Business Case: Aerofit - Descriptive Statistics & Probability

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
```

NumPy:

- NumPy is a powerful numerical computing library in Python.

Pandas:

- Pandas is a data manipulation and analysis library for Python

Seaborn:

- Seaborn is a statistical data visualization library based on Matplotlib.

Matplotlib:

- Matplotlib is a 2D plotting library for creating static, animated, and interactive visualizations in Python.

WorldCloud:

- WordCloud is a Python library used for creating word clouds, which are visual representations of text data.

These libraries are often used together in data science and analysis workflows to handle, manipulate, and visualize data effectively.

What does ‘good’ look like?

---

1. Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset

---

```
from google.colab import files
uploaded = files.upload()
#!gdown '1YgtfEN8M9vHu0xq2eO194o0aY_8T2EJh'
```

Choose Files no files selected — Upload widget is only available when the cell has been executed in the current browser session.

Please rerun this cell to enable.

Saving aerofit_treadmill.csv to aerofit_treadmill.csv

Downloading Dataset.....

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

- Data is read in Dataframe(df) format.

- ## The data type of all columns in the “customers” table

```
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
```

Insights

- The data type of all columns in the table.

Recommendations

- We want to display the data type of each column present in the dataset.
- We can see 2 Type of Data types.

Assumptions

--------Column - Data-Type--------

- Product - Object
- Age - Int64
- Gender - Object
- Education - Int64
- MaritalStatus - Object
- Usage - Int64
- Fitness - Int64
- Income - Int64
- Miles - Int64
- **Data-Type**
- Object- Holds addresses that refer to objects. You can assign any reference type (string, array, class, or interface) to an Object variable. An Object variable can also refer to data of any value type (numeric, Boolean, Char, Date, structure, or enumeration).
- Int64- The type int64 tells us that Python is storing each value within this column as a 64 bit integer. Holds signed 64-bit (8-byte) integers that range in value from -9223372036854775808 to 9223372036854775807.

- ## You can find the number of rows and columns given in the dataset

```
df.shape
```

```
(180, 9)
```

Insights

- You can find the number of rows and columns given in the dataset

Recommendations

- We want to find the shape of the dataset. We can use .shape

Assumptions

- Data contain 9 columns And 180 rows.

- ## Check for the missing values and find the number of missing values in each column

```
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
```

Insights

- Check for the missing values and find the number of missing values in each column

Recommendations

- We want to find any null values in columns, we can use .info()

Assumptions

- Data have NO NULL values in any columns.
- All the 9 columns have 180 non-null values and we know from ABOVE there is i80 rows. So, therefore there no null in Table

2. Detect Outliers

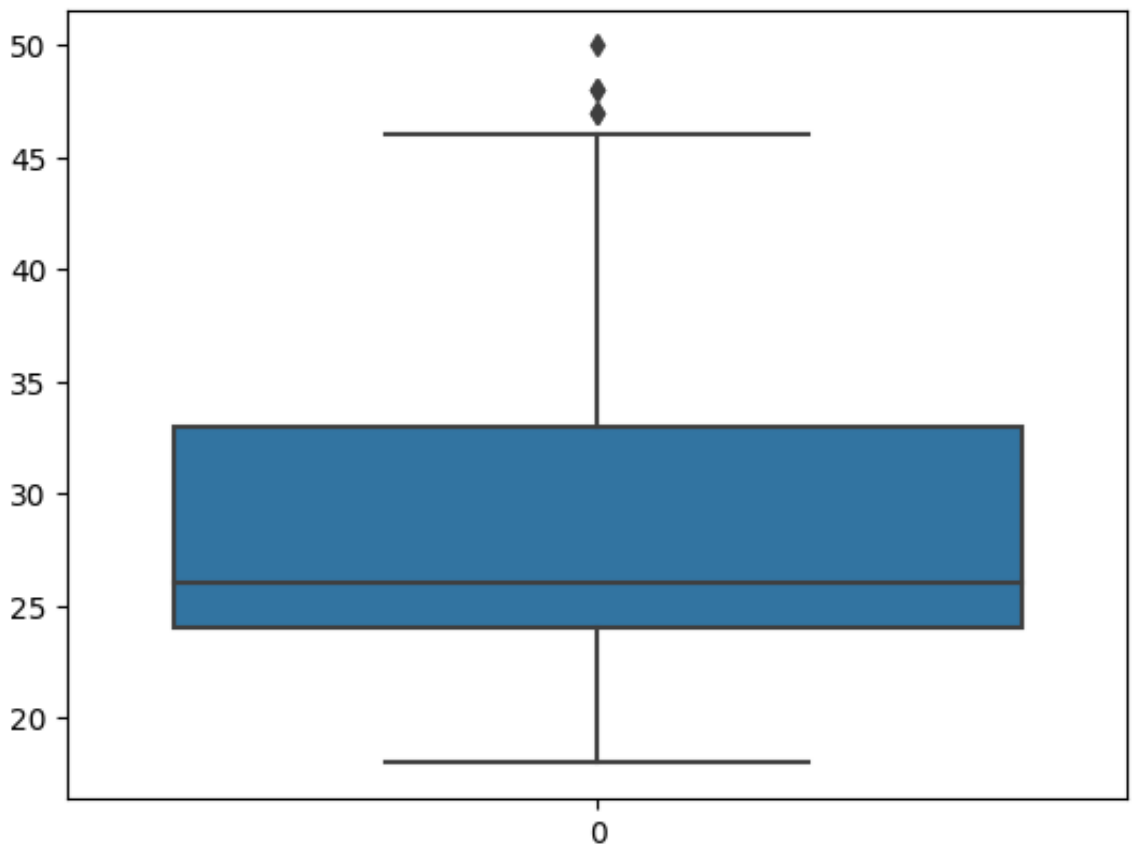
- ## Find the outliers for every continuous variable in the dataset

```
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

Column-'Age'

```
sns.boxplot(df["Age"])
plt.show()
```




```
Q3 = np.percentile(df["Age"], 75)
Q1 = np.percentile(df["Age"], 25)
iqr_Age = Q3 - Q1
iqr_Age
```

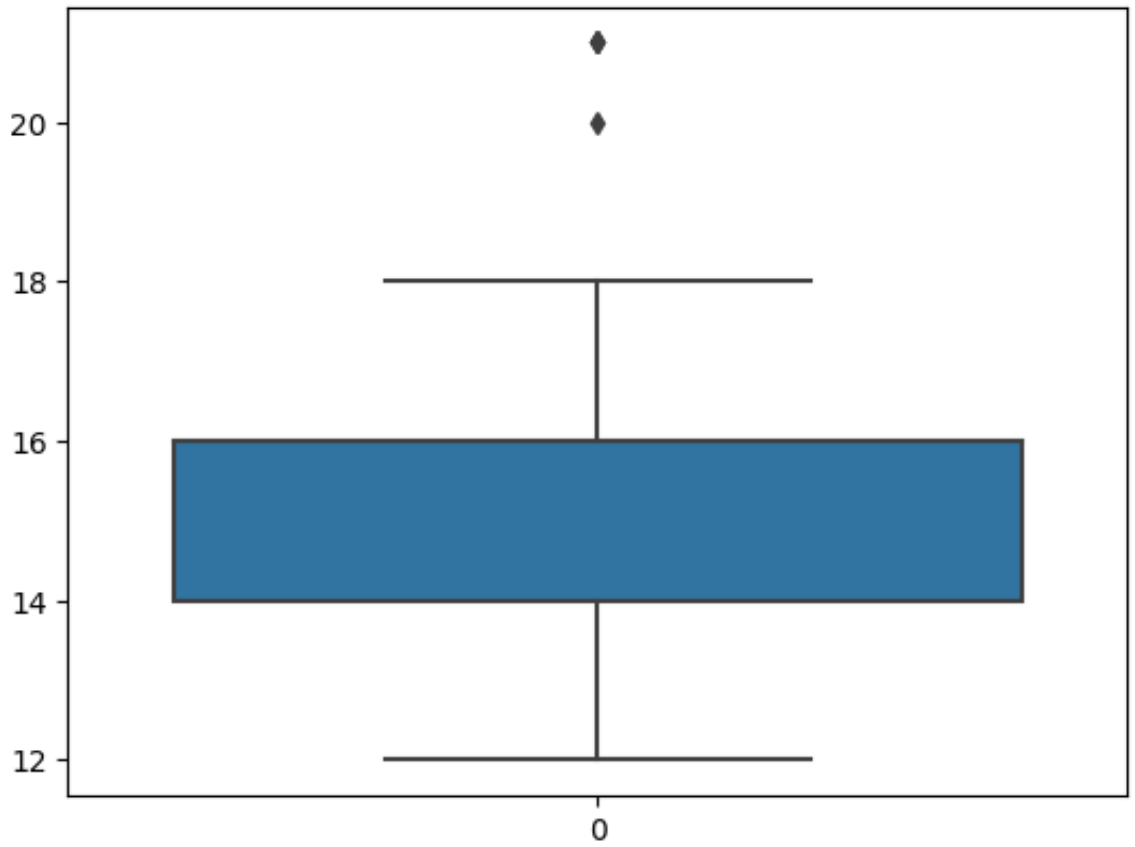
```
9.0
```

```
upper = Q3 + 1.5*iqr_Age
x= (df["Age"]>upper).sum()
print('No. of outliers in Age Column:',x,'Percentage of ounliers:',round((x/180)*100,2),"%")
```

```
No. of outliers in Age Column: 5 Percentage of ounliers: 2.78 %
```

Column-'Education'

```
sns.boxplot(df["Education"])
plt.show()
```




```
Q3 = np.percentile(df["Education"], 75)
Q1 = np.percentile(df["Education"], 25)
iqr_Education = Q3 - Q1
iqr_Education
```

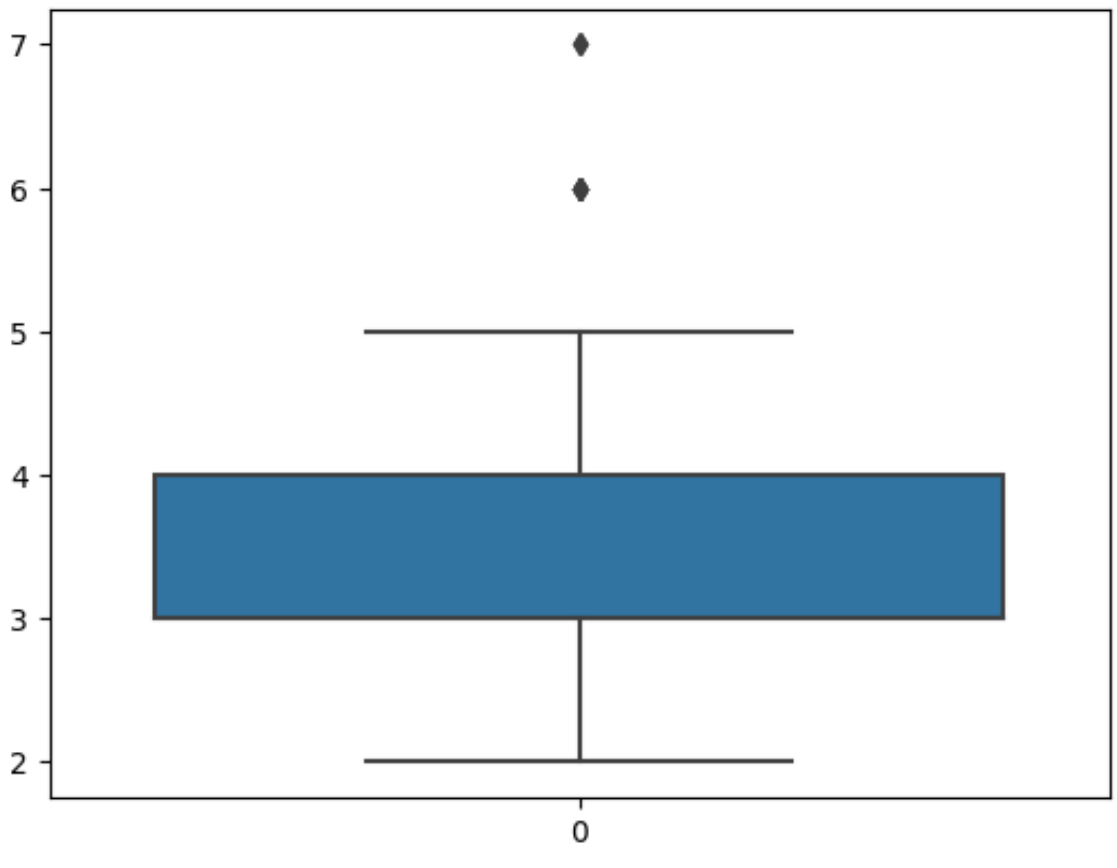
```
2.0
```

```
upper = Q3 + 1.5*iqr_Education
x= (df["Education"]>upper).sum()
print('No. of outliers in Education Column:',x,'Percentage of ounliers:',round((x/180)*100,2),"%")
```

```
No. of outliers in Education Column: 4 Percentage of ounliers: 2.22 %
```

Column-'Usage'

```
sns.boxplot(df["Usage"])
plt.show()
```



```
Q3 = np.percentile(df["Usage"], 75)
Q1 = np.percentile(df["Usage"], 25)
iqr_Usage = Q3 - Q1
iqr_Usage
```

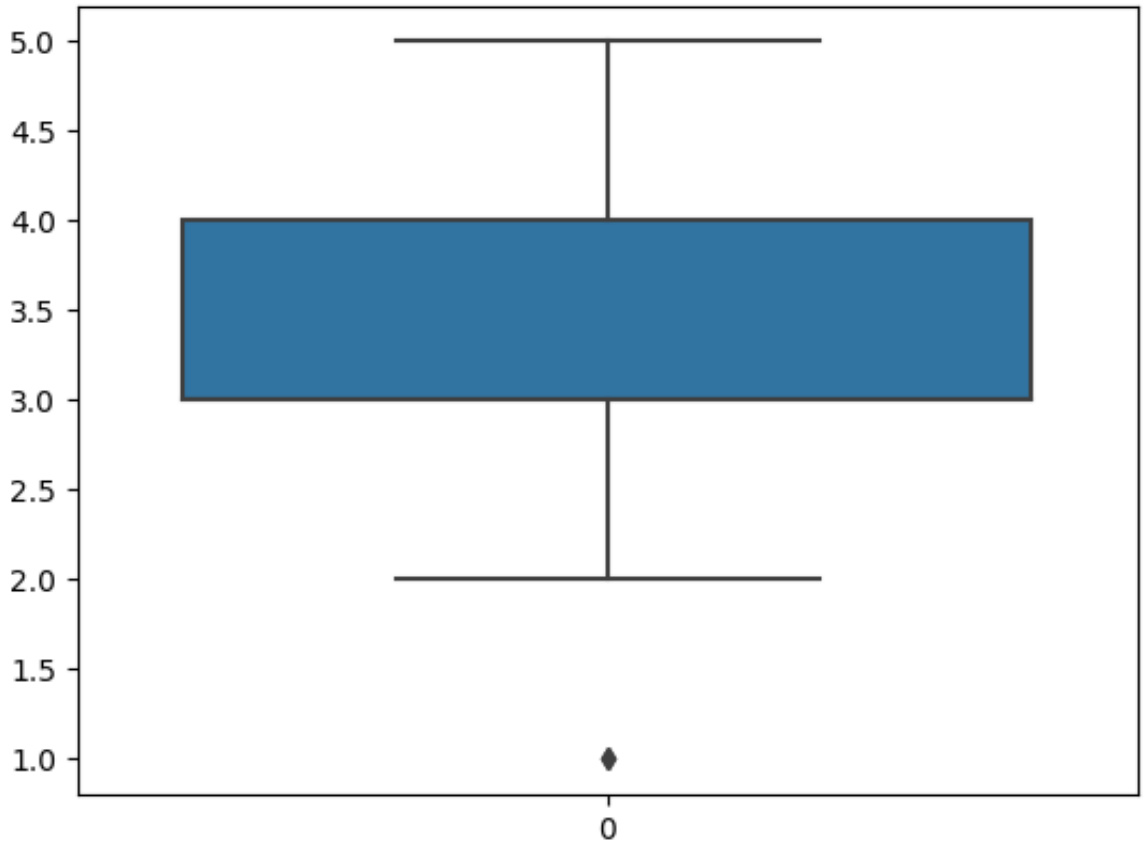
```
1.0
```

```
upper = Q3 + 1.5*iqr_Usage
x= (df["Usage"]>upper).sum()
print('No. of outliers in Usage Column:',x,'Percentage of ounliers:',round((x/180)*100,2),"%")
```

```
No. of outliers in Usage Column: 9 Percentage of ounliers: 5.0 %
```

Column-'Fitness'

```
sns.boxplot(df["Fitness"])
plt.show()
```




```
Q3 = np.percentile(df["Fitness"], 75)
Q1 = np.percentile(df["Fitness"], 25)
iqr_Fitness = Q3 - Q1
iqr_Fitness
```

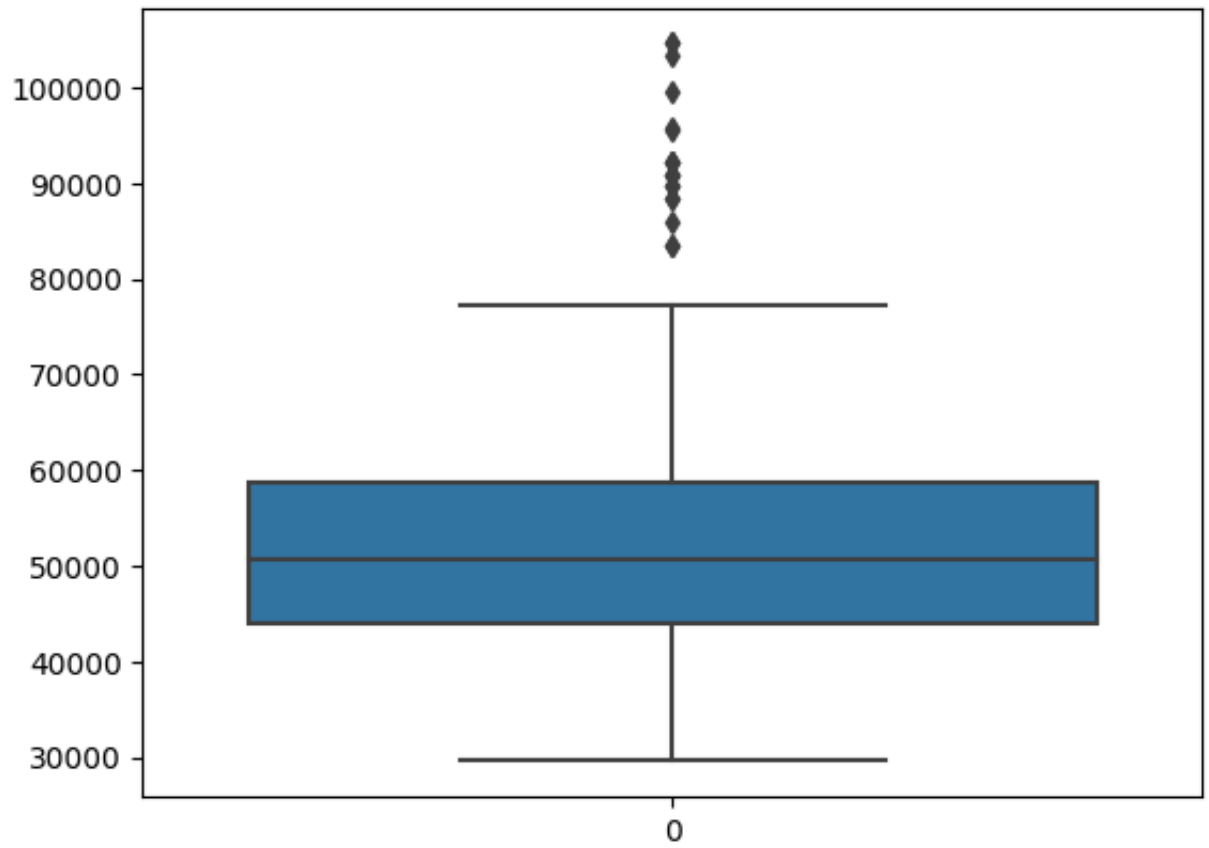
```
1.0
```

```
lower = Q1 - 1.5*iqr_Fitness
x= (df["Fitness"]<lower).sum()
print('No. of outliers in Fitness Column:',x,'Percentage of ounliers:',round((x/180)*100,2),"%")
```

```
No. of outliers in Fitness Column: 2 Percentage of ounliers: 1.11 %
```

Column-'Income'

```
sns.boxplot(df["Income"])
plt.show()
```




```
Q3 = np.percentile(df["Income"], 75)
Q1 = np.percentile(df["Income"], 25)
iqr_Income = Q3 - Q1
iqr_Income
```

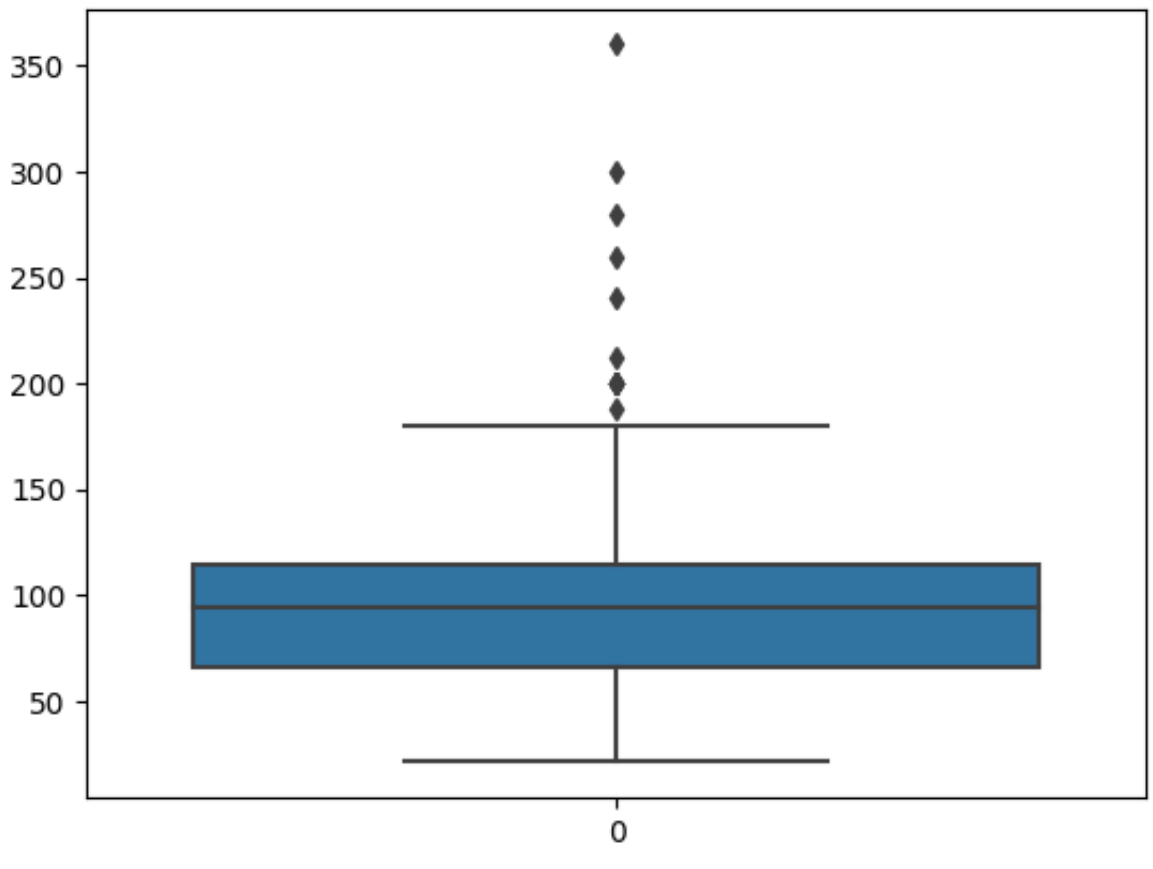
```
14609.25
```

```
upper = Q3 + 1.5*iqr_Income
x= (df["Income"]>upper).sum()
print('No. of outliers in Income Column:',x,'Percentage of ounliers:',round((x/180)*100,2),"%")
```

```
No. of outliers in Income Column: 19 Percentage of ounliers: 10.56 %
```

Column-'Miles'

```
sns.boxplot(df["Miles"])
plt.show()
```



```
Q3 = np.percentile(df["Miles"], 75)
Q1 = np.percentile(df["Miles"], 25)
iqr_Miles = Q3 - Q1
iqr_Miles
```

```
48.75
```

```
upper = Q3 + 1.5*iqr_Miles
x= (df["Miles"]>upper).sum()
print('No. of outliers in Miles Column:',x,'Percentage of ounliers:',round((x/180)*100,2),"%")
```

```
No. of outliers in Miles Column: 13 Percentage of ounliers: 7.22 %
```

Insights

- Find the outliers for every continuous variable in the dataset.

Recommendations

- We want to use boxplots to find the outliers in the given dataset.
- Generally we use 25-75 percentile to find Outliers. Thats why, I used 25 and 75 percentile to find outlies.

Assumptions

- No. of outliers in 'Age' Column: 5 and Percentage of ounliers: 2.78 %
- No. of outliers in 'Education' Column: 4 and Percentage of ounliers: 2.22 %
- No. of outliers in 'Usage' Column: 9 and Percentage of ounliers: 5.0 %
- No. of outliers in 'Fitness' Column: 2 and Percentage of ounliers: 1.11 %
- No. of outliers in 'Income' Column: 19 and Percentage of ounliers: 10.56 %
- No. of outliers in 'Miles' Column: 13 and Percentage of ounliers: 7.22 %

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

```
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

Column-'Age'

```
df_new=df
Q95 = np.percentile(df_new["Age"], 95)
Q5 = np.percentile(df_new["Age"], 5)
print('Age 5 Percentile:',Q5,'& 95 Percentile',Q95)
```

```
Age 5 Percentile: 20.0 & 95 Percentile 43.04999999999998
```

```
df_new['Age']=np.clip(df_new["Age"],Q5,Q95)
df_new
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.00	Male	14	Single	3	4	29562	112
1	KP281	20.00	Male	15	Single	2	3	31836	75
2	KP281	20.00	Female	14	Partnered	4	3	30699	66
3	KP281	20.00	Male	12	Single	3	3	32973	85
4	KP281	20.00	Male	13	Partnered	4	2	35247	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40.00	Male	21	Single	6	5	83416	200
176	KP781	42.00	Male	18	Single	5	4	89641	200
177	KP781	43.05	Male	16	Single	5	5	90886	160
178	KP781	43.05	Male	18	Partnered	4	5	104581	120
179	KP781	43.05	Male	18	Partnered	4	5	95508	180

180 rows × 9 columns

- Row 0,1,2,3 is change to 20(min),because data is value was less then 20.
- Row 177,178,179 is changed 43.05(max),because data is value was more then 43.05.

Column-'Education'

```
Q95 = np.percentile(df_new["Education"], 95)
Q5 = np.percentile(df_new["Education"], 5)
print('Education 5 Percentile:',Q5,'& 95 Percentile',Q95)
```

```
Education 5 Percentile: 14.0 & 95 Percentile 18.0
```

```
df_new['Education']=np.clip(df_new["Education"],Q5,Q95)
df_new
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.00	Male	14	Single	3	4	29562	112
1	KP281	20.00	Male	15	Single	2	3	31836	75
2	KP281	20.00	Female	14	Partnered	4	3	30699	66
3	KP281	20.00	Male	14	Single	3	3	32973	85
4	KP281	20.00	Male	14	Partnered	4	2	35247	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40.00	Male	18	Single	6	5	83416	200
176	KP781	42.00	Male	18	Single	5	4	89641	200
177	KP781	43.05	Male	16	Single	5	5	90886	160
178	KP781	43.05	Male	18	Partnered	4	5	104581	120
179	KP781	43.05	Male	18	Partnered	4	5	95508	180

180 rows × 9 columns

- Row 175 is changed to 18(max),because data is value was more then 18.

Column-'Usage'

```
Q95 = np.percentile(df_new["Usage"], 95)
Q5 = np.percentile(df_new["Usage"], 5)
print('Usage 5 Percentile:',Q5,'& 95 Percentile',Q95)
```

```
Usage 5 Percentile: 2.0 & 95 Percentile 5.049999999999983
```

```
df_new['Usage']=np.clip(df_new["Usage"],Q5,Q95)
df_new
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.00	Male	14	Single	3.00	4	29562	112
1	KP281	20.00	Male	15	Single	2.00	3	31836	75
2	KP281	20.00	Female	14	Partnered	4.00	3	30699	66
3	KP281	20.00	Male	14	Single	3.00	3	32973	85
4	KP281	20.00	Male	14	Partnered	4.00	2	35247	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40.00	Male	18	Single	5.05	5	83416	200
176	KP781	42.00	Male	18	Single	5.00	4	89641	200
177	KP781	43.05	Male	16	Single	5.00	5	90886	160
178	KP781	43.05	Male	18	Partnered	4.00	5	104581	120
179	KP781	43.05	Male	18	Partnered	4.00	5	95508	180

180 rows × 9 columns

Column-'Fitness

```
Q95 = np.percentile(df_new["Fitness"], 95)
Q5 = np.percentile(df_new["Fitness"], 5)
print('Fitness 5 Percentile:',Q5,'& 95 Percentile',Q95)
```

```
Fitness 5 Percentile: 2.0 & 95 Percentile 5.0
```

```
df_new['Fitness']=np.clip(df_new["Fitness"],Q5,Q95)
df_new
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.00	Male	14	Single	3.00	4	29562	112
1	KP281	20.00	Male	15	Single	2.00	3	31836	75
2	KP281	20.00	Female	14	Partnered	4.00	3	30699	66
3	KP281	20.00	Male	14	Single	3.00	3	32973	85
4	KP281	20.00	Male	14	Partnered	4.00	2	35247	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40.00	Male	18	Single	5.05	5	83416	200
176	KP781	42.00	Male	18	Single	5.00	4	89641	200
177	KP781	43.05	Male	16	Single	5.00	5	90886	160
178	KP781	43.05	Male	18	Partnered	4.00	5	104581	120
179	KP781	43.05	Male	18	Partnered	4.00	5	95508	180

180 rows × 9 columns

Column-'Income'

```
Q95 = np.percentile(df_new["Income"], 95)
Q5 = np.percentile(df_new["Income"], 5)
print('Income 5 Percentile:',Q5,'& 95 Percentile',Q95)
```

```
Income 5 Percentile: 34053.15 & 95 Percentile 90948.24999999999
```

```
df_new['Income']=np.clip(df_new["Income"],Q5,Q95)
df_new
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.00	Male	14	Single	3.00	4	34053.15	112
1	KP281	20.00	Male	15	Single	2.00	3	34053.15	75
2	KP281	20.00	Female	14	Partnered	4.00	3	34053.15	66
3	KP281	20.00	Male	14	Single	3.00	3	34053.15	85
4	KP281	20.00	Male	14	Partnered	4.00	2	35247.00	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40.00	Male	18	Single	5.05	5	83416.00	200
176	KP781	42.00	Male	18	Single	5.00	4	89641.00	200
177	KP781	43.05	Male	16	Single	5.00	5	90886.00	160
178	KP781	43.05	Male	18	Partnered	4.00	5	90948.25	120
179	KP781	43.05	Male	18	Partnered	4.00	5	90948.25	180

180 rows × 9 columns

- Row 0,1,2,3 changed income to 34053.15(min),because data is value was less then 34053.15.
- Row 178 and 179 changed income 90958.25(max),because data is value was more then 90958.25.

Column-'Miles'

```
Q95 = np.percentile(df_new["Miles"], 95)
Q5 = np.percentile(df_new["Miles"], 5)
print('Miles 5 Percentile:',Q5,'& 95 Percentile',Q95)
```

```
Miles 5 Percentile: 47.0 & 95 Percentile 200.0
```

```
df_new['Miles']=np.clip(df_new["Miles"],Q5,Q95)
df_new
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.00	Male	14	Single	3.00	4	34053.15	112
1	KP281	20.00	Male	15	Single	2.00	3	34053.15	75
2	KP281	20.00	Female	14	Partnered	4.00	3	34053.15	66
3	KP281	20.00	Male	14	Single	3.00	3	34053.15	85
4	KP281	20.00	Male	14	Partnered	4.00	2	35247.00	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40.00	Male	18	Single	5.05	5	83416.00	200
176	KP781	42.00	Male	18	Single	5.00	4	89641.00	200
177	KP781	43.05	Male	16	Single	5.00	5	90886.00	160
178	KP781	43.05	Male	18	Partnered	4.00	5	90948.25	120
179	KP781	43.05	Male	18	Partnered	4.00	5	90948.25	180

180 rows × 9 columns

- Row 4 is changed Miles to 47(min),because data is value was less then 47.
- Row 175 and 176 is change to 200(max),because data is value was more then 200.

Insights

- Remove/clip the data between the 5 percentile and 95 percentile.

Recommendations

- We want to use np.clip() for clipping the data.
- We want to find 5 And 95 percentile for each column and then clip the data.
- Age(Years):- 5 Percentile: 20.0 & 95 Percentile 43.04999999999998
- Education(Years):- 5 Percentile: 14.0 & 95 Percentile 18.0
- Usage:- 5 Percentile: 2.0 & 95 Percentile 5.049999999999983
- Fitness:- 5 Percentile: 2.0 & 95 Percentile 5.0
- Income:- 5 Percentile: 34053.15 & 95 Percentile 90948.24999999999
- Miles:- 5 Percentile: 47.0 & 95 Percentile 200.0

Assumptions

- Column-'Age'
  - Row 0,1,2,3 is change to 20(min),because data is value was less then 20.
  - Row 177,178,179 is changed 43.05(max),because data is value was more then 43.05.
- Column-'Education
  - Row 175 is changed to 18(max),because data is value was more then 18.
- Column-'Income'
  - Row 0,1,2,3 changed income to 34053.15(min),because data is value was less then 34053.15.
  - Row 178 and 179 changed income 90958.25(max),because data is value was more then 90958.25.
- Column-'Miles'
  - Row 4 is changed Miles to 47(min),because data is value was less then 47.
  - Row 175 and 176 is change to 200(max),because data is value was more then 200.
- Simlaraly In column 'Usage' And 'Fitness'.

3. Check if features like marital status, Gender, and age have any effect on the product purchased

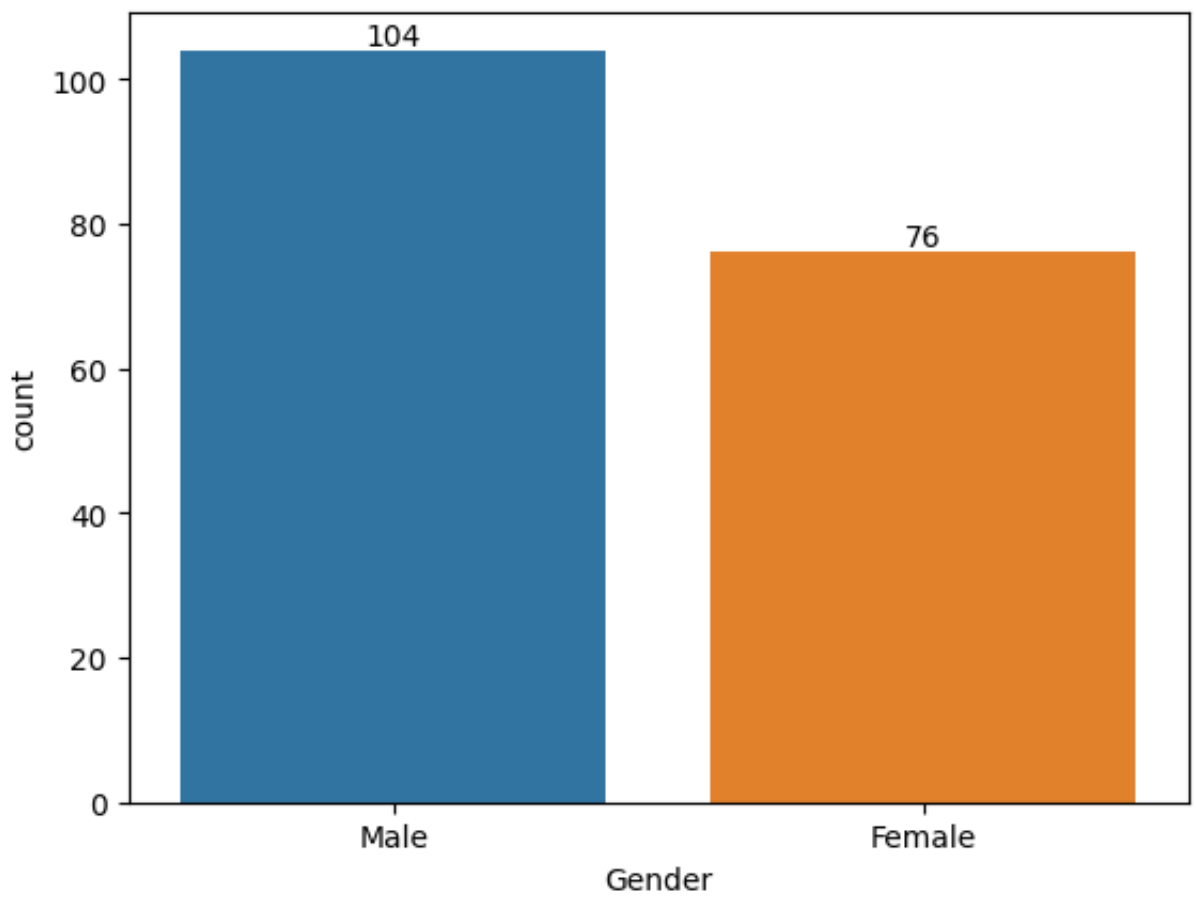
- **Find if there is any relationship between the categorical variables and the output variable in the data.**

```
df_new.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.0	Male	14	Single	3.0	4	34053.15	112
1	KP281	20.0	Male	15	Single	2.0	3	34053.15	75
2	KP281	20.0	Female	14	Partnered	4.0	3	34053.15	66
3	KP281	20.0	Male	14	Single	3.0	3	34053.15	85
4	KP281	20.0	Male	14	Partnered	4.0	2	35247.00	47

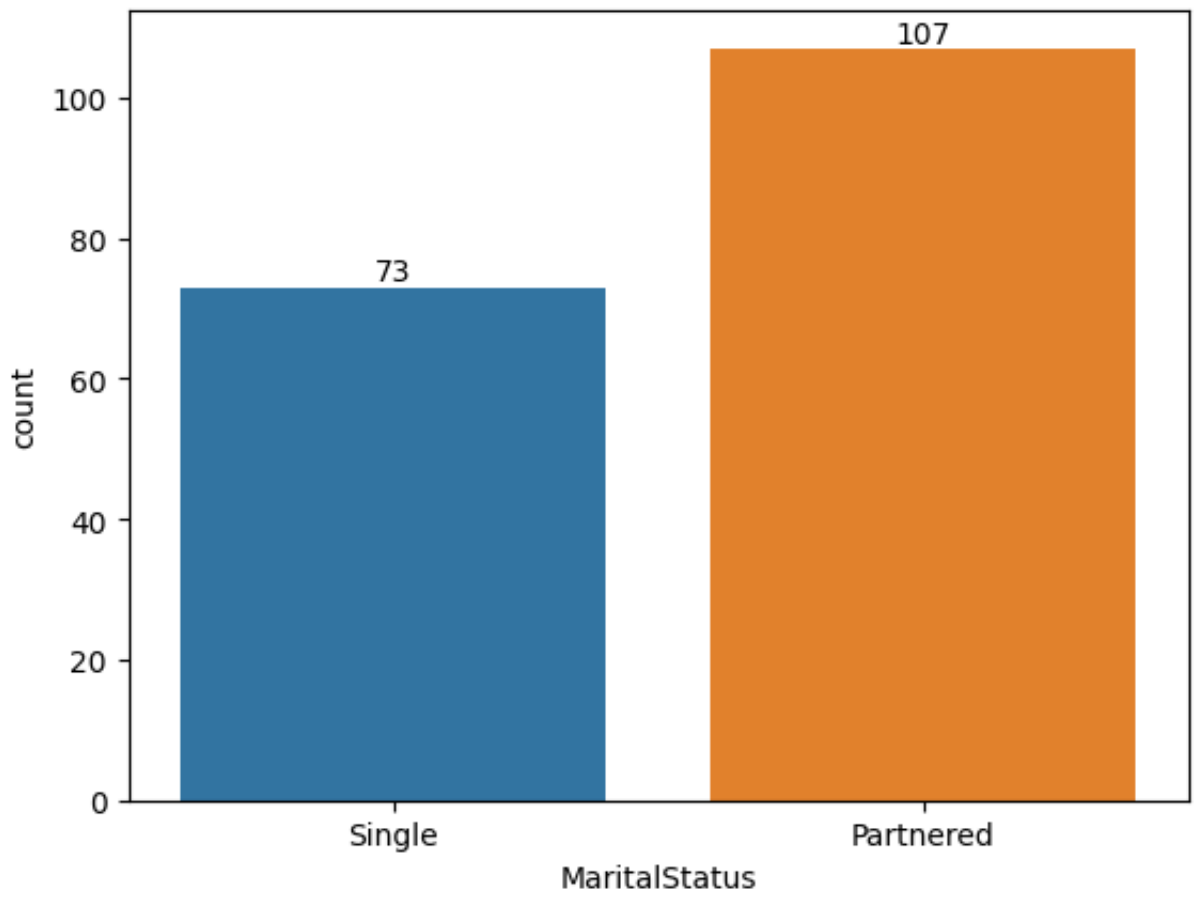
Categorical Variables

```
ax = sns.countplot(df_new, x="Gender")
ax.bar_label(ax.containers[0])
plt.show()
```



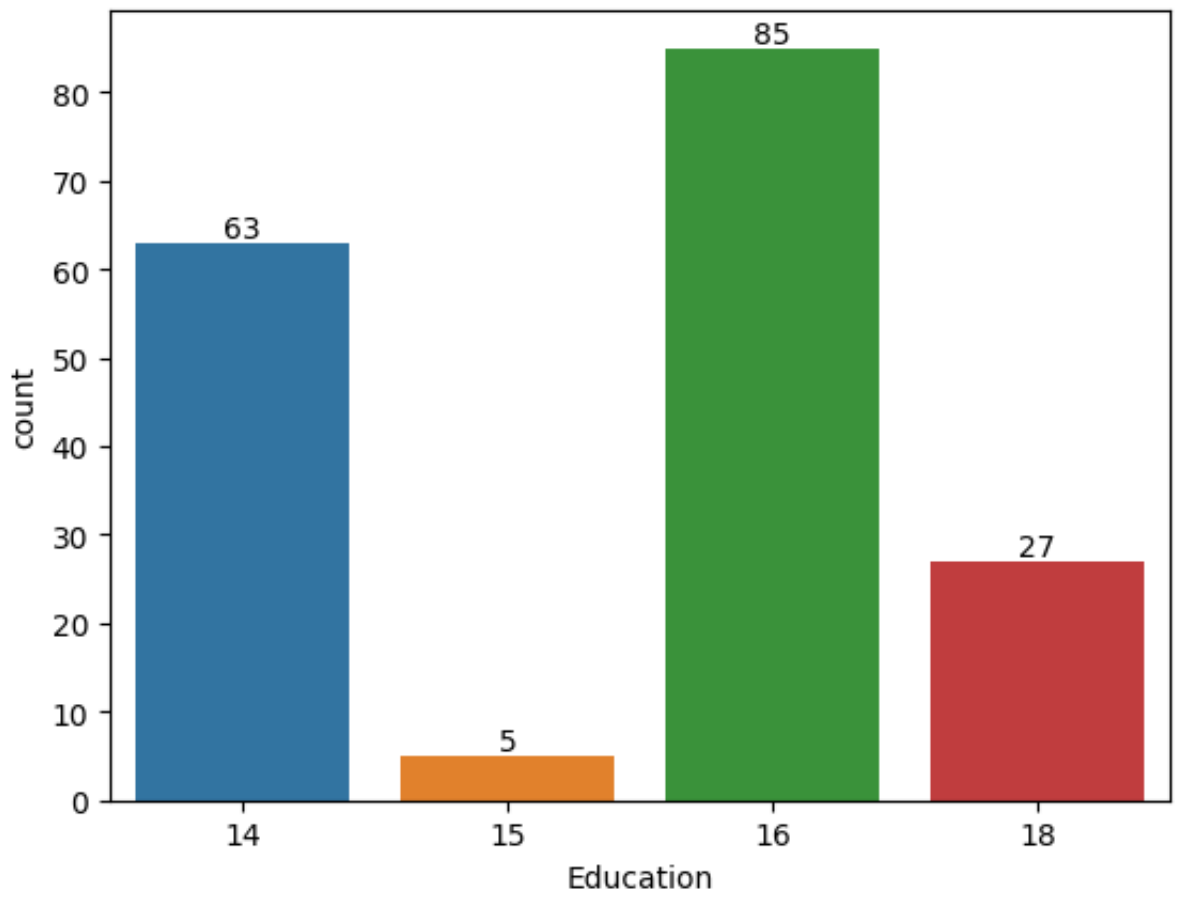
- There is 104 Male and 76 Female.

```
ax = sns.countplot(df_new, x="MaritalStatus")
ax.bar_label(ax.containers[0])
plt.show()
```



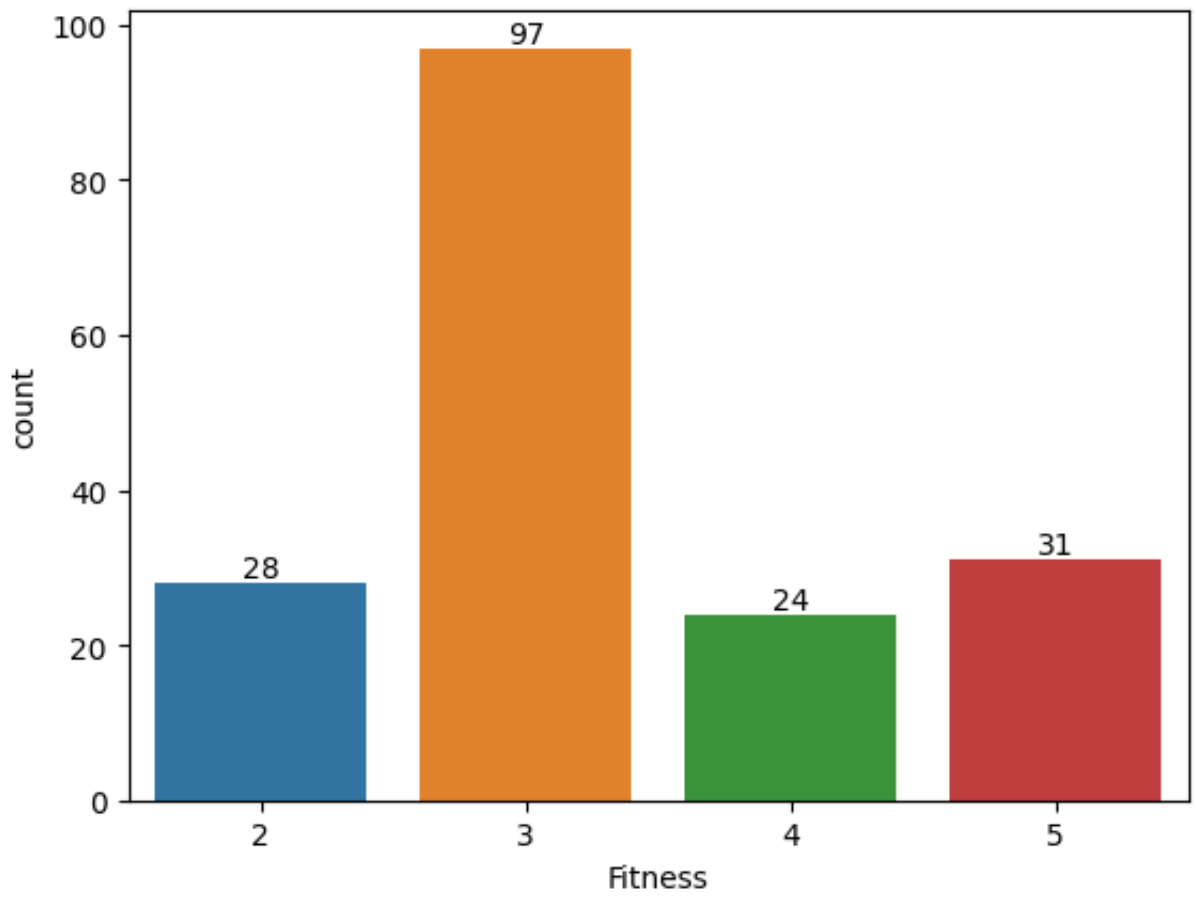
- Partnereds(107) are more then Singles(73).

```
ax = sns.countplot(df_new, x="Education")
ax.bar_label(ax.containers[0])
plt.show()
```



- There is 85 persons with 16 years if education.
- There is 63 persons with 14 years if education.
- There is 27 persons with 18 years if education.
- There is 5 persons with 15 years if education.

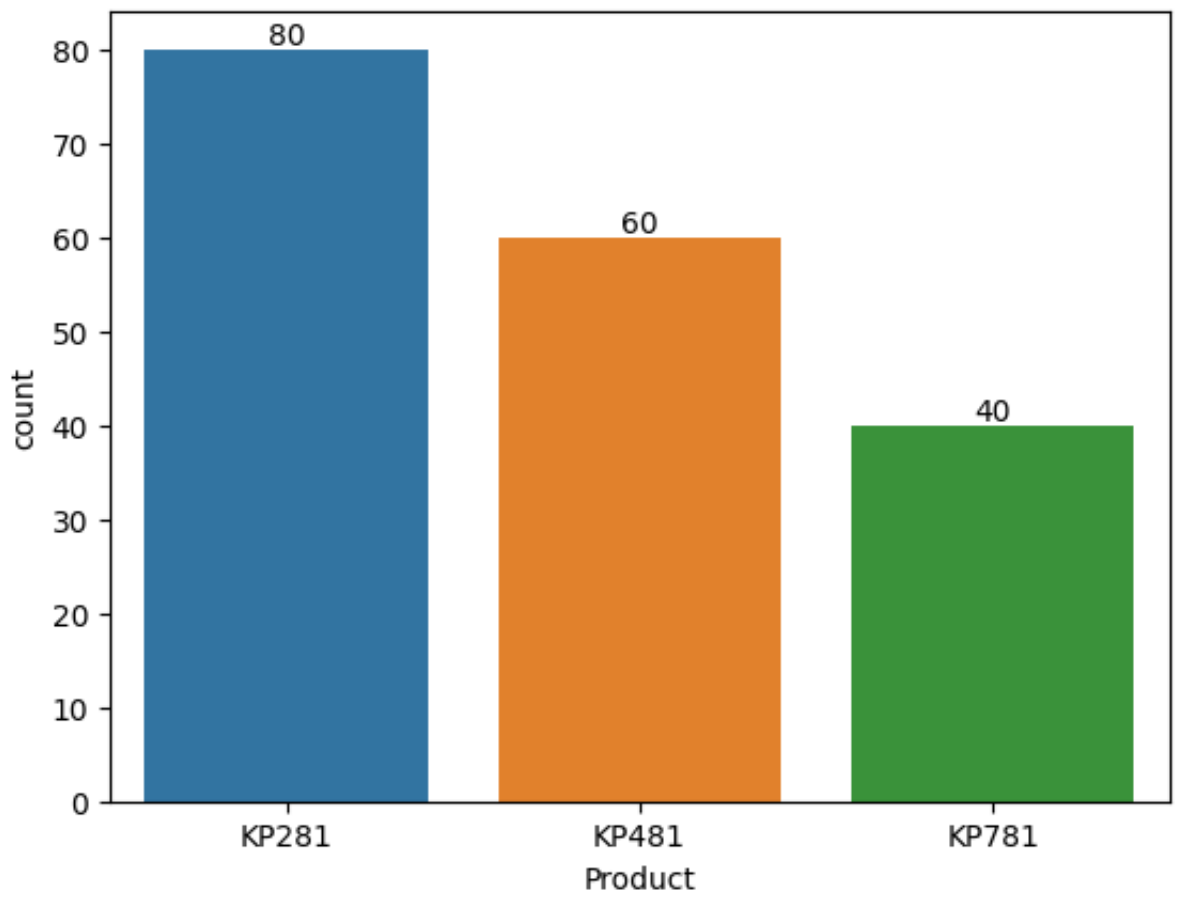
```
ax = sns.countplot(df_new, x="Fitness")
ax.bar_label(ax.containers[0])
plt.show()
```



1 is poor and 5 is Excellent

- There is 97 persons with 3 Fitness Score.
- There is 31 persons with 5 Fitness Score.
- There is 28 persons with 2 Fitness Score.
- There is 24 persons with 4 Fitness Score.

```
ax = sns.countplot(df_new, x="Product")
ax.bar_label(ax.containers[0])
plt.show()
```

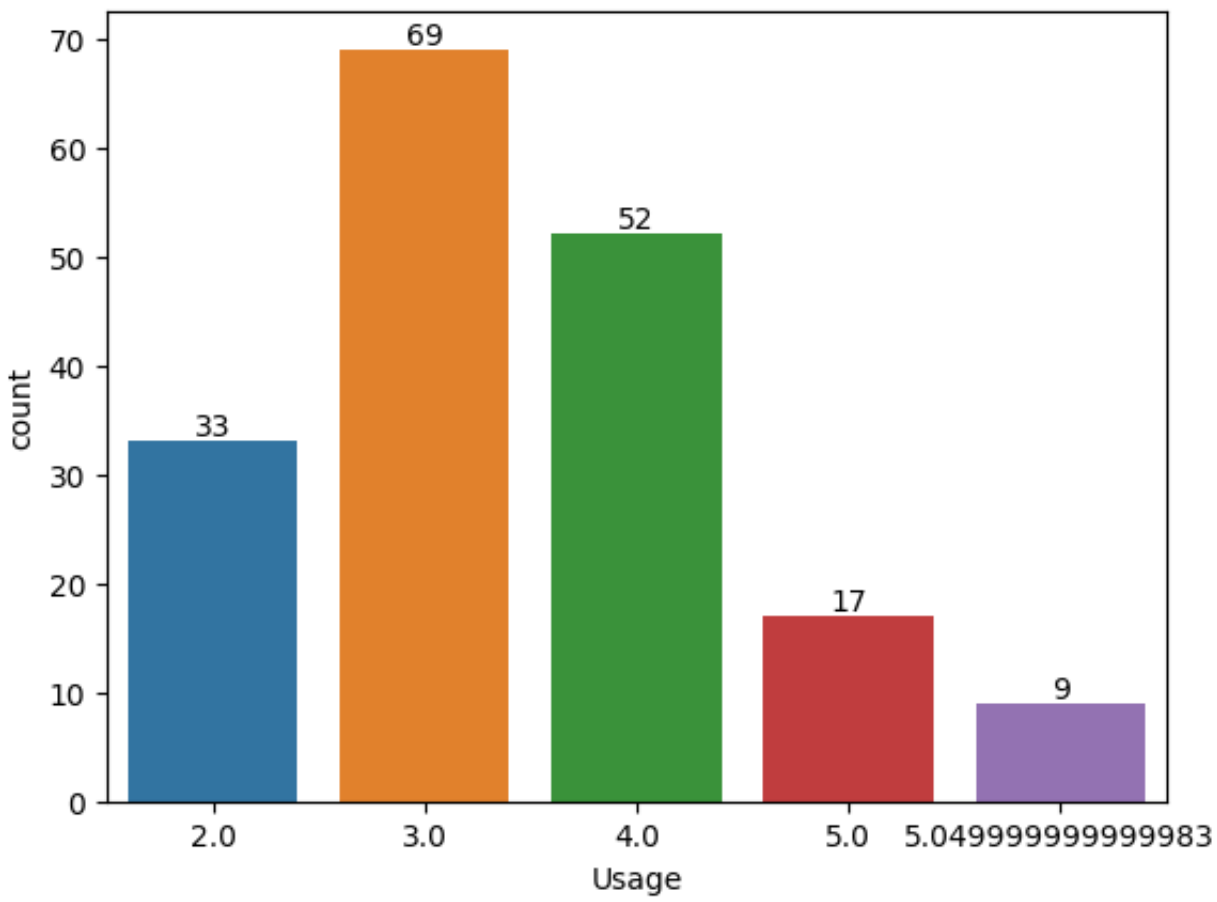


There is 3 type of products

----------(Count)

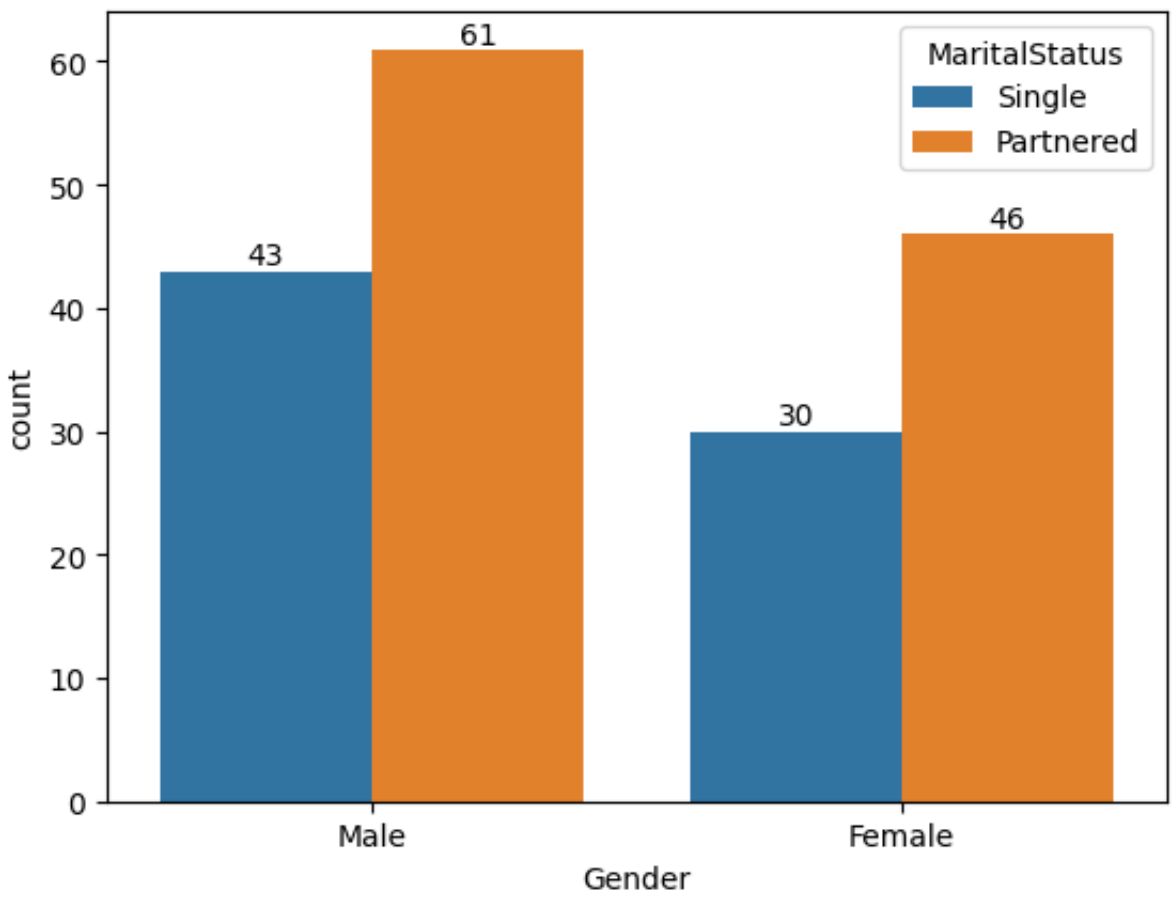
1. KP281 (80)
2. KP481 (60)
3. KP781 (40)

```
ax = sns.countplot(df_new, x="Usage")
ax.bar_label(ax.containers[0])
plt.show()
```



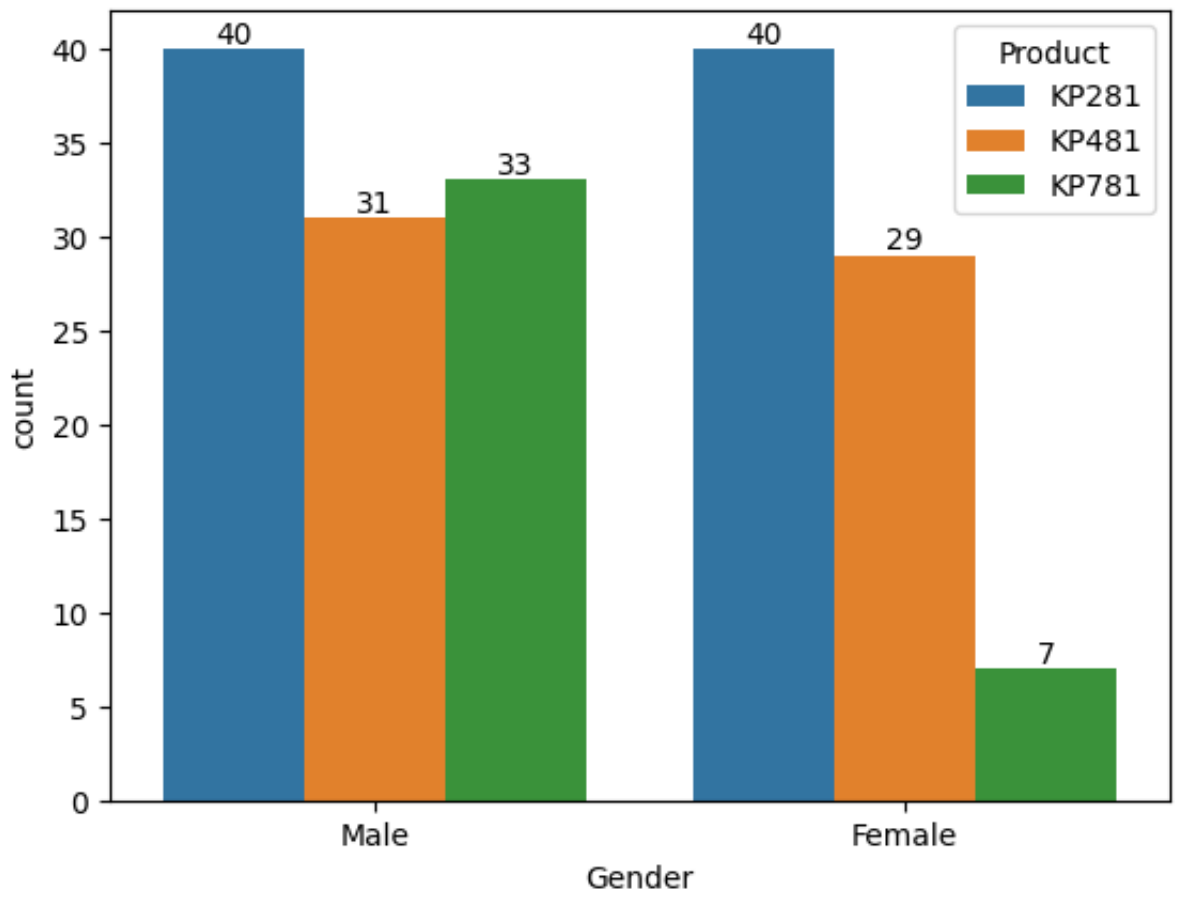
- There is 69 persons with 3 average number of times :plans to use the treadmill each week

```
ax = sns.countplot(df_new, x="Gender", hue='MaritalStatus')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

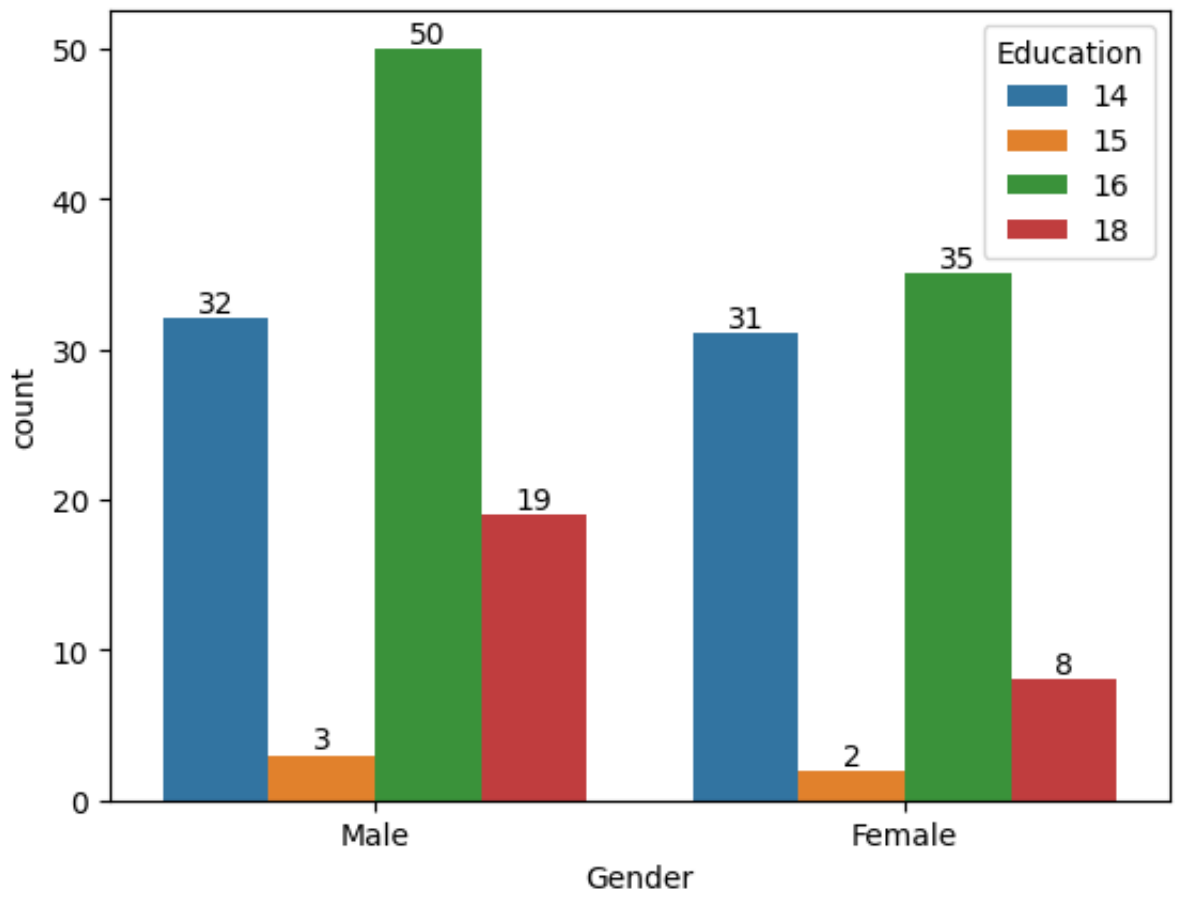


- At most 61 persons who are male and are partnered are using Aerofit product.

```
ax = sns.countplot(df_new, x="Gender", hue='Product')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

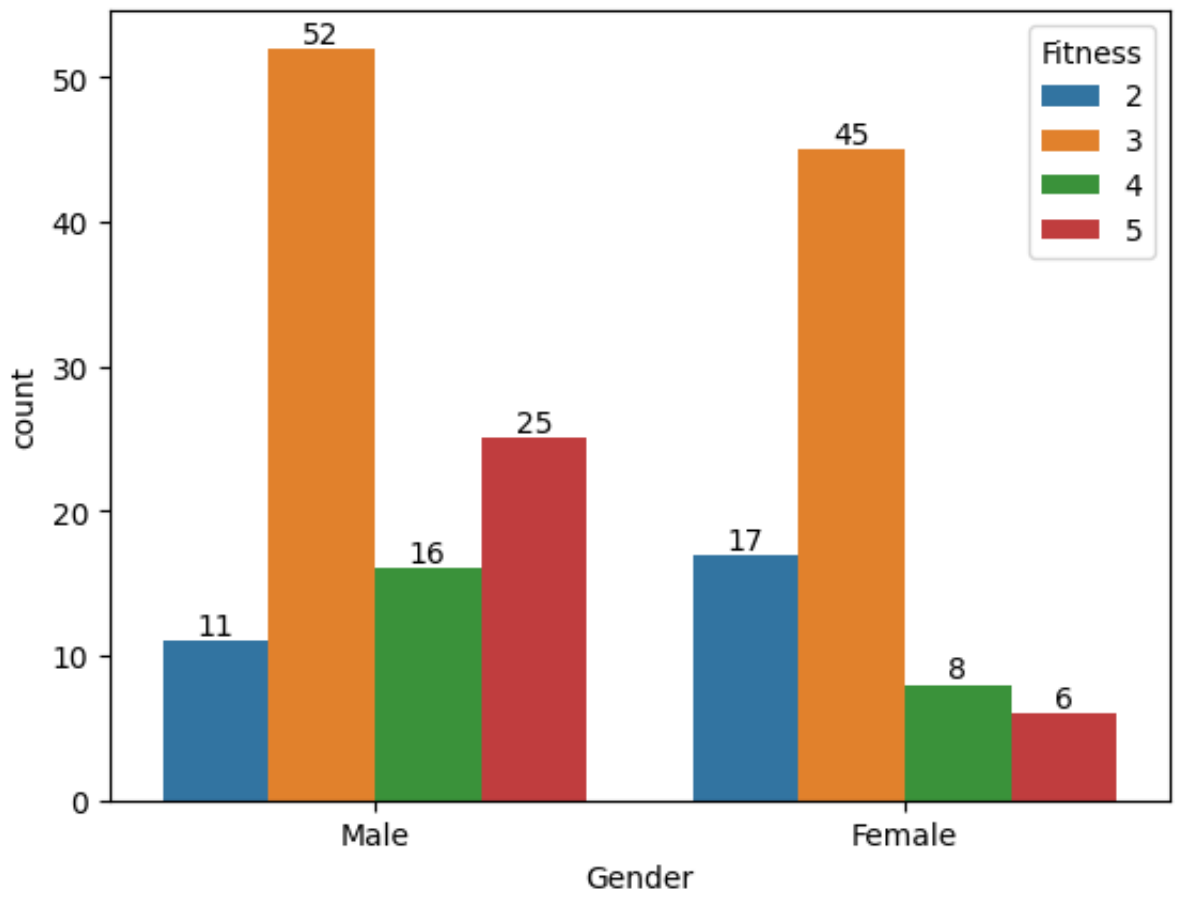



```
ax = sns.countplot(df_new, x="Gender", hue='Education')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



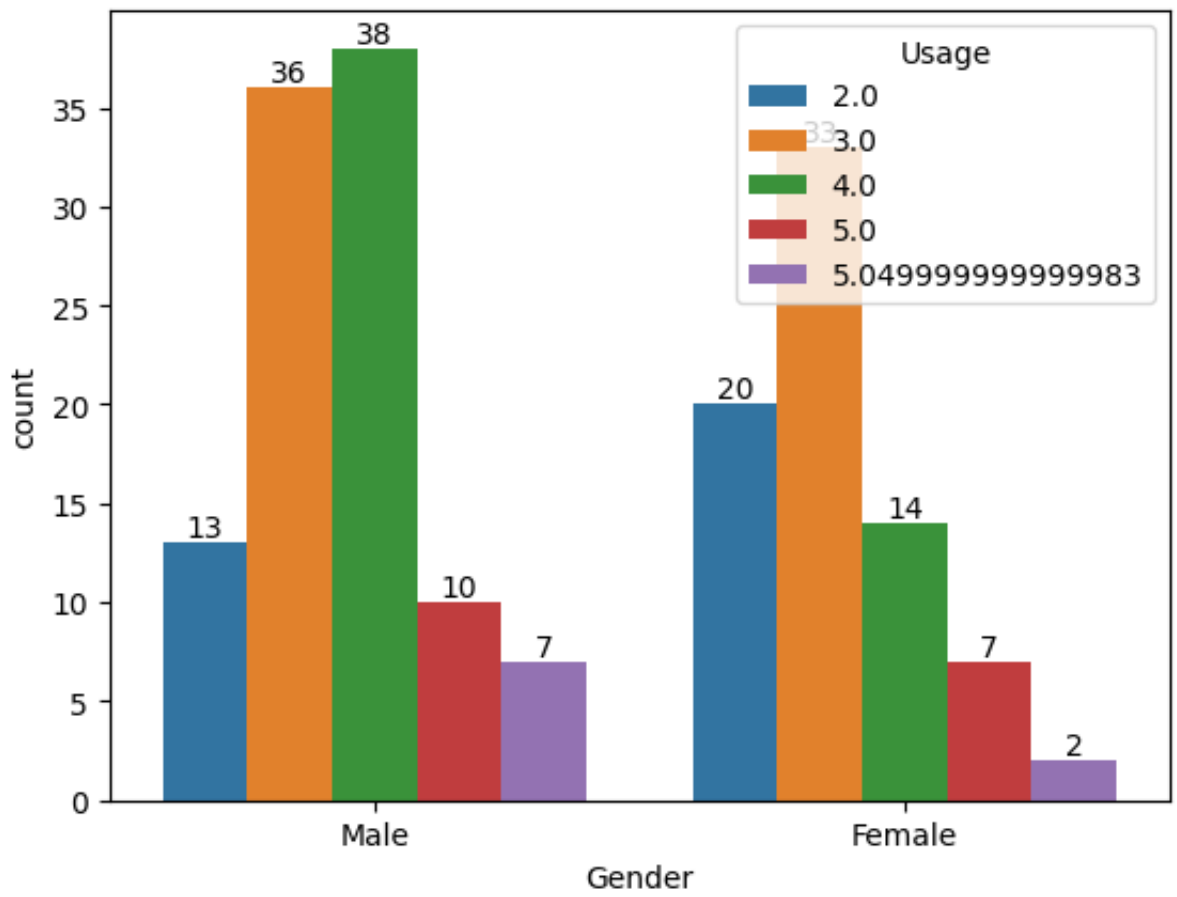
- At most 50 persons who are male and having 16 years od Education are using Aerofit product.
- At most 35 persons who are female and having 16 years od Education are using Aerofit product.

```
ax = sns.countplot(df_new, x="Gender", hue='Fitness')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



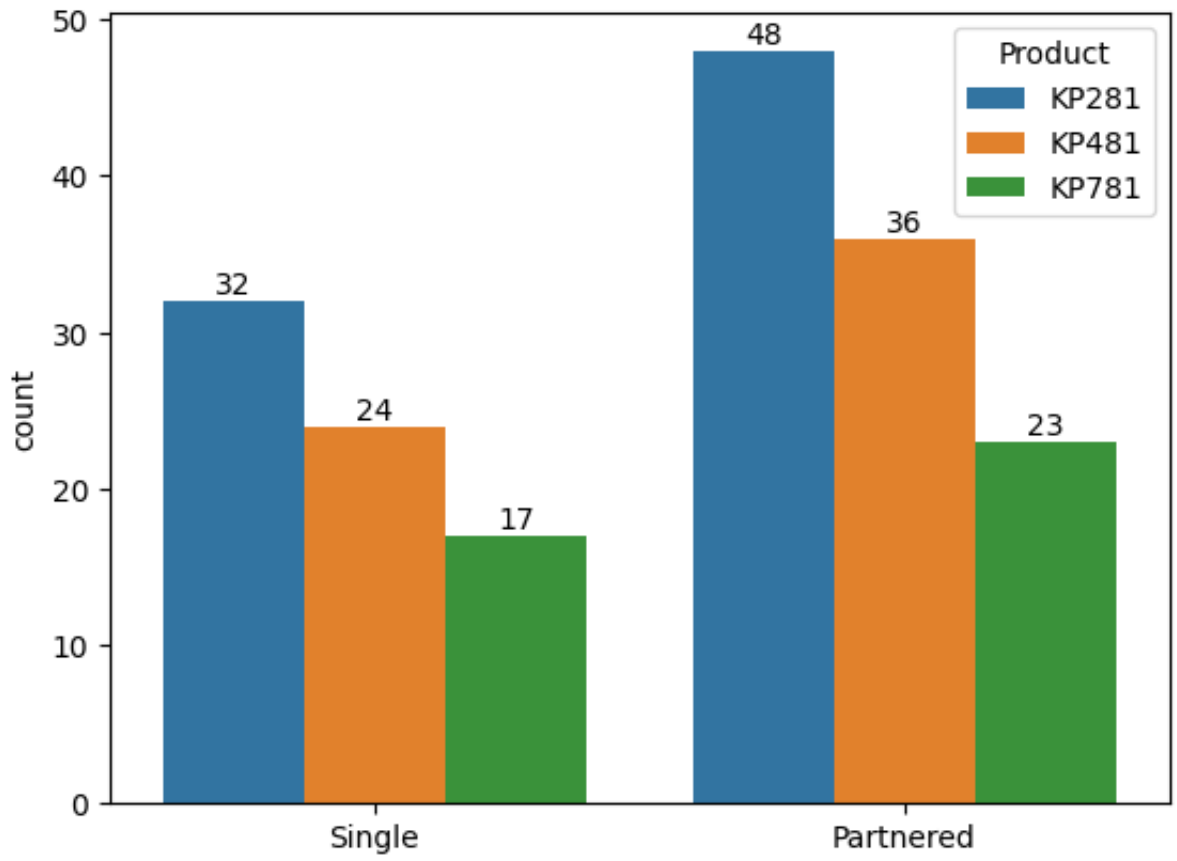
- At most 52 persons who are male and having 3 fitness score are using Aerofit product.
- At most 45 persons who are female and having 3 fitness score are using Aerofit product.

```
ax = sns.countplot(df_new, x="Gender", hue='Usage')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

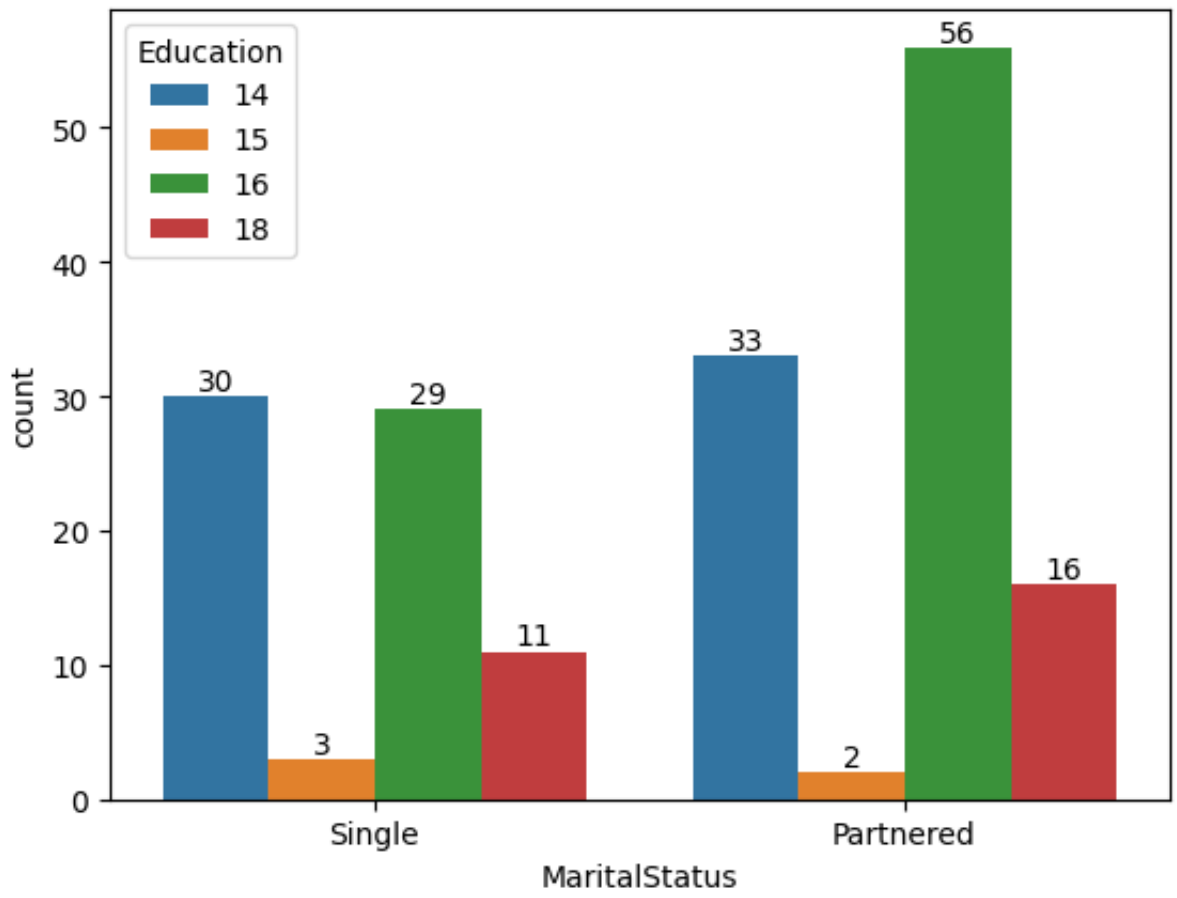


- Males are at most using product 4 times per week.(38)
- Females are at most using product 3 times per week.(33)

```
ax = sns.countplot(df_new, x="MaritalStatus", hue='Product')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

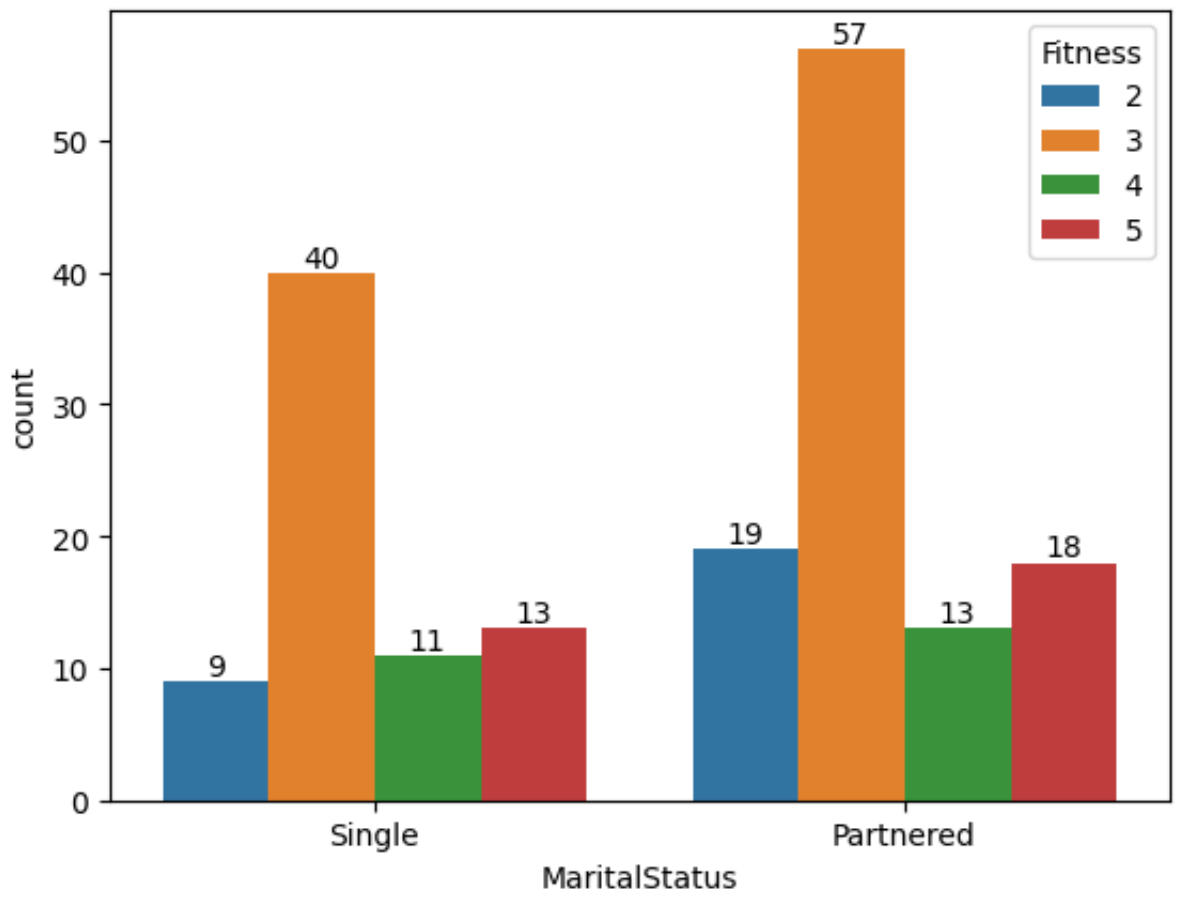



```
ax = sns.countplot(df_new, x="MaritalStatus", hue='Education')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



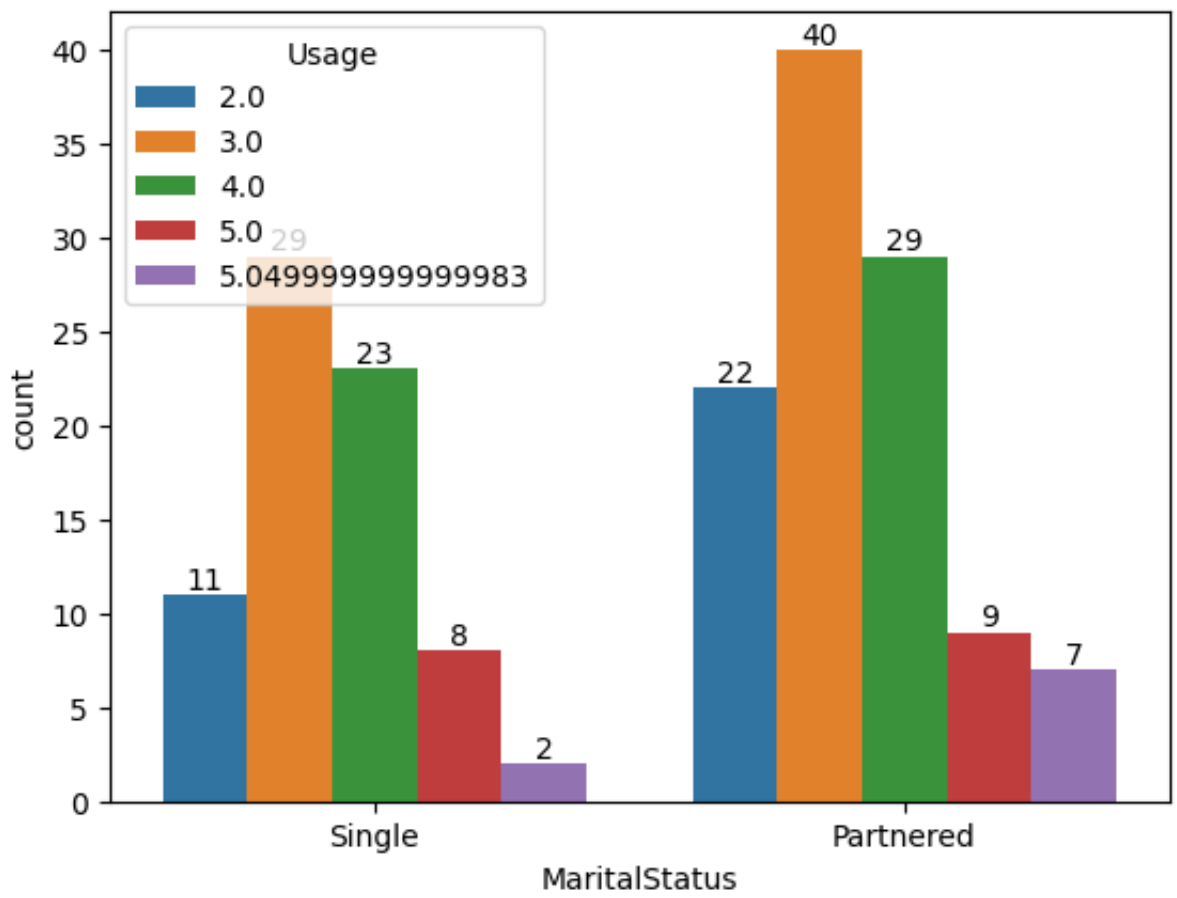
- Partnerd Are more Education Years.

```
ax = sns.countplot(df_new, x="MaritalStatus", hue='Fitness')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



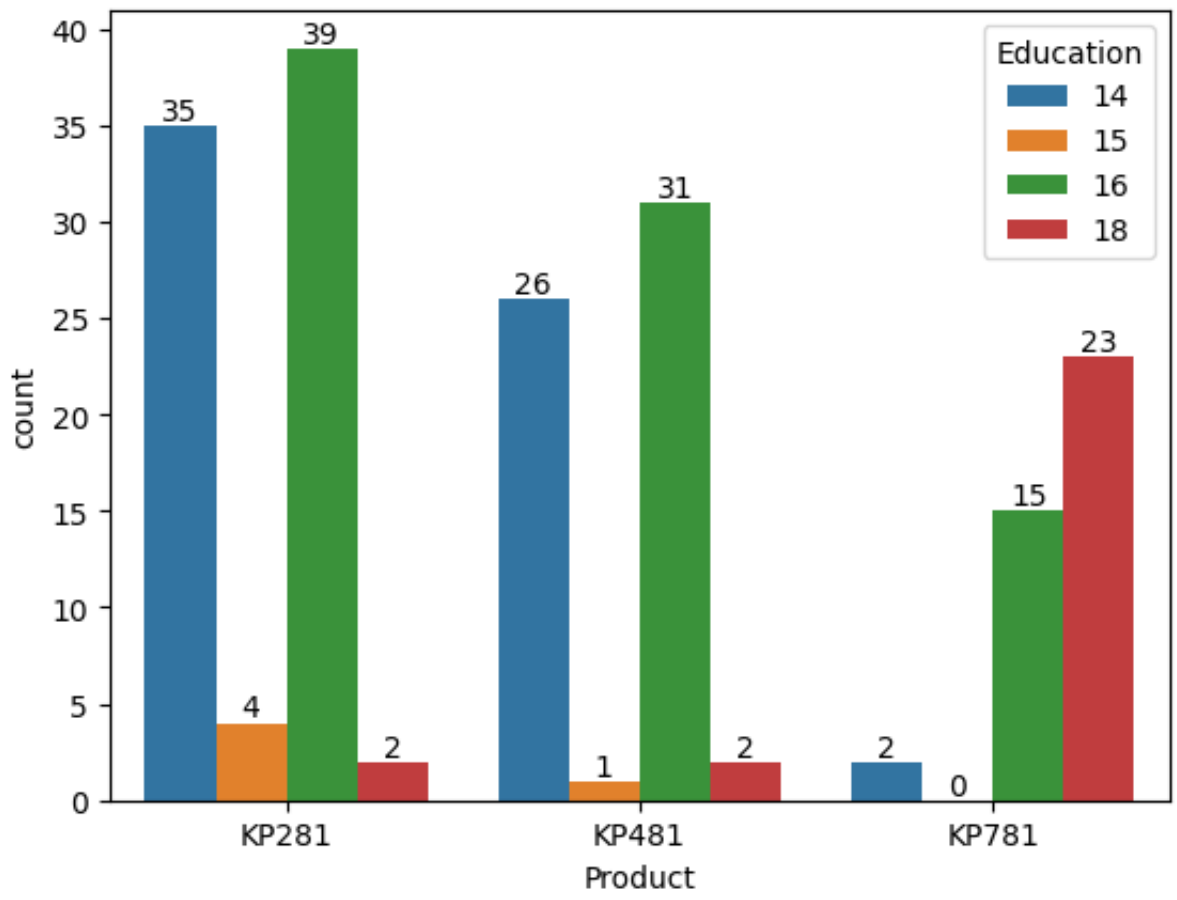
- Partnered Have more 3 fitness score

```
ax = sns.countplot(df_new, x="MaritalStatus", hue='Usage')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

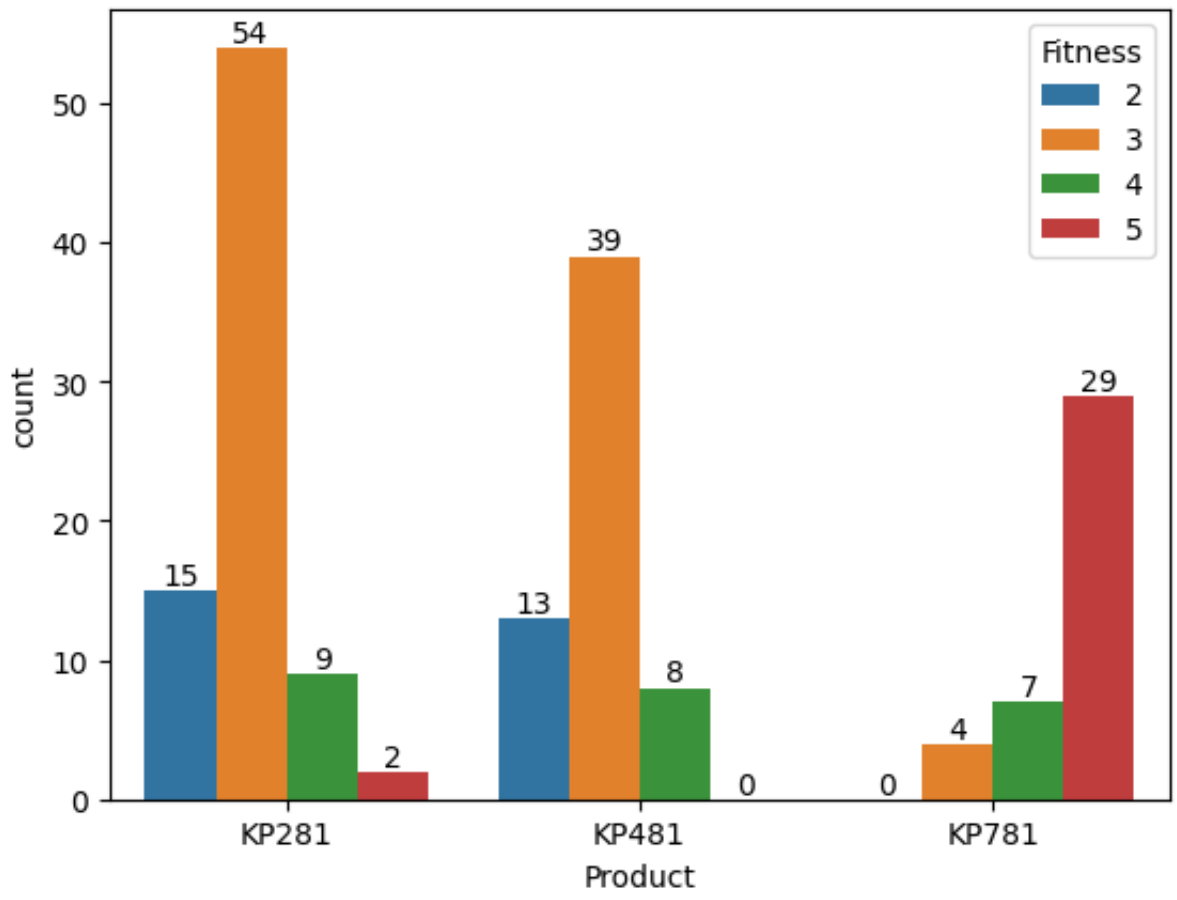


- Partnered have more Usage time per week

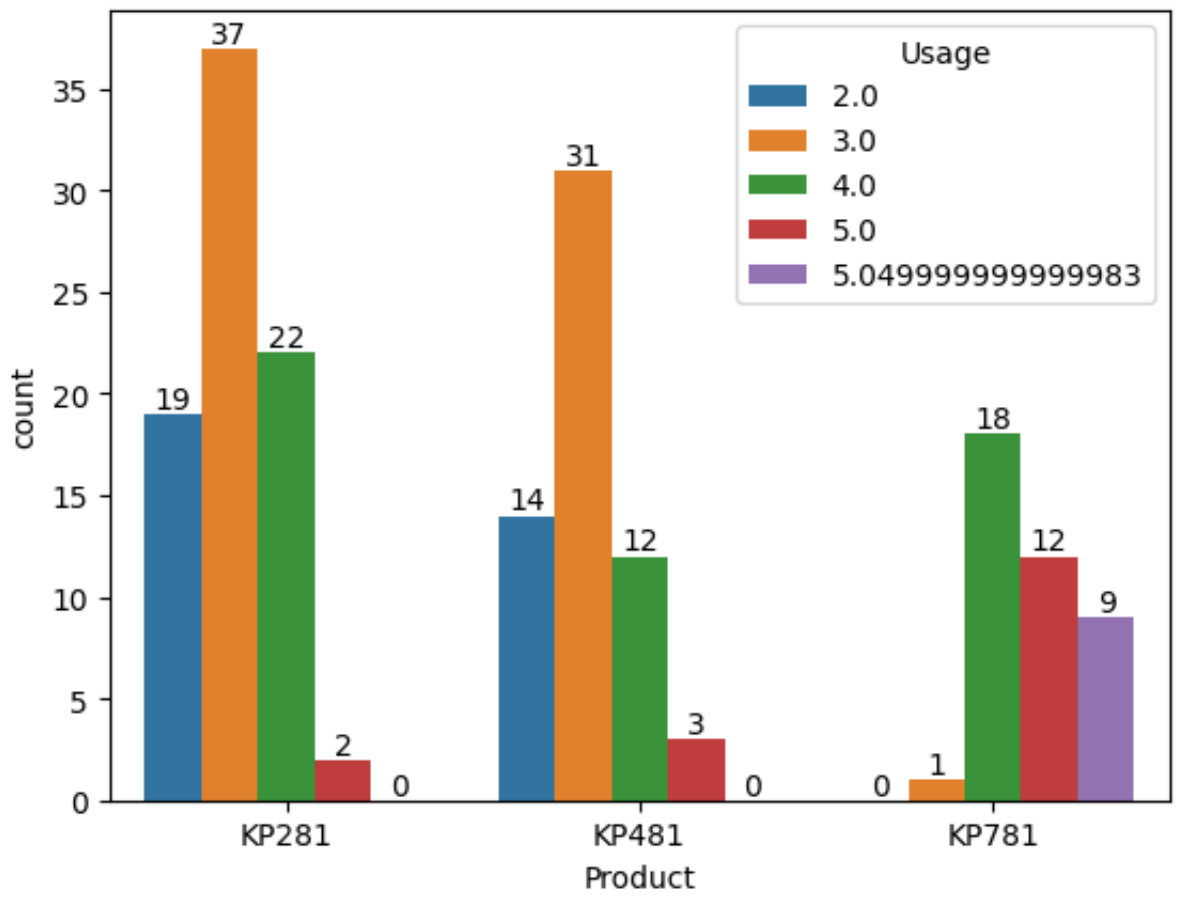
```
ax = sns.countplot(df_new, x="Product", hue='Education')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```




```
ax = sns.countplot(df_new, x="Product", hue='Fitness')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

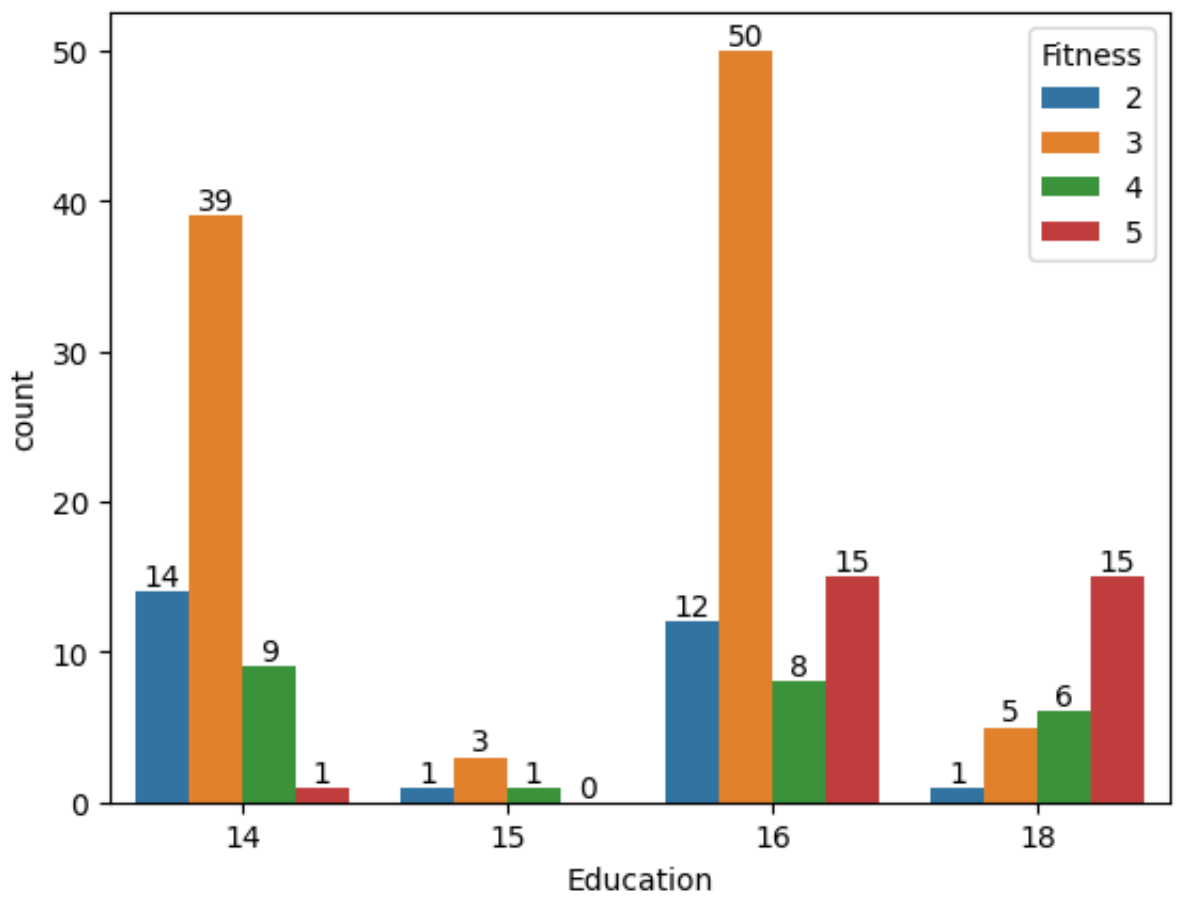


```
ax = sns.countplot(df_new, x="Product", hue='Usage')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



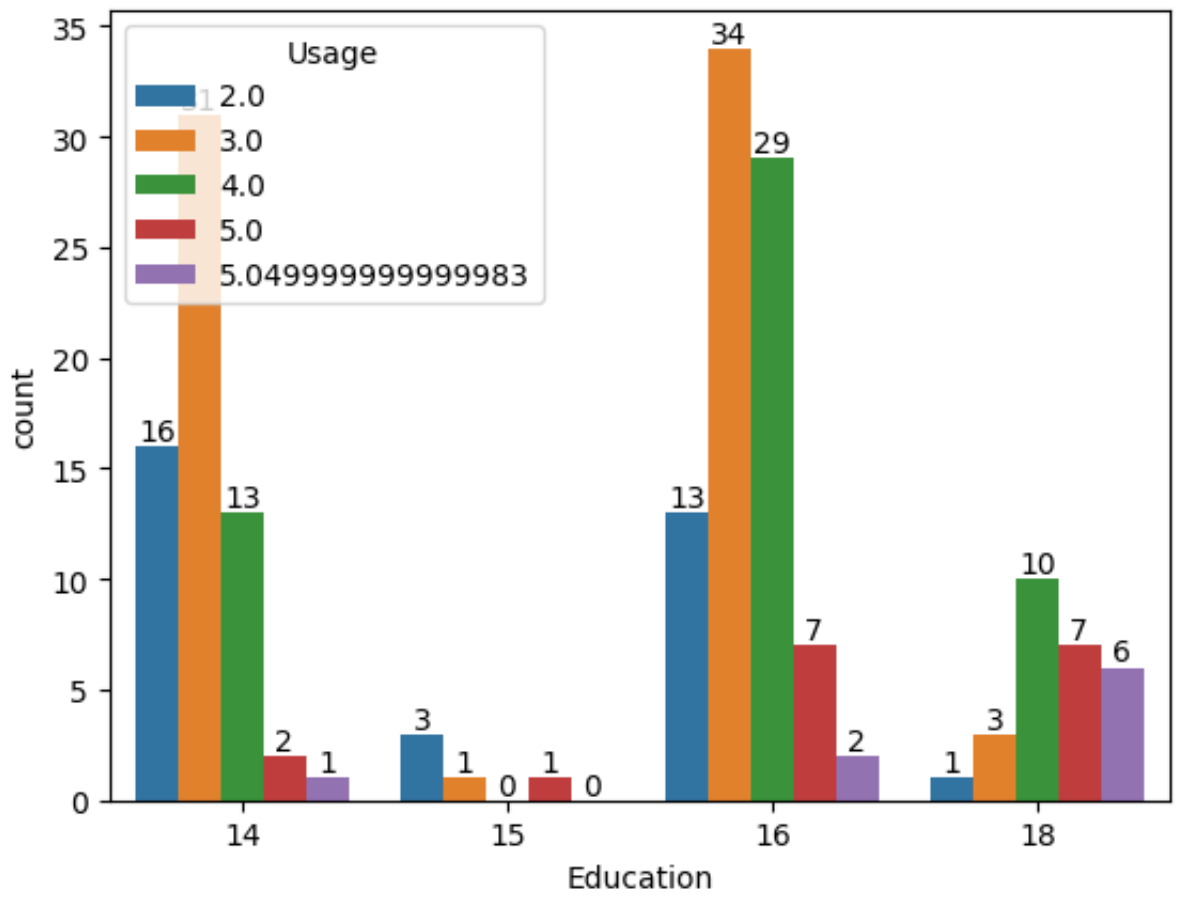
- Product KP281 have more 3 time usage per week

```
ax = sns.countplot(df_new, x="Education", hue='Fitness')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

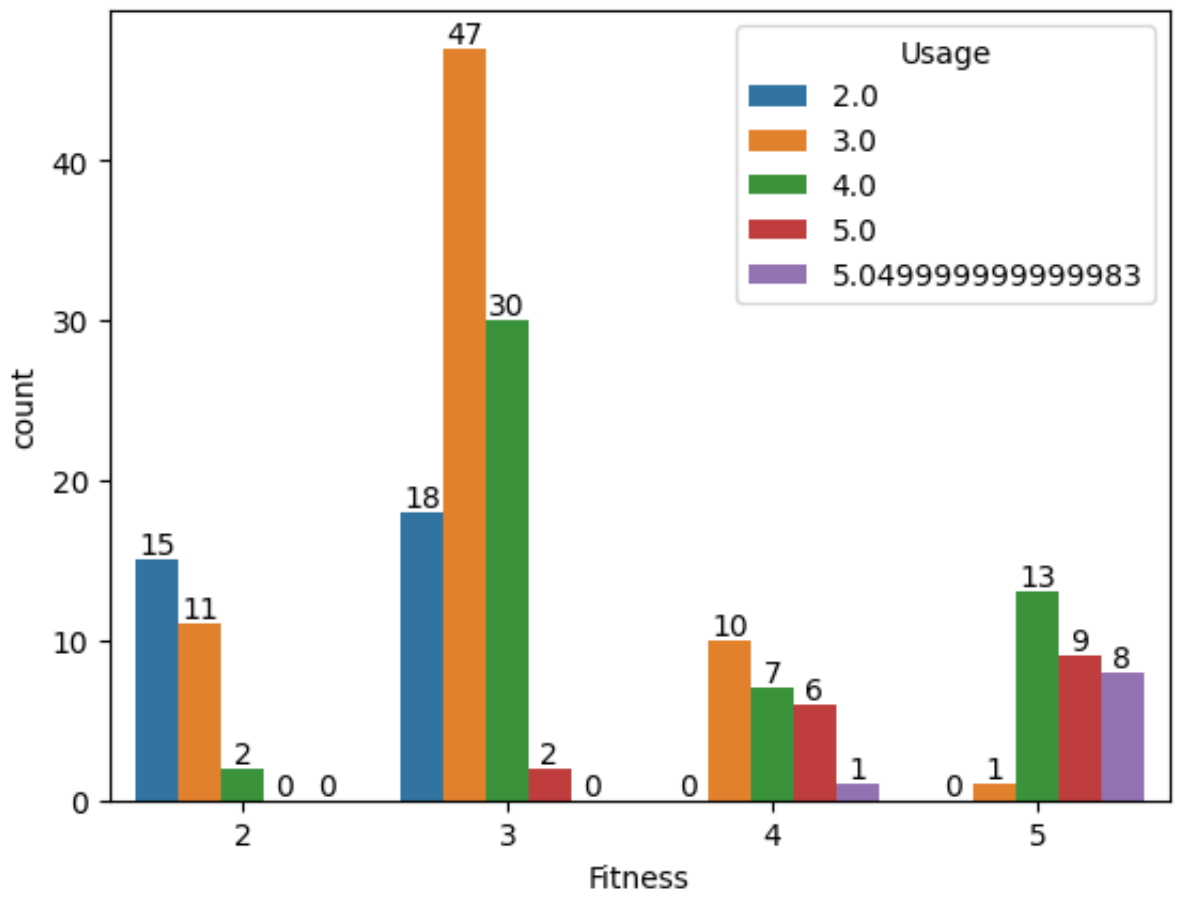


- People with 16 years of edcutaion have more 3 fitness score

```
ax = sns.countplot(df_new, x="Education", hue='Usage')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```

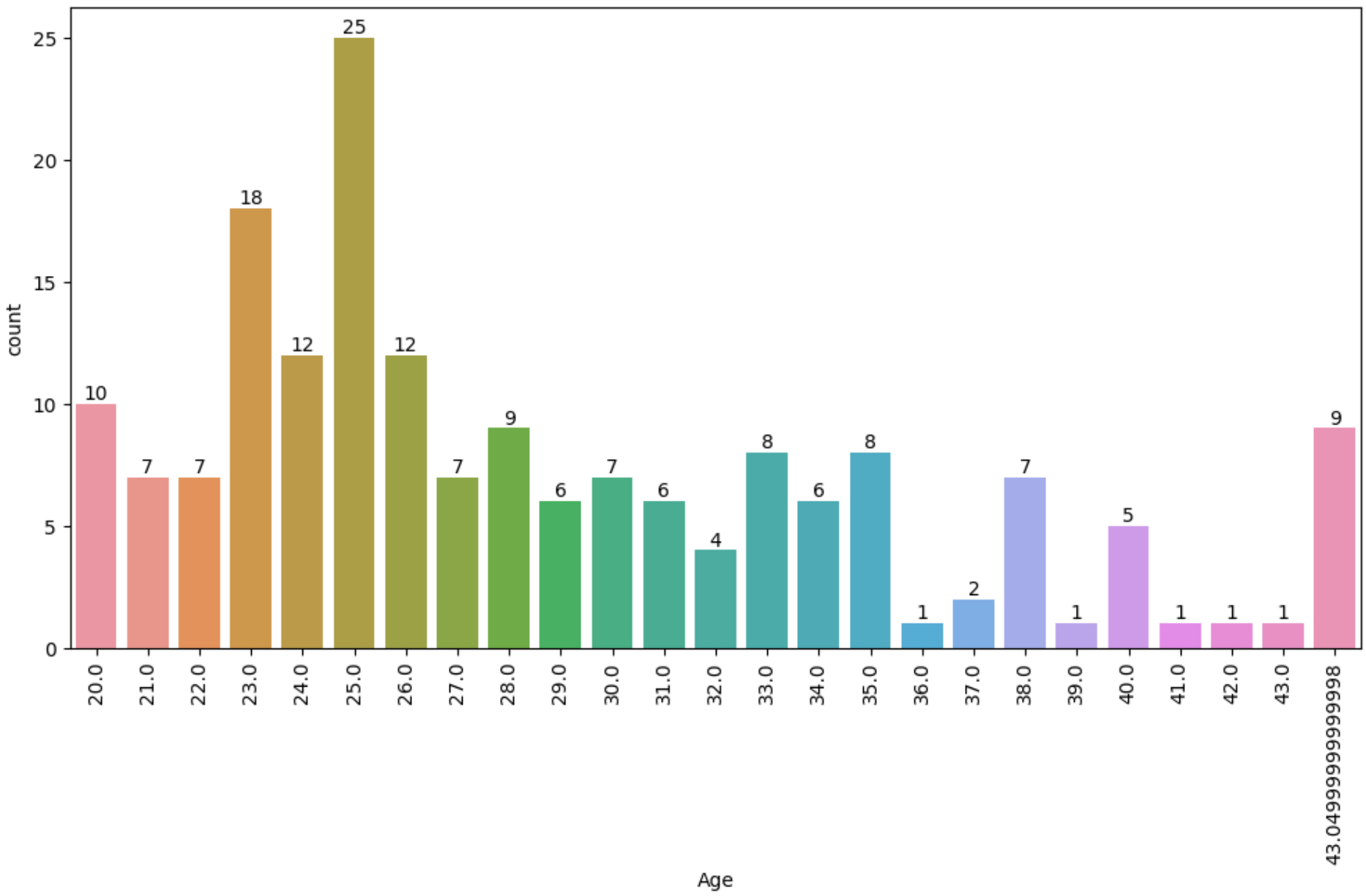



```
ax = sns.countplot(df_new, x="Fitness", hue='Usage')
for container in ax.containers:
    ax.bar_label(container)
plt.show()
```



Continuous Variables

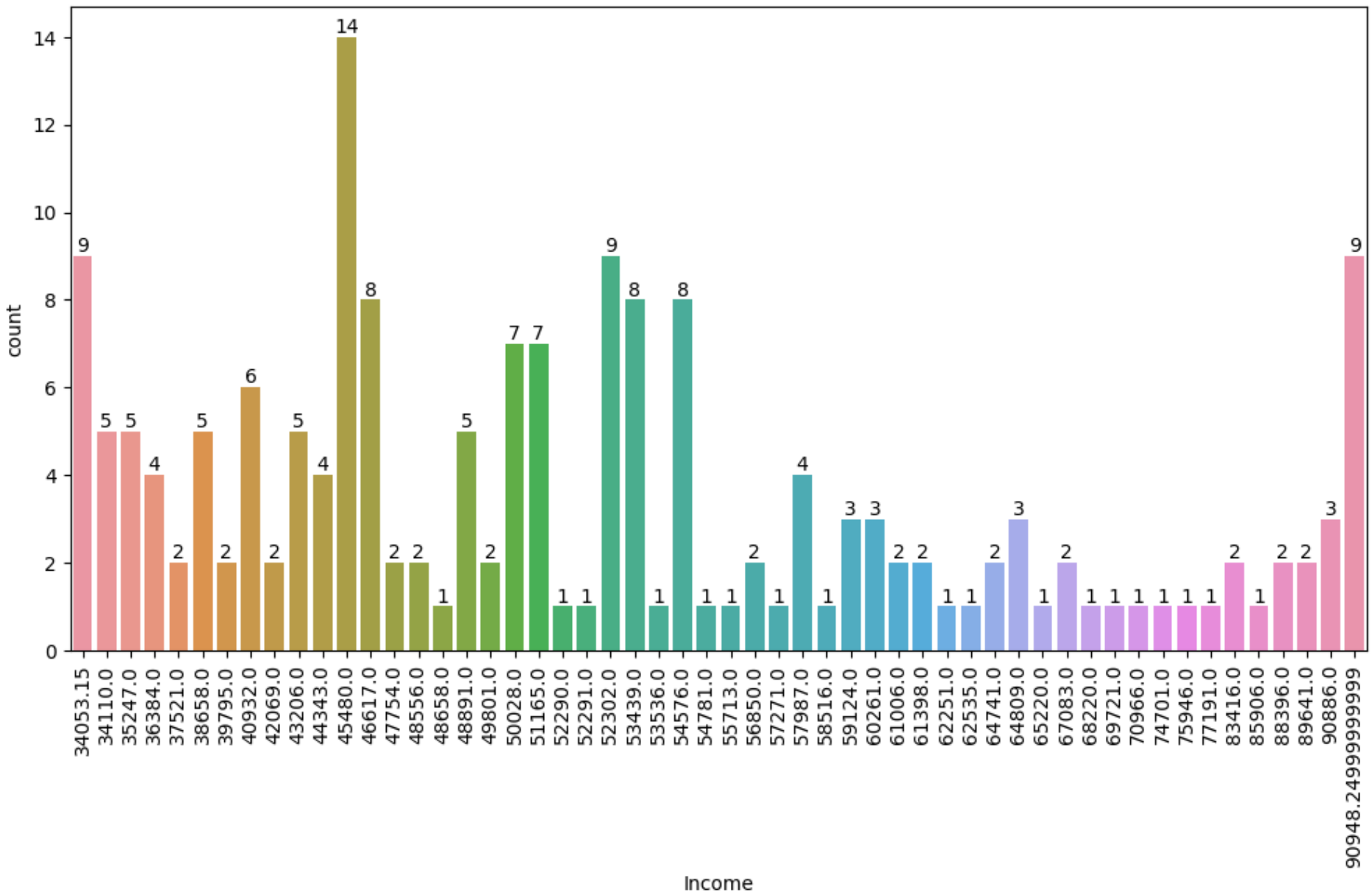
```
plt.figure(figsize=(12,6))
ax = sns.countplot(df_new, x="Age")
ax.bar_label(ax.containers[0])
plt.setp(ax.get_xticklabels(), rotation=90)
plt.show()
```



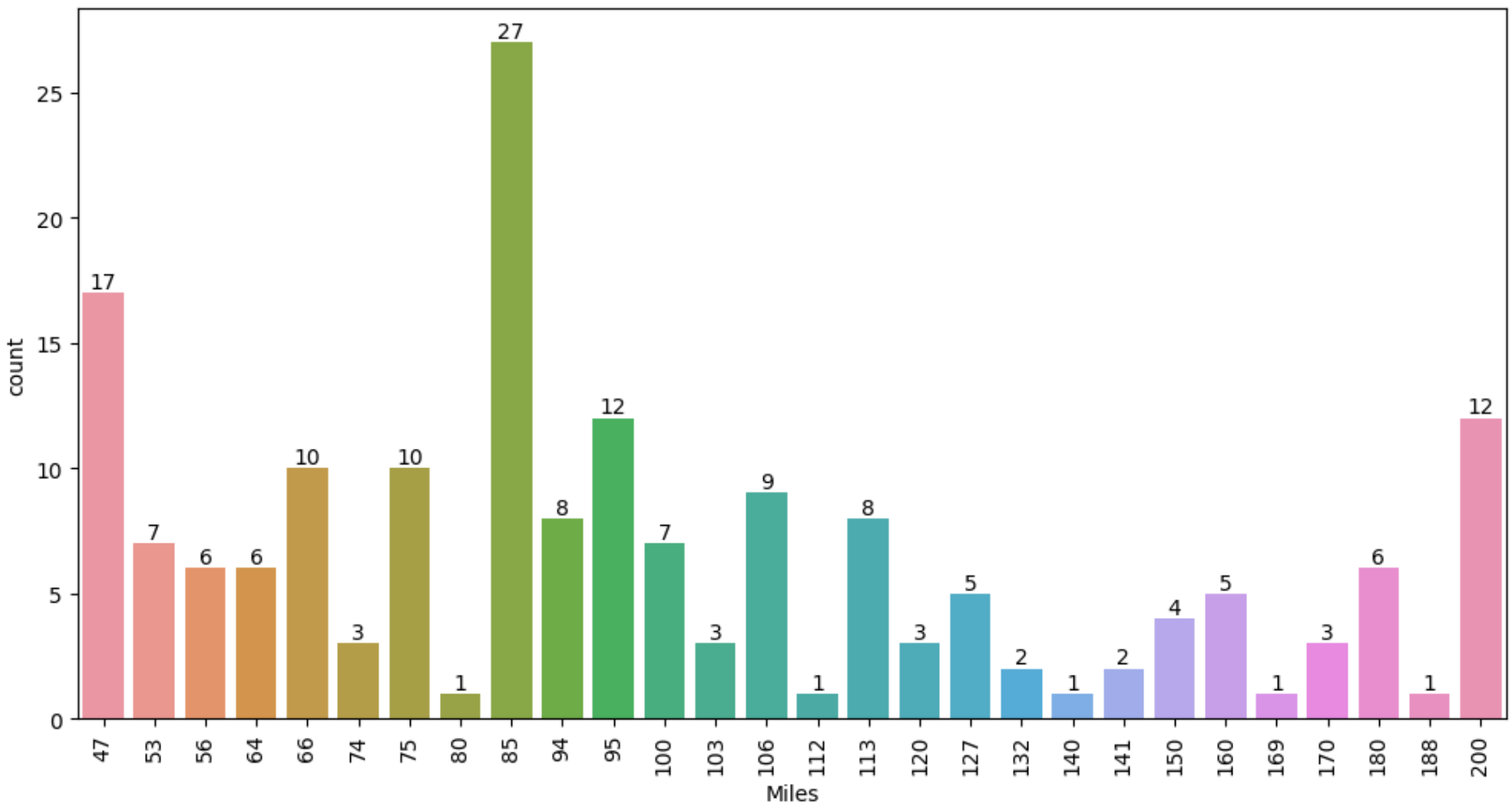
Highest

- 25 person of age 25 years.
- 18 person of age 23 years.
- 12 person each of age 24 and 26 years.
- As the Age increase no. of person decreases

```
plt.figure(figsize=(12,6))
ax = sns.countplot(df_new, x="Income")
ax.bar_label(ax.containers[0])
plt.setp(ax.get_xticklabels(), rotation=90)
plt.show()
```



```
plt.figure(figsize=(12,6))
ax = sns.countplot(df_new, x="Miles")
ax.bar_label(ax.containers[0])
plt.setp(ax.get_xticklabels(), rotation=90)
plt.show()
```



Insights

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

Recommendations

- We want you to use the count plot to find the relationship between categorical variables and output variables.
- We can use hue="-----".

Assumptions

- There is 104 Male and 76 Female.
- Partnereds(107) are more then Singles(73).
- There is 85 persons with 16 years if education.
- There is 63 persons with 14 years if education.
- There is 27 persons with 18 years if education.
- There is 5 persons with 15 years if education.

1 is poor and 5 is Excellent

- There is 97 persons with 3 Fitness Score.
- There is 31 persons with 5 Fitness Score.
- There is 28 persons with 2 Fitness Score.
- There is 24 persons with 4 Fitness Score.

There is 3 type of products

----------(Count)

1. KP281 (80)
2. KP481 (60)
3. KP781 (40)

- There is 69 persons with 3 average number of times :plans to use the treadmill each week
- At most 61 persons who are male and are partnered are using Aerofit product.
- At most 50 persons who are male and having 16 years od Education are using Aerofit product.
- At most 35 persons who are female and having 16 years od Education are using Aerofit product.
- At most 52 persons who are male and having 3 fitness score are using Aerofit product.
- At most 45 persons who are female and having 3 fitness score are using Aerofit product.
- Males are at most using product 4 times per week.(38)
- Females are at most using product 3 times per week.(33)
- Partnerd Are more Education Years.
- Partnered Have more 3 fitness score.
- Partnered have more Usage time per week.
- Product KP281 have more 3 time usage per week.
- People with 16 years of edcutaion have more 3 fitness score.

Highest

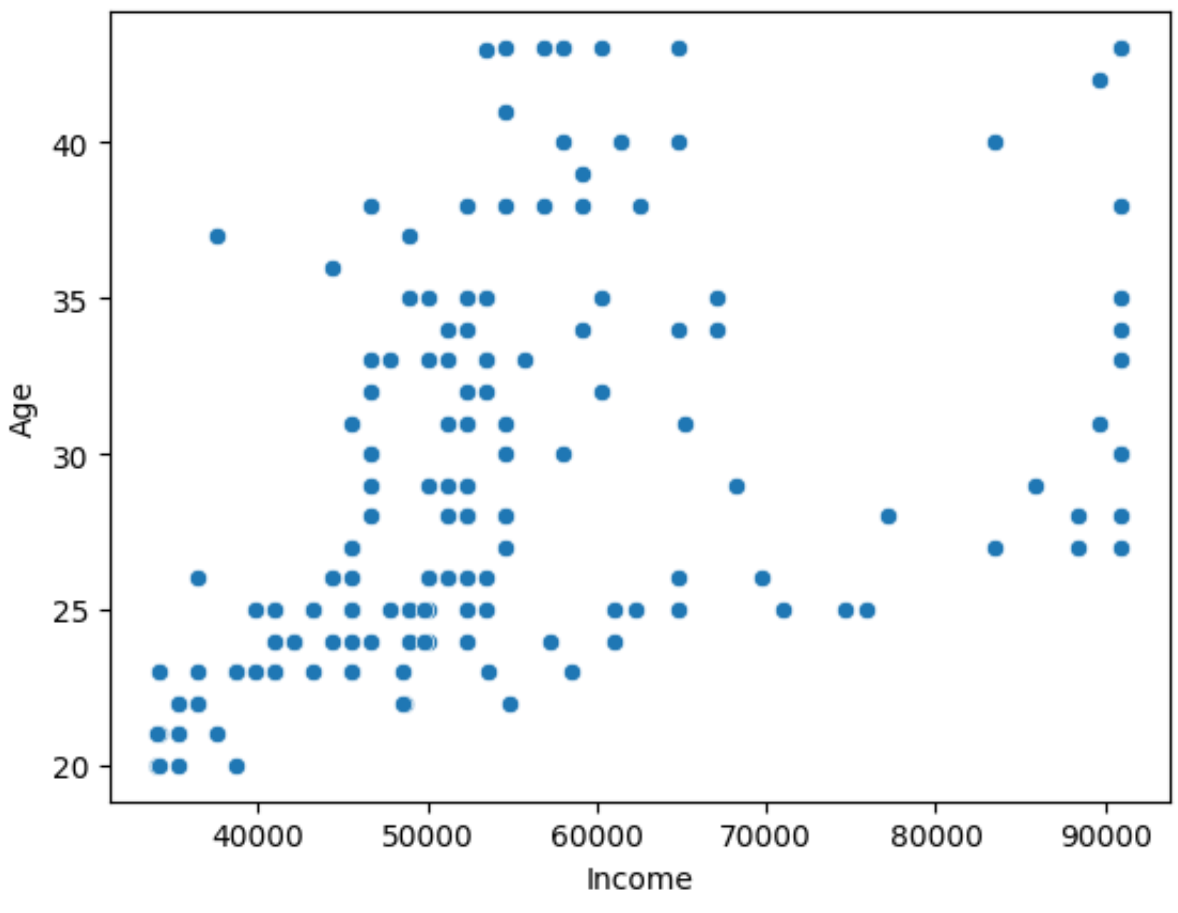
- 25 person of age 25 years.
- 18 person of age 23 years.
- 12 person each of age 24 and 26 years.
- As the Age increase no. of person decreases

Find if there is any relationship between the continuous variables and the output variable in the data.

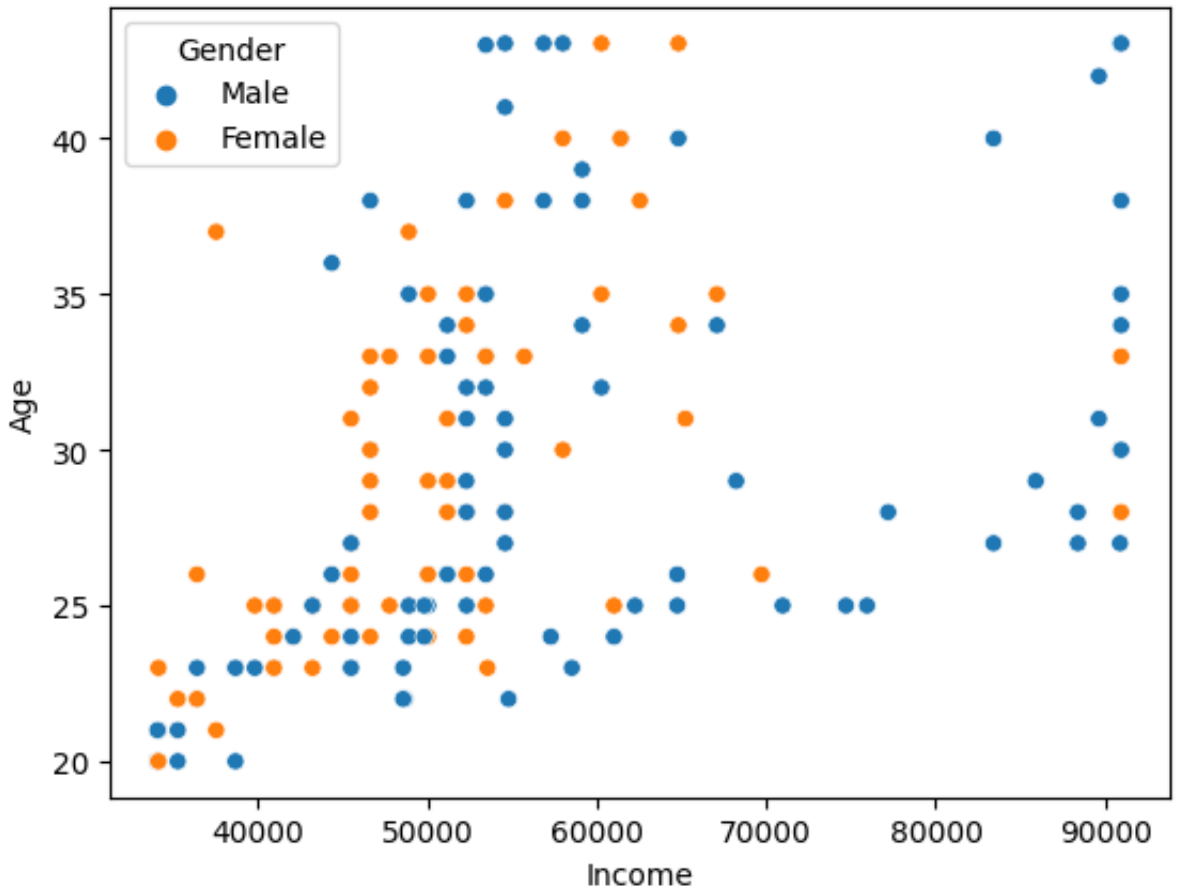
```
df_new.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.0	Male	14	Single	3.0	4	34053.15	112
1	KP281	20.0	Male	15	Single	2.0	3	34053.15	75
2	KP281	20.0	Female	14	Partnered	4.0	3	34053.15	66
3	KP281	20.0	Male	14	Single	3.0	3	34053.15	85
4	KP281	20.0	Male	14	Partnered	4.0	2	35247.00	47

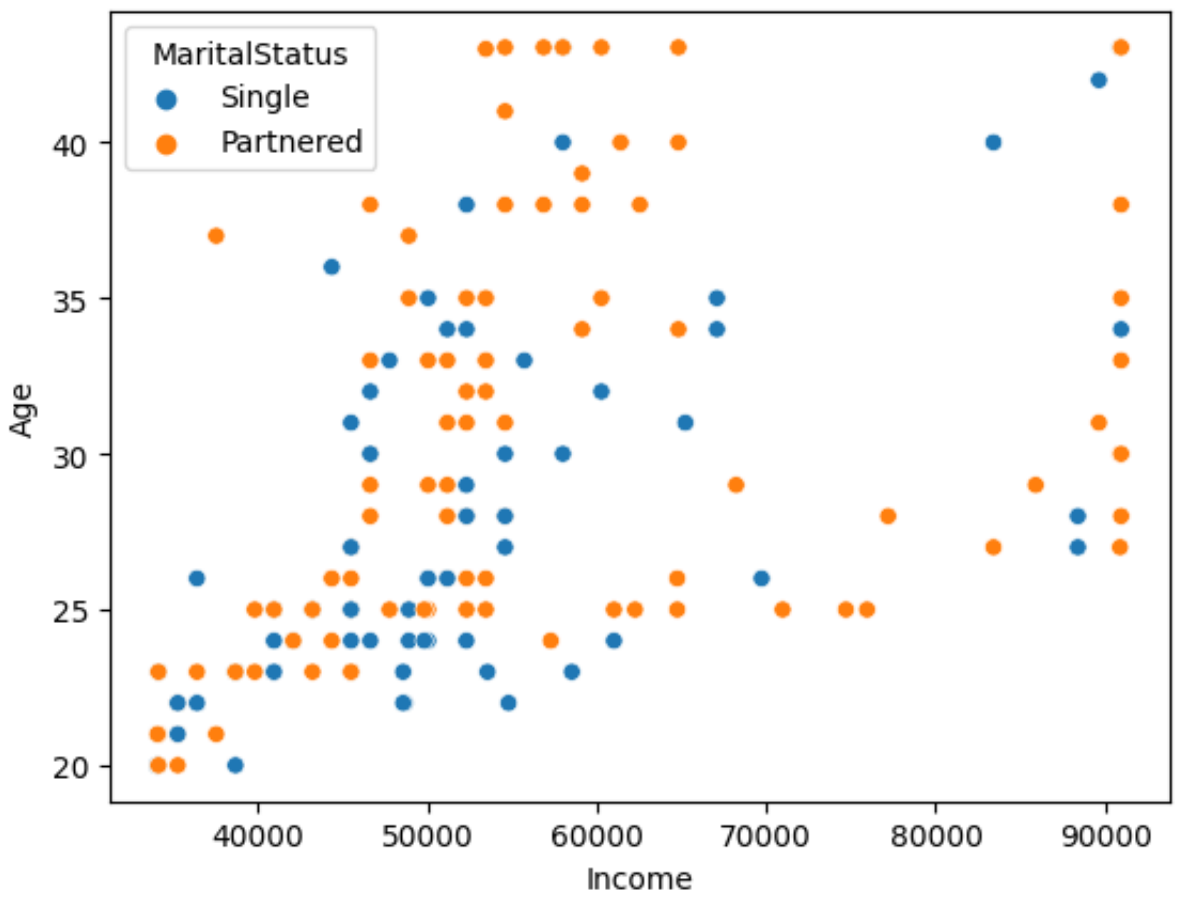
```
sns.scatterplot(data=df_new, y="Age", x="Income")
plt.show()
```



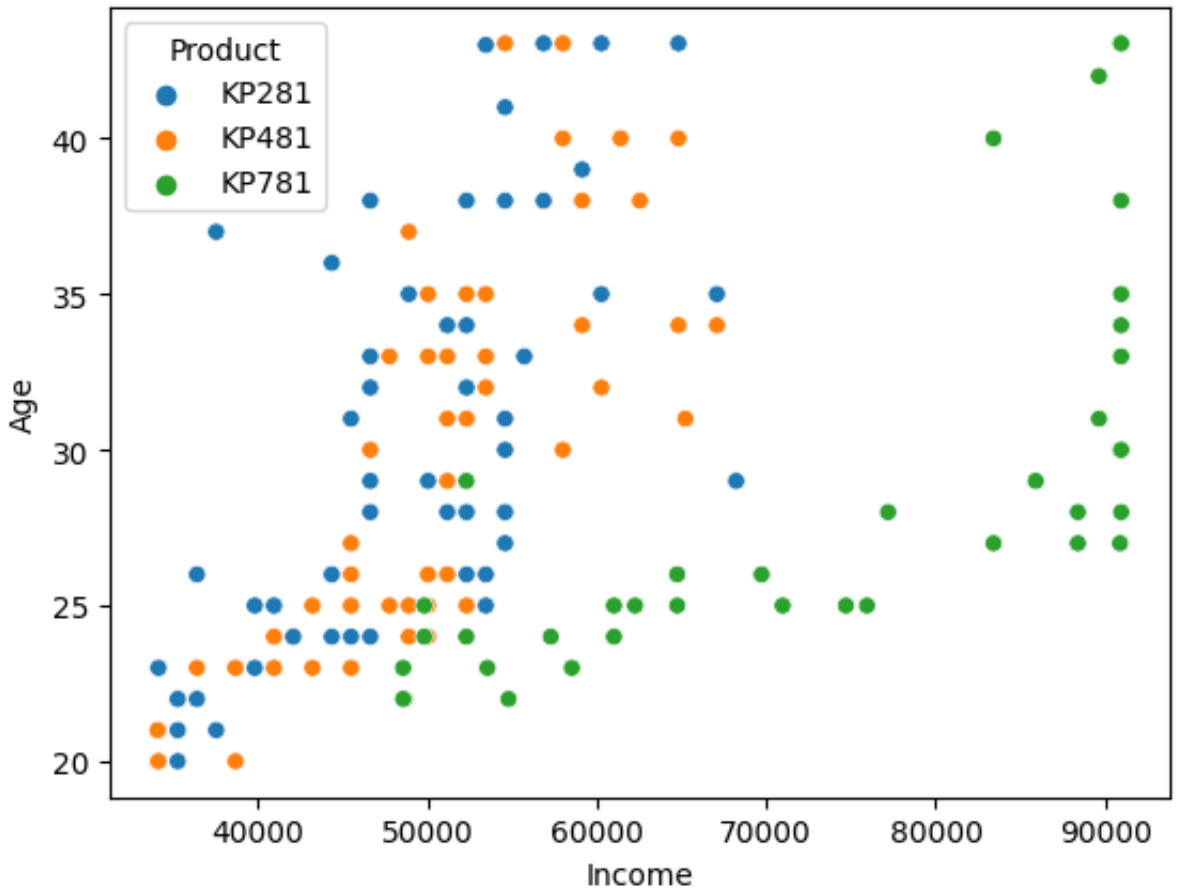
```
sns.scatterplot(data=df_new, y="Age", x="Income", hue='Gender')
plt.show()
```




```
sns.scatterplot(data=df_new, y="Age", x="Income", hue='MaritalStatus')
plt.show()
```

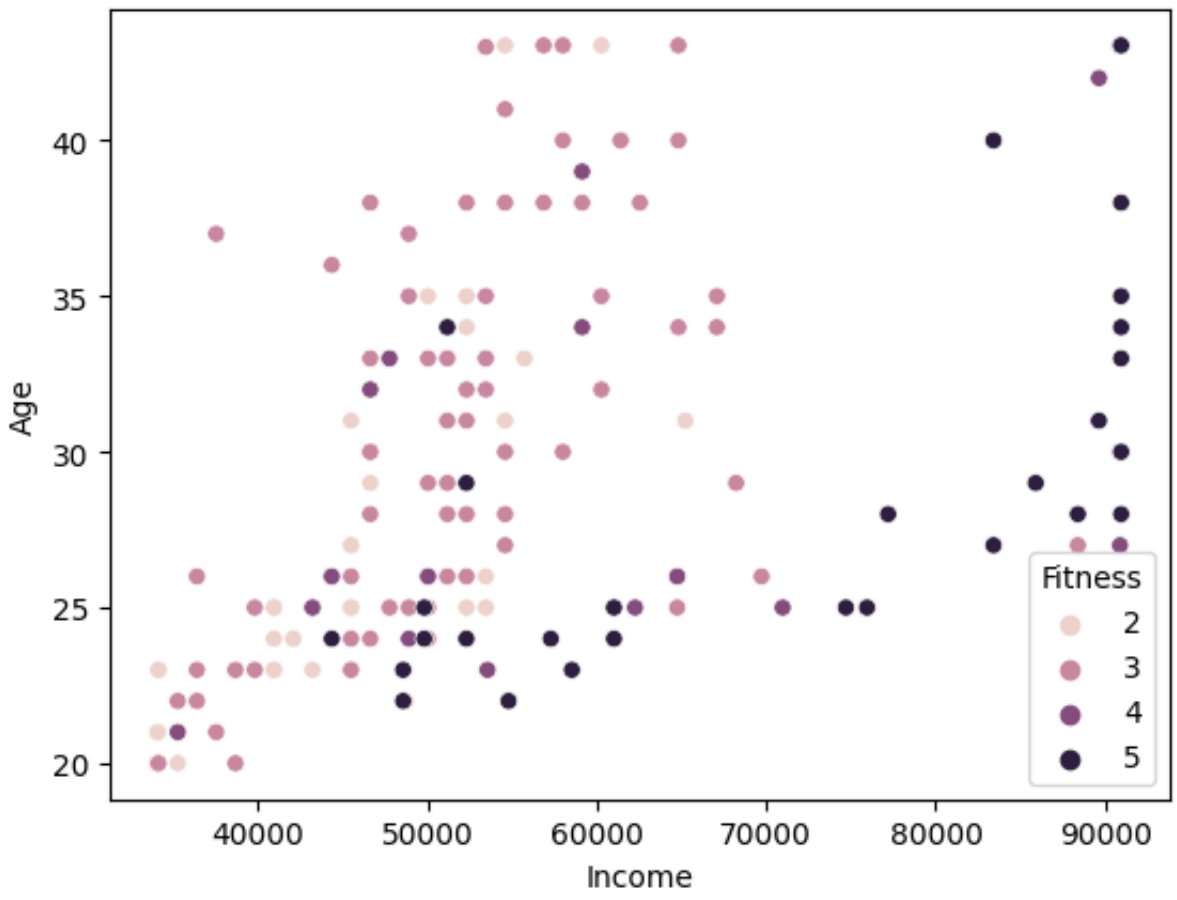


```
sns.scatterplot(data=df_new, y="Age", x="Income", hue='Product')
plt.show()
```



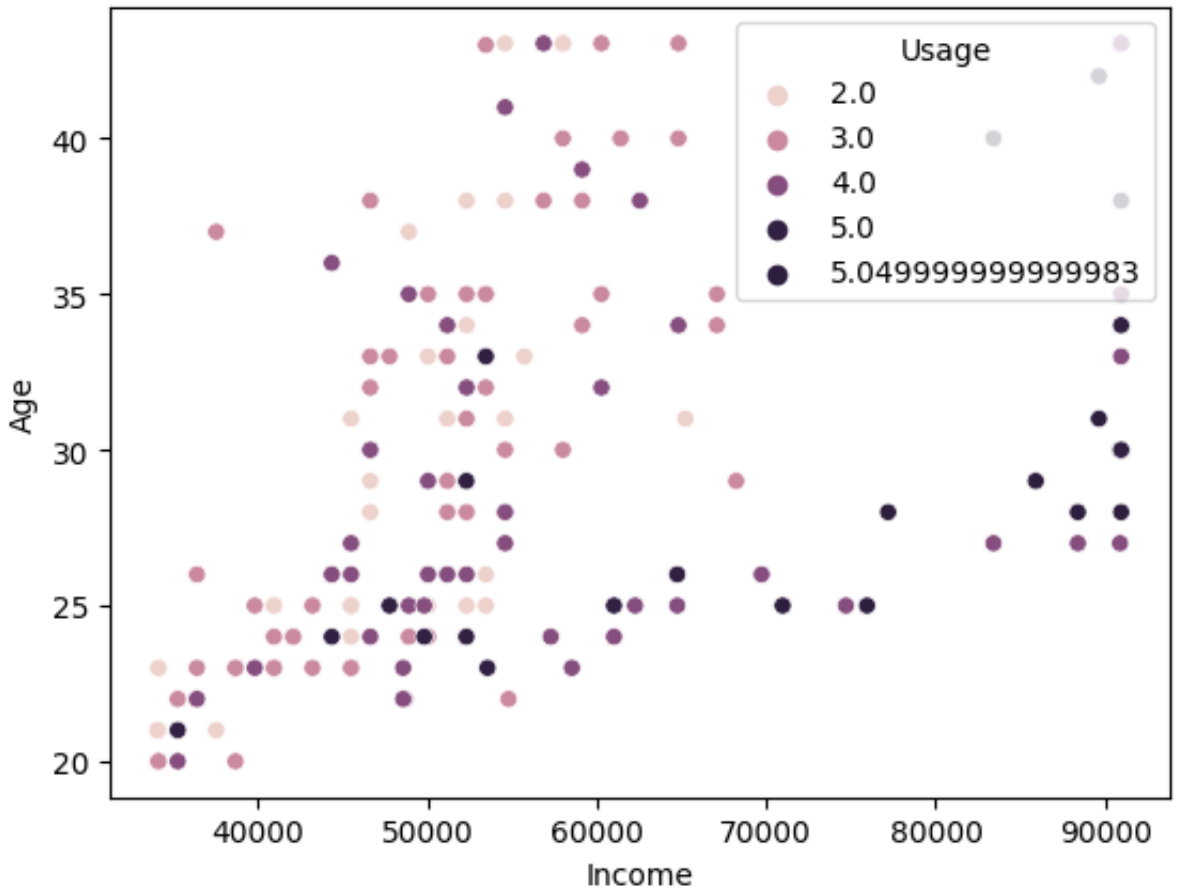
- High paid Income persons are only who buy expensive product(KP78). The KP781 treadmill has advanced features that sell for $2,500.

```
sns.scatterplot(data=df_new, y="Age", x="Income", hue='Fitness')
plt.show()
```

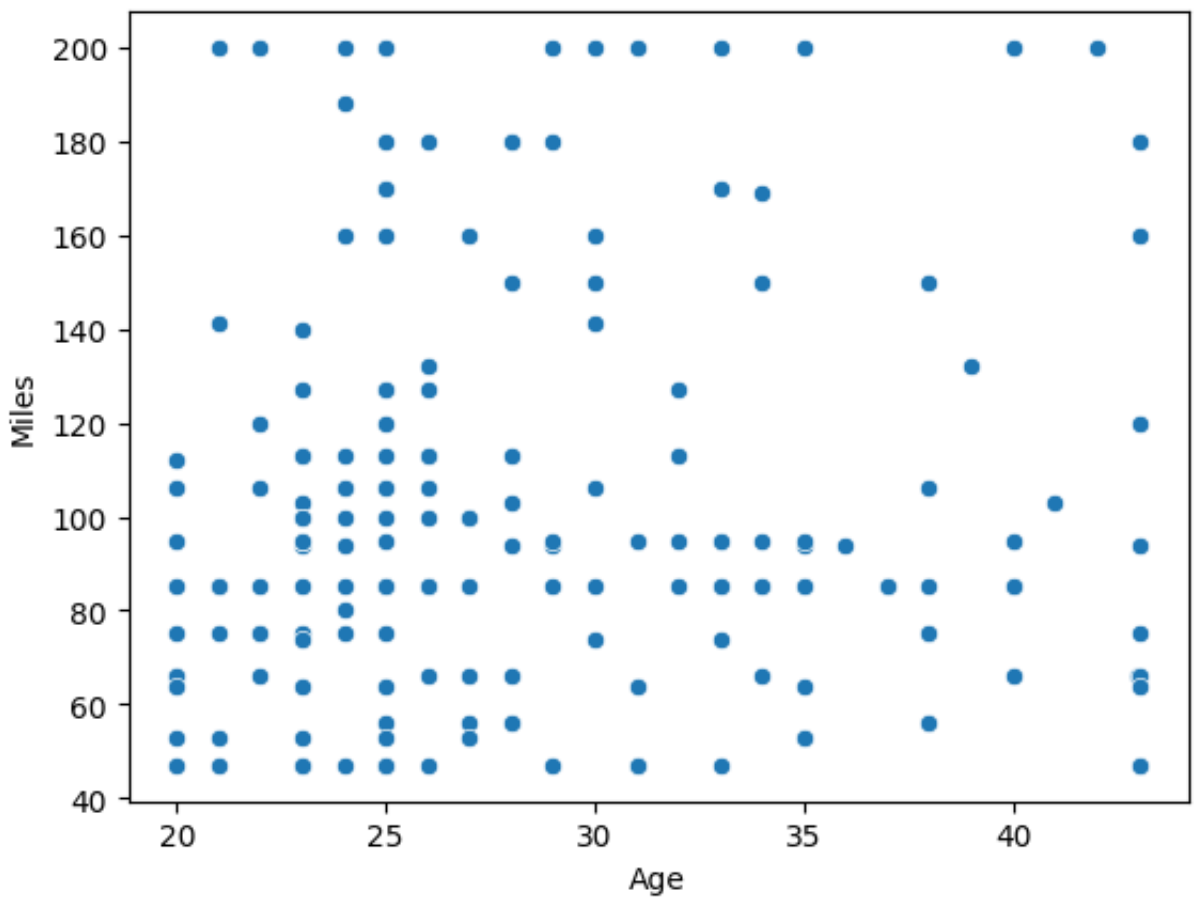


- Mostly high Income peoples have fitness score 5.

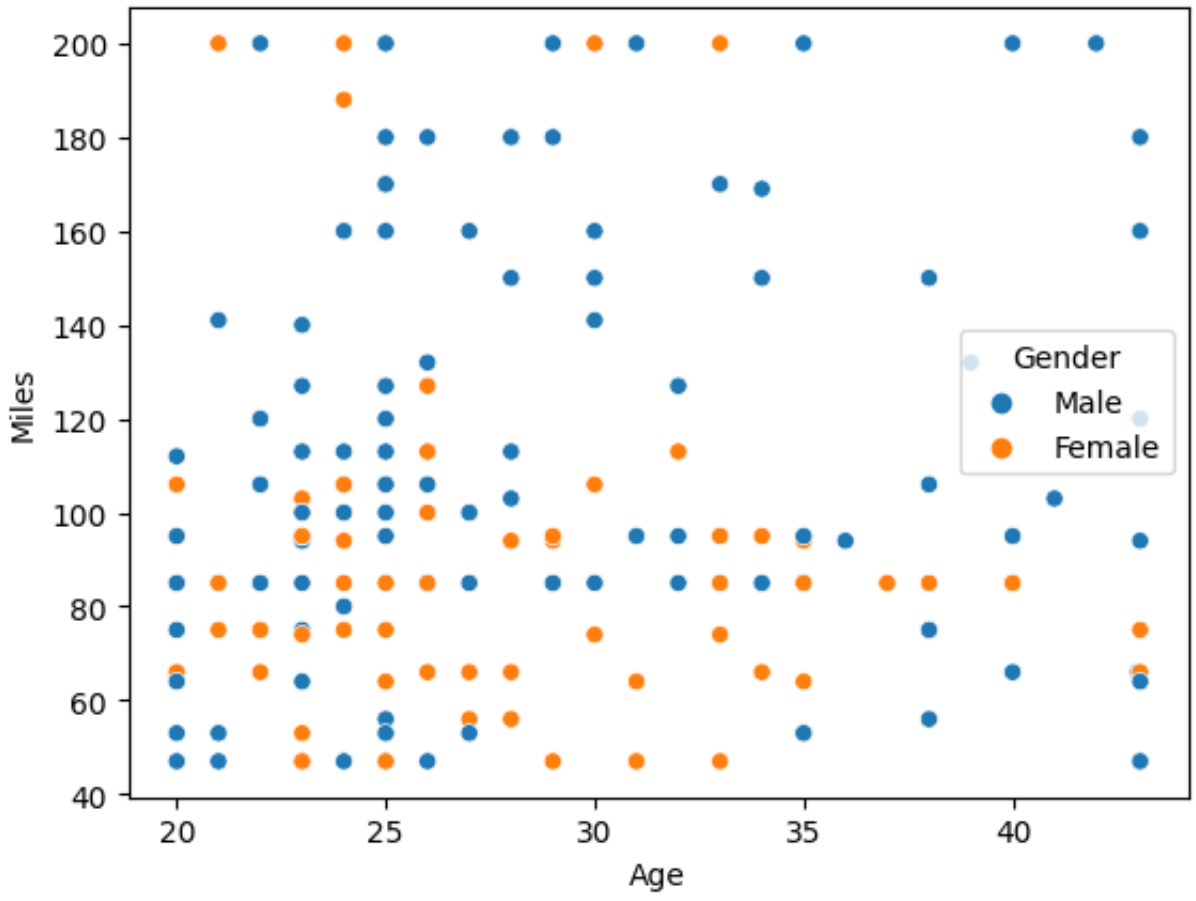
```
sns.scatterplot(data=df_new, y="Age", x="Income", hue='Usage')
plt.show()
```



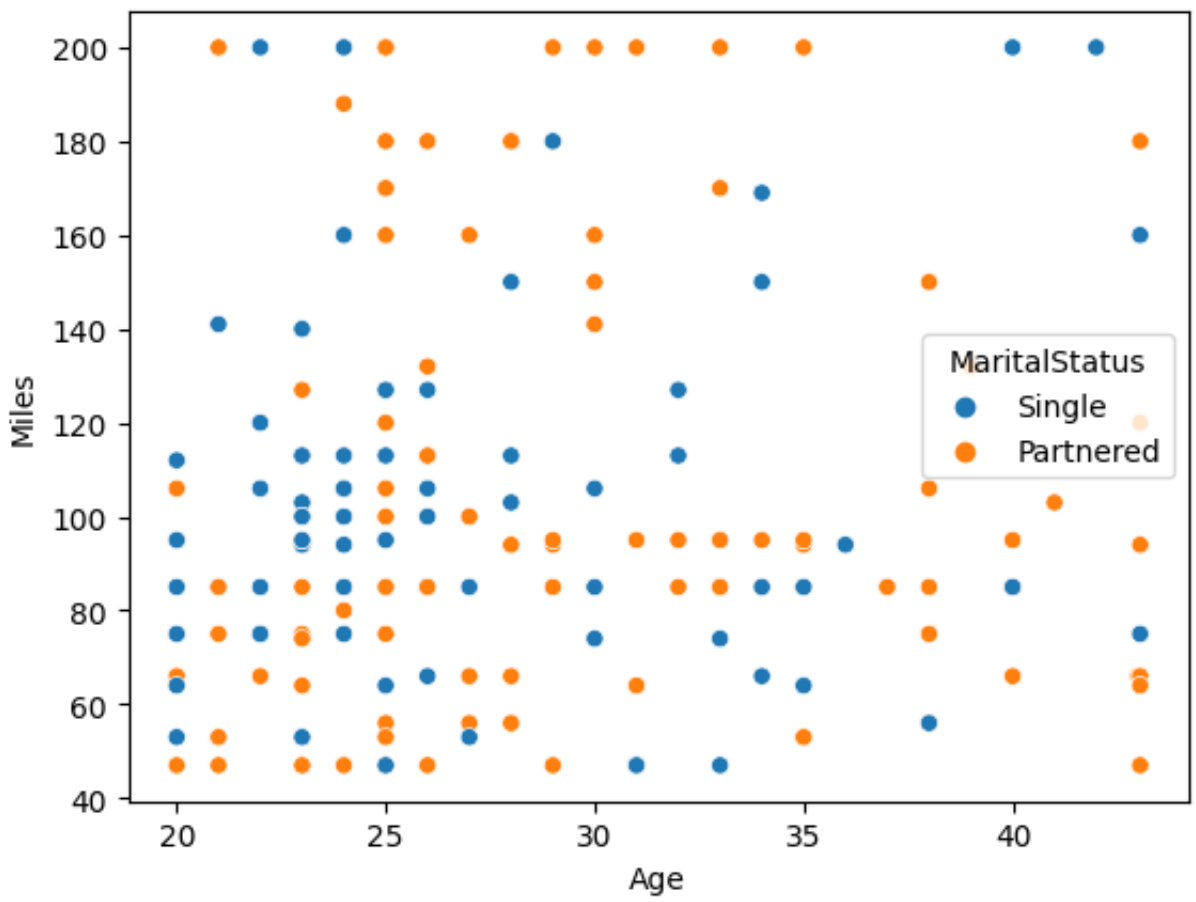

```
sns.scatterplot(data=df_new, x="Age", y="Miles")
plt.show()
```



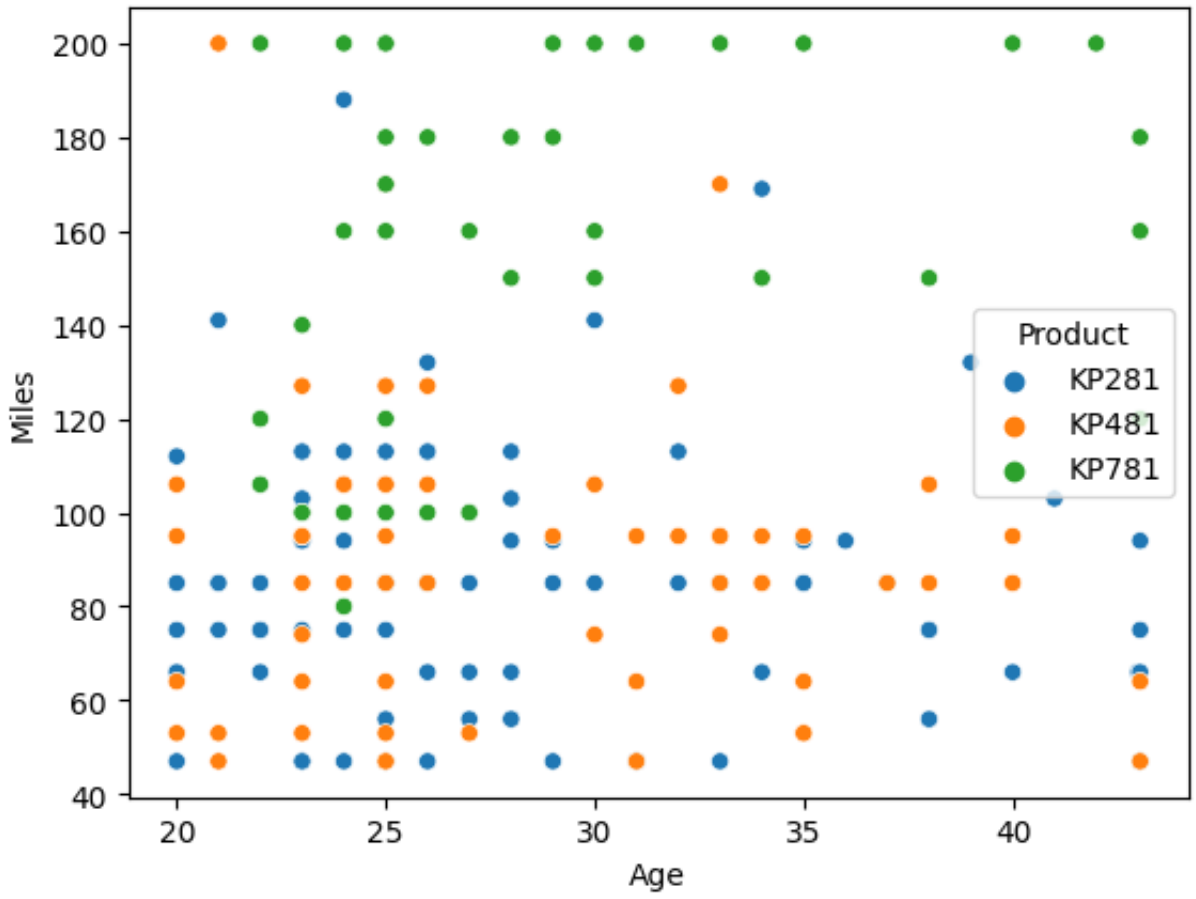
```
sns.scatterplot(data=df_new, x="Age", y="Miles", hue='Gender')
plt.show()
```




```
sns.scatterplot(data=df_new, x="Age", y="Miles", hue='MaritalStatus')
plt.show()
```

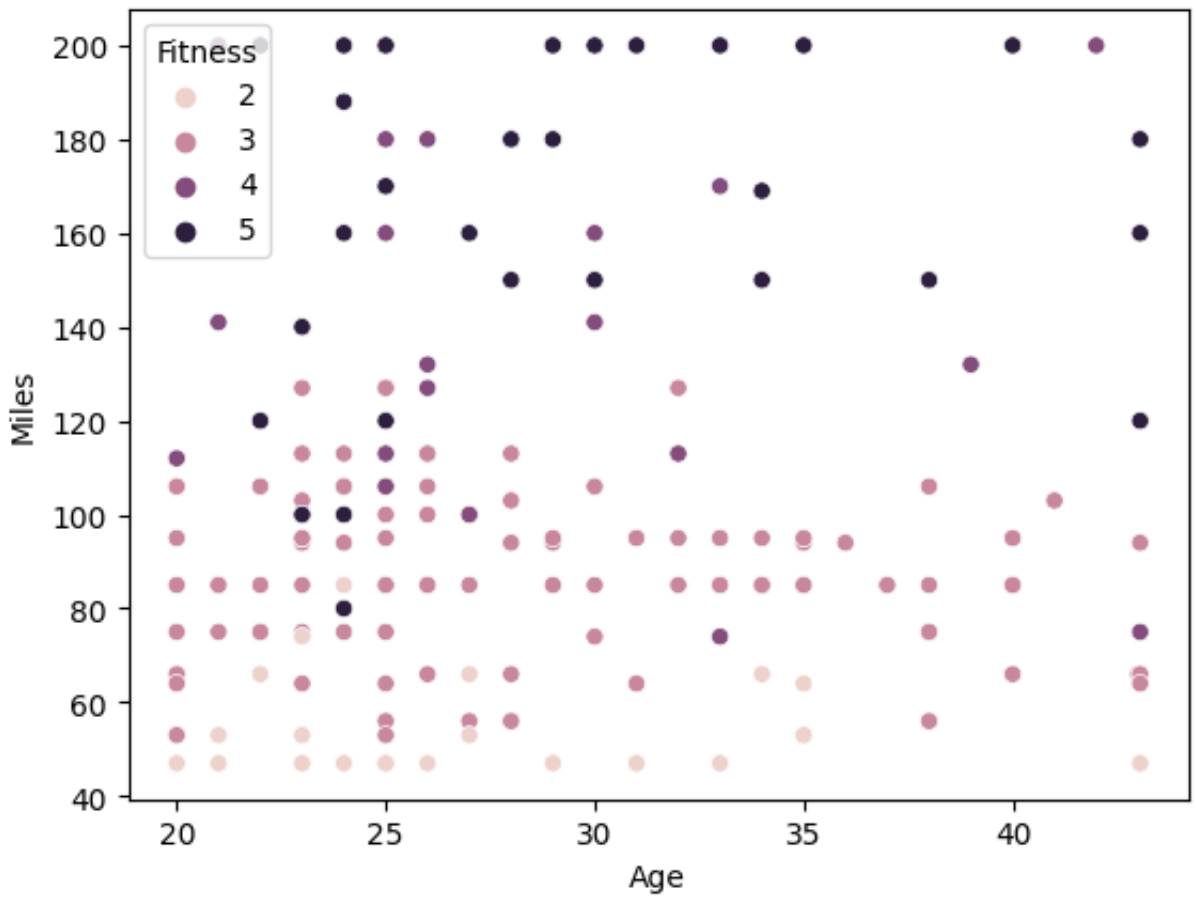


```
sns.scatterplot(data=df_new, x="Age", y="Miles", hue='Product')
plt.show()
```



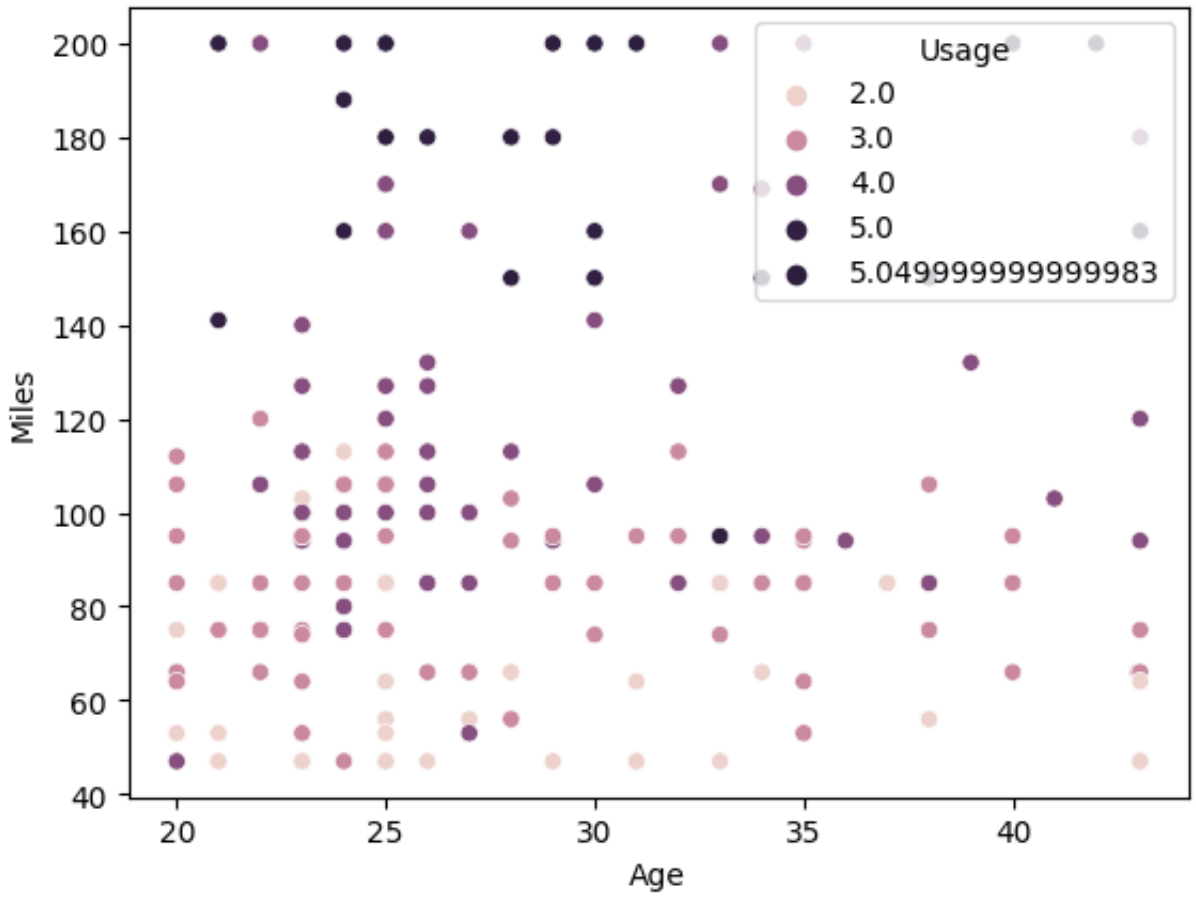
- Most expensive product KP781 have high miles Users.
- Cheap product KP281 is scattered Among all users.

```
sns.scatterplot(data=df_new, x="Age", y="Miles", hue='Fitness')
plt.show()
```



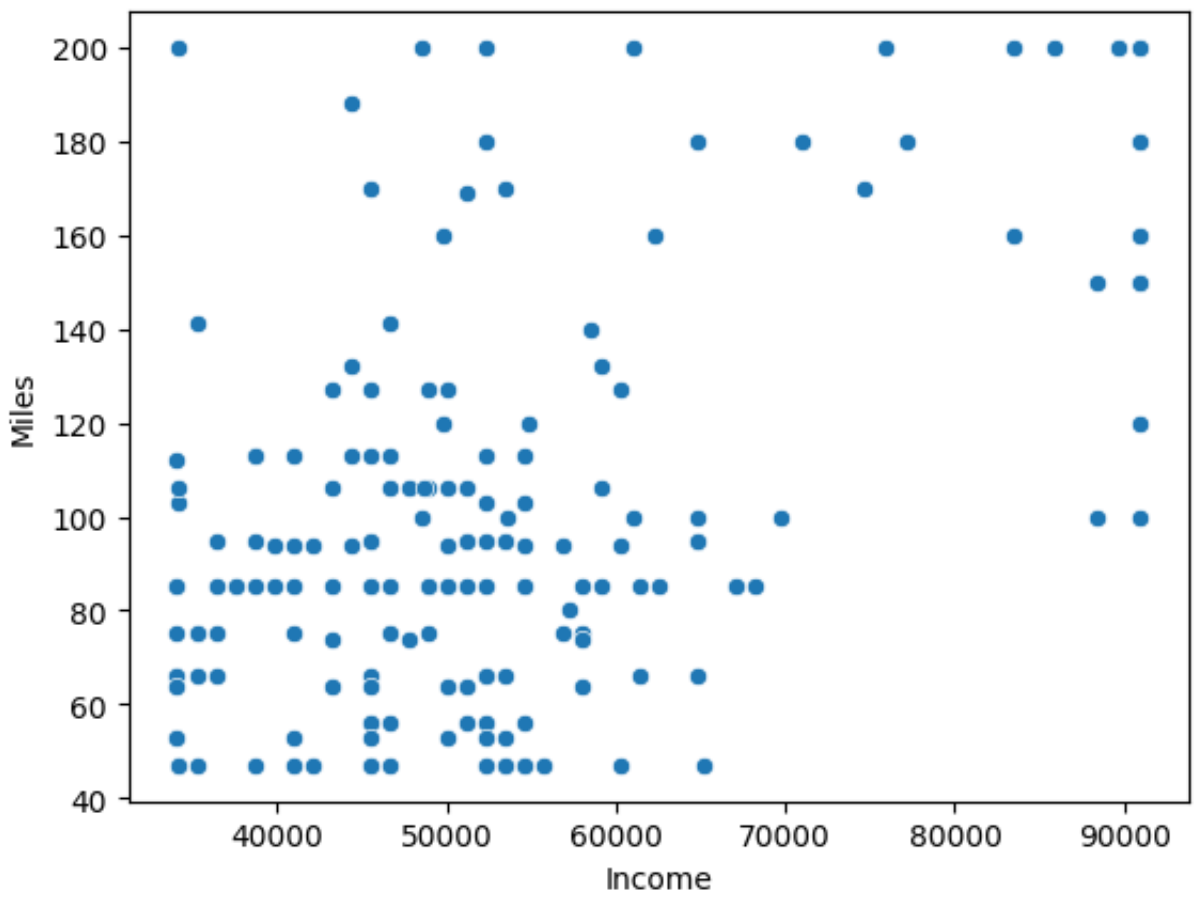
- Fitness Scorer 5 is mostly long miles runners.
- Fitness Scorer 2 is short miles runner.
- As the Fitness score is increasing 2 to 5, miles are also increasing.

```
sns.scatterplot(data=df_new, x="Age", y="Miles", hue='Usage')
plt.show()
```



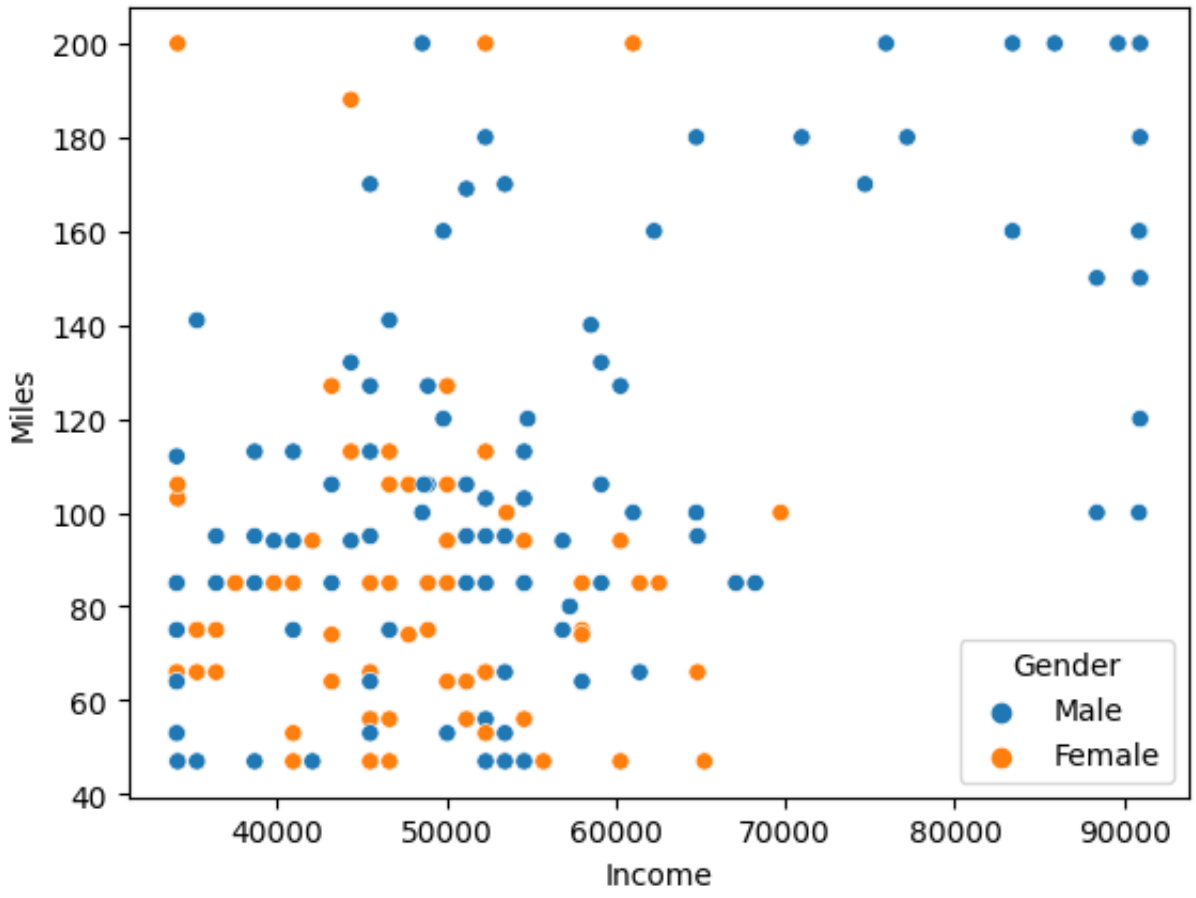
- Usage more then and equal to 5 time per week is mostly long miles runners.
- Usage 2 time per week is short miles runner.
- As the Usage per week is increasing 2 to 5, miles are also increasing.

```
sns.scatterplot(data=df_new, x="Income", y="Miles")
plt.show()
```



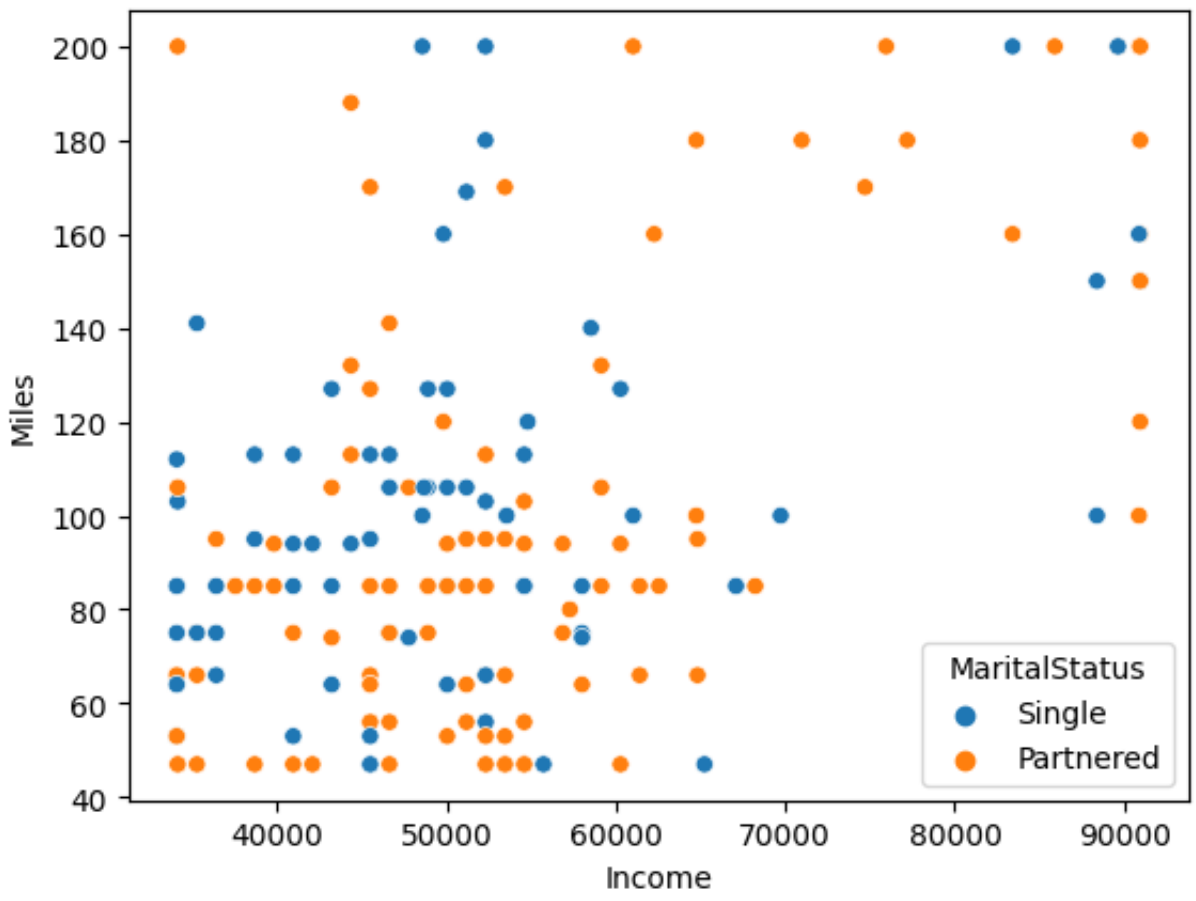
- Data is tigtly packed between 0-140 miles and 0-70000 Income.
- Some Data is scatter at high Income and high miles.

```
sns.scatterplot(data=df_new, x="Income", y="Miles", hue='Gender')
plt.show()
```



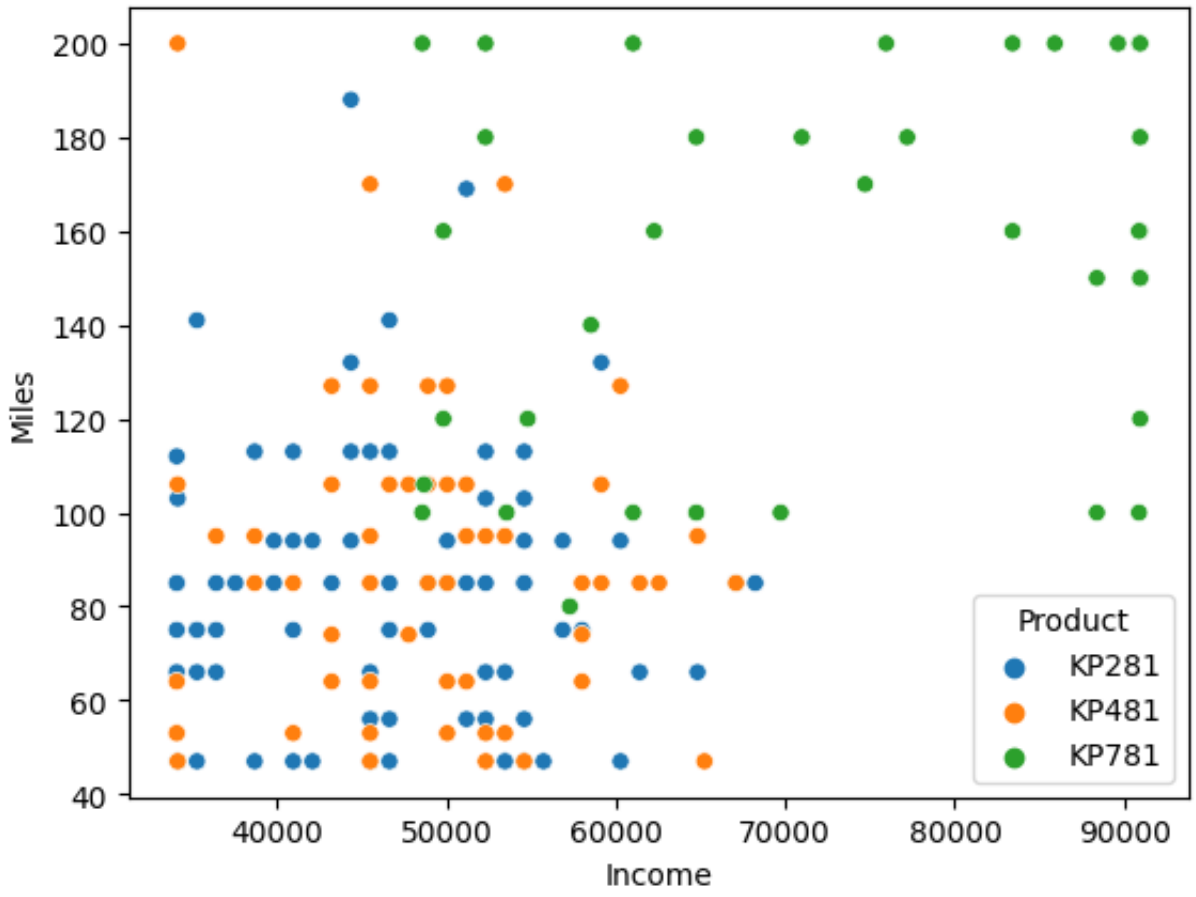
- Less No. of Females are high Mile runner. Mostly Males are high mile runner.

```
sns.scatterplot(data=df_new, x="Income", y="Miles", hue='MaritalStatus')
plt.show()
```



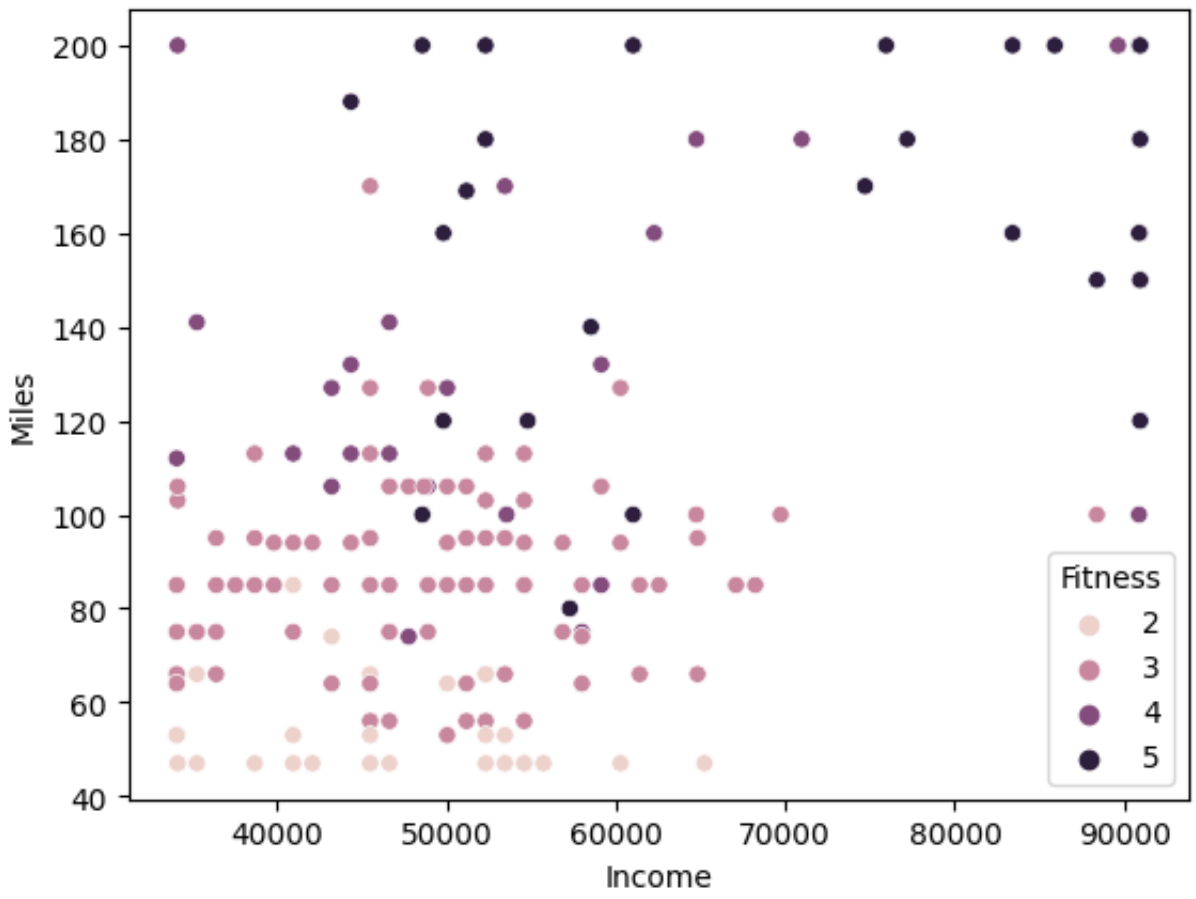
- Partnered People are well scattered then singles.
- Almost pantnered are double in population of high miles runner, then singles.

```
sns.scatterplot(data=df_new, x="Income", y="Miles", hue='Product')
plt.show()
```



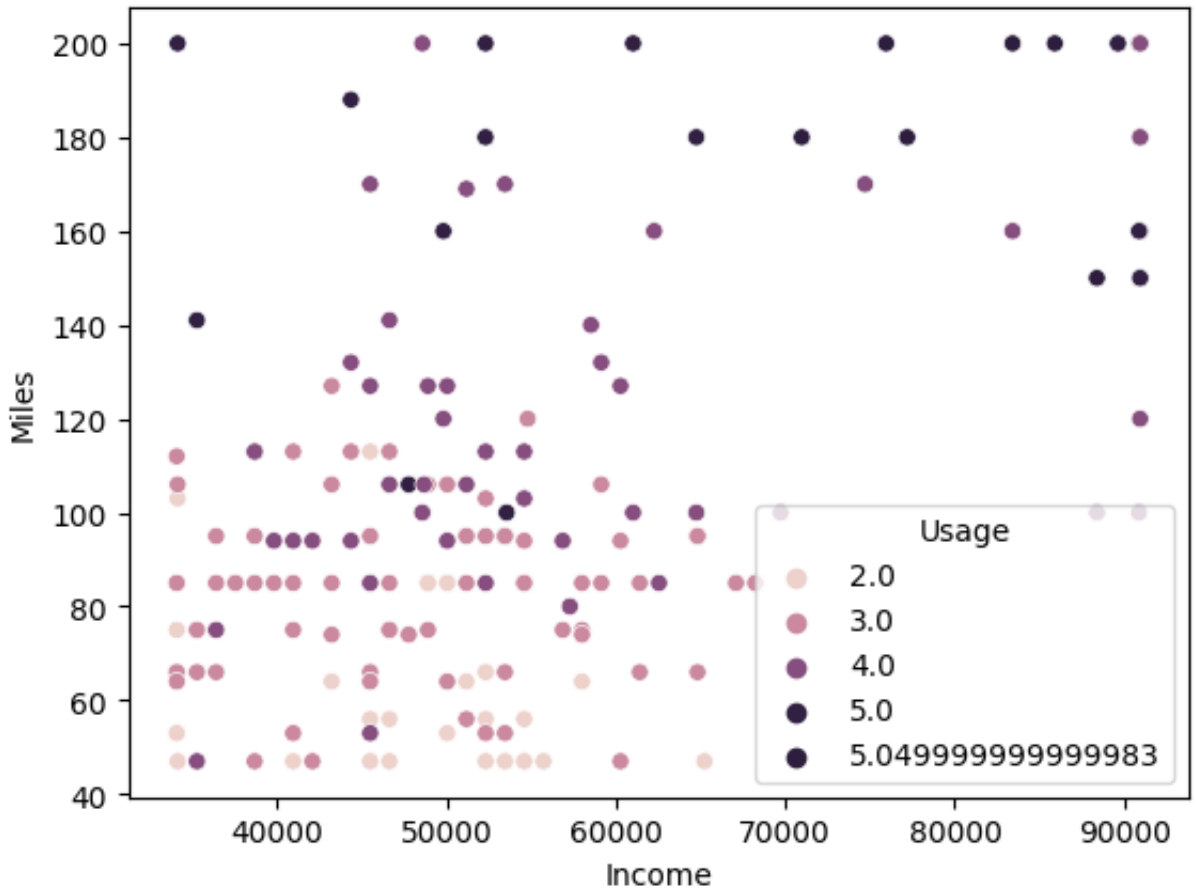
- Expensive product KP781 is mostly purchased by high miles runner and high income persons.
- And other 2 product(KP281 & KP481) is well scatterd among low to mid Miles vs Income.

```
sns.scatterplot(data=df_new, x="Income", y="Miles", hue='Fitness')
plt.show()
```



- Fitness Score 4 and 5 are mostly scored by high miles and high income users.
- Fitness Score 2 and 3 are mostly scored by low to mid miles and Income users.

```
sns.scatterplot(data=df_new, x="Income", y="Miles", hue='Usage')
plt.show()
```



- Usage rate(times per week) 4 and more then 5 are mostly scored by high miles and high income users.
- Usage rate(times per week) 2 and 3 are mostly scored by low to mid miles and Income users.

Insights

- Find if there is any relationship between the continuous variables and the output variable in the data.

Recommendations

- We want to use a scatter plot to find the relationship between continuous variables and output variables.
- We can use hue="------".

Assumptions

- High paid Income persons are only who buy expensive product(KP78). The KP781 treadmill has advanced features that sell for $2,500.
- Mostly high Income peoples have fitness score 5.
- Most expensive product KP781 have high miles Users.
- Cheap product KP281 is scattered Among all users.
- Fitness Scorer 5 is mostly long miles runners.
- Fitness Scorer 2 is short miles runner.
- As the Fitness score is increasing 2 to 5, miles are also increasing.
- Usage more then and equal to 5 time per week is mostly long miles runners.
- Usage 2 time per week is short miles runner.
- As the Usage per week is increasing 2 to 5, miles are also increasing.
- Data is tigtly packed between 0-140 miles and 0-70000 Income.
- Some Data is scatter at high Income and high miles.
- Less No. of Females are high Mile runner. Mostly Males are high mile runner.
- Partnered People are well scattered then singles.
- Almost pantnered are double in population of high miles runner, then singles.
- Expensive product KP781 is mostly purchased by high miles runner and high income persons.
- And other 2 product(KP281 & KP481) is well scatterd among low to mid Miles vs Income.
- Fitness Score 4 and 5 are mostly scored by high miles and high income users.
- Fitness Score 2 and 3 are mostly scored by low to mid miles and Income users.
- Usage rate(times per week) 4 and more then 5 are mostly scored by high miles and high income users.
- Usage rate(times per week) 2 and 3 are mostly scored by low to mid miles and Income users.

4. Representing the Probability

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

```
df_new.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.0	Male	14	Single	3.0	4	34053.15	112
1	KP281	20.0	Male	15	Single	2.0	3	34053.15	75
2	KP281	20.0	Female	14	Partnered	4.0	3	34053.15	66
3	KP281	20.0	Male	14	Single	3.0	3	34053.15	85
4	KP281	20.0	Male	14	Partnered	4.0	2	35247.00	47

```
pd.crosstab(df_new['Product'],df_new['Product'], margins=True, margins_name='Total', normalize=True)
```

Product	KP281	KP481	KP781	Total
Product				
KP281	0.444444	0.000000	0.000000	0.444444
KP481	0.000000	0.333333	0.000000	0.333333
KP781	0.000000	0.000000	0.222222	0.222222
Total	0.444444	0.333333	0.222222	1.000000

Insights

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

Recommendations

- We want to use the pandas crosstab to find the marginal probability of each product.

Assumptions

- KP281- 80 Quntity - 44.44%
- KP481- 60 Quntity - 33.33%
- Kp781- 40 Quntity - 22.22%

- **Find the conditional probability that an event occurs given that another event has occurred. (Example: given that a customer is female, what is the probability she'll purchase a KP481)**

```
pd.crosstab(df_new['Product'],df_new['Gender'],margins=True, margins_name='Total', normalize=True)
```

Gender	Female	Male	Total
Product			
KP281	0.222222	0.222222	0.444444
KP481	0.161111	0.172222	0.333333
KP781	0.038889	0.183333	0.222222
Total	0.422222	0.577778	1.000000

- 42.22% Are Female.
- 57.78% Are Male.
- Mostly Female 22.22% have purchase KP281(Cheapest).
- Mostly Male 22.22% have purchase KP281(Cheapest).

```
pd.crosstab(df_new['Product'],df_new['MaritalStatus'], margins=True, margins_name='Total', normalize=True)
```

MaritalStatus	Partnered	Single	Total
Product			
KP281	0.266667	0.177778	0.444444
KP481	0.200000	0.133333	0.333333
KP781	0.127778	0.094444	0.222222
Total	0.594444	0.405556	1.000000

- There is 59.44% are Partnered and 40.56% are Single
- Mostly 26.67% Partnered have KP281.
- 20% Partnered have KP481.
- Mostly 17.78% single have KP281.

```
pd.crosstab(df_new['Product'],df_new['Education'], margins=True, margins_name='Total', normalize=True)
```

Education	14	15	16	18	Total
Product					
KP281	0.194444	0.022222	0.216667	0.011111	0.444444
KP481	0.144444	0.005556	0.172222	0.011111	0.333333
KP781	0.011111	0.000000	0.083333	0.127778	0.222222
Total	0.350000	0.027778	0.472222	0.150000	1.000000

- 47.22% have 16 Year Education.

```
pd.crosstab(df_new['Product'],df_new['Fitness'], margins=True, margins_name='Total', normalize=True)
```

Fitness	2	3	4	5	Total
Product					
KP281	0.083333	0.300000	0.050000	0.011111	0.444444
KP481	0.072222	0.216667	0.044444	0.000000	0.333333
KP781	0.000000	0.022222	0.038889	0.161111	0.222222
Total	0.155556	0.538889	0.133333	0.172222	1.000000

- 53.89% have fitness score 3.

```
pd.crosstab(df_new['Product'],df_new['Usage'],margins=True, margins_name='Total', normalize=True)
```

Usage	2.0	3.0	4.0	5.0	5.049999999999983	Total
Product						
KP281	0.105556	0.205556	0.122222	0.011111	0.00	0.444444
KP481	0.077778	0.172222	0.066667	0.016667	0.00	0.333333
KP781	0.000000	0.005556	0.100000	0.066667	0.05	0.222222
Total	0.183333	0.383333	0.288889	0.094444	0.05	1.000000

```
pd.crosstab(df_new['Age'],df_new['Product'],margins=True, margins_name='Total', normalize=True)
```

Product	KP281	KP481	KP781	Total
Age				
20.0	0.033333	0.022222	0.000000	0.055556
21.0	0.022222	0.016667	0.000000	0.038889
22.0	0.022222	0.000000	0.016667	0.038889
23.0	0.044444	0.038889	0.016667	0.100000
24.0	0.027778	0.016667	0.022222	0.066667
25.0	0.038889	0.061111	0.038889	0.138889
26.0	0.038889	0.016667	0.011111	0.066667
27.0	0.016667	0.005556	0.016667	0.038889
28.0	0.033333	0.000000	0.016667	0.050000
29.0	0.016667	0.005556	0.011111	0.033333
30.0	0.011111	0.011111	0.016667	0.038889
31.0	0.011111	0.016667	0.005556	0.033333
32.0	0.011111	0.011111	0.000000	0.022222
33.0	0.011111	0.027778	0.005556	0.044444
34.0	0.011111	0.016667	0.005556	0.033333
35.0	0.016667	0.022222	0.005556	0.044444
36.0	0.005556	0.000000	0.000000	0.005556
37.0	0.005556	0.005556	0.000000	0.011111
38.0	0.022222	0.011111	0.005556	0.038889
39.0	0.005556	0.000000	0.000000	0.005556
40.0	0.005556	0.016667	0.005556	0.027778
41.0	0.005556	0.000000	0.000000	0.005556
42.0	0.000000	0.000000	0.005556	0.005556
43.0	0.005556	0.000000	0.000000	0.005556
43.04999999999998	0.022222	0.011111	0.016667	0.050000
Total	0.444444	0.333333	0.222222	1.000000

We can see at what customer age, He/She buying Which product.

```
pd.crosstab(df_new['Miles'],df_new['Product'],margins=True, margins_name='Total', normalize=True)
```

Product	KP281	KP481	KP781	Total
Miles				
47	0.066667	0.027778	0.000000	0.094444
53	0.000000	0.038889	0.000000	0.038889
56	0.033333	0.000000	0.000000	0.033333
64	0.000000	0.033333	0.000000	0.033333
66	0.055556	0.000000	0.000000	0.055556
74	0.000000	0.016667	0.000000	0.016667
75	0.055556	0.000000	0.000000	0.055556
80	0.000000	0.000000	0.005556	0.005556
85	0.088889	0.061111	0.000000	0.150000
94	0.044444	0.000000	0.000000	0.044444
95	0.000000	0.066667	0.000000	0.066667
100	0.000000	0.000000	0.038889	0.038889
103	0.016667	0.000000	0.000000	0.016667
106	0.000000	0.044444	0.005556	0.050000
112	0.005556	0.000000	0.000000	0.005556
113	0.044444	0.000000	0.000000	0.044444
120	0.000000	0.000000	0.016667	0.016667
127	0.000000	0.027778	0.000000	0.027778
132	0.011111	0.000000	0.000000	0.011111
140	0.000000	0.000000	0.005556	0.005556
141	0.011111	0.000000	0.000000	0.011111
150	0.000000	0.000000	0.022222	0.022222
160	0.000000	0.000000	0.027778	0.027778
169	0.005556	0.000000	0.000000	0.005556
170	0.000000	0.011111	0.005556	0.016667
180	0.000000	0.000000	0.033333	0.033333
188	0.005556	0.000000	0.000000	0.005556
200	0.000000	0.005556	0.061111	0.066667
Total	0.444444	0.333333	0.222222	1.000000

```
pd.crosstab(df_new['Income'],df_new['Product'],margins=True, margins_name='Total', normalize=True).head(20)
```

Product	KP281	KP481	KP781	Total
Income				
34053.15	0.033333	0.016667	0.000000	0.050000
34110.0	0.011111	0.016667	0.000000	0.027778
35247.0	0.027778	0.000000	0.000000	0.027778
36384.0	0.016667	0.005556	0.000000	0.022222
37521.0	0.011111	0.000000	0.000000	0.011111
38658.0	0.016667	0.011111	0.000000	0.027778
39795.0	0.011111	0.000000	0.000000	0.011111
40932.0	0.022222	0.011111	0.000000	0.033333
42069.0	0.011111	0.000000	0.000000	0.011111
43206.0	0.005556	0.022222	0.000000	0.027778
44343.0	0.022222	0.000000	0.000000	0.022222
45480.0	0.027778	0.050000	0.000000	0.077778
46617.0	0.038889	0.005556	0.000000	0.044444
47754.0	0.000000	0.011111	0.000000	0.011111
48556.0	0.000000	0.000000	0.011111	0.011111
48658.0	0.000000	0.000000	0.005556	0.005556
48891.0	0.011111	0.016667	0.000000	0.027778
49801.0	0.000000	0.000000	0.011111	0.011111
50028.0	0.011111	0.027778	0.000000	0.038889
51165.0	0.016667	0.022222	0.000000	0.038889

Insights

- Find the conditional probability that an event occurs given that another event has occurred. (Example: given that a customer is female, what is the probability she'll purchase a KP481)
- Find the probability that the customer buys a product based on each column.

Recommendations

- Based on previous crosstab values you find the probability.

Assumptions

- 42.22% Are Female.
- 57.78% Are Male.
- Mostly Female 22.22% have purchase KP281(Cheapest).
- Mostly Male 22.22% have purchase KP281(Cheapest).
- There is 59.44% are Partnered and 40.56% are Single
- Mostly 26.67% Partnered have KP281.
- 20% Partnered have KP481.
- Mostly 17.78% single have KP281.
- 47.22% have 16 Year Education.
- 53.89% have fitness score 3.

5. Check the correlation among different factors

- ### Find the correlation between the given features in the table.

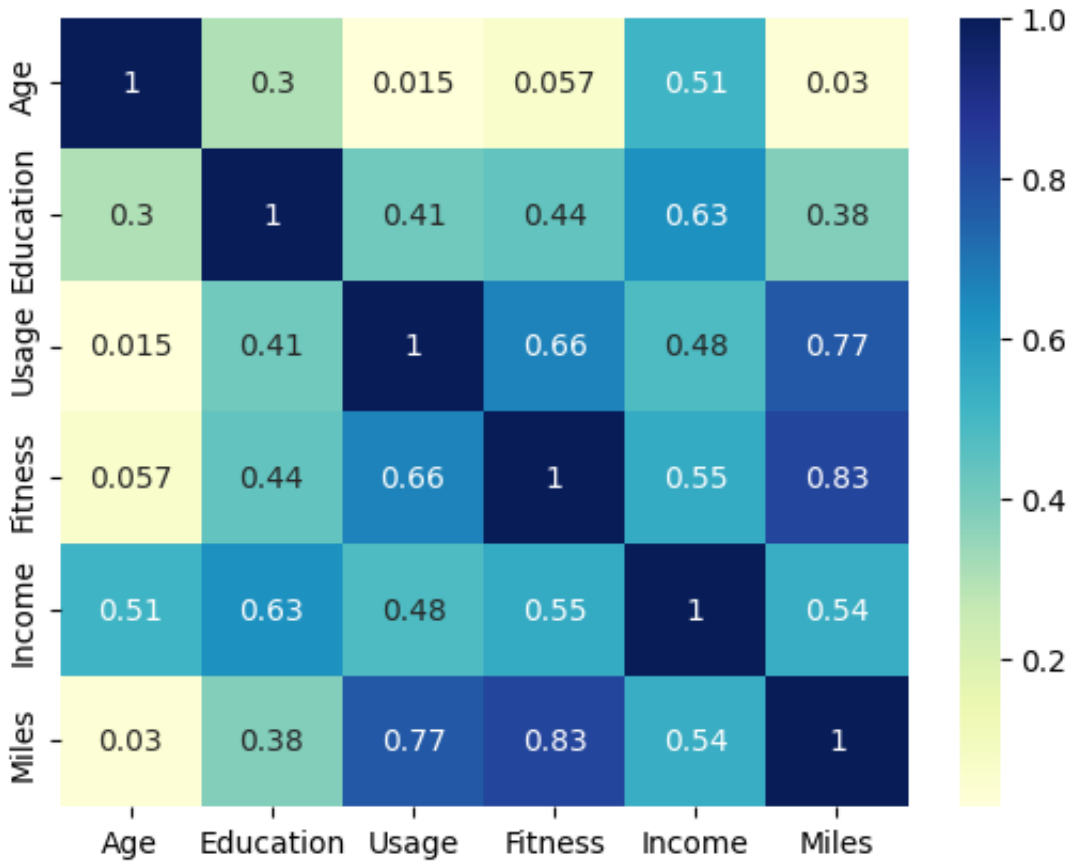
```
df_new.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.0	Male	14	Single	3.0	4	34053.15	112
1	KP281	20.0	Male	15	Single	2.0	3	34053.15	75
2	KP281	20.0	Female	14	Partnered	4.0	3	34053.15	66
3	KP281	20.0	Male	14	Single	3.0	3	34053.15	85
4	KP281	20.0	Male	14	Partnered	4.0	2	35247.00	47

```
df_new.corr(numeric_only = True)
```

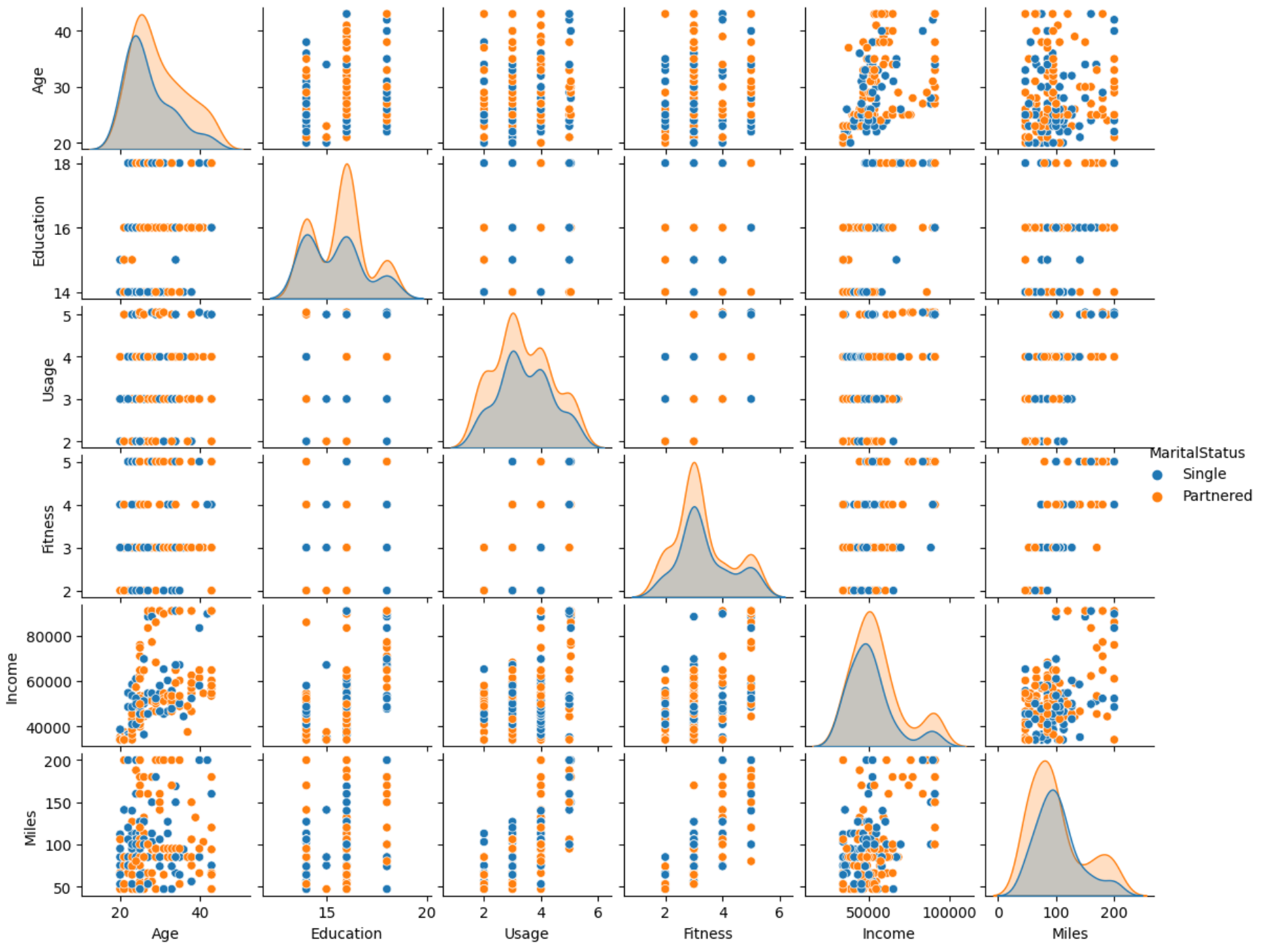
	Age	Education	Usage	Fitness	Income	Miles
Age	1.000000	0.301971	0.015394	0.057361	0.514362	0.029636
Education	0.301971	1.000000	0.413600	0.441082	0.628597	0.377294
Usage	0.015394	0.413600	1.000000	0.661978	0.481608	0.771030
Fitness	0.057361	0.441082	0.661978	1.000000	0.546998	0.826307
Income	0.514362	0.628597	0.481608	0.546998	1.000000	0.537297
Miles	0.029636	0.377294	0.771030	0.826307	0.537297	1.000000

```
sns.heatmap(df_new.corr(numeric_only = True), cmap="YlGnBu", annot=True)
plt.show()
```



```
ax=sns.pairplot(df_new,hue='Gender')
ax.fig.set_size_inches(12,9)
plt.show()
```

```
ax=sns.pairplot(df_new,hue='MaritalStatus')
ax.fig.set_size_inches(12,9)
plt.show()
```



```
ax=sns.pairplot(df_new,hue='Product')
ax.fig.set_size_inches(12,9)
plt.show()
```

Insights

- Find the correlation between the given features in the table

Recommendations

- We can use the heatmap and corr function to find the correlation between the variables.
- In a correlation heatmap, each variable is represented by a row and a column, and the cells show the correlation between them. The color of each cell represents the strength and direction of the correlation, with darker colors indicating stronger correlations.
- Darker colors indicate stronger correlations, while lighter colors indicate weaker correlations.
- Positive correlations (when one variable increases, the other variable tends to increase).
- Negative correlations (when one variable increases, the other variable tends to decrease).

Assumptions

- All are positive Correlations.
- Education And Income have 63% Correlation.
- Age and Income have 51% Correlation.
- Usage and Fitness have 66% Correlation.
- Usage and Miles have 77% Correlation.
- Fitness And Income have 55% Correlation.
- Fitness And Miles have 83% Correlation.
- Income and Miles have 54% Correlation.

Customer profiling and recommendation

- ### Make customer profilings for each and every product.

```
df_new.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.0	Male	14	Single	3.0	4	34053.15	112
1	KP281	20.0	Male	15	Single	2.0	3	34053.15	75
2	KP281	20.0	Female	14	Partnered	4.0	3	34053.15	66
3	KP281	20.0	Male	14	Single	3.0	3	34053.15	85
4	KP281	20.0	Male	14	Partnered	4.0	2	35247.00	47

```
event_dictionary = {'KP281': 1500, 'KP481': 1750, 'KP781': 2500}
df['Package'] = df['Product'].map(event_dictionary)
df_new
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Package
0	KP281	20.00	Male	14	Single	3.00	4	34053.15	112	1500
1	KP281	20.00	Male	15	Single	2.00	3	34053.15	75	1500
2	KP281	20.00	Female	14	Partnered	4.00	3	34053.15	66	1500
3	KP281	20.00	Male	14	Single	3.00	3	34053.15	85	1500
4	KP281	20.00	Male	14	Partnered	4.00	2	35247.00	47	1500
...	...	...	...	...	...	...	...	...	...	...
175	KP781	40.00	Male	18	Single	5.05	5	83416.00	200	2500
176	KP781	42.00	Male	18	Single	5.00	4	89641.00	200	2500
177	KP781	43.05	Male	16	Single	5.00	5	90886.00	160	2500
178	KP781	43.05	Male	18	Partnered	4.00	5	90948.25	120	2500
179	KP781	43.05	Male	18	Partnered	4.00	5	90948.25	180	2500

180 rows × 10 columns

- Adding Package column in Dataframe

```
df_gender=df_new[['Product','Gender']]
df_gender.groupby('Product').count()
```

Product	Gender
KP281	80
KP481	60
KP781	40

```
df_gender.groupby(['Product','Gender']).size()
```

```
Product  Gender
KP281    Female    40
         Male      40
KP481    Female    29
         Male      31
KP781    Female     7
         Male      33
dtype: int64
```

```
df_age=df_new[['Product','Age']]
df_age.groupby(['Product','Age']).size()
```

```
     Product  Age
     KP281    20.00     6
              21.00     4
              22.00     4
              23.00     8
              24.00     5
              25.00     7
              26.00     7
              27.00     3
              28.00     6
              29.00     3
              30.00     2
              31.00     2
              32.00     2
              33.00     2
              34.00     2
              35.00     3
              36.00     1
              37.00     1
              38.00     4
              39.00     1
              40.00     1
              41.00     1
              43.00     1
              43.05     4
     KP481    20.00     4
              21.00     3
              23.00     7
              24.00     3
              25.00    11
              26.00     3
              27.00     1
              29.00     1
              30.00     2
              31.00     3
              32.00     2
              33.00     5
              34.00     3
              35.00     4
              37.00     1
              38.00     2
              40.00     3
              43.05     2
     KP781    22.00     3
              23.00     3
              24.00     4
              25.00     7
              26.00     2
              27.00     3
              28.00     3
              29.00     2
              30.00     3
              31.00     1
              33.00     1
              34.00     1
              35.00     1
              38.00     1
              40.00     1
              42.00     1
              43.05     3
```

- Age 25 persons are taking more profit then other ages.

```
df_income=df_new[['Product','Income']]
df_income.groupby(['Product','Income']).size().head(30)
```

```
Product  Income
KP281    34053.15    6
         34110.00    2
         35247.00    5
         36384.00    3
         37521.00    2
         38658.00    3
         39795.00    2
         40932.00    4
         42069.00    2
         43206.00    1
         44343.00    4
         45480.00    5
         46617.00    7
         48891.00    2
         50028.00    2
         51165.00    3
         52302.00    6
         53439.00    3
         54576.00    7
         55713.00    1
         56850.00    2
         57987.00    1
         59124.00    1
         60261.00    2
         61398.00    1
         64809.00    1
         67083.00    1
         68220.00    1
KP481    34053.15    3
         34110.00    3
dtype: int64
```

```
df_totcost=df_new[['Product','Package']]
df_totcost.groupby('Product').sum()
```

Product	Package
KP281	120000
KP481	105000
KP781	100000

- There is 80 KP281 units and each cost 1500, so the total cost is 120000.
- There is 60 KP481 units and each cost 1750, so the total cost is 105000.
- There is 40 KP481 units and each cost 2500, so the total cost is 100000.

```
df_totcost1=df_new[['Product','Gender','Package']]
df_totcost1.groupby(['Product','Gender']).sum()
```

Product	Gender	Package
KP281	**Female**	60000
	Male	60000
KP481	**Female**	50750
	Male	54250
KP781	**Female**	17500
	Male	82500

- There is 40 Male who buy KP281, so total cost is 60000.
- There is 40 Female who buy KP281, so total cost is 60000.========Total for KP281= 120000
- There is 31 Male who buy KP481, so total cost is 54250.
- There is 29 Female who buy KP481, so total cost is 50750.========Total for KP481= 105000
- There is 33 Male who buy KP781, so total cost is 82500.
- There is 7 Female who buy K7281, so total cost is 17500.========Total for KP781= 100000

```
pd.crosstab(df_new['Age'],df_new['Product'],margins=True, margins_name='Total')
```

Product	KP281	KP481	KP781	Total
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
41.0	1	0	0	1
42.0	0	0	1	1
43.0	1	0	0	1
43.04999999999998	4	2	3	9
Total	80	60	40	180

Age VS Product

- At what Age Customer Buying which product.

```
pd.crosstab(df_new['Miles'],df_new['Product'],margins=True, margins_name='Total')
```

Product	KP281	KP481	KP781	Total
Miles				
47	12	5	0	17
53	0	7	0	7
56	6	0	0	6
64	0	6	0	6
66	10	0	0	10
74	0	3	0	3
75	10	0	0	10
80	0	0	1	1
85	16	11	0	27
94	8	0	0	8
95	0	12	0	12
100	0	0	7	7
103	3	0	0	3
106	0	8	1	9
112	1	0	0	1
113	8	0	0	8
120	0	0	3	3
127	0	5	0	5
132	2	0	0	2
140	0	0	1	1
141	2	0	0	2
150	0	0	4	4
160	0	0	5	5
169	1	0	0	1
170	0	2	1	3
180	0	0	6	6
188	1	0	0	1
200	0	1	11	12
Total	80	60	40	180

Miles VS Product

- IF CUSTOMER buying product, then how much miles he/she runs.

```
df_mari=df_new[['Product','MaritalStatus']]
df_mari.groupby(['Product','MaritalStatus']).size()
```

```
Product  MaritalStatus
KP281    Partnered        48
         Single           32
KP481    Partnered        36
         Single           24
KP781    Partnered        23
         Single           17
dtype: int64
```

```
df_totcost2=df_new[['Product','MaritalStatus','Package']]
df_totcost2.groupby(['Product','MaritalStatus']).sum()
```

		Package
Product	**MaritalStatus**	
KP281	**Partnered**	72000
	Single	48000
KP481	**Partnered**	63000
	Single	42000
KP781	**Partnered**	57500
	Single	42500

- Partnered are more in no. the Single.
    - So Partnered are taking more profit the Singles

```
df_totcost3=df_new[['Product','Fitness','Package']]
df_totcost3.groupby(['Product','Fitness']).sum()
```

		Package
Product	**Fitness**	
KP281	**2**	22500
	3	81000
	4	13500
	5	3000
KP481	**2**	22750
	3	68250
	4	14000
KP781	**3**	10000
	4	17500
	5	72500

```
pd.crosstab(df_new['Fitness'],df_new['Product'],margins=True, margins_name='Total')
```

Product	KP281	KP481	KP781	Total
Fitness				
2	15	13	0	28
3	54	39	4	97
4	9	8	7	24
5	2	0	29	31
Total	80	60	40	180

Fitness Score Analysis

- KP281
  - 54 person have 3 fitness score. Its profit for both company(81000) and Customer.
  - But less no. 5 fitness score.
- KP481
  - 39 person have 3 fitness score. Its profit for both company(68250) and Customer.
  - No one have 5 fitness score.
- KP781
  - I thing this most profitable product for company and Customer.
  - 29 persons with 5 fitness score.

```
df_totcost4=df_new[['Product','Usage','Package']]
df_totcost4.groupby(['Product','Usage']).sum()
```

		Package
Product	Usage	
KP281	**2.00**	28500
	3.00	55500
	4.00	33000
	5.00	3000
KP481	**2.00**	24500
	3.00	54250
	4.00	21000
	5.00	5250
KP781	**3.00**	2500
	4.00	45000
	5.00	30000
	5.05	22500

```
pd.crosstab(df_new['Usage'],df_new['Product'],margins=True, margins_name='Total')
```

Product	KP281	KP481	KP781	Total
Usage				
2.0	19	14	0	33
3.0	37	31	1	69
4.0	22	12	18	52
5.0	2	3	12	17
5.049999999999983	0	0	9	9
Total	80	60	40	180

Usage per week Analysis

- KP281
  - 37 person are using product 3 Times a week. Its profit for Customer.
  - 22 person are using product 4 Times a week. Its profit for Customer.
  - But less no. for 5 or more then 5 per week. which is less profitable for product vs usage.(2 people only)
- KP481
  - 31 person are using product 3 Times a week. Its profit for Customer.
  - 12 person are using product 4 Times a week. Its profit for Customer.
  - But less no. for 5 or more then 5 per week. which is less profitable for product vs usage.(3 people only)
- KP781
  - 18 person are using product 4 Times a week. Its profit for Customer.
  - 12 person are using product 5 Times a week. Its profit for Customer.
  - But more no. for 5 or more then 5 per week. which is most profitable in product vs usage.(21 people)

Insights

- Make customer profilings for each and every product.

Recommendations

- We want to find at What age, gender, and income group but product the KP281

Assumptions

- Age 25 persons are taking more profit then other ages.
- There is 80 KP281 units and each cost 1500, so the total cost is 120000.
- There is 60 KP481 units and each cost 1750, so the total cost is 105000.
- There is 40 KP481 units and each cost 2500, so the total cost is 100000.
- There is 40 Male who buy KP281, so total cost is 60000.
- There is 40 Female who buy KP281, so total cost is 60000.========Total for KP281= 120000
- There is 31 Male who buy KP481, so total cost is 54250.
- There is 29 Female who buy KP481, so total cost is 50750.========Total for KP481= 105000
- There is 33 Male who buy KP781, so total cost is 82500.
- There is 7 Female who buy K7281, so total cost is 17500.========Total for KP781= 100000
- Partnered are more in no. the Single.
    - So Partnered are taking more profit the Singles.

Fitness Score Analysis

- KP281
    - 54 person have 3 fitness score. Its profit for both company(81000) and Customer.
    - But less no. 5 fitness score.
- KP481
    - 39 person have 3 fitness score. Its profit for both company(68250) and Customer.
    - No one have 5 fitness score.
- KP781
    - I thing this most profitable product for company and Customer.
    - 29 persons with 5 fitness score.

Usage per week Analysis

- KP281
    - 37 person are using product 3 Times a week. Its profit for Customer.
    - 22 person are using product 4 Times a week. Its profit for Customer.
    - But less no. for 5 or more then 5 per week. which is less profitable for product vs usage.(2 people only)
- KP481
    - 31 person are using product 3 Times a week. Its profit for Customer.
    - 12 person are using product 4 Times a week. Its profit for Customer.
    - But less no. for 5 or more then 5 per week. which is less profitable for product vs usage.(3 people only)
- KP781
    - 18 person are using product 4 Times a week. Its profit for Customer.
    - 12 person are using product 5 Times a week. Its profit for Customer.
    - But more no. for 5 or more then 5 per week. which is most profitable in product vs usage.(21 people)

Colab Link:- https://colab.research.google.com/drive/1olgRpcLGfJ4DZoBX2C9Wd9Qd5wdJtXy-?usp=sharing

Dataset Link:- https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749