Business Case Study: Leading Fitness Equipment Brand

Context:

This particular business case focuses on the operations of a leading brand in the field of fitness equipment, providing a product range including machines such as treadmills, exercise bikes, and other gym/fitness accessories to cater to the needs of all categories of people. This case study aims to identify the characteristics of the target audience for each type of treadmill offered by the company, and provide data driven insights and actionable business recommendations about treadmills suggestions to the new customers.

This case study report contains the solutions to the problem statements (as Python queries by employing Descriptive Statistics & Probability, sample output of the queries, followed by insights and recommendations. As part of the confidentiality agreement, the name of the retailer brand, the actual dataset and problem statements are not included in this report.

Google Colab Notebook-Python File - This Python project involves exploratory data analysis of a dataset from this treadmill brand. The code is importing necessary libraries such as numpy, pandas, seaborn, and matplotlib.

****

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

Shape of the dataset and Data type of all columns

Code:

```
data = pd.read_csv("treadmill.csv")
data.shape
data.info()
```

```
data.shape

(180, 9)

data.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: There are a total of 180 rows and 9 columns. All the columns – Product, Age, Gender, Education, MaritalStatus, Income, Usage, Fitness, Miles have zero null entries. Except the columns Product, Gender, and MaritalStatus, rest of the columns have datatype as integer. These categorical columns have object datatype.

2. Are there any missing values?

```
data.isnull().sum()
```

There are no missing or duplicate values in the dataset

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

3. Quick Look at unique values from different columns

```
columns = ['Product','MaritalStatus','Usage','Fitness','Education','Age']
for i in columns:
        print(f"{i}   -> {len(data[i].unique())}, {data[i].unique()}")
```

```
Product   -> 3, ['KP281' 'KP481' 'KP781']
MaritalStatus   -> 2, ['Single' 'Partnered']
Usage   -> 6, [3 2 4 5 6 7]
Fitness   -> 5, [4 3 2 1 5]
Education   -> 8, [14 15 12 13 16 18 20 21]
Age   -> 32, [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]
```

Insights:
1. There are 3 different treadmills products.
2. Age of customers ranges from 18 to 50
3.Education in years is from 12 -21
4. There are both Partnered and single customers
5. Fitness level of customers from 1 – 5
6. Usage of treadmill is from 2 days to 7 days a week

4. Detect Outliers (using boxplot, "describe" method by checking the difference between mean and median)

Code: `data.describe()`

```
data.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

Now, get the difference between mean and median for each numerical column to identify outliers:

```
(data.describe().loc['mean'] - data.describe().loc['50%']).abs()
```

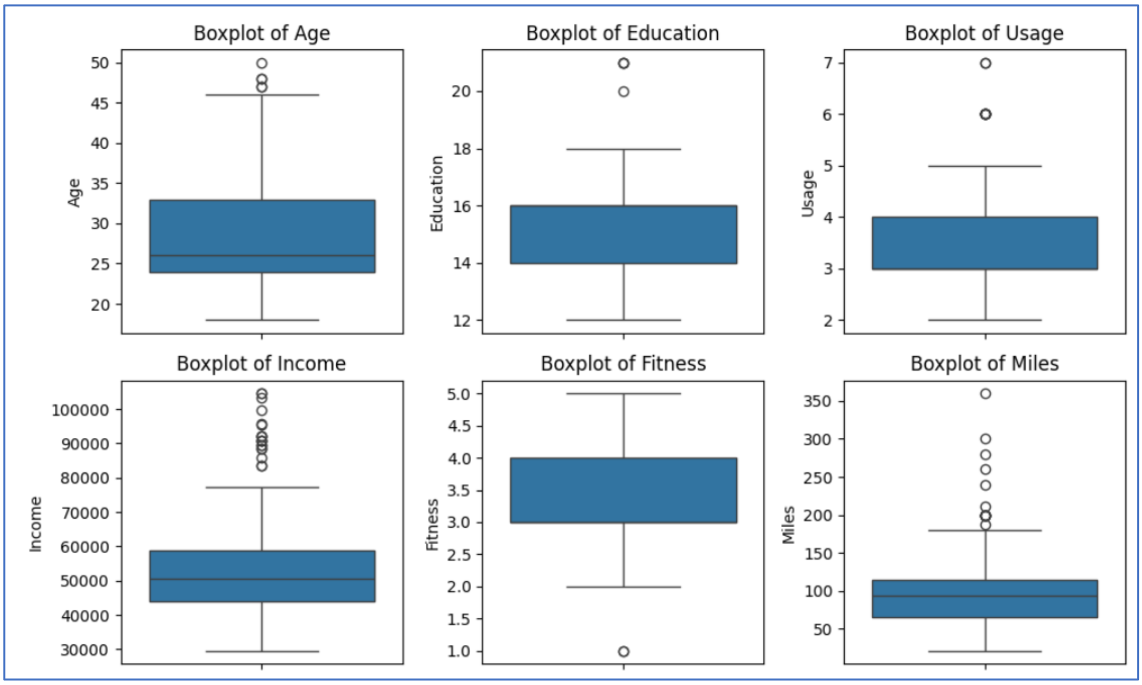
```
Age               2.641389
Education         0.427778
Usage             0.396944
Fitness           0.322222
Income         2880.570000
Miles             7.088889
dtype: float64
```

Insights: Each column seems to have some difference between mean and median indicating the presence of outliers. The columns - Income, Miles and Age seems to have the highest difference. The mean is higher than the median for all columns except Education, indicating right skewness (positive skew). The standard deviation for Income and Miles column (16,506 and 51.8 respectively) is slightly higher than their IQRs (14610 and 48.75 respectively) indicating a large spread and possible outliers. The mean is much higher than the median for Income column, indicating strong right skewness.

Identifying outliers using Boxplots for all continuous variables.

Code:

```
continuous_vars = ['Age', 'Education', 'Usage', 'Income', 'Fitness', 'Miles']

plt.figure(figsize=(10,6))
for i, var in enumerate(continuous_vars, 1):
    plt.subplot(2, 3, i)
    sns.boxplot(y=data[var])
    plt.title(f'Boxplot of {var}')

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



As shown on the plots, the columns - Income, Miles and Age columns have the most number of outliers. To identify the outlier data in all the continuous columns:

Code:

```
def find_outliers(data, column):
```

```
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_whisker = Q1 - 1.5 * IQR
    upper_whisker = Q3 + 1.5 * IQR
    outliers = data[(data[column]<lower_whisker) | (data[column]>upper_whisker)]

    return outliers
```

This `find_outliers` function can now be applied on all 6 columns to get respective outliers data. The data outside the lower and upper whisker values will be considered outliers.

Results:

```
find_outliers(data, "Age")
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
78	KP281	47	Male	16	Partnered	4	3	56850	94
79	KP281	50	Female	16	Partnered	3	3	64809	66
139	KP481	48	Male	16	Partnered	2	3	57987	64
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

```
find_outliers(data, "Usage")
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
154	KP781	25	Male	18	Partnered	6	4	70966	180
155	KP781	25	Male	18	Partnered	6	5	75946	240
162	KP781	28	Female	18	Partnered	6	5	92131	180
163	KP781	28	Male	18	Partnered	7	5	77191	180
164	KP781	28	Male	18	Single	6	5	88396	150
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
170	KP781	31	Male	16	Partnered	6	5	89641	260
175	KP781	40	Male	21	Single	6	5	83416	200

```
find_outliers(data, "Income")
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
159	KP781	27	Male	16	Partnered	4	5	83416	160
160	KP781	27	Male	18	Single	4	3	88396	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
162	KP781	28	Female	18	Partnered	6	5	92131	180
164	KP781	28	Male	18	Single	6	5	88396	150
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
168	KP781	30	Male	18	Partnered	5	4	103336	160
169	KP781	30	Male	18	Partnered	5	5	99601	150
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
172	KP781	34	Male	16	Single	5	5	92131	150
173	KP781	35	Male	16	Partnered	4	5	92131	360
174	KP781	38	Male	18	Partnered	5	5	104581	150
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

```
find_outliers(data, "Miles")
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
23	KP281	24	Female	16	Partnered	5	5	44343	188
84	KP481	21	Female	14	Partnered	5	4	34110	212
142	KP781	22	Male	18	Single	4	5	48556	200
148	KP781	24	Female	16	Single	5	5	52291	200
152	KP781	25	Female	18	Partnered	5	5	61006	200
155	KP781	25	Male	18	Partnered	6	5	75946	240
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
173	KP781	35	Male	16	Partnered	4	5	92131	360
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200

```
find_outliers(data, "Fitness")
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
14	KP281	23	Male	16	Partnered	3	1	38658	47
117	KP481	31	Female	18	Single	2	1	65220	21

```
find_outliers(data, "Education")
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
156	KP781	25	Male	20	Partnered	4	5	74701	170
157	KP781	26	Female	21	Single	4	3	69721	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
175	KP781	40	Male	21	Single	6	5	83416	200

Now, clip the data between the 5 percentile and 95 percentile by retaining all rows. This allows to set lower and upper bounds for the values in the DataFrame. i.e. it sets the values that are below the 5th percentile to the 5th percentile value, and those above the 95th percentile to the 95th percentile value.

Code:

```
def clip_outliers(data, columns):
    clipped_data = data.copy()
    for column in columns:
```

```
        lower_bound = round(data[column].quantile(0.05))
        upper_bound = round(data[column].quantile(0.95))
        clipped_data[column]=data[column].clip(lower =lower_bound,upper = upper_bound)
    return clipped_data

cleaned_data = clip_outliers(data, ['Income', 'Miles', 'Age', 'Education',
'Fitness', 'Usage'])
print("Clipped DataFrame:")
cleaned_data
```

Results ->

Clipped DataFrame:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20	Male	14	Single	3	4	34053	112
1	KP281	20	Male	15	Single	2	3	34053	75
2	KP281	20	Female	14	Partnered	4	3	34053	66
3	KP281	20	Male	14	Single	3	3	34053	85
4	KP281	20	Male	14	Partnered	4	2	35247	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	18	Single	5	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	43	Male	16	Single	5	5	90886	160
178	KP781	43	Male	18	Partnered	4	5	90948	120
179	KP781	43	Male	18	Partnered	4	5	90948	180

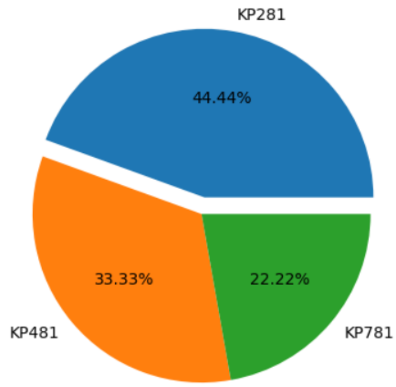
180 rows × 9 columns

Note: We will use this clipped_data for all further analysis

5. Which is most sold Model/Product?

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

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



KP281 treadmill model is the most sold model.

6. Representing the marginal probability - What percent of customers have purchased KP281, KP481, or KP781

Code:

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

marginal_probabilities = crosstab.loc['All', :]
print(marginal_probabilities)
crosstab
```

```
Gender
Female    0.422222
Male      0.577778
All       1.000000
Name: All, dtype: float64
```

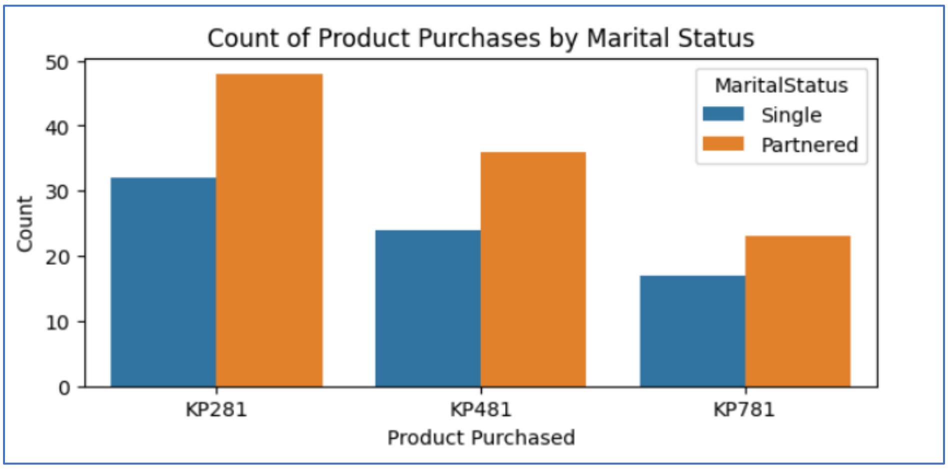
Gender / Product	Female	Male	All
KP281	0.222222	0.222222	0.444444
KP481	0.161111	0.172222	0.333333
KP781	0.038889	0.183333	0.222222
All	0.422222	0.577778	1.000000

Insights: 44.4% of total 180 customers have purchased KP281, 33.3% of total have purchased KP481 and remaining 22.2% have purchased KP781.

7. Check if features like marital status, gender, age etc have any effect on the product purchased Countplots can help visualize the relationships between the categorical variables and the product purchased.

   Count of Product Purchases by Marital Status:

```
plt.figure(figsize = (7,3))
sns.countplot(data = cleaned_data, x = "Product", hue = "MaritalStatus")
plt.xlabel('Product Purchased')
plt.ylabel('Count')
plt.title('Count of Product Purchases by Marital Status')
plt.show()
```



   Value Counts for MaritalStatus: `cleaned_data["MaritalStatus"].value_counts()`

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

Insights: The plot suggests that partnered individuals tend to purchase treadmills more frequently across all product categories compared to single individuals. This trend is most pronounced for KP281 and KP481. The difference is relatively negligible in KP781. In total, there are 107 Partnered and 73 single customers. This means 59.4% of the customers who purchased treadmill are partnered.

Value Counts for Gender: `cleaned_data["Gender"].value_counts()`

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

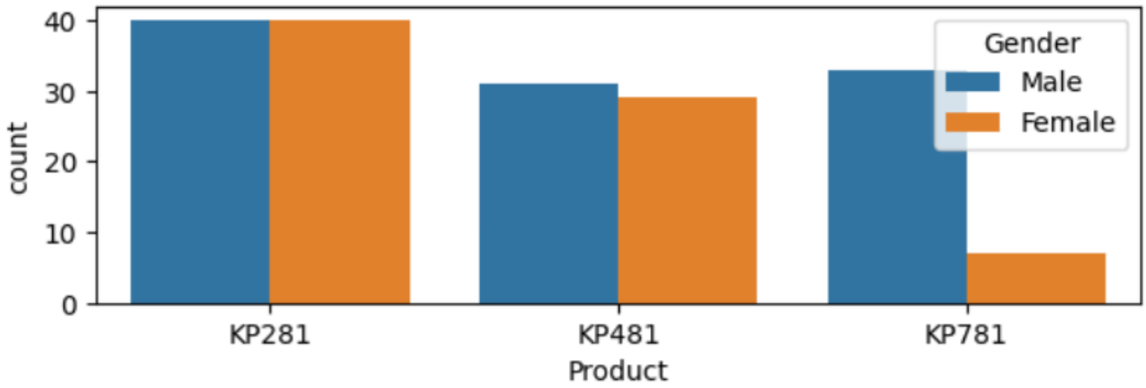
Count of Product Purchases by Gender:

```
print(cleaned_data.groupby(["Product", "Gender"]).count().Age)
```

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

Countplot:

```
plt.figure(figsize = (7,2))
sns.countplot(data = cleaned_data, x = "Product", hue = "Gender")
plt.show()
```

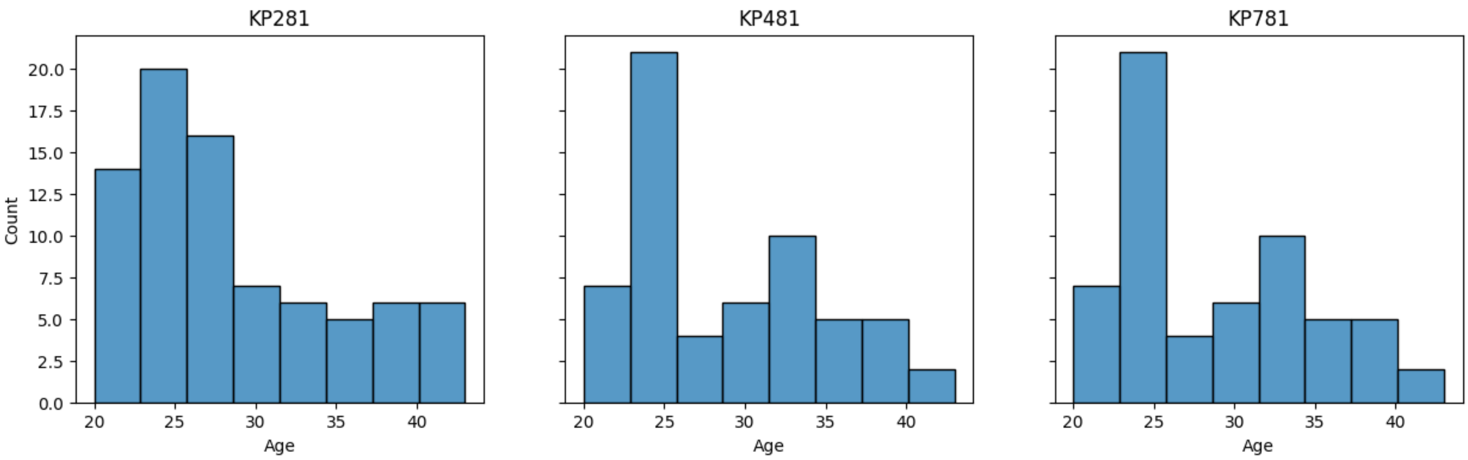


Insights: There are 76 female and 104 males customers. This means 57.8% of the customers who purchased treadmill are Males. More Male customers are buying treadmill compared to female customers. The product KP281 was equally brought my male and female. For the products KP481, the difference in male and female numbers are negligible as compared to KP781 where male numbers are significantly higher than females.

Effects of Age on Product Purchased:

Code:

```
axes = plt.subplots(1, 3, figsize=(15,4), sharey=True)
plt.subplot(1, 3, 1)
product_1 = cleaned_data[cleaned_data["Product"] == "KP281"]
sns.histplot(data = product_1, x= "Age", bins= 8)
plt.title("KP281")
plt.subplot(1, 3, 2)
product_2 = cleaned_data[cleaned_data["Product"] == "KP481"]
sns.histplot(data = product_2, x= "Age", bins= 8)
plt.title("KP481")
plt.subplot(1, 3, 3)
product_3 = cleaned_data[cleaned_data["Product"] == "KP481"]
sns.histplot(data = product_3, x= "Age", bins= 8)
plt.title("KP781")
```



Insights: For all 3 products, most customers are from the age group of 25-30.

8. Bi-variate Analysis for:
   - Product & Age
   - Product & Income
   - Product & Education
   - Product & Usage
   - Product & Fitness
   - Product & Miles

   Code:

```
cleaned_data.groupby("Product").agg(
    Product_count = ('Gender', 'count'),
    mean_Age = ("Age", "mean"),
    mean_Income = ("Income", "mean"),
    mean_Miles = ("Miles", "mean"),
    mean_Usage = ("Usage", "mean"),
    mean_Fitness = ("Fitness", "mean"),
    mean_Education = ("Education", "mean")).reset_index()
```

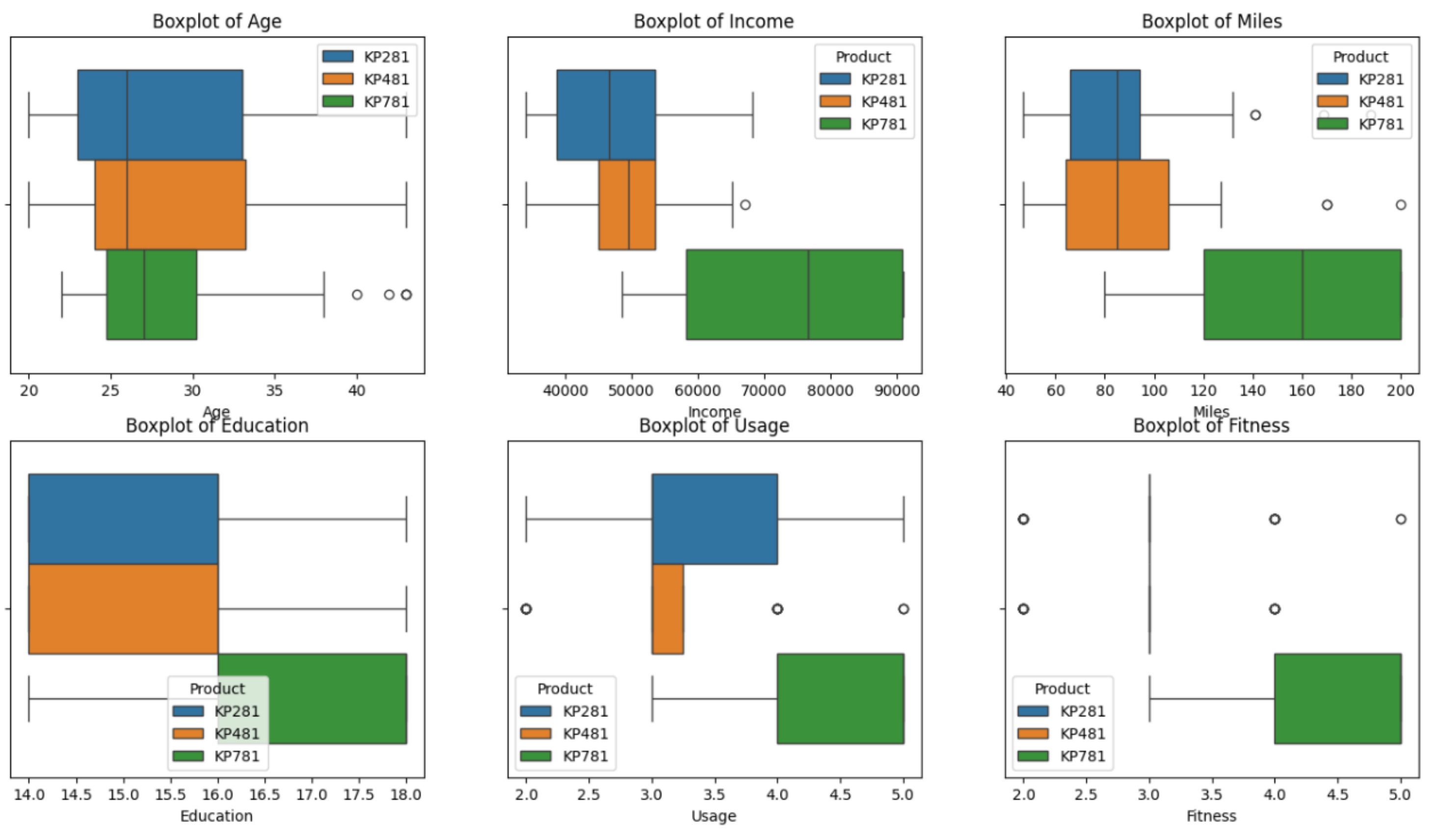
	Product	Product_count	mean_Age	mean_Income	mean_Miles
0	KP281	80	28.425	46584.300	83.125
1	KP481	60	28.800	49046.600	88.500
2	KP781	40	28.825	73908.225	155.900

	Product	Product_count	mean_Usage	mean_Fitness	mean_Education
0	KP281	80	3.087500	2.975000	15.125000
1	KP481	60	3.066667	2.916667	15.183333
2	KP781	40	4.500000	4.625000	17.050000

Visualisation through Boxplots:

```
axes = plt.figure(figsize=(17,9))

plt.subplot(2, 3, 1)
sns.boxplot(x='Age', data=cleaned_data, hue = 'Product')
```

```
plt.legend(loc='upper right')
plt.title('Boxplot of Age')

plt.subplot(2, 3, 2)
sns.boxplot(x='Income', data=cleaned_data, hue = 'Product')
plt.title('Boxplot of Income')

plt.subplot(2, 3, 3)
sns.boxplot(x='Miles', data=cleaned_data, hue = 'Product')
plt.title('Boxplot of Miles')

plt.subplot(2, 3, 4)
sns.boxplot(x='Education', data=cleaned_data, hue = 'Product')
plt.title('Boxplot of Education')

plt.subplot(2, 3, 5)
sns.boxplot(x='Usage', data=cleaned_data, hue = 'Product')
plt.title('Boxplot of Usage')

plt.subplot(2, 3, 6)
sns.boxplot(x='Fitness', data=cleaned_data, hue = 'Product')
plt.title('Boxplot of Fitness')

plt.show()
```



Insights:

1. Age of customers buying KP281 and KP481 is between 20-35, whereas customers buying KP781 are primarily in 25-30
2. Customers with higher income and more education have purchased KP781 model.
3. Customers with lower income purchase KP281 and KP481 model. The lower cost of these treadmill may have encouraged them to buy it.

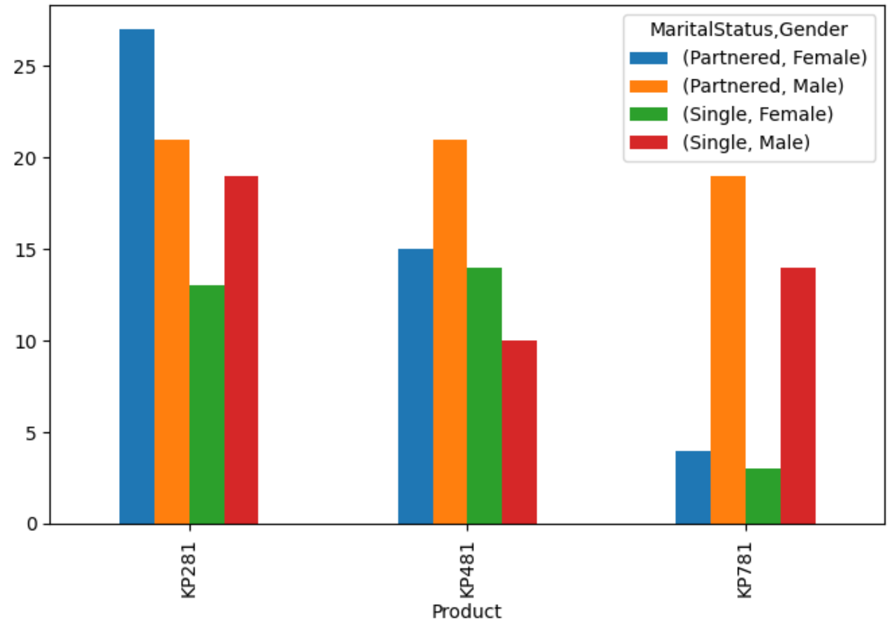
4. Customers who bought KP481 model expecting to use Treadmill less frequently but to run more miles a week.
5. Customer purchasing KP781 plan to use it more frequently, run more miles and have high self-rated fitness. These are mostly male customers. They are likely more serious about their fitness routines. They also have higher education and income. The KP781 might offer features that cater to high-intensity and frequent usage, such as better durability, advanced features, and higher performance. Individuals who are health conscious or professional athletes/trainers might require a treadmill that can withstand rigorous and frequent use. Higher education often correlates with a greater awareness of the importance of fitness and the benefits of investing in quality equipment. Higher-income customers have more disposable income to spend on premium products.

9. Multivariate Analysis

   Code:

```
multi = pd.crosstab(index=cleaned_data["Product"],
        columns=[cleaned_data["MaritalStatus"],
        cleaned_data["Gender"]],)
multi.plot(kind='bar',figsize=(8,5))
```

<Axes: xlabel='Product'>



Insights:

1. Partnered Female mostly bought KP281 Model
2. KP781 is mostly bought by Partnered Male
3. Single Female customers bought KP481 model more than Single Male customers.
4. Partnered Males almost equally bought all 3 models.
5. There are more single males buying Treadmill than single Females.
6. The majority of our buyers are men.
10. What is the probability of a male customer buying a KP781 treadmill?

Code:
```
Male = cleaned_data[cleaned_data["Gender"] == "Male"]
crosstab = pd.crosstab(Male['Product'], Male['Gender'], margins=True,
normalize=True)

conditional_probabilities = crosstab.loc[:, ["Male"]]
print(conditional_probabilities)
```

```
Gender        Male
Product
KP281     0.384615
KP481     0.298077
KP781     0.317308
All       1.000000
```

Insights: The probability of a male customer buying a KP781 treadmill is 31.7%. The probability of them buying KP281 is the highest, which is 38.4%. Probability of buying KP481 is the lowest (29%).

11. What is the probability of a female customer buying a KP781 treadmill?
Code:
```
Female = cleaned_data[cleaned_data["Gender"] == " Female "]
crosstab = pd.crosstab(Male['Product'], Female ['Gender'], margins=True,
normalize=True)

conditional_probabilities = crosstab.loc[:, ["Male"]]
print(conditional_probabilities)
```

```
Gender      Female
Product
KP281     0.526316
KP481     0.381579
KP781     0.092105
All       1.000000
```

Insights: The probability of a female customer buying a KP781 treadmill is as low as 9.2%. The probability of them buying KP281 is the highest, which is 52.6%. Probability of buying KP481 is the second highest (38%).

12. Check correlation among different factors using heat maps or pair plots.

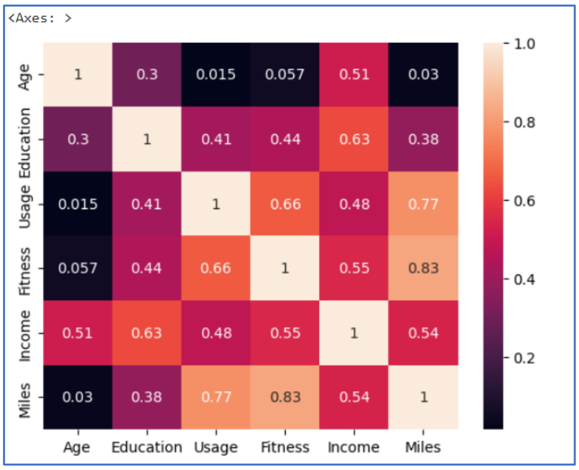
```
Code:
data_corr = cleaned_data.corr(numeric_only = True)
```

Results:

	Age	Education	Usage	Fitness	Income	Miles
Age	1.000000	0.301984	0.015218	0.057314	0.514407	0.029719
Education	0.301984	1.000000	0.412484	0.441082	0.628597	0.377294
Usage	0.015218	0.412484	1.000000	0.660556	0.478615	0.769234
Fitness	0.057314	0.441082	0.660556	1.000000	0.546997	0.826307
Income	0.514407	0.628597	0.478615	0.546997	1.000000	0.537296
Miles	0.029719	0.377294	0.769234	0.826307	0.537296	1.000000

```
sns.heatmap(data_corr, annot = True)
```

Results:



Insights: Fitness and Miles are the most correlated (positively) by 0.83%. As the expected average number of miles to walk/run have increased, people have also rated themselves with higher fitness ratings. Usage and Miles have the second highest positive correlation by 0.77%.

Customer Profiling

1. KP281
    - 44.4% customers brought KP281. Making it the model with the highest demand.
    - Average customer income is 46.5K
    - There are same numbers of Male and Female customers purchasing this, hence this model is not gender specific
    - Partnered Females mostly bought this model.
    - Average age of customer who purchases TKP281 is 28.5, Median is 26.
    - They expect to use treadmill 3-4 times a week.
    - Customers who purchased the KP781 treadmill generally rate their fitness as average. They might be seeking a reliable treadmill that meets their basic exercise needs without requiring advanced features.
    - The KP781 might offer straightforward functionality, making it user-friendly for those who are not looking for complex features.
    - Since this model is priced reasonably, it might attract customers who want a basic treadmill that fits within their budget.

2. KP481
    - 33.3% customers brought KP481. Making it the second most popular product.
    - Average Income of the customer is 49K
    - Average age of customer who purchases this model is 28.8.
    - The income of this group is almost same as KP281 model.
    - Customers who bought KP481 model expecting to use Treadmill less frequently but to run more miles a week.
    - Partnered Males forms the largest customer base of this model.
3. KP781
    - Product made only 22 % of sales.
    - Average Income of the customer is 74K
    - Average age of customer who purchases this model is 28.8.
    - This treadmill seems to be more popular with customer having higher income and who are Partnered Males. This model is costlier compared to other two.
    - Customers who purchase the KP781 often rate their fitness level highly and plan to use the treadmill more frequently. This might mean the treadmill is perceived as suitable for serious fitness enthusiasts, rather than casual users.

Recommendations:

KP281 & KP481:

- **Target Audience**: These models attract individuals with an income below approximately $50,000, likely due to their affordability.

- **Marketing Strategy**: Position the KP281 and KP481 as budget-friendly treadmills that offer excellent value for money. Emphasize that these models provide all the essential features needed for a great workout experience, making them ideal for cost-conscious customers. Market them as accessible and reliable fitness solutions for everyone

KP781:

- **Target Audience**: The KP781 appeals to professionals and athletes who are willing to invest in high-end fitness equipment.

- **Marketing Strategy**: Promote the KP781 as a premium treadmill designed for serious fitness enthusiasts. Highlight its advanced features, superior build quality, and performance capabilities. Create a luxurious brand image that positions the KP781 as the go-to choice for those seeking top-tier fitness equipment. Emphasize its suitability for rigorous training and professional use, appealing to customers who value excellence and luxury in their workout gear.