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

df = pd.read_csv('/content/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

```
print("No of rows:",df.shape[0],"\nNo of columns:",df.shape[1])
```

```
No of rows: 180
No of columns: 9
```

```
#shows the columns name, count of values and null colums, Data type
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
```

```
#for numerric column, finding row count, mean, median, min value, max value, standard deviation
df.describe().T
```

	count	mean	std	min	25%	50%	75%	
Age	180.0	28.788889	6.943498	18.0	24.00	26.0	33.00	
Education	180.0	15.572222	1.617055	12.0	14.00	16.0	16.00	
Usage	180.0	3.455556	1.084797	2.0	3.00	3.0	4.00	
Fitness	180.0	3.311111	0.958869	1.0	3.00	3.0	4.00	
Income	180.0	53719.577778	16506.684226	29562.0	44058.75	50596.5	58668.00	104

```
#for object data type, finding count of rows, number od uniques values and its frequency in the given data
df.describe(include='object').T
```

	count	unique	top	freq
Product	180	3	KP281	80
Gender	180	2	Male	104
MaritalStatus	180	2	Partnered	107

```
df['Product'].unique()
```

```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

```
df['Product'].nunique()
```

```
3
```

```
(df['Product'].value_counts(normalize=True)*100).round(2)
```

```
KP281    44.44
KP481    33.33
KP781    22.22
Name: Product, dtype: float64
```

```
df['Age'].unique()
```

```
array([18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 43, 44, 46, 47, 50, 45, 48, 42])
```

```
df['Age'].nunique()
```

```
32
```

```
(df['Age'].value_counts(normalize=True)*100).round(2)
```

```
25    13.89
23    10.00
24     6.67
26     6.67
28     5.00
35     4.44
33     4.44
30     3.89
38     3.89
21     3.89
22     3.89
27     3.89
31     3.33
34     3.33
29     3.33
20     2.78
40     2.78
32     2.22
19     2.22
48     1.11
37     1.11
45     1.11
47     1.11
46     0.56
50     0.56
18     0.56
44     0.56
43     0.56
41     0.56
39     0.56
36     0.56
42     0.56
Name: Age, dtype: float64
```

```
df['Gender'].unique()
```

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

```
df['Gender'].nunique()
```

```
2
```

```
(df['Gender'].value_counts(normalize=True)*100).round(2)
```

```
Male      57.78
Female    42.22
Name: Gender, dtype: float64
```

```
df['MaritalStatus'].unique()
```

```
     array(['Single', 'Partnered'], dtype=object)
```

```
df['MaritalStatus'].nunique()
```

```
     2
```

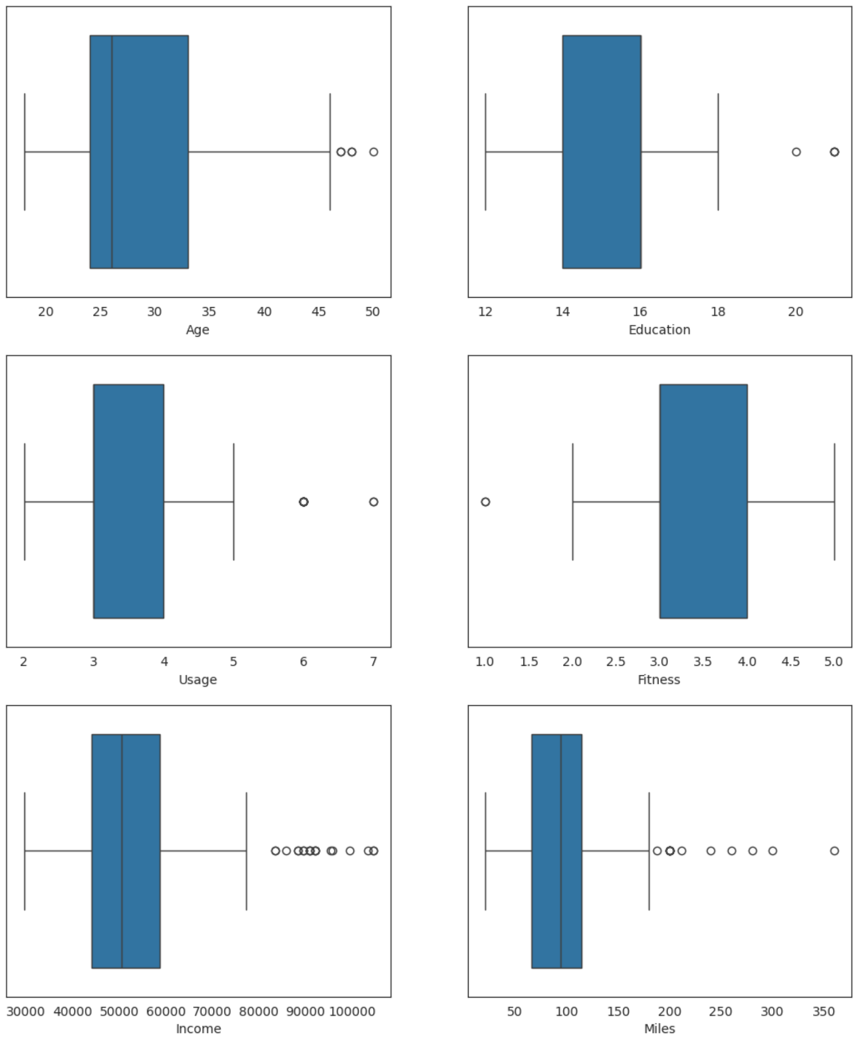
```
(df['MaritalStatus'].value_counts(normalize=True)*100).round(2)
```

```
     Partnered    59.44
     Single       40.56
     Name: MaritalStatus, dtype: float64
```

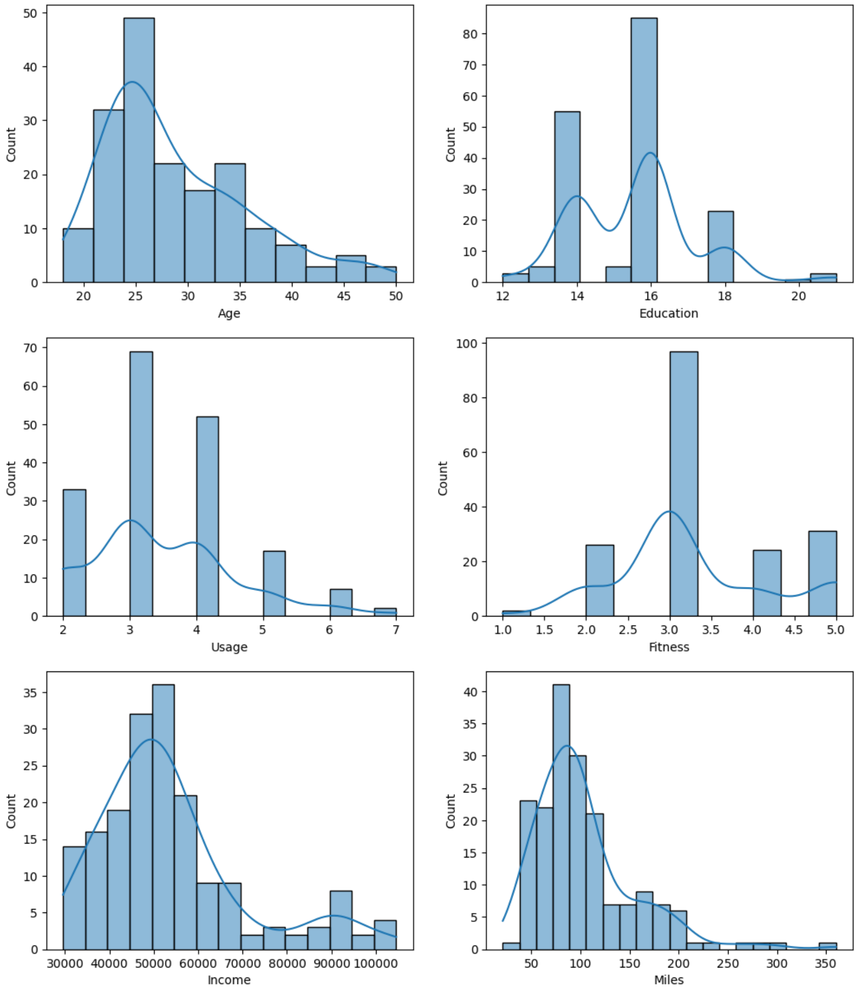
```
# Finding Missing values

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

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

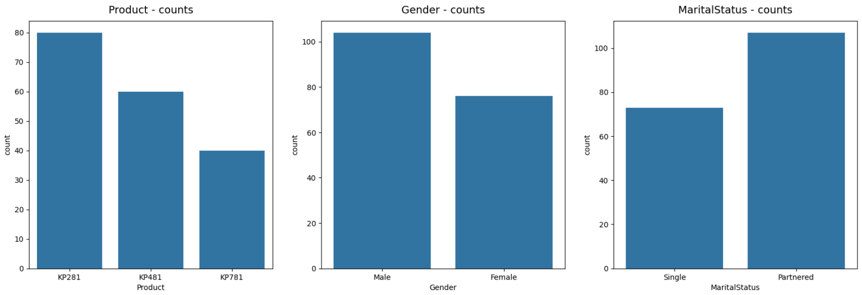
```
#Univariate Analysis

#outlier detectiong using Boxplot

fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))
fig.subplots_adjust(top=1.2)

sns.boxplot(data=df, x="Age", orient='h', ax=axis[0,0])
sns.boxplot(data=df, x="Education", orient='h', ax=axis[0,1])
sns.boxplot(data=df, x="Usage", orient='h', ax=axis[1,0])
sns.boxplot(data=df, x="Fitness", orient='h', ax=axis[1,1])
sns.boxplot(data=df, x="Income", orient='h', ax=axis[2,0])
sns.boxplot(data=df, x="Miles", orient='h', ax=axis[2,1])
plt.show()
```



```
#Understanding the distribution of the data for the quantitative attributes:Age,Education,Usage,Fitness,Income,Miles

fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))
fig.subplots_adjust(top=1.2)

sns.histplot(data=df, x="Age", kde=True, ax=axis[0,0])
sns.histplot(data=df, x="Education", kde=True, ax=axis[0,1])
sns.histplot(data=df, x="Usage", kde=True, ax=axis[1,0])
sns.histplot(data=df, x="Fitness", kde=True, ax=axis[1,1])
sns.histplot(data=df, x="Income", kde=True, ax=axis[2,0])
sns.histplot(data=df, x="Miles", kde=True, ax=axis[2,1])
plt.show()
```



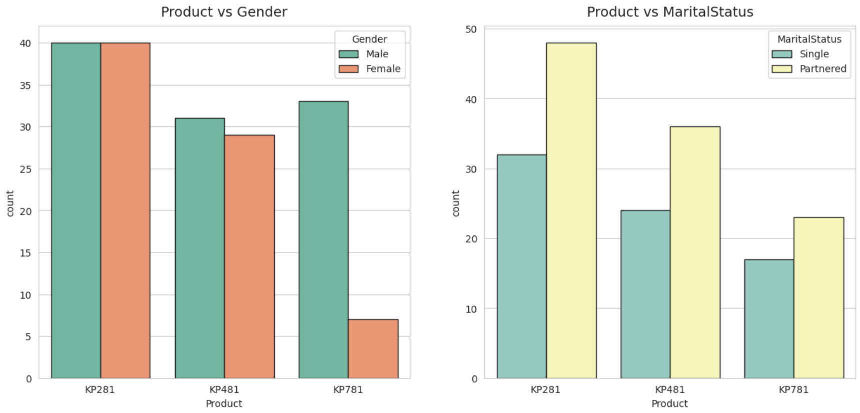
```
#Understanding the distribution of the data for the qualitative attributes: Product,Gender,MaritalStatus

fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(20, 6))
sns.countplot(data=df, x='Product', ax=axs[0])
sns.countplot(data=df, x='Gender', ax=axs[1])
sns.countplot(data=df, x='MaritalStatus', ax=axs[2])

axs[0].set_title("Product - counts", pad=10, fontsize=14)
axs[1].set_title("Gender - counts", pad=10, fontsize=14)
axs[2].set_title("MaritalStatus - counts", pad=10, fontsize=14)
plt.show()
```



```
df1 = df[['Product', 'Gender', 'MaritalStatus']].melt()
df1.groupby(['variable', 'value'])[['value']].count() / len(df)
```

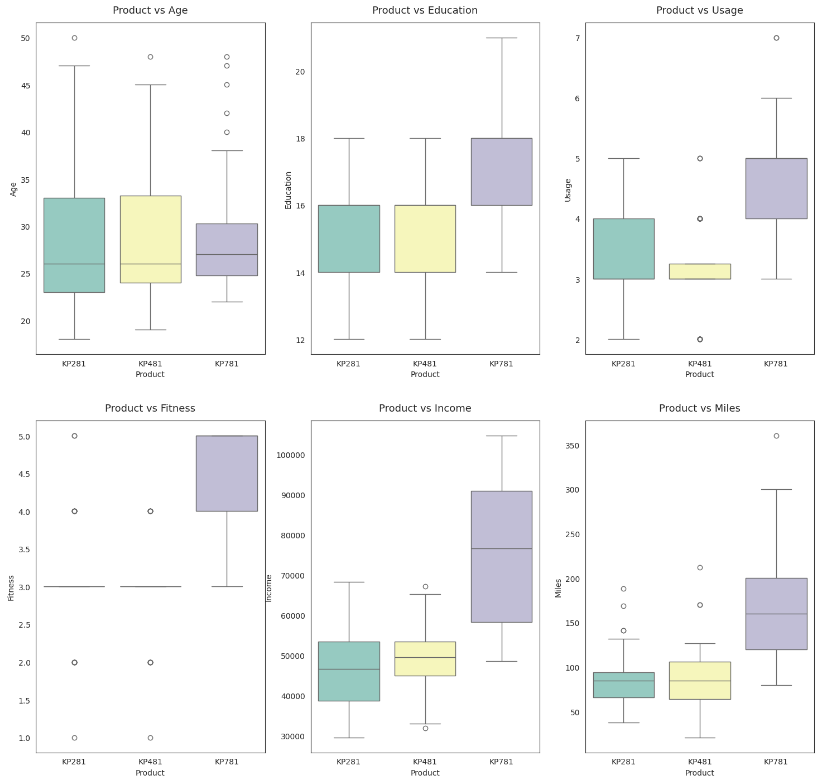
		value
variable	value	
Gender	Female	0.422222
	Male	0.577778
MaritalStatus	Partnered	0.594444
	Single	0.405556
Product	KP281	0.444444
	KP481	0.333333
	KP781	0.222222

```
#Bivariate Analysis
#Checking if features - Gender or MaritalStatus have any effect on the product purchased.

sns.set_style(style='whitegrid')
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15, 6.5))
sns.countplot(data=df, x='Product', hue='Gender', edgecolor="0.15", palette='Set2', ax=axs[0])
sns.countplot(data=df, x='Product', hue='MaritalStatus', edgecolor="0.15", palette='Set3', ax=axs[1])
axs[0].set_title("Product vs Gender", pad=10, fontsize=14)
axs[1].set_title("Product vs MaritalStatus", pad=10, fontsize=14)
plt.show()
```



```
# Checking if following features have any effect on the product purchased: Age,Education,Usage,Fitness,Income,Miles

attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(18, 12))
fig.subplots_adjust(top=1.2)
count = 0
for i in range(2):
    for j in range(3):
        sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i,j], palette='Set3')
        axs[i,j].set_title(f"Product vs {attrs[count]}", pad=12, fontsize=13)
        count += 1
```

```
  sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i,j], palette='Set3')
<ipython-input-33-6aee39b5577c>:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v

  sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i,j], palette='Set3')
<ipython-input-33-6aee39b5577c>:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v

  sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i,j], palette='Set3')
<ipython-input-33-6aee39b5577c>:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v

  sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i,j], palette='Set3')
<ipython-input-33-6aee39b5577c>:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v

  sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i,j], palette='Set3')
```


Product vs Age
Age
KP281
KP481
KP781
Product
Product vs Education
Education
KP281
KP481
KP781
Product
Product vs Usage
Usage
KP281
KP481
KP781
Product
Product vs Fitness
Fitness
KP281
KP481
KP781
Product
Product vs Income
Income
KP281
KP481
KP781
Product
Product vs Miles
Miles
KP281
KP481
KP781
Product