Aerofit Business case study

Aerofit : Aerofit is a top brand known for its fitness equipment. The company offers a variety of products, including treadmills, exercise bikes, gym machines, and fitness accessories. These products are designed to meet the fitness needs of people from all walks of life.

Business Problem: The market research team at AeroFit aims to determine the characteristics of the target audience for each type of treadmill offered by the company. This will help them provide better recommendations of the threadmills to new customers.

Importing the required libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
!pip install pandas_profiling
```

```
  Downloading visions-0.7.6-py3-none-any.whl (104 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 104.8/104.8 kB 5.2 MB/s eta 0:00:00
Requirement already satisfied: numpy<2,>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling) (
Collecting htmlmin==0.1.12 (from ydata-profiling->pandas_profiling)
  Downloading htmlmin-0.1.12.tar.gz (19 kB)
  Preparing metadata (setup.py) ... done
Collecting phik<0.13,>=0.11.1 (from ydata-profiling->pandas_profiling)
  Downloading phik-0.12.4-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (686 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 686.1/686.1 kB 9.3 MB/s eta 0:00:00
Requirement already satisfied: requests<3,>=2.24.0 in /usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling
Requirement already satisfied: tqdm<5,>=4.48.2 in /usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling) (4
Requirement already satisfied: seaborn<0.14,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profili
Collecting multimethod<2,>=1.4 (from ydata-profiling->pandas_profiling)
  Downloading multimethod-1.12-py3-none-any.whl (10 kB)
Requirement already satisfied: statsmodels<1,>=0.13.2 in /usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profil
Collecting typeguard<5,>=3 (from ydata-profiling->pandas_profiling)
  Downloading typeguard-4.3.0-py3-none-any.whl (35 kB)
Collecting imagehash==4.3.1 (from ydata-profiling->pandas_profiling)
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl (296 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 296.5/296.5 kB 10.8 MB/s eta 0:00:00
Requirement already satisfied: wordcloud>=1.9.1 in /usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling) (
Collecting dacite>=1.8 (from ydata-profiling->pandas_profiling)
  Downloading dacite-1.8.1-py3-none-any.whl (14 kB)
Requirement already satisfied: numba<1,>=0.56.0 in /usr/local/lib/python3.10/dist-packages (from ydata-profiling->pandas_profiling) (
Requirement already satisfied: PyWavelets in /usr/local/lib/python3.10/dist-packages (from imagehash==4.3.1->ydata-profiling->pandas_
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from imagehash==4.3.1->ydata-profiling->pandas_prof
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2<3.2,>=2.11.1->ydata-profiling-
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-profilir
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-profiling->p
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-profili
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-profili
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-profiling
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-profilir
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-prof
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba<1,>=0.56.0->ydata-pr
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas!=1.4.0,<3,>1.1->ydata-profiling->
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas!=1.4.0,<3,>1.1->ydata-profiling
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.10/dist-packages (from phik<0.13,>=0.11.1->ydata-profiling->p
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata-profiling->
Requirement already satisfied: pydantic-core==2.20.1 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata-profiling->p
Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata-profiling
```

```
Installing collected packages: htmlmin, typeguard, multimethod, dacite, imagehash, visions, phik, ydata-profiling, pandas_profiling
Successfully installed dacite-1.8.1 htmlmin-0.1.12 imagehash-4.3.1 multimethod-1.12 pandas_profiling-3.6.6 phik-0.12.4 typeguard-4.3.
```

```
from ydata_profiling import ProfileReport
```

Downloading Aerofit Dataset:

```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749
To: /content/aerofit_treadmill.csv?1639992749
100% 7.28k/7.28k [00:00<00:00, 17.5MB/s]
```

```
df = pd.read_csv('/content/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

Next steps: Generate code with df View recommended plots

1. Data Analysis :

```
df.shape
```

```
(180, 9)
```

```
df.size
```

```
1620
```

```
df.dtypes
```

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

```
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
```

```
df.duplicated().sum()
```

```
0
```

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

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

```
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

Next steps: Generate code with df View recommended plots

```
df.tail()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
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

Insights On Analysis:

1. Observed that there are 180 rows and 9 columns
2. Observed that there are no duplicate values and also there are no null values present in the given dataset.

Statiscal Infomation:

```
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.describe(include="object")
```

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

```
df.describe(include='all')
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

```
ProfileReport(df)
```

Summarize dataset: 100% 43/43 [00:14<00:00, 1.78it/s, Completed]

Generate report structure: 100% 1/1 [00:12<00:00, 12.22s/it]

Render HTML: 100% 1/1 [00:00<00:00, 1.09it/s]

Overview

Dataset statistics

Number of variables	9
Number of observations	180
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	12.8 KiB
Average record size in memory	72.7 B

Variable types

Categorical	4
Numeric	5

Alerts

`Age` is highly overall correlated with `Income`	High correlation
`Education` is highly overall correlated with `Income`	High correlation
`Fitness` is highly overall correlated with `Miles` and 2 other fields (Miles, Product, Usage)	High correlation
`Income` is highly overall correlated with `Age` and 1 other fields (Age, Education)	High correlation
`Miles` is highly overall correlated with `Fitness` and 1 other fields (Fitness, Usage)	High correlation
`Product` is highly overall correlated with `Fitness`	High correlation
`Usage` is highly overall correlated with `Fitness` and 1 other fields (Fitness, Miles)	High correlation

```
df.nunique()
```

```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

```
df.value_counts()
```

```
Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  Miles
KP281    18   Male    14         Single         3      4        29562   112      1
KP481    30   Female  13         Single         4      3        46617   106      1
         31   Female  16         Partnered      2      3        51165   64       1
                      18         Single         2      1        65220   21       1
              Male    16         Partnered      3      3        52302   95       1
                                                                                ..
KP281    34   Female  16         Single         2      2        52302   66       1
              Male    16         Single         4      5        51165   169      1
         35   Female  16         Partnered      3      3        60261   94       1
                      18         Single         3      3        67083   85       1
KP781    48   Male    18         Partnered      4      5        95508   180      1
Name: count, Length: 180, dtype: int64
```

```
df.columns
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

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

```
Fitness
3    97
5    31
2    26
4    24
1     2
Name: count, dtype: int64
```

```
df["Income"].value_counts()
```

```
Income
45480    14
52302     9
46617     8
54576     8
53439     8
         ..
65220     1
55713     1
68220     1
30699     1
95508     1
Name: count, Length: 62, dtype: int64
```

```
def income_level(x):
  if x>=60000:
    return "High"
  elif x>=30000 and x<60000:
    return "Medium"
  else:
    return "Low"
```

```
df["Income_Range"]=df["Income"].apply(income_level)
df
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Income_Range
0	KP281	18	Male	14	Single	3	4	29562	112	Low
1	KP281	19	Male	15	Single	2	3	31836	75	Medium
2	KP281	19	Female	14	Partnered	4	3	30699	66	Medium
3	KP281	19	Male	12	Single	3	3	32973	85	Medium
4	KP281	20	Male	13	Partnered	4	2	35247	47	Medium
...	...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200	High
176	KP781	42	Male	18	Single	5	4	89641	200	High
177	KP781	45	Male	16	Single	5	5	90886	160	High
178	KP781	47	Male	18	Partnered	4	5	104581	120	High
179	KP781	48	Male	18	Partnered	4	5	95508	180	High

180 rows × 10 columns

Next steps: Generate code with df View recommended plots

```
bins = [18,25,35,50]

#Creating lables for the bins
lables = ['18-25','26-35','36-50']
#Creating new column in the dataframe
df["Age_Group"]=pd.cut(df["Age"],bins=bins,labels=lables,include_lowest=True)

df
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Income_Range	Age_Group
0	KP281	18	Male	14	Single	3	4	29562	112	Low	18-25
1	KP281	19	Male	15	Single	2	3	31836	75	Medium	18-25
2	KP281	19	Female	14	Partnered	4	3	30699	66	Medium	18-25
3	KP281	19	Male	12	Single	3	3	32973	85	Medium	18-25
4	KP281	20	Male	13	Partnered	4	2	35247	47	Medium	18-25
...	...	...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200	High	36-50
176	KP781	42	Male	18	Single	5	4	89641	200	High	36-50
177	KP781	45	Male	16	Single	5	5	90886	160	High	36-50
178	KP781	47	Male	18	Partnered	4	5	104581	120	High	36-50
179	KP781	48	Male	18	Partnered	4	5	95508	180	High	36-50

180 rows × 11 columns

Next steps: Generate code with df View recommended plots

```
df.describe(include="object")
```

	Product	Gender	MaritalStatus	Income_Range
count	180	180	180	180
unique	3	2	2	3
top	KP281	Male	Partnered	Medium
freq	80	104	107	137

Statistical Summary : -

Age Distribution: The participants have an average age of about 28.79 years, with ages ranging from 18 to 50 years. Most participants, falling within the 25th to 75th percentile, are aged between 24 and 33 years.

Education: On average, participants have completed approximately 15.57 years of education. The minimum education level is 12 years, and the maximum is 21 years. The middle 50% of participants have between 14 and 16 years of education.

Usage and Fitness: Participants typically use fitness equipment around 3.46 times per week and rate their fitness level at an average of 3.31 on a scale of 1 to 5. The standard deviations for usage and fitness level are 1.08 and 0.96, respectively, indicating relatively consistent patterns.

Income: The average annual income of participants is about $53,719, with incomes ranging from$29,562 to $104,581. There is a wide variation in income, with a standard deviation of approximately$16,506.

Miles: Participants travel an average of 103.19 miles per week using fitness equipment. The distances range from 21 to 360 miles per week, with the middle 50% covering between 66 and 114.75 miles.

Product: There are three unique products. The product with the code 'KP281' is the most common, appearing 80 times in the data.

Gender: The dataset includes two genders: Male and Female. Males are more frequent, appearing 104 times.

Marital Status: There are two marital statuses: Partnered and Single. The most common status is Partnered, occurring 107 times.

Income Range: Three income ranges are identified: Low, Medium, and High. The Medium income range is the most prevalent, appearing 137 times.

Detect Outliers

```
continuous_var = ['Age', 'Income', 'Usage', 'Fitness', 'Miles']
arr = {'5th percentile': 5, '25th percentile or Q1': 25, '50th percentile or Q2': 50, '75th percentile or Q3': 75,
       '95th percentile': 95}
for key, value in arr.items():
  for var in continuous_var:
```

```
    print(f'{var} -> {key} : {np.percentile(df[var], value):.2f}')
```

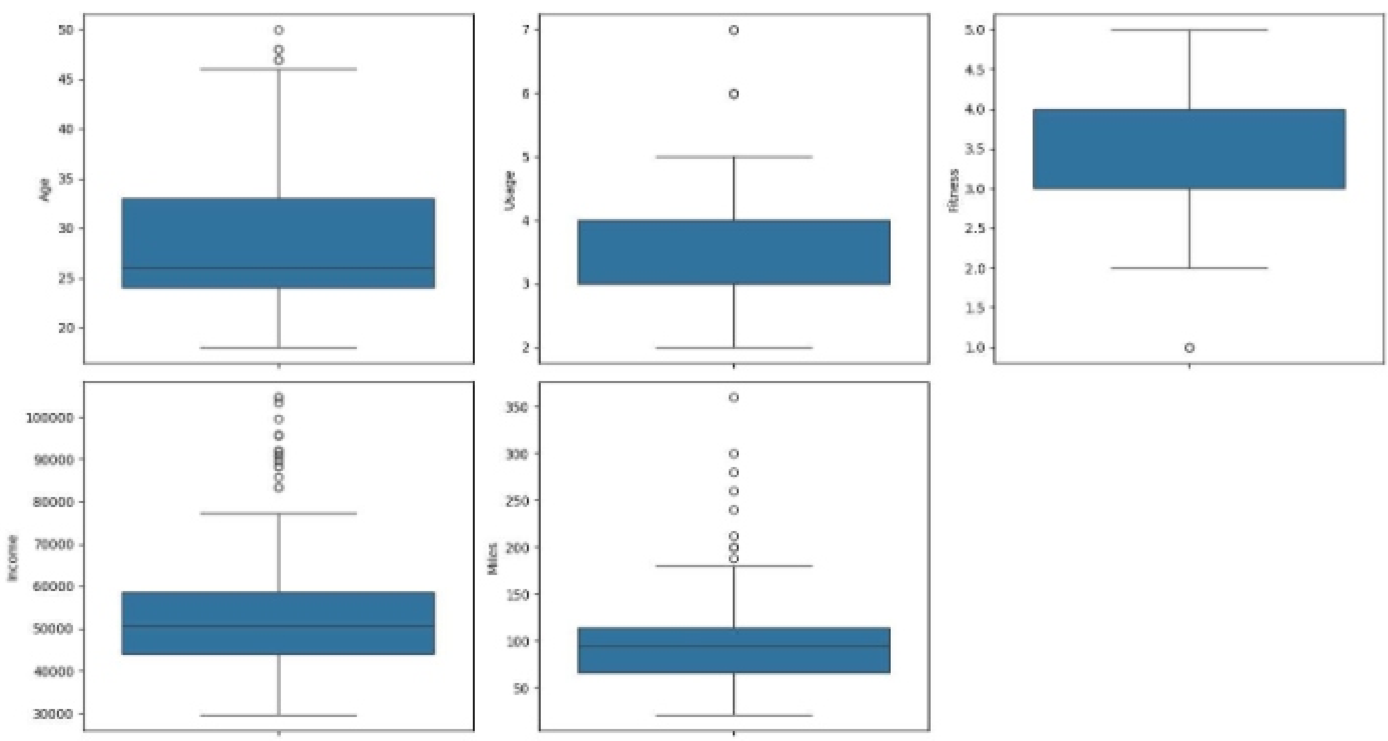
```
Age -> 5th percentile : 20.00
Income -> 5th percentile : 34053.15
Usage -> 5th percentile : 2.00
Fitness -> 5th percentile : 2.00
Miles -> 5th percentile : 47.00
Age -> 25th percentile or Q1 : 24.00
Income -> 25th percentile or Q1 : 44058.75
Usage -> 25th percentile or Q1 : 3.00
Fitness -> 25th percentile or Q1 : 3.00
Miles -> 25th percentile or Q1 : 66.00
Age -> 50th percentile or Q2 : 26.00
Income -> 50th percentile or Q2 : 50596.50
Usage -> 50th percentile or Q2 : 3.00
Fitness -> 50th percentile or Q2 : 3.00
Miles -> 50th percentile or Q2 : 94.00
Age -> 75th percentile or Q3 : 33.00
Income -> 75th percentile or Q3 : 58668.00
Usage -> 75th percentile or Q3 : 4.00
Fitness -> 75th percentile or Q3 : 4.00
Miles -> 75th percentile or Q3 : 114.75
Age -> 95th percentile : 43.05
Income -> 95th percentile : 90948.25
Usage -> 95th percentile : 5.05
Fitness -> 95th percentile : 5.00
Miles -> 95th percentile : 200.00
```

```
for var in continuous_var:
    # Calculate the IQR for the variable
    Q1 = np.percentile(df[var], arr['25th percentile or Q1'])
    Q3 = np.percentile(df[var], arr['75th percentile or Q3'])
    percentile_95 = np.percentile(df[var], arr['95th percentile'])
    IQR = Q3 - Q1

    # Define the outlier thresholds
    lower_threshold = Q1 - 1.5 * IQR
    upper_threshold = Q3 + 1.5 * IQR

    # Find the outliers for the variable
    outliers = df[(df[var] < lower_threshold) | (df[var] > upper_threshold)]

    # Calculate the percentage of outliers
    outlier_percentage = round(len(outliers) / len(df[var]) * 100, 2 )

    # Output the percentage of outliers
    print(f"IQR for {var}: {IQR}")
    print(f"Outlier above this Q3 {var} : {upper_threshold}")
    print(f"Percentage of outliers for {var}: {outlier_percentage}% \n")
```

```
IQR for Age: 9.0
Outlier above this Q3 Age : 46.5
Percentage of outliers for Age: 2.78%

IQR for Income: 14609.25
Outlier above this Q3 Income : 80581.875
Percentage of outliers for Income: 10.56%

IQR for Usage: 1.0
Outlier above this Q3 Usage : 5.5
Percentage of outliers for Usage: 5.0%

IQR for Fitness: 1.0
Outlier above this Q3 Fitness : 5.5
Percentage of outliers for Fitness: 1.11%

IQR for Miles: 48.75
Outlier above this Q3 Miles : 187.875
Percentage of outliers for Miles: 7.22%
```

2 a. Finding the outliers for every continuous variable in the given dataset:

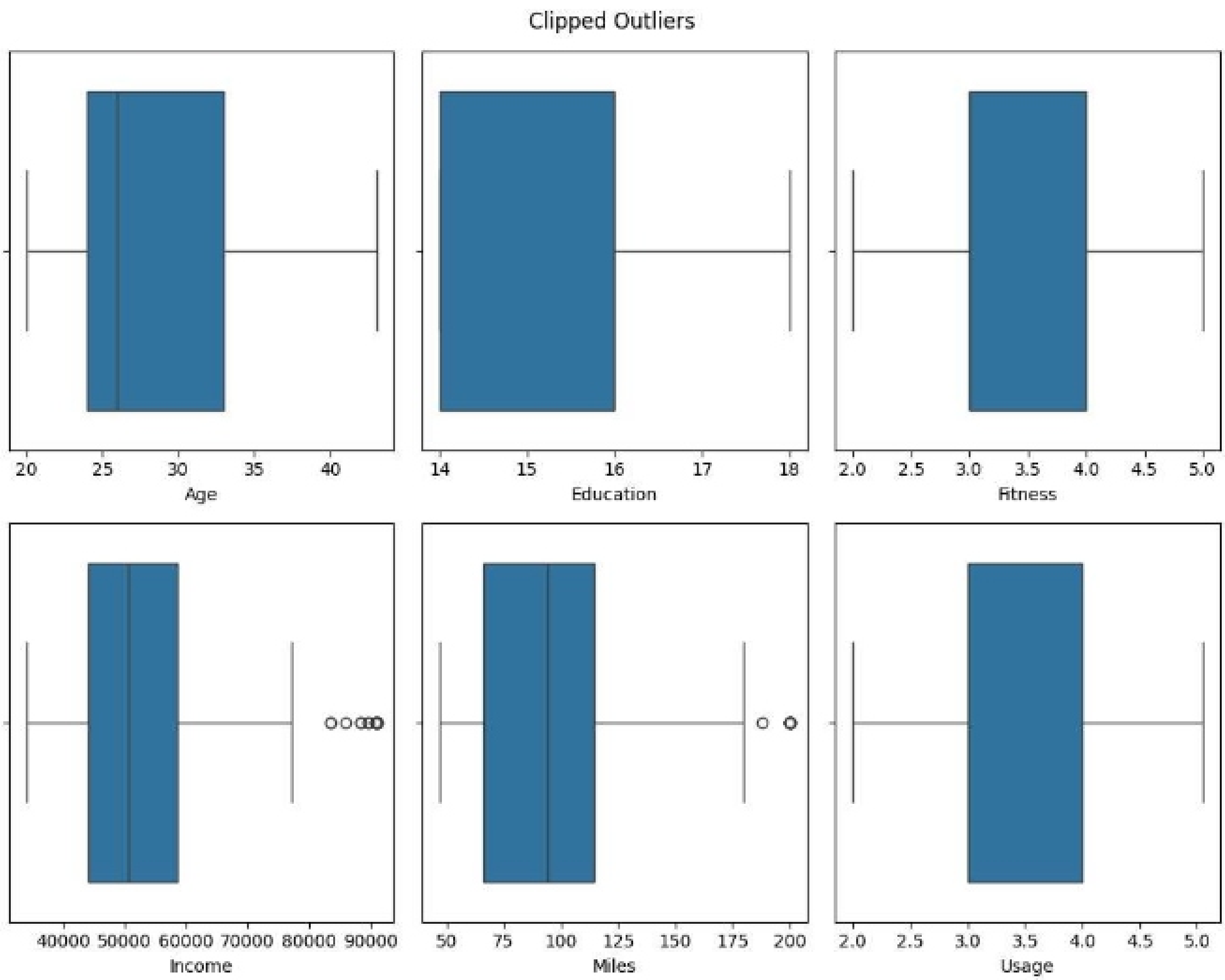
```
plt.figure(figsize=(15,8))

# Box Plot for Age
plt.subplot(2,3,1)
sns.boxplot(df['Age'])

# Box Plot for Usage
plt.subplot(2,3,2)
sns.boxplot(df['Usage'])

#Box Plot for Fitness
plt.subplot(2,3,3)
sns.boxplot(df['Fitness'])

#Box Plot for Income
plt.subplot(2,3,4)
sns.boxplot(df['Income'])

#Box Plot for Miles
plt.subplot(2,3,5)
sns.boxplot(df['Miles'])

plt.tight_layout()
plt.show()
```



b. Removing/clipping the data between the 5 percentile and 95 percentile

```
clipped_age = np.clip(df['Age'], np.percentile(df['Age'], 5), np.percentile(df['Age'], 95))
clipped_education = np.clip(df['Education'], np.percentile(df['Education'], 5), np.percentile(df['Education'], 95))
clipped_income = np.clip(df['Income'], np.percentile(df['Income'], 5), np.percentile(df['Income'], 95))
clipped_usage = np.clip(df['Usage'], np.percentile(df['Usage'], 5), np.percentile(df['Usage'], 95))
clipped_miles = np.clip(df['Miles'], np.percentile(df['Miles'], 5), np.percentile(df['Miles'], 95))
clipped_fitness = np.clip(df['Fitness'], np.percentile(df['Fitness'], 5), np.percentile(df['Fitness'], 95))

fig,ax =plt.subplots(2,3,figsize=(10,8))
plt.suptitle("Clipped Outliers")

plt.subplot(2,3,1)
sns.boxplot(data=df,x=clipped_age)

plt.subplot(2,3,2)
sns.boxplot(data=df,x=clipped_education)
```

```
plt.subplot(2,3,3)
sns.boxplot(data=df,x=clipped_fitness)

plt.subplot(2,3,4)
sns.boxplot(df,x=clipped_income)

plt.subplot(2,3,5)
sns.boxplot(data=df,x=clipped_miles)

plt.subplot(2,3,6)
sns.boxplot(data=df,x=clipped_usage)

plt.tight_layout()
plt.show()
```



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

a. Using the count plot finding the relationship between categorical variables and output variables.

b.Finding the relationship between the continuous variables and the output variable in the data.

Graphical Analysis:-

```
#Categorical Columns = Product,Marital_Status,Gender
df.groupby('Product')['MaritalStatus'].value_counts()
```

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

```
df.groupby('MaritalStatus')['Product'].value_counts()
```

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

```
df.groupby('Product')['Gender'].value_counts()
```

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

```
df.groupby('Gender')['Product'].value_counts()
```

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

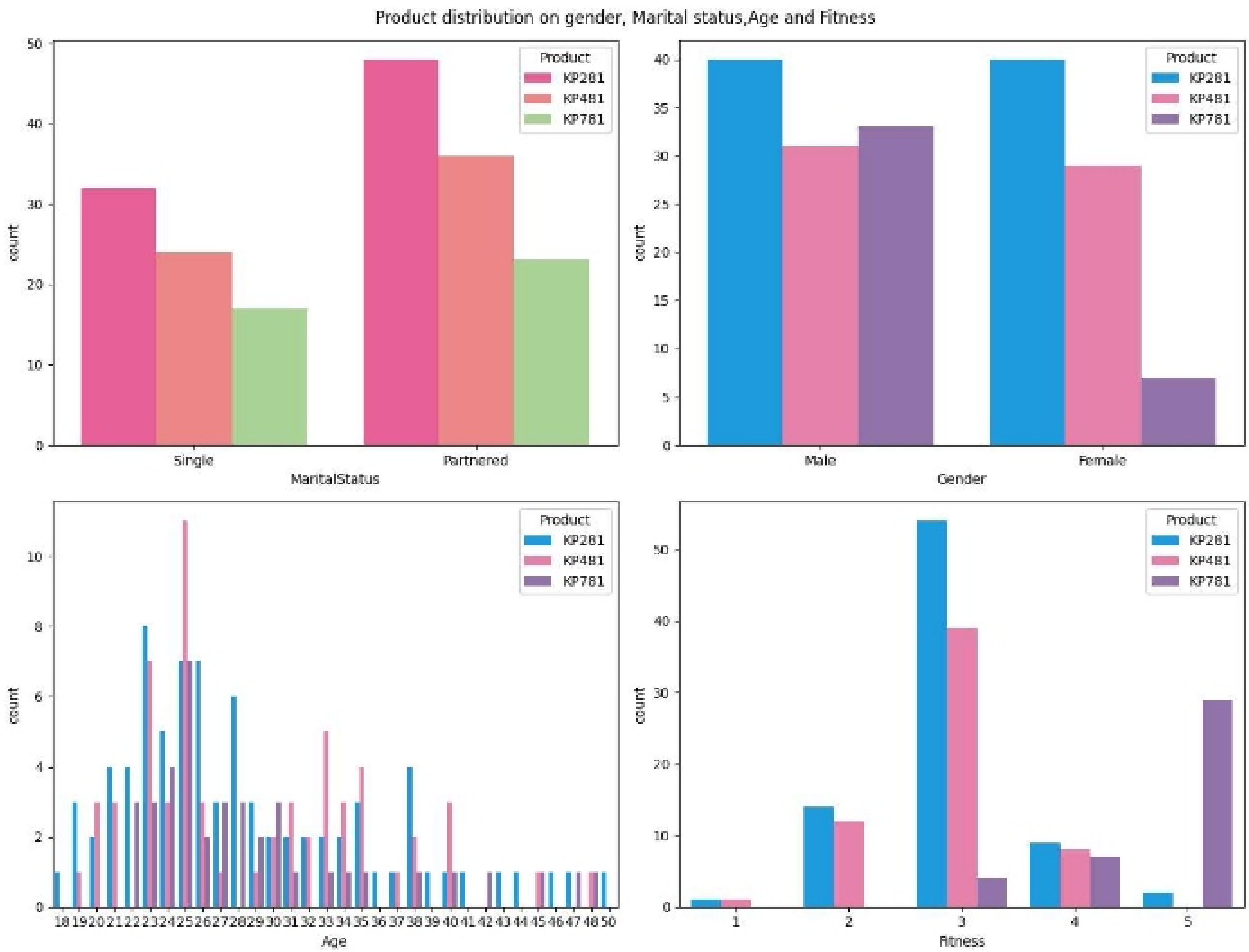
```
df.groupby('Product')['Age_Group'].value_counts()
```

```
Product  Age_Group
KP281    18-25        34
         26-35        32
         36-50        14
KP481    18-25        28
         26-35        24
         36-50         8
KP781    18-25        17
         26-35        17
         36-50         6
Name: count, dtype: int64
```

```
df.groupby('Age')['Product'].value_counts()
```

```
Age  Product
18   KP281      1
19   KP281      3
     KP481      1
20   KP481      3
     KP281      2
               ..
47   KP781      1
     KP281      1
48   KP481      1
     KP781      1
50   KP281      1
Name: count, Length: 68, dtype: int64
```

```
plt.figure(figsize =(13,10))
plt.suptitle('Product distribution on gender, Marital status,Age and Fitness')

plt.subplot(2,2,1)
sns.countplot(data = df, x='MaritalStatus', hue='Product', palette=['#FF4B91', '#FF7676', '#A8DF8E'])

plt.subplot(2,2,2)
sns.countplot(data = df, x='Gender', hue='Product', palette=['#00A9FF', '#F875AA', '#916DB3'])

plt.subplot(2,2,3)
sns.countplot(data = df, x='Age', hue='Product', palette=['#00A9FF', '#F875AA', '#916DB3'])


plt.subplot(2,2,4)
sns.countplot(data = df, x='Fitness', hue='Product', palette=['#00A9FF', '#F875AA', '#916DB3'])

plt.tight_layout()
plt.show()
```




```
df["Product"].value_counts()
```

```
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
```

```
df["Gender"].value_counts()
```

```
Gender
Male      104
```

```
Female     76
Name: count, dtype: int64
```

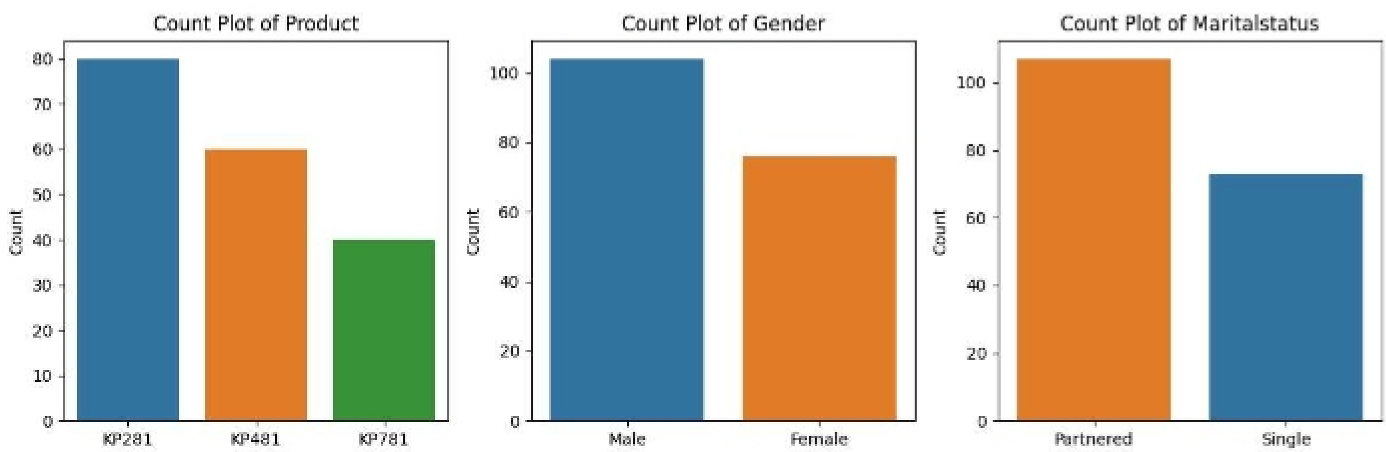
```
df["MaritalStatus"].value_counts()
```

```
MaritalStatus
Partnered    107
Single        73
Name: count, dtype: int64
```

```
#Non-graphical analysis: Value counts for each categorical variable
categorical_columns= ['Product', 'Gender', 'MaritalStatus']
for column in categorical_columns:
    print(f"{df[column].value_counts()}\n")
```

```
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64

Gender
Male      104
Female     76
Name: count, dtype: int64

MaritalStatus
Partnered    107
Single        73
Name: count, dtype: int64
```

```
 # Countplots for each categorical variable
fig,axes = plt.subplots(1, 3,figsize=(12, 4))
for i, column in enumerate(categorical_columns):
    order = df[column].value_counts().index[:10]
    sns.countplot(x=column, data=df, order=order, ax=axes[i], hue=column)
    axes[i].set_title(f'Count Plot of {column.capitalize()}')
    axes[i].set_xlabel('')
    axes[i].set_ylabel('Count')
    axes[i].tick_params(axis='y',labelsize=10)
    axes[i].tick_params(axis='x',labelsize=10)
plt.tight_layout()
plt.show()
```



Checking the unique values for columns

```
for i in df.columns:
    print(f'Unique Values in {i} column are :-\n {df[i].unique()}\n')
    print('.'*80)
```

```
Unique Values in Product column are :-
 ['KP281' 'KP481' 'KP781']

................................................................................
Unique Values in Age column are :-
```

```
 [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]

................................................................................
Unique Values in Gender column are :-
 ['Male' 'Female']

................................................................................
Unique Values in Education column are :-
 [14 15 12 13 16 18 20 21]

................................................................................
Unique Values in MaritalStatus column are :-
 ['Single' 'Partnered']

................................................................................
Unique Values in Usage column are :-
 [3 2 4 5 6 7]

................................................................................
Unique Values in Fitness column are :-
 [4 3 2 1 5]

................................................................................
Unique Values in Income column are :-
 [ 29562  31836  30699  32973  35247  37521  36384  38658  40932  34110
   39795  42069  44343  45480  46617  48891  53439  43206  52302  51165
   50028  54576  68220  55713  60261  67083  56850  59124  61398  57987
   64809  47754  65220  62535  48658  54781  48556  58516  53536  61006
   57271  52291  49801  62251  64741  70966  75946  74701  69721  83416
   88396  90886  92131  77191  52290  85906 103336  99601  89641  95866
 104581  95508]

................................................................................
Unique Values in Miles column are :-
 [112  75  66  85  47 141 103  94 113  38 188  56 132 169  64  53 106  95
 212  42 127  74 170  21 120 200 140 100  80 160 180 240 150 300 280 260
 360]

................................................................................
Unique Values in Income_Range column are :-
 ['Low' 'Medium' 'High']

................................................................................
Unique Values in Age_Group column are :-
 ['18-25', '26-35', '36-50']
Categories (3, object): ['18-25' < '26-35' < '36-50']

................................................................................
```

Checking the number of unique values for columns

```
for i in df.columns:
    print('Number of Unique Values in',i,'column :', df[i].nunique())
    print('-'*70)
```

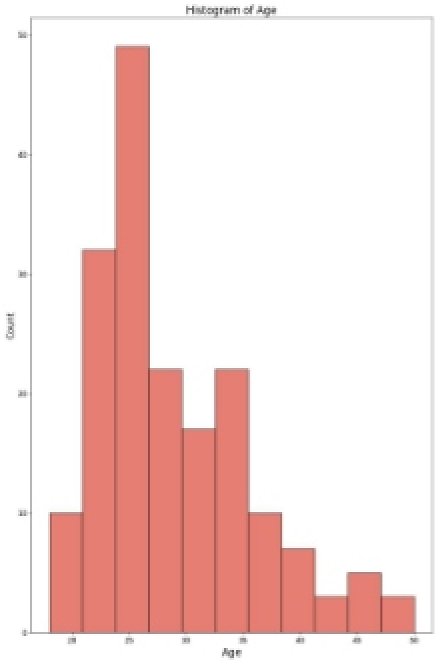
```
Number of Unique Values in Product column : 3
----------------------------------------------------------------------
Number of Unique Values in Age column : 32
----------------------------------------------------------------------
Number of Unique Values in Gender column : 2
----------------------------------------------------------------------
Number of Unique Values in Education column : 8
----------------------------------------------------------------------
Number of Unique Values in MaritalStatus column : 2
----------------------------------------------------------------------
Number of Unique Values in Usage column : 6
----------------------------------------------------------------------
Number of Unique Values in Fitness column : 5
----------------------------------------------------------------------
Number of Unique Values in Income column : 62
----------------------------------------------------------------------
Number of Unique Values in Miles column : 37
----------------------------------------------------------------------
Number of Unique Values in Income_Range column : 3
----------------------------------------------------------------------
Number of Unique Values in Age_Group column : 3
----------------------------------------------------------------------
```

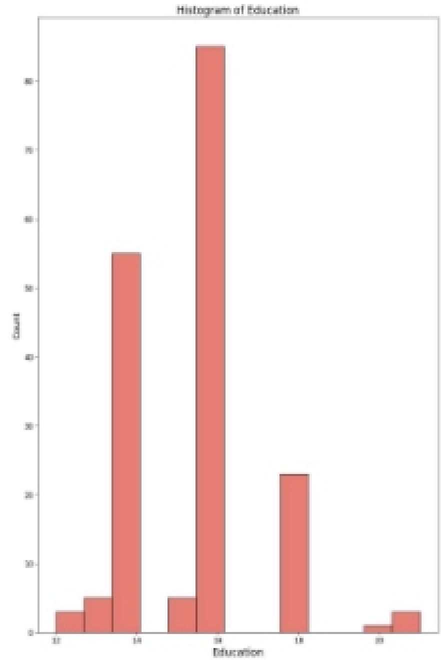
```
continuous_var = ['Age', 'Education', 'Income', 'Usage', 'Fitness', 'Miles']
for column in continuous_var:
    print(f"{column}\n{df[column].value_counts().sort_values(ascending=False)}")
```

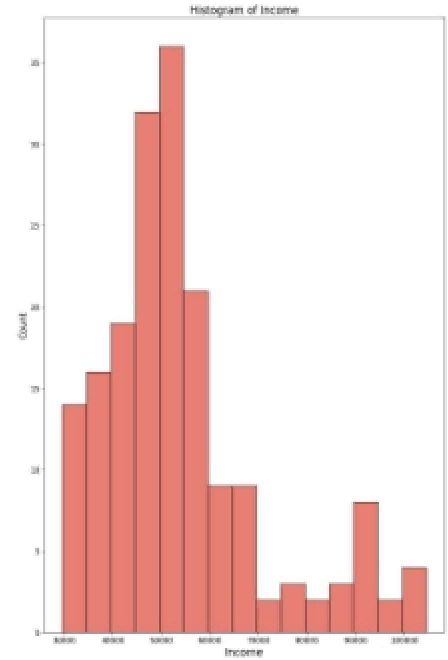
```
Name: count, Length: 62, dtype: int64
Usage
Usage
3    69
4    52
2    33
5    17
6     7
7     2
Name: count, dtype: int64
Fitness
Fitness
3    97
5    31
2    26
4    24
1     2
Name: count, dtype: int64
Miles
Miles
85     27
95     12
66     10
75     10
47      9
106     9
94      8
113     8
53      7
100     7
56      6
64      6
180     6
200     6
127     5
160     5
42      4
150     4
120     3
103     3
38      3
170     3
74      3
132     2
141     2
280     1
260     1
300     1
240     1
112     1
212     1
80      1
140     1
21      1
169     1
188     1
360     1
Name: count, dtype: int64
```

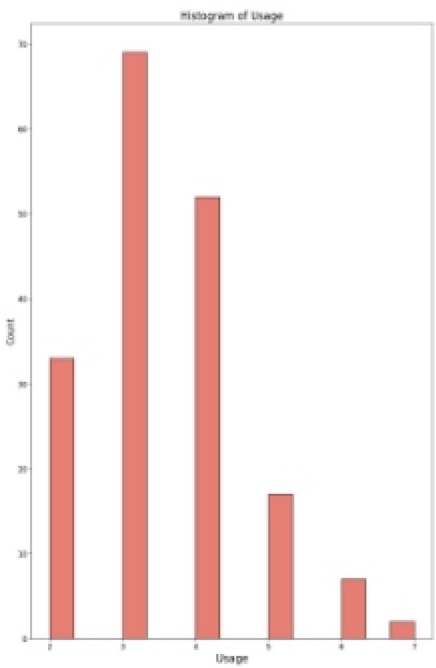
For Graphical Analysis

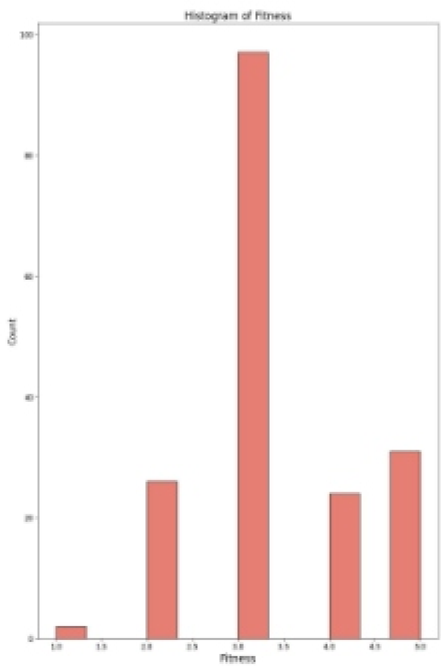
```
# Hisplot for Continuous Variable
sns.set_palette('Spectral')
fig, axes = plt.subplots(2,3, figsize=(40, 40))
axes = axes.flatten()

for i, column in enumerate(continuous_var):
    sns.histplot(df[column], ax=axes[i])
    axes[i].set_title(f'Histogram of {column.capitalize()}', fontsize= 17)
    axes[i].set_ylabel('Count', fontsize=15)
    axes[i].set_xlabel(column.capitalize(), fontsize=17 )
    axes[i].tick_params(axis='both', labelsize=12)
```

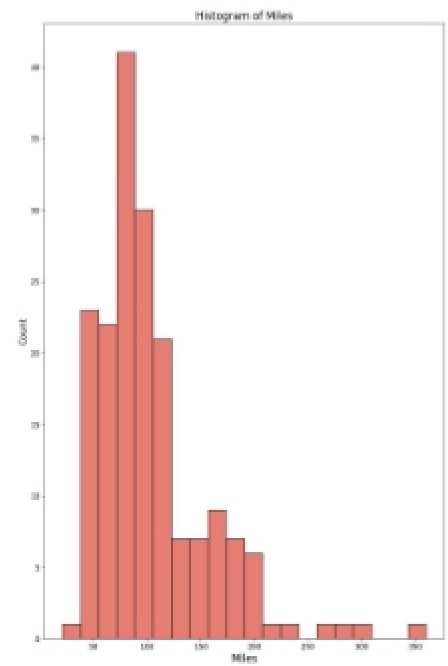












Insights :

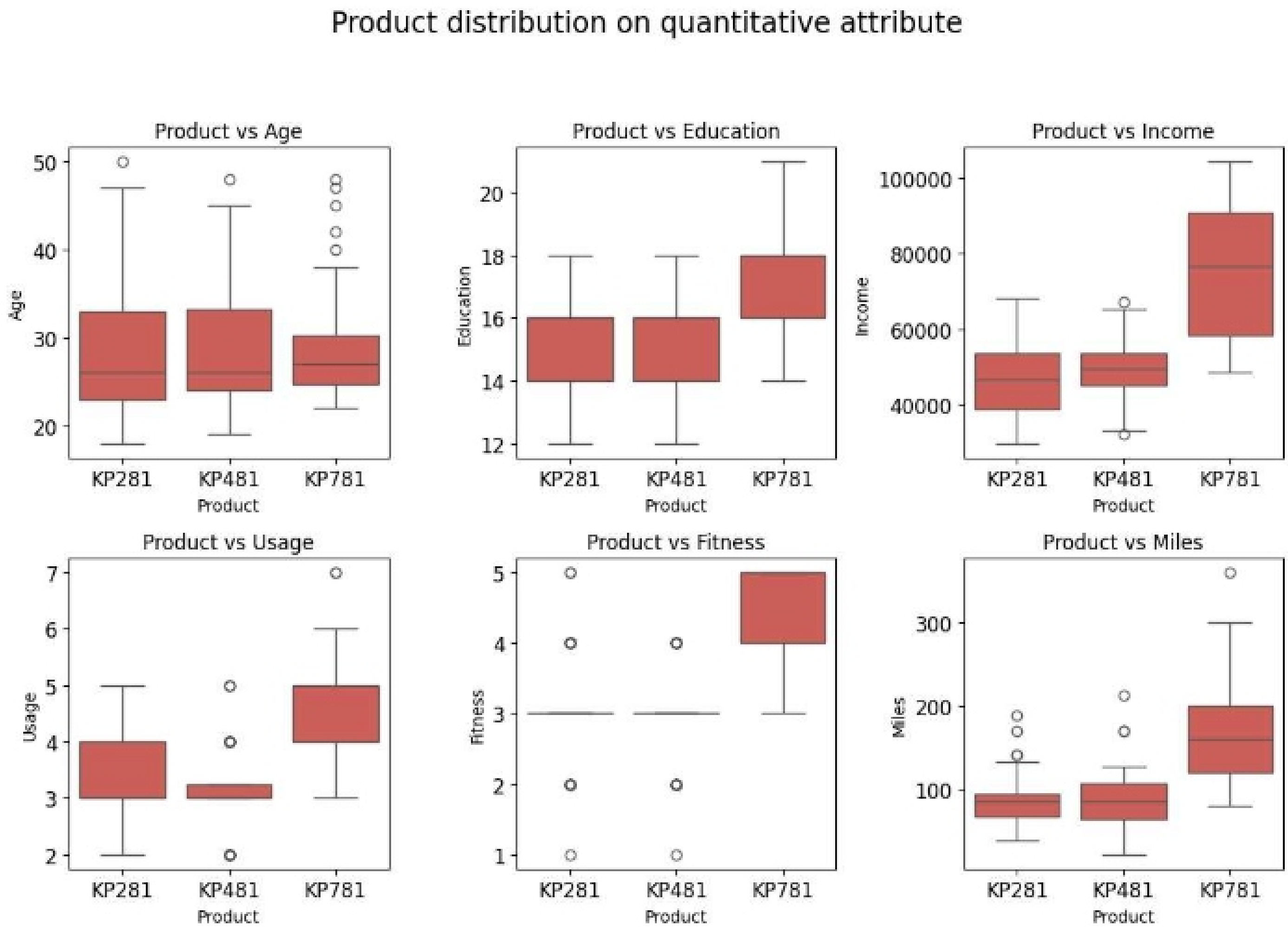
Gender Distribution:

There are more male customers compared to female customers.

Partnership Status: Partnered customers are more prevalent.

Product Preference and Fitness Rating: Product KP281 is the most frequently purchased by customers who rate their fitness level as 3, indicating they are moderate-fitness individuals.

```
# Product distribution on quantitative attribute
fig,axes = plt.subplots(2,3,figsize=(11,8))
plt.suptitle('Product distribution on quantitative attribute\n\n', fontsize=17)
axes = axes.flatten()

for i, column in enumerate(continuous_var):
    sns.boxplot(y=df[column], x =df['Product'],ax=axes[i])
    axes[i].set_title(f'Product vs {column.capitalize()}')
    axes[i].tick_params(axis='y',labelsize=12)
    axes[i].tick_params(axis='x',labelsize=12)
plt.tight_layout()
plt.show()
```



Insights & Observations:

Product vs Age:

The median age of customers purchasing products KP281 and KP481 is the same. Customers aged between 25 and 30 are more likely to buy the KP781 product.

Product vs Education:

Customers with more than 16 years of education are more likely to purchase the KP781 product. Customers with 16 or fewer years of education have equal chances of purchasing KP281 or KP481.

Product vs Usage:

Customers planning to use the treadmill more than 4 times a week are more likely to purchase the KP781 product.

Product vs Fitness:

Customers with a fitness level of 3 or higher have a higher chance of purchasing the KP781 product.

Product vs Income:

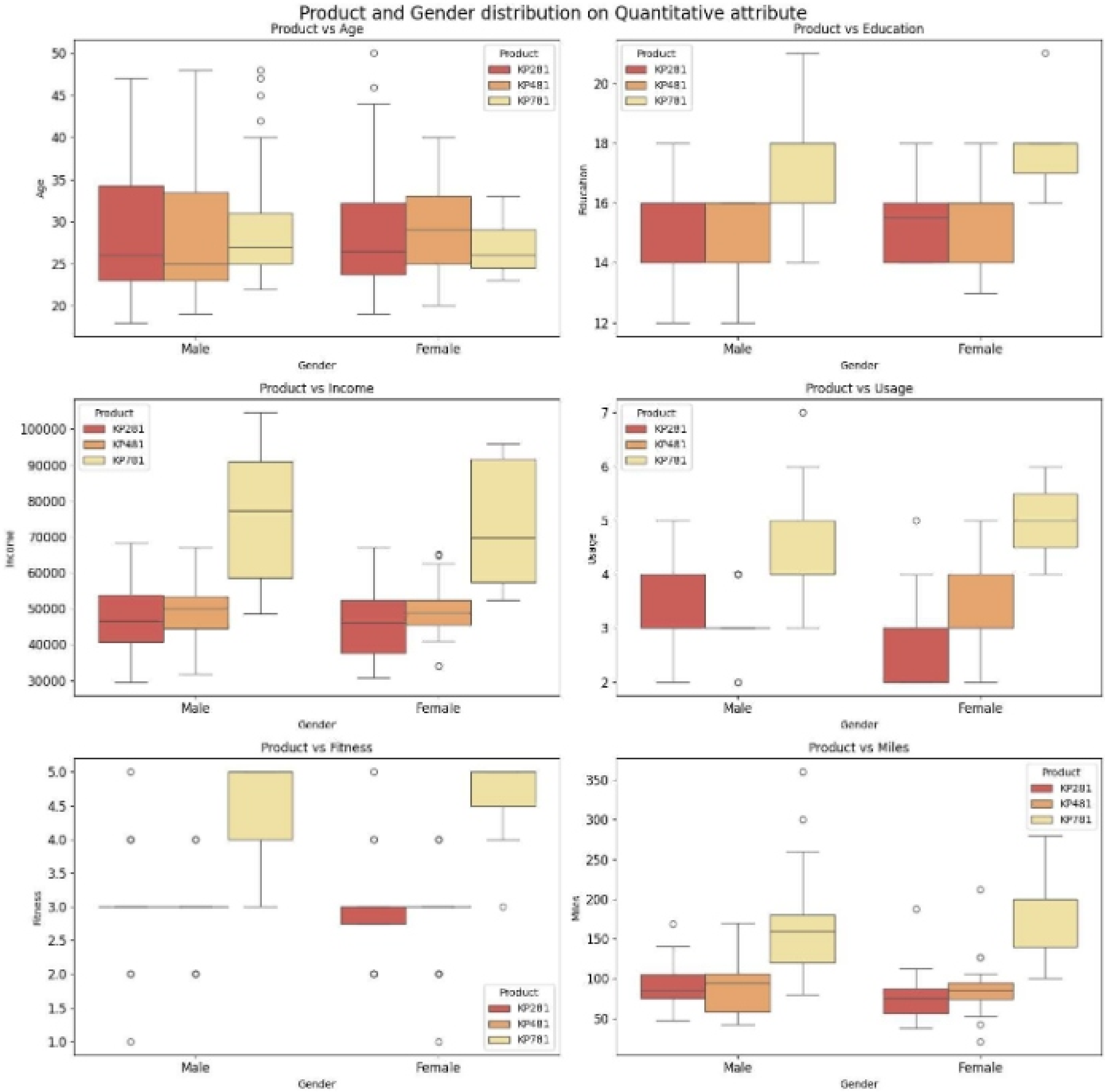
Customers with an income of $60,000 or more are more likely to purchase the KP781 product.

Product vs Miles:

Customers expecting to walk or run more than 120 miles per week are more likely to buy the KP781 product.

Multivariate Analysis

```
fig,axes = plt.subplots(3,2,figsize=(15,15))
plt.suptitle('Product and Gender distribution on Quantitative attribute', fontsize=17)
axes = axes.flatten()

for i, column in enumerate(continuous_var):
    sns.boxplot(y=df[column], x =df['Gender'],ax=axes[i], hue=df['Product'])
    axes[i].set_title(f'Product vs {column.capitalize()}')
    axes[i].tick_params(axis='y',labelsize=12)
    axes[i].tick_params(axis='x',labelsize=12)
plt.tight_layout()
plt.show()
```



Insights & Observations:-

Product vs Gender and Usage:

Female customers who plan to use the treadmill 3-4 times a week are more likely to buy the KP481 product.

4.Representing the Probability:

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

Hint: Using the pandas crosstab to find the marginal probability of each product.

```
marginal_probability = df['Product'].value_counts() / len(df['Product'])*100
round(marginal_probability,2)
```

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

Insights & Observations

Based on the data:

Product Popularity:

KP281 is the most popular treadmill, preferred by approximately 44.44% of customers. KP481 is the second most popular, with 33.33% of customers preferring it. KP781 is chosen by 22.22% of customers.

Usage Frequency:

Customers who plan to use the treadmill more than 4 times a week are more inclined to choose the KP781.

Fitness Level:

Customers with a higher fitness level (3 or above) are more likely to select the KP781.

Income Level:

A higher income (equal to or greater than $60,000) is a significant factor for customers choosing the KP781 over other options.

Expected Usage (Miles):

Customers expecting to walk or run more than 120 miles per week tend to prefer the KP781.

These insights can inform marketing and product positioning strategies by identifying potential target segments for each treadmill product.

Find the probability that the customer buys a product based on each column

Finding the probability based on previous crosstab values

```
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Income_Range	Age_Group
0	KP281	18	Male	14	Single	3	4	29562	112	Low	18-25
1	KP281	19	Male	15	Single	2	3	31836	75	Medium	18-25
2	KP281	19	Female	14	Partnered	4	3	30699	66	Medium	18-25
3	KP281	19	Male	12	Single	3	3	32973	85	Medium	18-25
4	KP281	20	Male	13	Partnered	4	2	35247	47	Medium	18-25

Next steps: Generate code with df View recommended plots

```
#binning the age values into categories

age_bin = [17,25,35,45,float('inf')]
bin_labels = ['17-25', '25-35', '35-45', '45+']
df['age_group'] = pd.cut(df['Age'],bins = age_bin ,labels = bin_labels)

# binning the income values into categories
income_bin = [0,40000,60000,80000,float('inf')]
income_bin_labels = ['Low Income','Moderate Income','High Income','Very High Income']

df['Income_Range'] = pd.cut(df['Income'],bins = income_bin ,labels = income_bin_labels)

# binning the miles values into categories
miles_range = [0,70,100,200,float('inf')]
miles_bin_label = ['Light', 'Moderate', 'Active', 'Fitness Enthusiast ']
df['miles_group'] = pd.cut(df['Miles'],bins = miles_range,labels = miles_bin_label)


df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Income_Range	Age_Group	age_group	miles_group
0	KP281	18	Male	14	Single	3	4	29562	112	Low Income	18-25	17-25	Active
1	KP281	19	Male	15	Single	2	3	31836	75	Low Income	18-25	17-25	Moderate
2	KP281	19	Female	14	Partnered	4	3	30699	66	Low Income	18-25	17-25	Light
3	KP281	19	Male	12	Single	3	3	32973	85	Low Income	18-25	17-25	Moderate
4	KP281	20	Male	13	Partnered	4	2	35247	47	Low Income	18-25	17-25	Light

Next steps: Generate code with df View recommended plots

```
# Calculate the probability of buying a product based on each column
probability_of_buy = {}

for column in df.columns:
    if column not in ( 'Product', 'Age', 'Income', 'Miles'):
        probability_of_buy[column] = pd.crosstab(index=df['Product'], columns=df[column], margins =True, normalize=True).round(2)

# Display the probabilities
for column, prob in probability_of_buy.items():
    print(f"\nProbability of buying a product based on {column}:")
    print('-' * 70)
    print(f'{prob}\n')
```

```
----------------------------------------------------------------------
Usage       2     3     4     5     6     7   All
Product
KP281    0.11  0.21  0.12  0.01  0.00  0.00  0.44
KP481    0.08  0.17  0.07  0.02  0.00  0.00  0.33
KP781    0.00  0.01  0.10  0.07  0.04  0.01  0.22
All      0.18  0.38  0.29  0.09  0.04  0.01  1.00


Probability of buying a product based on Fitness:
----------------------------------------------------------------------
Fitness     1     2     3     4     5   All
Product
KP281    0.01  0.08  0.30  0.05  0.01  0.44
KP481    0.01  0.07  0.22  0.04  0.00  0.33
KP781    0.00  0.00  0.02  0.04  0.16  0.22
All      0.01  0.14  0.54  0.13  0.17  1.00


Probability of buying a product based on Income_Range:
----------------------------------------------------------------------
Income_Range  Low Income  Moderate Income  High Income  Very High Income   All
```

```
KP481         0.10      0.14    0.08                  0.01  0.33
KP781         0.00      0.04    0.15                  0.03  0.22
All           0.26      0.38    0.33                  0.03  1.00
```

Insights & Observations:

Based on the probabilities, we can observe the following insights:

Gender: The highest number of customers are male compared to female customers.

Education:

KP281:

Customers with education level 14 (some college education) have the highest probability of purchasing the KP281 treadmill. Customers with education levels 16 (graduate degree) and 18 (professional degree) also show a relatively high probability of purchasing KP281.

KP481: Customers with education level 14 (some college education) have the highest probability of purchasing the KP481 treadmill. Customers with education levels 16 (graduate degree) and 18 (professional degree) also show a relatively high probability of purchasing KP481.

KP781: Customers with education level 18 (professional degree) have the highest probability of purchasing the KP781 treadmill. Customers with education levels 15 (college degree) and 16 (graduate degree) also show a relatively high probability of purchasing KP781.

Overall, customers with higher education levels (such as graduate degrees and professional degrees) tend to have a higher probability of purchasing all three treadmill products. However, customers with some college education (education level 14) also show a significant probability for both KP281 and KP481.

Marital Status:

Partnered customers have a higher probability of purchasing all three treadmill products compared to single customers.

Usage:

Customers who plan to use the treadmill 3-4 times a week have a higher probability of purchasing the KP281 treadmill. Those who plan to use it 5+ times a week have a higher probability of purchasing the KP781 treadmill.

Fitness:

Customers with higher fitness levels (3-5) have a higher probability of purchasing the KP281 treadmill. Customers with lower fitness levels (1-2) have a higher probability of purchasing the KP781 treadmill.

Lifestyle:

Light Activity (0 to 70 miles per week): Overall probability of purchasing any treadmill: 26% KP281: 16% KP481: 10% KP781: 0% Moderate Activity (71 to 100 miles per week): Overall probability of purchasing any treadmill: 38% KP281: 19% KP481: 14% KP781: 4% Active Lifestyle (100 to 200 miles per week): Overall probability of purchasing any treadmill: 33% KP281: 10% KP481: 8% KP781: 15% Fitness Enthusiasts (more than 200 miles per week): Overall probability of purchasing any treadmill: 3%

Age Group:

Customers in the age group 17-25 have a higher probability of purchasing the KP281 treadmill. Other age groups show similar probabilities for all three products.

Income Range:

Moderate and high-income customers have a higher probability of purchasing the KP281 and KP481 treadmills. Low-income customers have a higher probability of purchasing the KP781 treadmill. Very high-income customers have a higher probability of purchasing the KP781 and KP481 treadmills.

Miles Group:

Customers who categorize themselves as fitness enthusiasts have a higher probability of purchasing the KP781 treadmill. Other miles groups show similar probabilities for all three products. These insights can be useful for targeted marketing strategies, product development, and pricing decisions.

Finding 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)

```
def p_prod_given_gender(gender, print_marginal=False):
    if gender != "Female" and gender != "Male":
        return "Invalid Gender value."
    df1 = pd.crosstab(df['Gender'], columns=[df['Product']])
    p_781 = df1['KP781'][gender] / df1.loc[gender].sum()
    p_481 = df1['KP481'][gender] / df1.loc[gender].sum()
    p_281 = df1['KP281'][gender] / df1.loc[gender].sum()

    if print_marginal:
        print(f"P(Male): {df1.loc['Male'].sum()/len(df):.2f}")
        print(f"P(Female): {df1.loc['Female'].sum()/len(df):.2f}\n")
    print(f"P(KP781/{gender}): {p_781:.2f}")
    print(f"P(KP481/{gender}): {p_481:.2f}")
    print(f"P(KP281/{gender}): {p_281:.2f}\n")

p_prod_given_gender('Male', True)
p_prod_given_gender('Female')
```

```
P(Male): 0.58
P(Female): 0.42

P(KP781/Male): 0.32
P(KP481/Male): 0.30
P(KP281/Male): 0.38

P(KP781/Female): 0.09
P(KP481/Female): 0.38
P(KP281/Female): 0.53
```

INSIGHTS & OBSERVATIONS -

Among male customers, there is a higher probability of purchasing KP281 compared to KP781 or KP481.

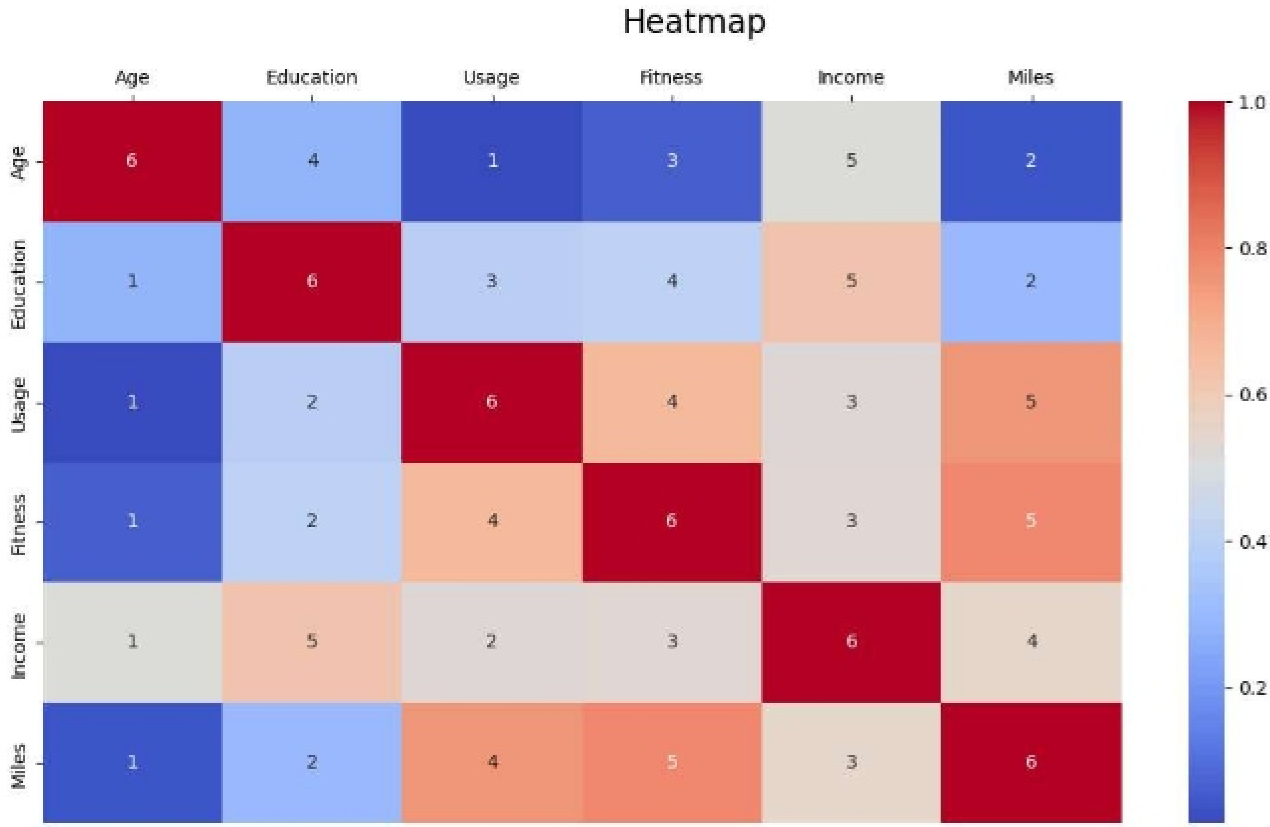
Among female customers, there is a higher probability of purchasing KP281 compared to KP481, but the probability of purchasing KP781 is the lowest.

The conditional probabilities provide insights into the likelihood of customers purchasing specific products based on their gender.

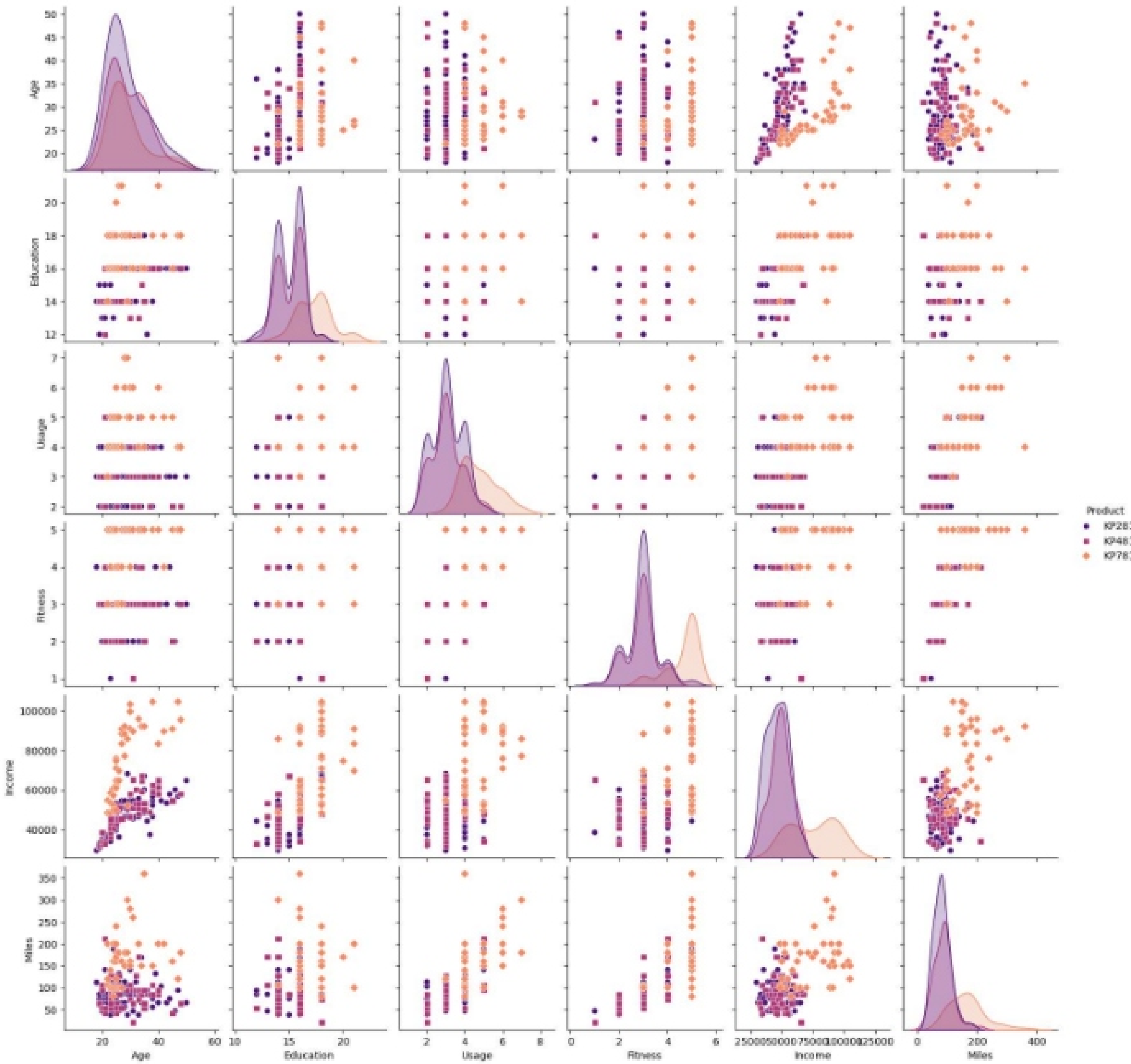
5. Check the correlation among different factors

Check the correlation among different factors

```
correlation_matrix = df.corr(method='pearson', numeric_only = True)

# Display the heatmap of the correlation matrix:
plt.figure(figsize=(13,7))
plt.suptitle('Heatmap', fontsize= 17)
sns.heatmap(correlation_matrix, annot=correlation_matrix.rank(axis="columns"), cmap='coolwarm').xaxis.tick_top()
plt.show()
```




```
# Display the Pairplot of the correlation matrix:
sns.pairplot(df, hue ='Product', palette= 'magma', markers=["o", "s", "D"])
plt.show()
```



Insights :

Age and Income: There is a positive correlation between age and income, indicating that as age increases, income tends to rise as well.

Education and Income: Higher levels of education are associated with higher income levels, which is a well-established trend.

Education and Fitness: Education also correlates with fitness rating and treadmill usage, suggesting that more educated individuals tend to have better fitness levels and use fitness equipment more regularly.

Treadmill Usage and Fitness: There's a strong positive correlation between treadmill usage and fitness level. This means that frequent use of treadmills is linked to higher fitness levels and the ability to cover more distance.

Income and Purchasing Patterns: Income influences education levels and the preference for treadmills with greater mileage capacity. Higher-income individuals are more likely to invest in higher-quality equipment.

Limited Influence of Age: Age shows weaker correlations with other variables compared to income and education. It suggests that age alone may not significantly impact income, fitness levels, or treadmill usage patterns.

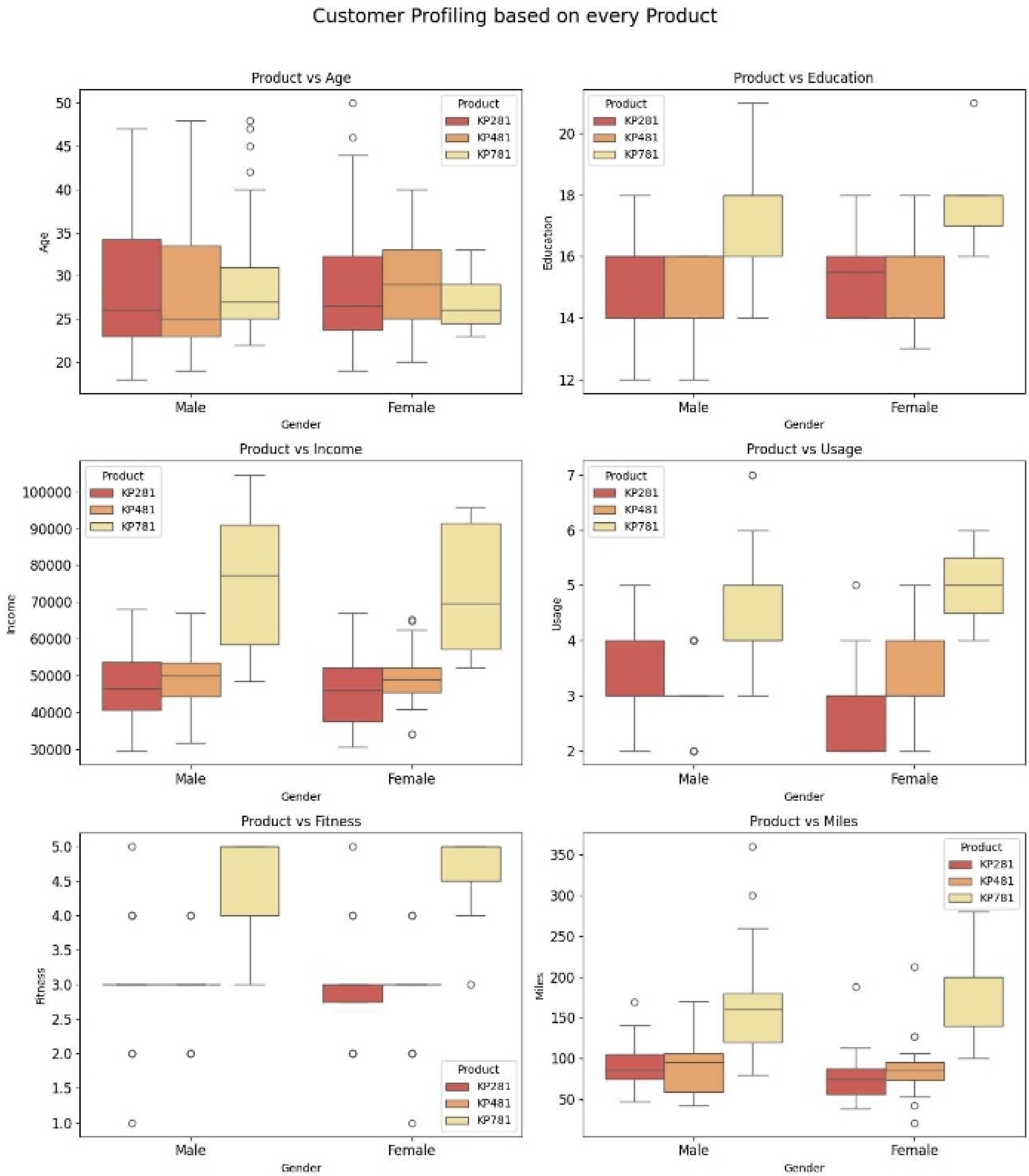
Role of Education: Education plays a significant role across various factors, positively correlating with income and moderately with fitness and treadmill usage. This indicates a propensity among higher-educated individuals for fitness engagement and higher income.

These insights underscore the importance of income, education, and regular treadmill usage in shaping fitness levels and consumer behavior related to fitness equipment.

6. Customer profiling and recommendation :

Making customer profilings for each and every product.

```
fig,axes = plt.subplots(3,2,figsize=(13,15))
plt.suptitle('Customer Profiling based on every Product\n\n', fontsize=17)
axes = axes.flatten()

for i, column in enumerate(continuous_var):
    sns.boxplot(y=df[column], x =df['Gender'],ax=axes[i], hue=df['Product'])
    axes[i].set_title(f'Product vs {column.capitalize()}')
    axes[i].tick_params(axis='y',labelsize=12)
    axes[i].tick_params(axis='x',labelsize=12)
plt.tight_layout()
plt.show()
```



Customer Profiling and Recommendations:

6.1 INSIGHTS & OBSERVATIONS - Customer profiling for each product involves creating a detailed description and understanding of the target customers who are likely to purchase that specific product. It helps in identifying the characteristics, preferences, and behaviors of the target audience for effective marketing and sales strategies.

Based on above analysis:-

• Probability of purchase of KP281 = 44% • Probability of purchase of KP481 = 33% • Probability of purchase of KP781 = 22%

6.2 Customer Profile for KP281 Tread mill:

Age of customer mainly between 18 to 35 years with few between 35 to 50 years

Education level of customer 13 years and above

Annual Income of customer below USD 60,000

Weekly Usage - 2 to 4 times

Fitness Scale - 2 to 4

Weekly Running Mileage to 50 miles

6.3 Customer Profile for KP481 Treadmill:

Age of customer mainly between 18 to 35 years with few between 35 to 50 years

Education level of customer 13 years and above

Annual Income of customer between USD 40,000 to USD 80,000

Weekly Usage - 2 to 4 times

Fitness Scale - 2 to 4

Weekly Running Mileage to 200 miles

6.4 Customer Profile f or KP781 Treadmill:

Gender - Male

Age of customer between 18 to 35 years

Education level of customer 15 years and above

Annual Income of customer USD 80,000 and above

Weekly Usage - 4 to 7 times

Fitness Scale - 3 to 5

Weekly Running Mil effectiveness.

Recommendations - Based on the analysis of the provided data, here are some recommendations :

Marketing Strategy: Focus on targeting customers with higher fitness levels by promoting the benefits of using fitness equipment regularly. Emphasize how regular usage can contribute to improving fitness and overall health. Based on the provided analysis, it would be beneficial to focus marketing efforts for KP281 towards females and lower-income customers. This is because the analysis showed a positive correlation between usage and fitness level, indicating that individuals who use fitness equipment more frequently tend to have higher fitness levels. By targeting females, the marketing efforts can communicate the benefits of using the KP281 to achieve their fitness goals. In addition, targeting lower-income customers aligns with the notable association between income and education, suggesting that customers with lower incomes may have pursued less education and may prefer a more affordable treadmill like the KP281.

On the other hand, for the KP781, it is recommended to target higher-income and possibly male customers. The analysis revealed a positive correlation between income and both education and miles covered. This suggests that customers with higher incomes may have pursued more education and might prefer treadmills that offer longer mileage, such as the KP781. By targeting higher-income customers, the marketing efforts can highlight the advanced features, longer mileage, and potentially higher quality of the KP781 to appeal to their preferences and desire for a high-performance treadmill.

Product Development: Consider developing treadmill models that offer longer mileage for customers with higher incomes. This can cater to their preference for treadmills that allow them to cover more distance and potentially attract this customer segment. Use the data on product preferences and conditional probabilities to guide product development. If KP281 is popular among certain groups, consider enhancing its features or affordability for wider appeal. For KP781, explore ways to cater to higher-income customers' fitness needs.

Pricing Strategy: Adjust pricing strategies accordingly based on the income levels of the target customer segment. Higher-income individuals may be willing to pay more for advanced treadmill features and better overall quality.

Education Campaign: Develop educational content to promote the link between education, income, and fitness. Highlight how higher education levels can lead to higher incomes and a greater likelihood of engaging in fitness activities. Show how using treadmills can be a part of an overall active and healthy lifestyle.

Customer Segmentation: Segment the customer base based on their activity lifestyles, income levels, and education levels. This will help tailor marketing messages and product offerings to each segment's specific needs and preferences.

Partnerships: Collaborate with fitness influencers or organizations that target customers with higher fitness levels or higher incomes. This can help to expand brand reach and credibility among the target audience.

Customer Insights: Continuously collect customer feedback and usage data to gain insights into customer preferences, needs, and satisfaction levels. This will enable a more customer-centric approach to product development and marketing efforts.

Continuous Improvement: Regularly review and analyze data to identify any emerging trends or changes in customer behavior. This will allow for timely adjustments to marketing strategies and product offerings, ensuring the company stays aligned with customer needs and preferences.

Overall, these recommendations focus on targeting specific customer segments, aligning product development with customer preferences, and utilizing education and marketing tactics to drive sales and brand loyalty.

To create customer profiles for each product, we can follow these steps:

Defining the product: Clearly identify and describe the specific product for which we want to create customer profiles.

Conduct market research: Gather data and insights about the market, industry, and customer demographics related to the product. This can include conducting surveys, analyzing customer feedback, studying competitors, and researching industry trends.

Identify target audience: Based on the product's features, benefits, and value propositions, define the target audience that is most likely to have a need or desire for the product. Consider demographic factors like age, gender, location, income, profession, and lifestyle.

Evaluate customer characteristics: Understand the psychographic factors of target audience, including their interests, hobbies, values, attitudes, opinions, and buying behaviors. This can be done through interviews, focus groups, or analyzing existing customer data.

Create customer profiles: Compile the information gathered to create detailed customer profiles or buyer personas for each product. Include demographics, psychographics, motivations, challenges, goals, buying habits, and preferred communication channels.

Use customer profiles for marketing: Utilize the customer profiles to tailor marketing messages, content, and channels to effectively reach and engage the target audience. This allows for better product positioning, personalized marketing campaigns, and improved customer acquisition and retention rates.

Aerofit End of the Business Case

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit