3	4	29562	112	20.0	34053.15	11
1	KP281	19	Male	15	Single	2	3	31836	75	20.0	34053.15	7
2	KP281	19	Female	14	Partnered	4	3	30699	66	20.0	34053.15	6
3	KP281	19	Male	12	Single	3	3	32973	85	20.0	34053.15	8
4	KP281	20	Male	13	Partnered	4	2	35247	47	20.0	35247.00	4

```
bins=[10000,20000,30000,40000,50000,60000,70000,80000]
labels=['<10k','10k-20k','20k-30k','30k-40k','40k-50k','50k-60k','60k+']
df['Income_bins'] = pd.cut(df['remove_income'],bins=bins,labels=labels)
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	remove_age	remove_income	remove_mile
0	KP281	18	Male	14	Single	3	4	29562	112	20.0	34053.15	11
1	KP281	19	Male	15	Single	2	3	31836	75	20.0	34053.15	7
2	KP281	19	Female	14	Partnered	4	3	30699	66	20.0	34053.15	6
3	KP281	19	Male	12	Single	3	3	32973	85	20.0	34053.15	8
4	KP281	20	Male	13	Partnered	4	2	35247	47	20.0	35247.00	4

Joins and Probability

> Marginal Joint Conditional

```
pd.crosstab(index=df['Gender'],columns=df['Product'])
```

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

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

Product	KP281	KP481	KP781	All
Gender				
Female	40	29	7	76
Male	40	31	33	104
All	80	60	40	180

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

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

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

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	48	36	23	107
Single	32	24	17	73
All	80	60	40	180

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

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	26.666667	20.000000	12.777778	59.444444
Single	17.777778	13.333333	9.444444	40.555556
All	44.444444	33.333333	22.222222	100.000000

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

```
Usage
3    69
4    52
2    33
5    17
6     7
7     2
Name: count, dtype: int64
```

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

Product	KP281				KP481				KP781				All
Usage	2	3	4	5	2	3	4	5	3	4	5	6	7	
Gender														
Female	13	19	7	1	7	14	5	3	0	2	3	2	0	76
Male	6	18	15	1	7	17	7	0	1	16	9	5	2	104
All	19	37	22	2	14	31	12	3	1	18	12	7	2	180

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

Gender	Product	Usage 2	3	4	5	6	7	All
Female	KP281	13	19	7	1	0	0	40
	KP481	7	14	5	3	0	0	29
	KP781	0	0	2	3	2	0	7
Male	KP281	6	18	15	1	0	0	40
	KP481	7	17	7	0	0	0	31
	KP781	0	1	16	9	5	2	33
All		33	69	52	17	7	2	180

```
#Joint and Conditional Probability
```

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

Gender	Product KP281	KP481	KP781	All
Female	22.222222	16.111111	3.888889	42.222222
Male	22.222222	17.222222	18.333333	57.777778
All	44.444444	33.333333	22.222222	100.000000

Joint Probability : Probability of buying KP481 by a Female is 16%

Prob of purchasing KP281 is 44.44%.

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

Gender	Product KP281	KP481	KP781
Female	52.631579	38.157895	9.210526
Male	38.461538	29.807692	31.730769
All	44.444444	33.333333	22.222222

Conditional Probability.

What is the probability of a person buying KP481 given that its a Female - 38.16% --> P(KP481 | Female) What is the probability fo a person

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

Product	KP281	KP481	KP781	All
remove_age				
20.0	6	4	0	10
21.0	4	3	0	7
22.0	4	0	3	7
23.0	8	7	3	18
24.0	5	3	4	12
25.0	7	11	7	25
26.0	7	3	2	12
27.0	3	1	3	7
28.0	6	0	3	9
29.0	3	1	2	6
30.0	2	2	3	7
31.0	2	3	1	6
32.0	2	2	0	4
33.0	2	5	1	8
34.0	2	3	1	6
35.0	3	4	1	8
36.0	1	0	0	1
37.0	1	1	0	2
38.0	4	2	1	7
39.0	1	0	0	1
40.0	1	3	1	5