Business Case: Aerofit

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

```
df = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749")
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

shape of data

```
[5] #shape of the data
    print("Rows in the data:",df.shape[0])
    print("Columns in the data:",df.shape[1])
```

```
Rows in the data: 180
Columns in the data: 9
```

```
[ ] #data types of all the attributes
    df.dtypes
```

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

```python
# Statical Analysis
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
```

```python
#statistical summary
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

2.Non-Graphical Analysis: Value counts and unique attributes

```python
#count and unique value for product
print("Count of the product:",df["Product"].value_counts()),df["Product"].unique()
```

```
Count of the product: KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
(None, array(['KP281', 'KP481', 'KP781'], dtype=object))
```

```python
#count and unique value for gender
print("Count of gender:",df["Gender"].value_counts()),df["Gender"].unique()
```

```
Count of gender: Male      104
Female     76
Name: Gender, dtype: int64
(None, array(['Male', 'Female'], dtype=object))
```

```python
#count of MaritalStatus
print("count of MaritalStatus:",df["MaritalStatus"].value_counts()),df["MaritalStatus"].unique()
```

```
count of MaritalStatus: Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
(None, array(['Single', 'Partnered'], dtype=object))
```

```
[ ] #Value Count
    df.value_counts(normalize=True)
```

```
Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  Miles
KP281    18   Male    14         Single         3      4        29562   112      0.005556
KP481    30   Female  13         Single         4      3        46617   106      0.005556
         31   Female  16         Partnered      2      3        51165   64       0.005556
                      18         Single         2      1        65220   21       0.005556
              Male    16         Partnered      3      3        52302   95       0.005556
                                                                                   ...
KP281    34   Female  16         Single         2      2        52302   66       0.005556
              Male    16         Single         4      5        51165   169      0.005556
         35   Female  16         Partnered      3      3        60261   94       0.005556
                      18         Single         3      3        67083   85       0.005556
KP781    48   Male    18         Partnered      4      5        95508   180      0.005556
Length: 180, dtype: float64
```

```
[ ] #unique attributes
    df.drop_duplicates().head()
```

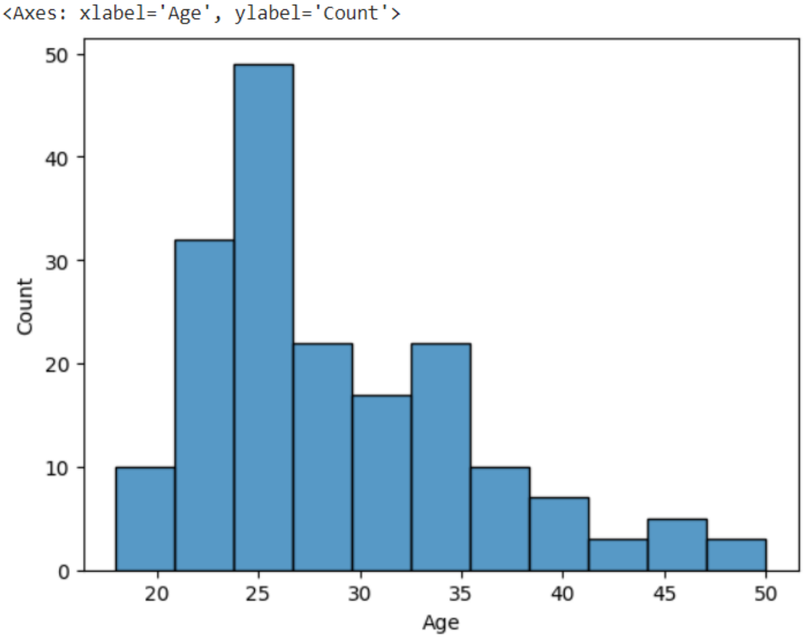
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

3.Visual Analysis - Univariate & Bivariate

(a)For continuous variable(s): Distplot, countplot, histogram for univariate analysis

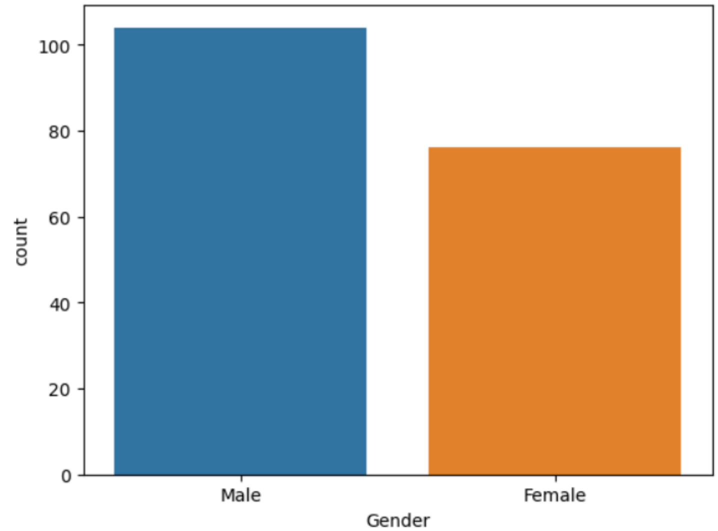
```
[ ] sns.histplot(df['Age'])
```

<Axes: xlabel='Age', ylabel='Count'>



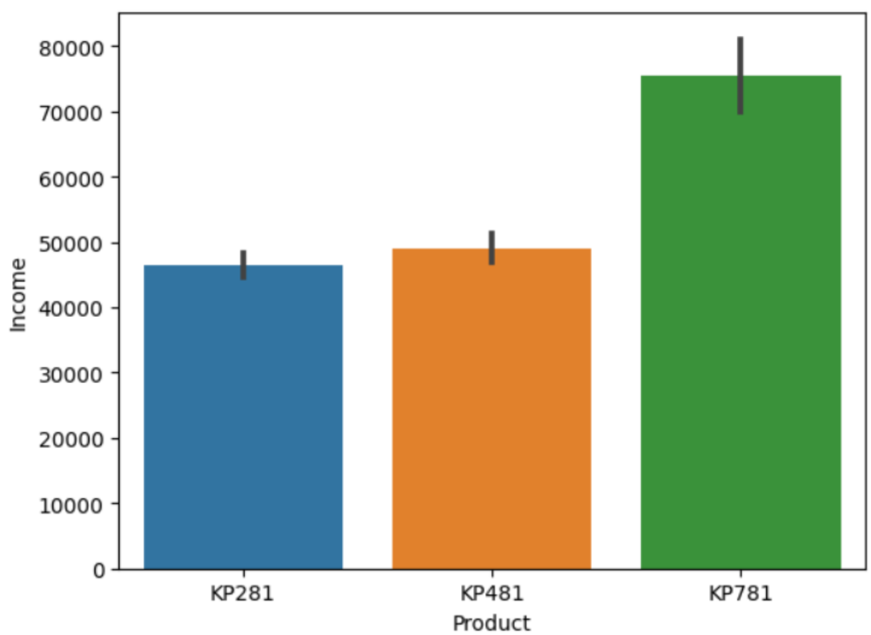
```
[ ]  sns.countplot(x=df["Gender"])
```

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



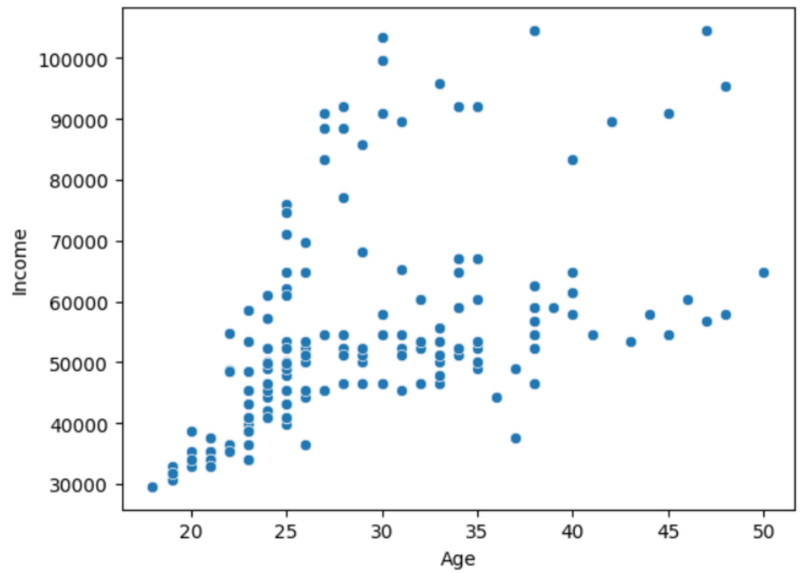

```
[ ]  #bivariate
     #Categorical vs Numerical
     sns.barplot(x = df['Product'], y = df['Income'])
```

<Axes: xlabel='Product', ylabel='Income'>



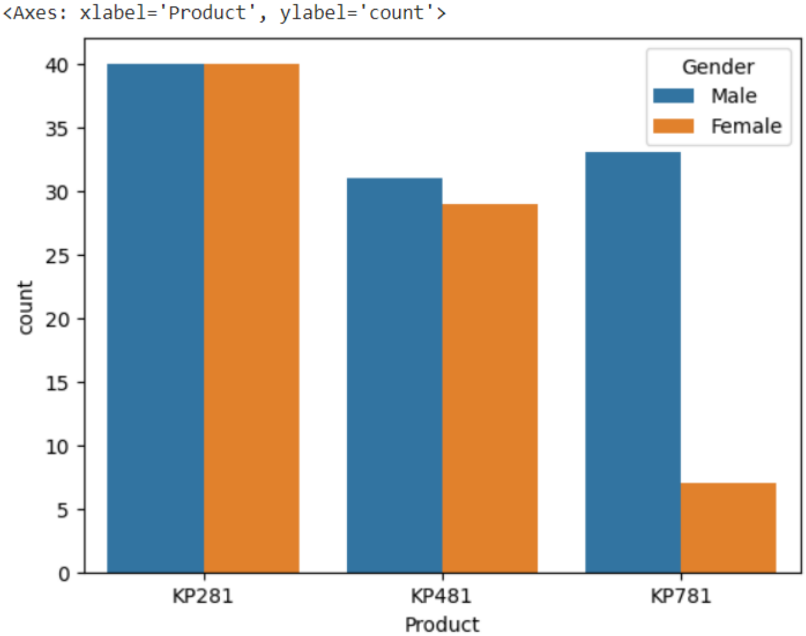
```
[ ] #Numerical vs Numerical
    sns.scatterplot(x = df['Age'], y = df['Income'])
```

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




```
[ ] #Categorical vs Categorical
    sns.countplot(x = df['Product'], hue = df['Gender'])
```

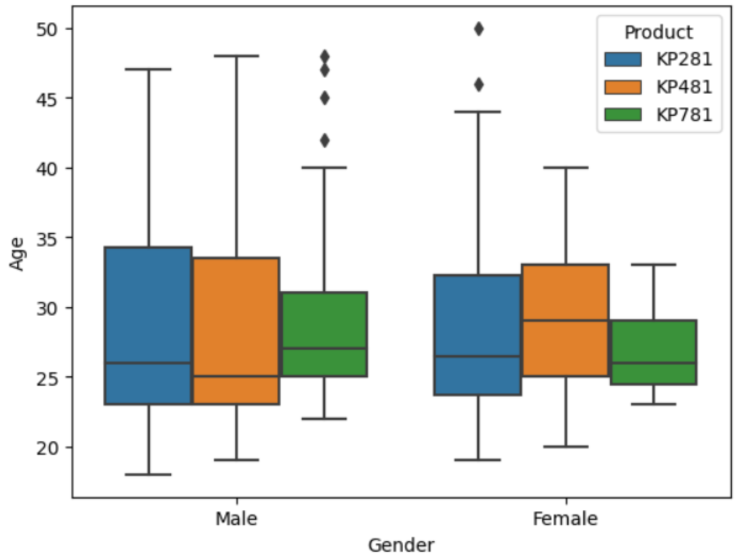
<Axes: xlabel='Product', ylabel='count'>



(b) For categorical variable(s): Boxplot

```
[ ] sns.boxplot(x= "Gender" ,y="Age",hue="Product",data=df)
```

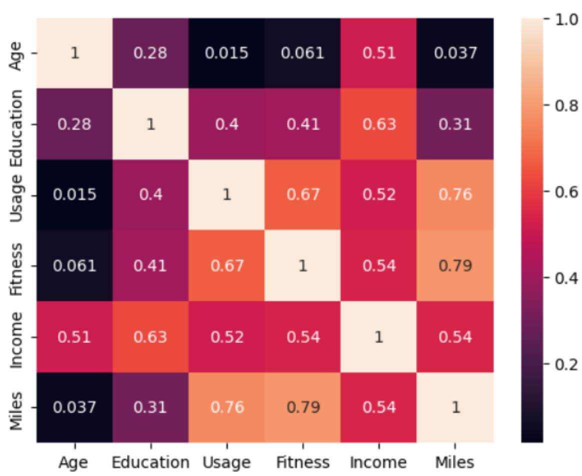
```
<Axes: xlabel='Gender', ylabel='Age'>
```



(c)For correlation: Heatmaps, Pairplots

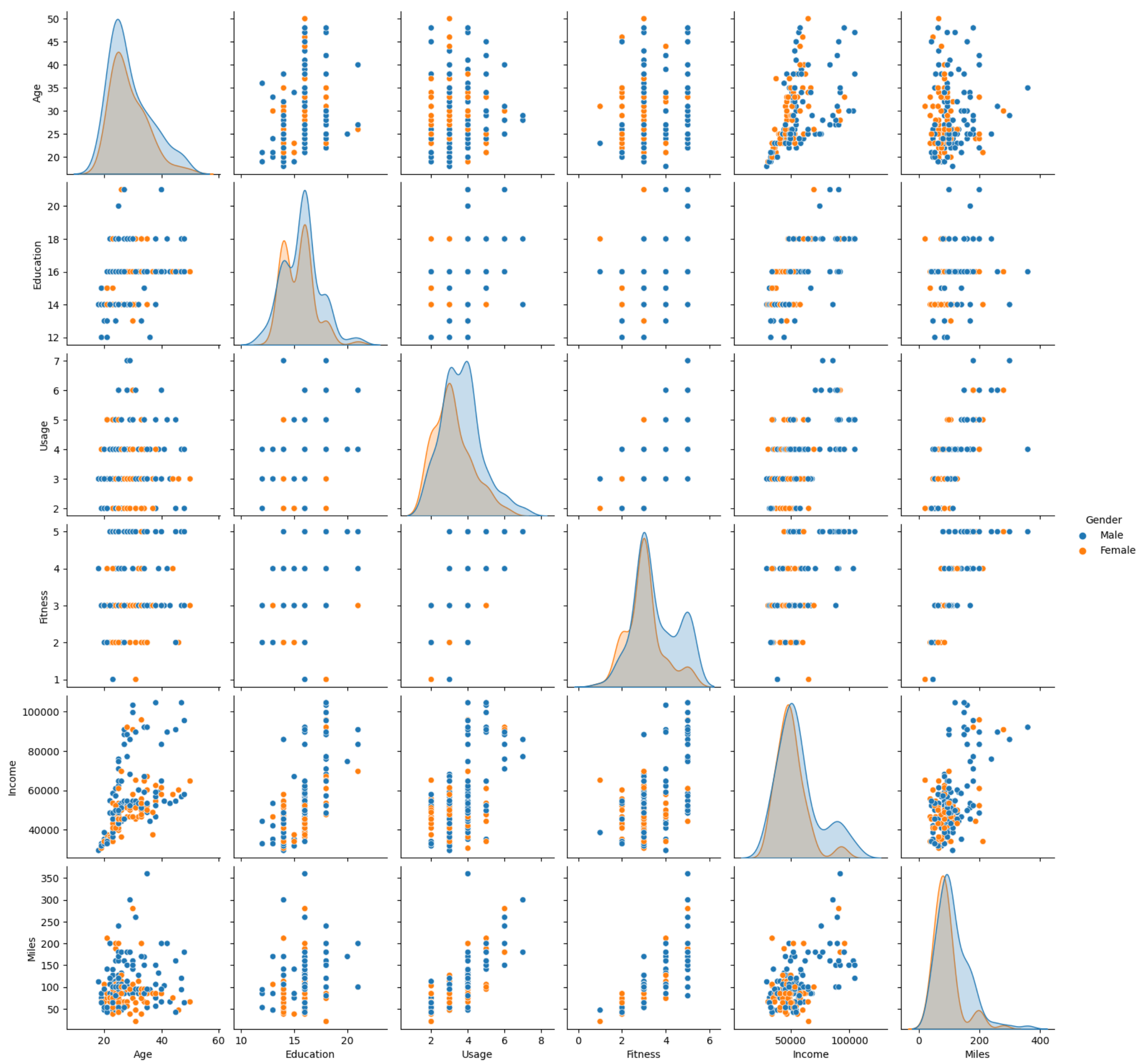
```
[ ] sns.heatmap(df.corr(), annot=True)
```

```
<ipython-input-59-6dc1c4c1753e>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will
  sns.heatmap(df.corr(), annot=True)
<Axes: >
```



{x} 18s

```
sns.pairplot(df, hue = 'Gender')
```



(4) Missing Value & Outlier Detection

```
[ ] df.isnull().values.any()
```

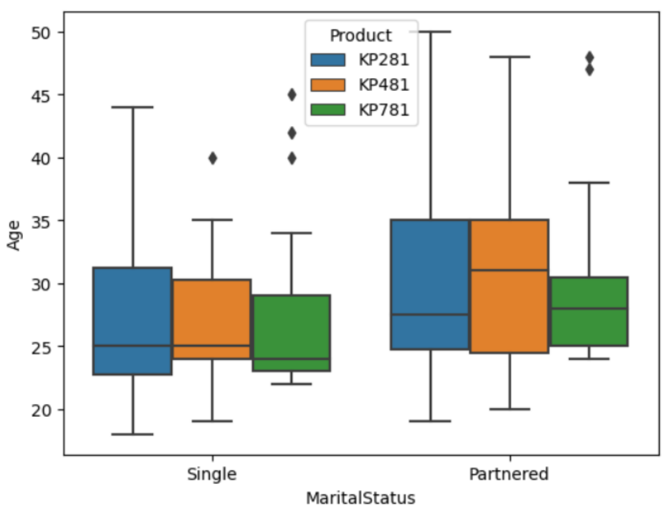
```
False
```

```
[ ] df.isnull().sum()
```

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

```
[ ] sns.boxplot(x = 'MaritalStatus', y = 'Age' , hue = 'Product', data = df)
```

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



(5) Business Insights based on Non-Graphical and Visual Analysis. Comments on the range of attributes

There are some attributes like - numerical and categorical present in the data.

Under numeric attributes like - Minimum and Maximum values, Range, Mean and Median , Standard deviation etc.

Under Categorical attributes like -- Outliers, Unique Values, Most common values, Correlation

Comments for each univariate and bivariate plot

univariate

histplot - shows the distribution of Age. It appears to be approximately normal with a peak around. There is no outliers.

countplot - shows the distribution of Gender.It appears to be approximately normal with a peak around. There is no outliers.

Bivariate

Barplot - shows distribution between Product vs Income.

Scatterplot - shows distribution between Age vs Income.

Countplot - shows distribution between Product vs Gender.

#Business Insights

1. Individuals in a committed relationship are more inclined to purchase machines.

2. The KP281 product category dominated sales in both genders, accounting for the majority of the purchases

3. There are Male outliers in KP781 product and Female outliers in KP281 product category.

4. The majority of sales for our premium product, KP781, were in the Male category.

5.The machines are more likely to be purchased by individuals aged between 20 and 35.