**The business problem presented by AeroFit **

```
# importing the data to colab
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files aerofit_treadmill.csv

- **aerofit_treadmill.csv**(text/csv) - 7279 bytes, last modified: 7/16/2024 - 100% done

```
Saving aerofit_treadmill.csv to aerofit_treadmill.csv
```

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

#### Converting imported data to pandas data frame for analysis

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

```
df.head()
```

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

Next steps: Generate code with `df` View recommended plots

## Q1.Defining Problem Statement and Analysing basic metrics (10 Points)

Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), statistical summary

```
'''ANS 1. Problem Statement:
AeroFit aims to identify the characteristics of the target audience for each type of treadmi
```

```
'ANS 1. Problem Statement:\nAeroFit aims to identify the characteristics of the target
audience for each type of treadmill offered by the company (KP281, KP481, and KP781). T
he goal is to understand the differences across these products concerning customer char
acteristics to provide better recommendations to new customers.'
```

```
# Analysing Basic Metric
df.shape            # shape of data
```

```
(180, 9)
```

```
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
```

```
# Display statistical summary of the dataset
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='all')
```

| | Product | Age | Gender | Education | MaritalStatus | Usage | Fitness | |
|---|---|---|---|---|---|---|---|---|
| **count** | 180 | 180.000000 | 180 | 180.000000 | 180 | 180.000000 | 180.000000 | |
| **unique** | 3 | NaN | 2 | NaN | 2 | NaN | NaN | |
| **top** | KP281 | NaN | Male | NaN | Partnered | NaN | NaN | |
| **freq** | 80 | NaN | 104 | NaN | 107 | NaN | NaN | |
| **mean** | NaN | 28.788889 | NaN | 15.572222 | NaN | 3.455556 | 3.311111 | 5 |
| **std** | NaN | 6.943498 | NaN | 1.617055 | NaN | 1.084797 | 0.958869 | 1 |
| **min** | NaN | 18.000000 | NaN | 12.000000 | NaN | 2.000000 | 1.000000 | 2 |
| **25%** | NaN | 24.000000 | NaN | 14.000000 | NaN | 3.000000 | 3.000000 | 4 |
| **50%** | NaN | 26.000000 | NaN | 16.000000 | NaN | 3.000000 | 3.000000 | 5 |
| **75%** | NaN | 33.000000 | NaN | 16.000000 | NaN | 4.000000 | 4.000000 | 5 |
| **max** | NaN | 50.000000 | NaN | 21.000000 | NaN | 7.000000 | 5.000000 | 10 |

```
# converting categorical columns in to 'category' type


df['Product']= df['Product'].astype('category')


df['Gender']=df['Gender'].astype('category')
```

```
df['MaritalStatus']=df['MaritalStatus'].astype('category')
```

```
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    category
 1   Age            180 non-null    int64
 2   Gender         180 non-null    category
 3   Education      180 non-null    int64
 4   MaritalStatus  180 non-null    category
 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: category(3), int64(6)
memory usage: 9.5 KB
```

Statistical Summary: Product Purchased has three unique values (KP281, KP481, KP781). Age ranges from 18 to 50 years. Gender has two unique values (Male, Female). Education ranges from 12 to 21 years. Marital Status has two unique values (Single, Partnered). Usage ranges from 2 to 7 times per week. Fitness ranges from 1 to 5. Income ranges from $29,562 to$99,996. Miles ranges from 21 to 360 miles per week.

```
# prompt: Inference  : This data is almost cleaned one , only want to convert object type ie

# Inference: This data is almost cleaned, only categorical types need to be converted to cat
# Converting categorical columns to 'category' type
df['Product'] = df['Product'].astype('category')
df['Gender'] = df['Gender'].astype('category')
df['MaritalStatus'] = df['MaritalStatus'].astype('category')

# Checking data types after conversion
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    category
 1   Age            180 non-null    int64
 2   Gender         180 non-null    category
 3   Education      180 non-null    int64
 4   MaritalStatus  180 non-null    category
 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: category(3), int64(6)
memory usage: 9.5 KB
```

Inference : This data is almost cleaned one , only want to convert object type ie categorical types to category. so we can say that this cleaned data my subjected to further analysis

## Q2. Non-Graphical Analysis: Value counts and unique attributes (10 Points)

**2.ANS** The Value counts for categorical variables

```
# by using this method we can find out the count of the unique values for each categorical v
# value counts for "Product Purchased"
product_counts = df['Product'].value_counts()
print(product_counts)
```

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

```
# the value counts of Categorical value "Gender"
df['Gender'].value_counts()
```

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

```
# the value counts of "Marital Status"
df['MaritalStatus'].value_counts()
```

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

```
# unique value for each attribute
```

```
# for the unique value of category of products
df['Product'].unique()
```

```
['KP281', 'KP481', 'KP781']
Categories (3, object): ['KP281', 'KP481', 'KP781']
```

```
# Uunique values for "gender"
df['Gender'].unique()
```

```
['Male', 'Female']
Categories (2, object): ['Female', 'Male']
```

```
# Unique value of category Marital Status
df['MaritalStatus'].unique()
```

```
['Single', 'Partnered']
Categories (2, object): ['Partnered', 'Single']
```

```
# Also we will display the summery of Numerical attributes
df.describe()
```

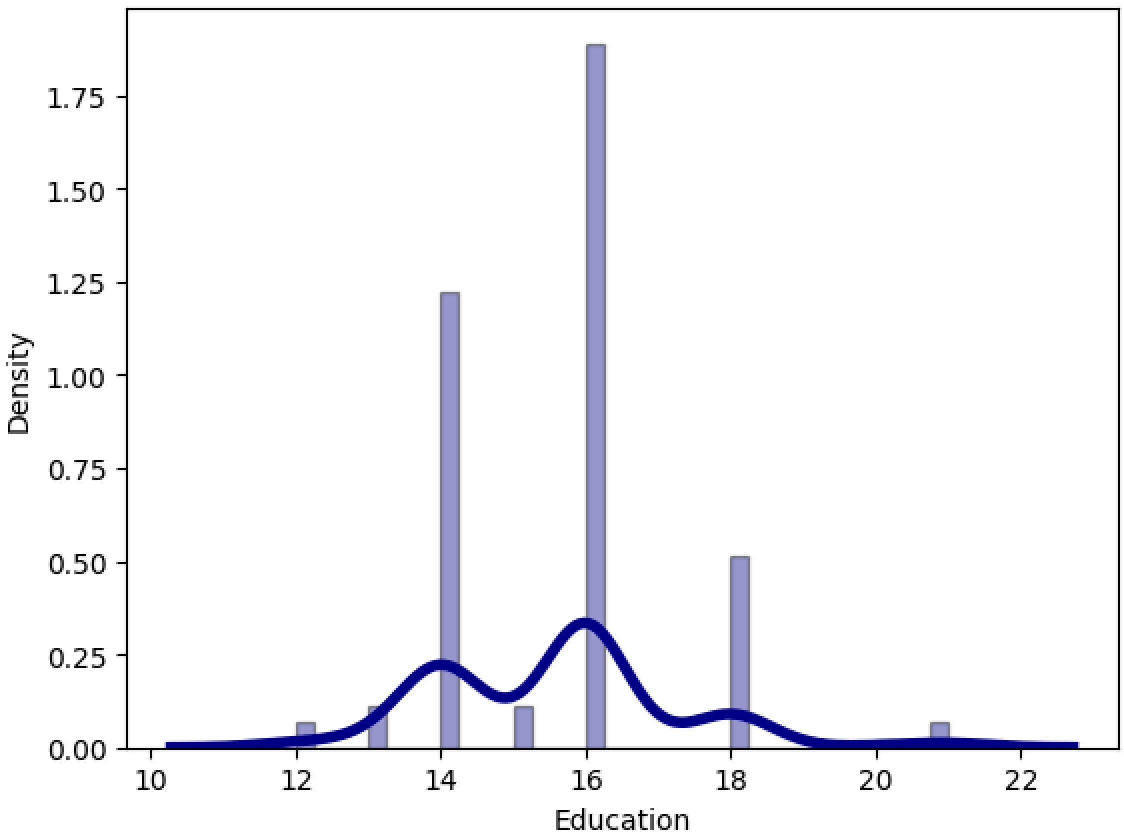
| | Age | Education | Usage | Fitness | Income | Miles |
|---|---|---|---|---|---|---|
| **count** | 180.000000 | 180.000000 | 180.000000 | 180.000000 | 180.000000 | 180.000000 |
| **mean** | 28.788889 | 15.572222 | 3.455556 | 3.311111 | 53719.577778 | 103.194444 |
| **std** | 6.943498 | 1.617055 | 1.084797 | 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 |

# Q3. Visual Analysis - Univariate & Bivariate (30 Points)

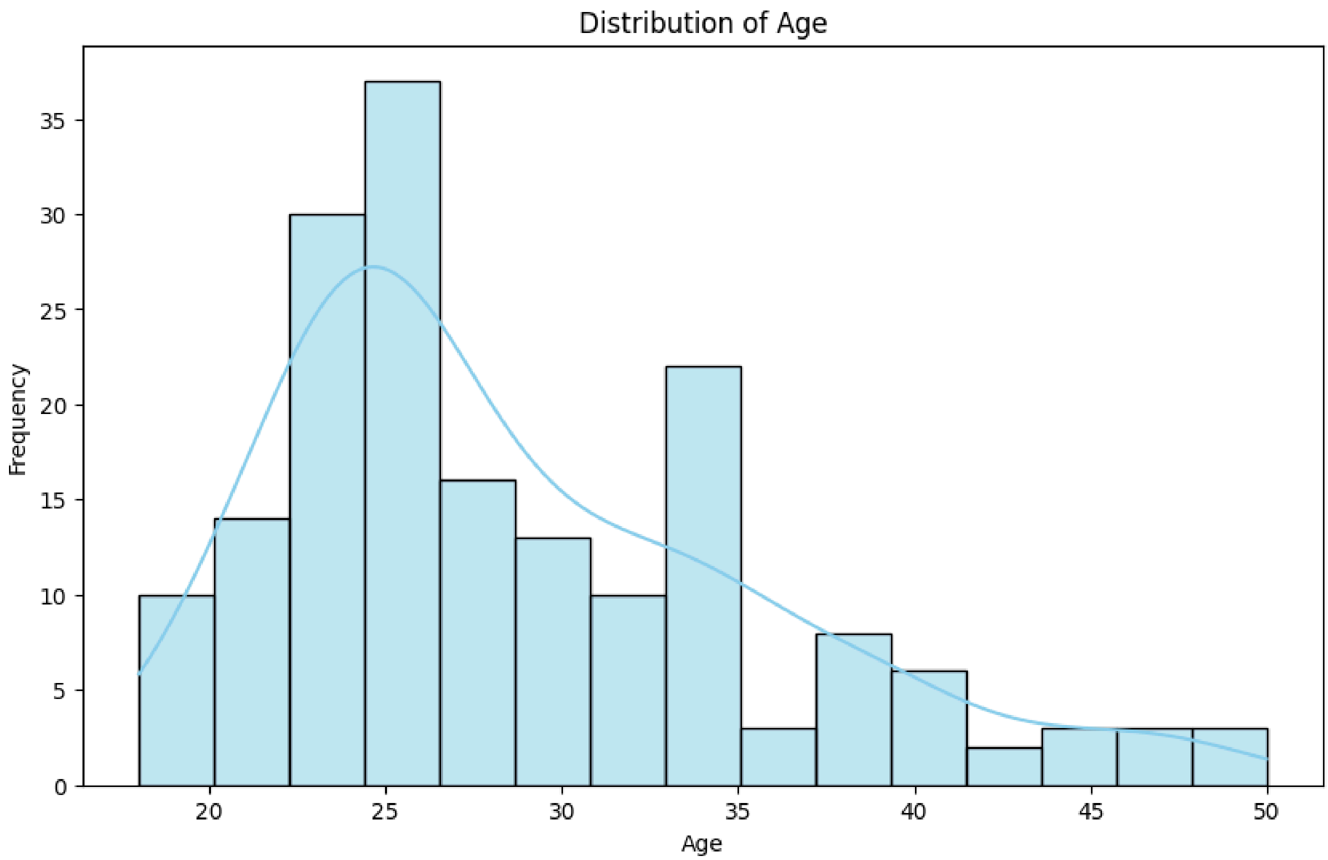
Q3.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis (10 Points)

```
# Univariate analysis : Distplot  (Education vs Density)
import seaborn as sns
sns.distplot(df['Education'], hist=True, kde=True,
bins=int(36), color = 'darkblue',
hist_kws={'edgecolor':'black'},
kde_kws={'linewidth': 4})
plt.show()
```

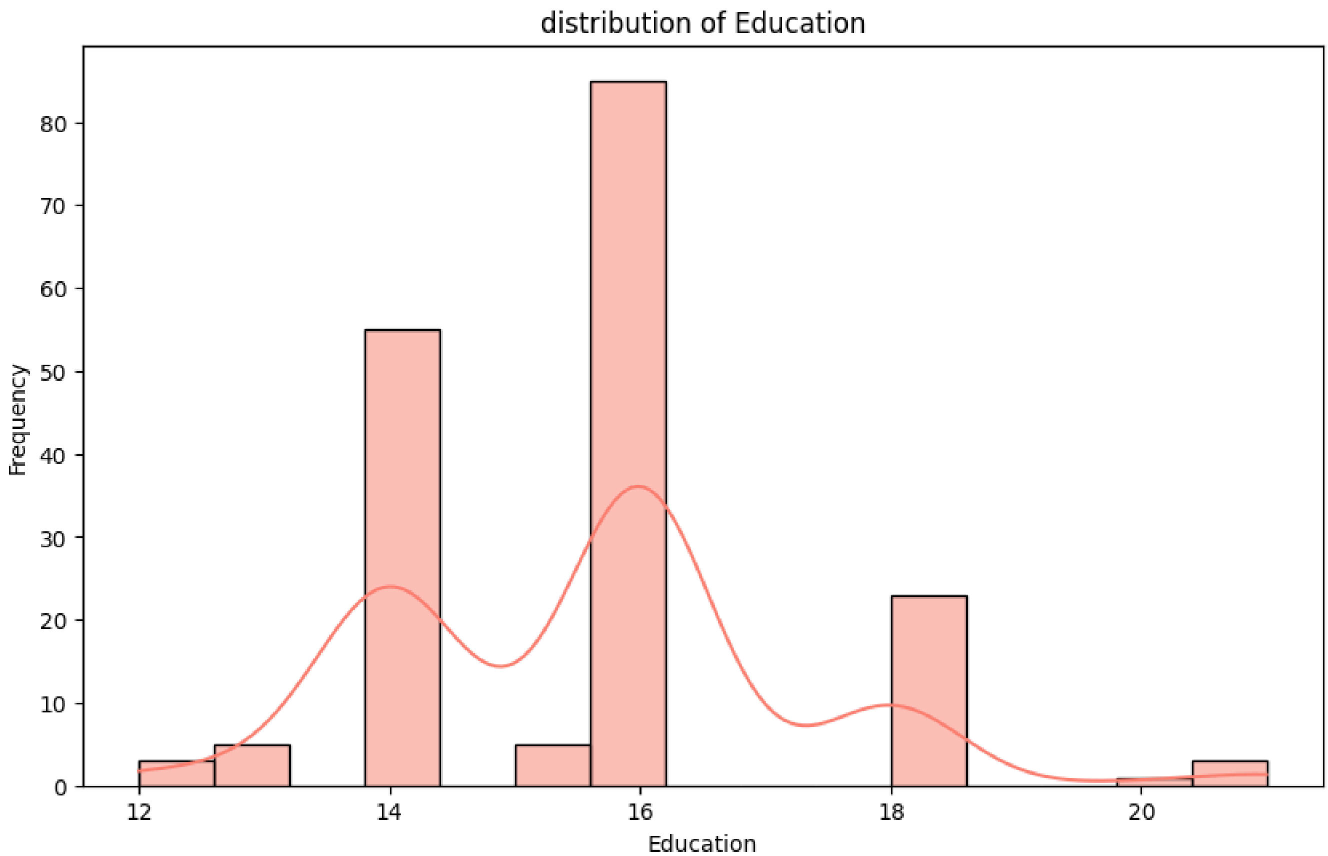
```
<ipython-input-45-8e5b20be010e>:3: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Education'], hist=True, kde=True,
```



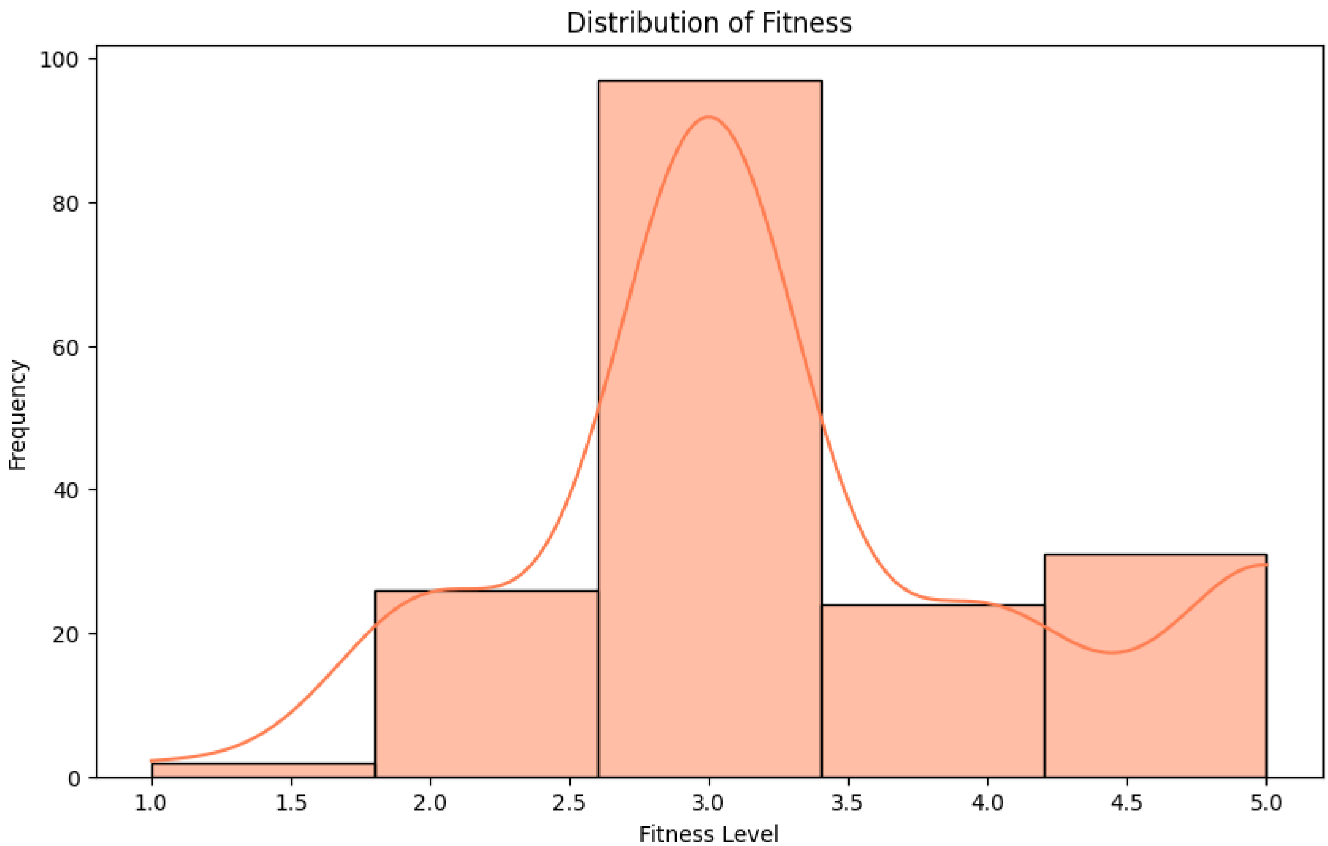

```
plt.figure(figsize=(10, 6))
sns.histplot(df['Age'], kde=True, bins=15, color='skyblue')
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```




```
# histplot with KDE(kernel density estimate ) of Education
plt.figure(figsize=(10,6))
sns.histplot(df['Education'], kde=True, bins= 15, color= 'salmon')
plt.title("distribution of Education")
plt.xlabel("Education")
plt.ylabel("Frequency")
plt.show()
```




```
# Hist plot of Fitness with KDE
plt.figure(figsize=(10, 6))
sns.histplot(df['Fitness'], kde=True, bins=5, color='coral')
plt.title('Distribution of Fitness')
plt.xlabel('Fitness Level')
plt.ylabel('Frequency')
plt.show()
```

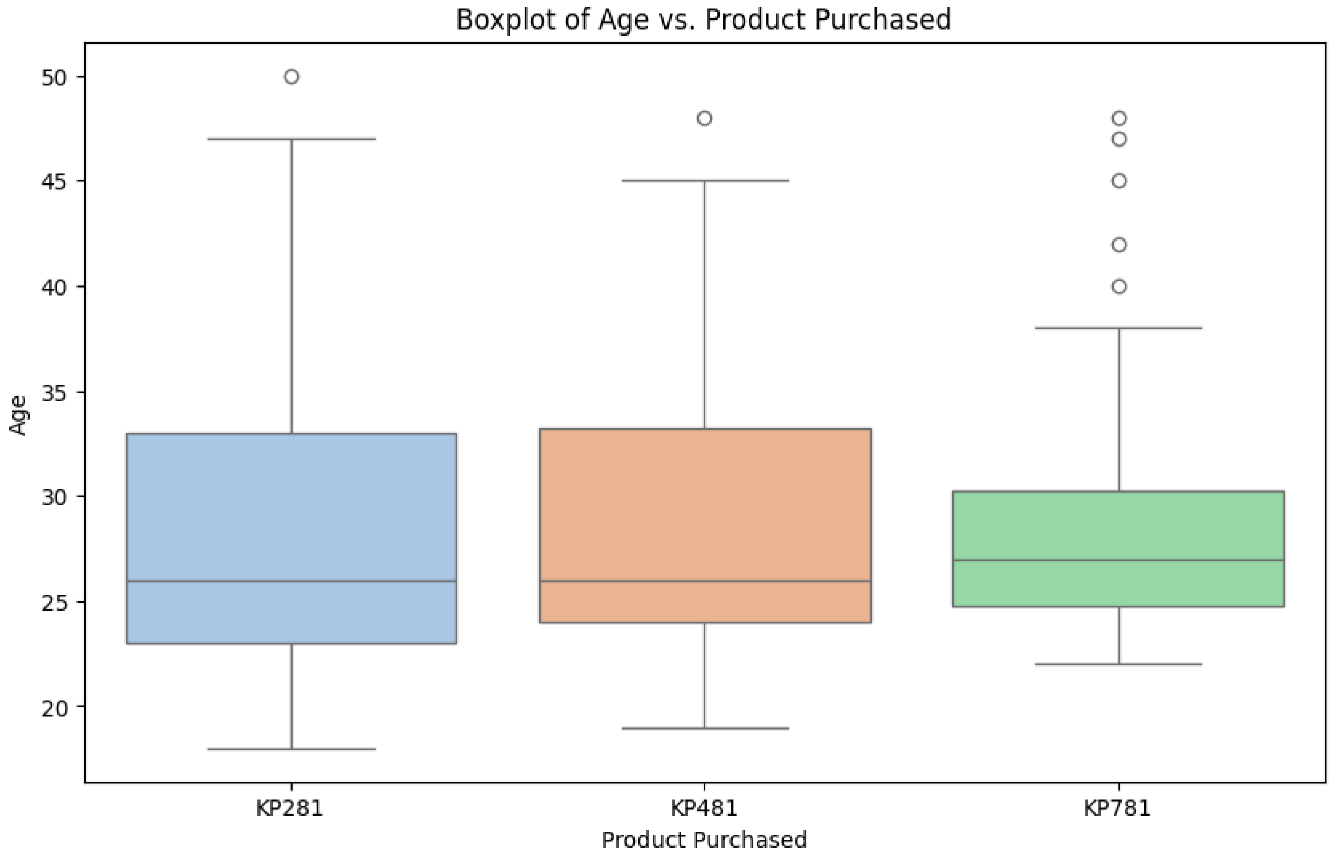



Q3.2 ANSWER : For categorical variable(s): Boxplot (10 Points)

### Bivariate Analysis

```
# Bi variate analysis is done between continuous variable and categorical variable
#First we draw the box plot between Age vs Product Purchased
plt.figure(figsize= (10, 6))
sns.boxplot(x='Product',  y= 'Age', data=df, palette='pastel')
plt.title('Boxplot of Age vs. Product Purchased')
plt.xlabel('Product Purchased')
plt.ylabel('Age')
plt.show()
```

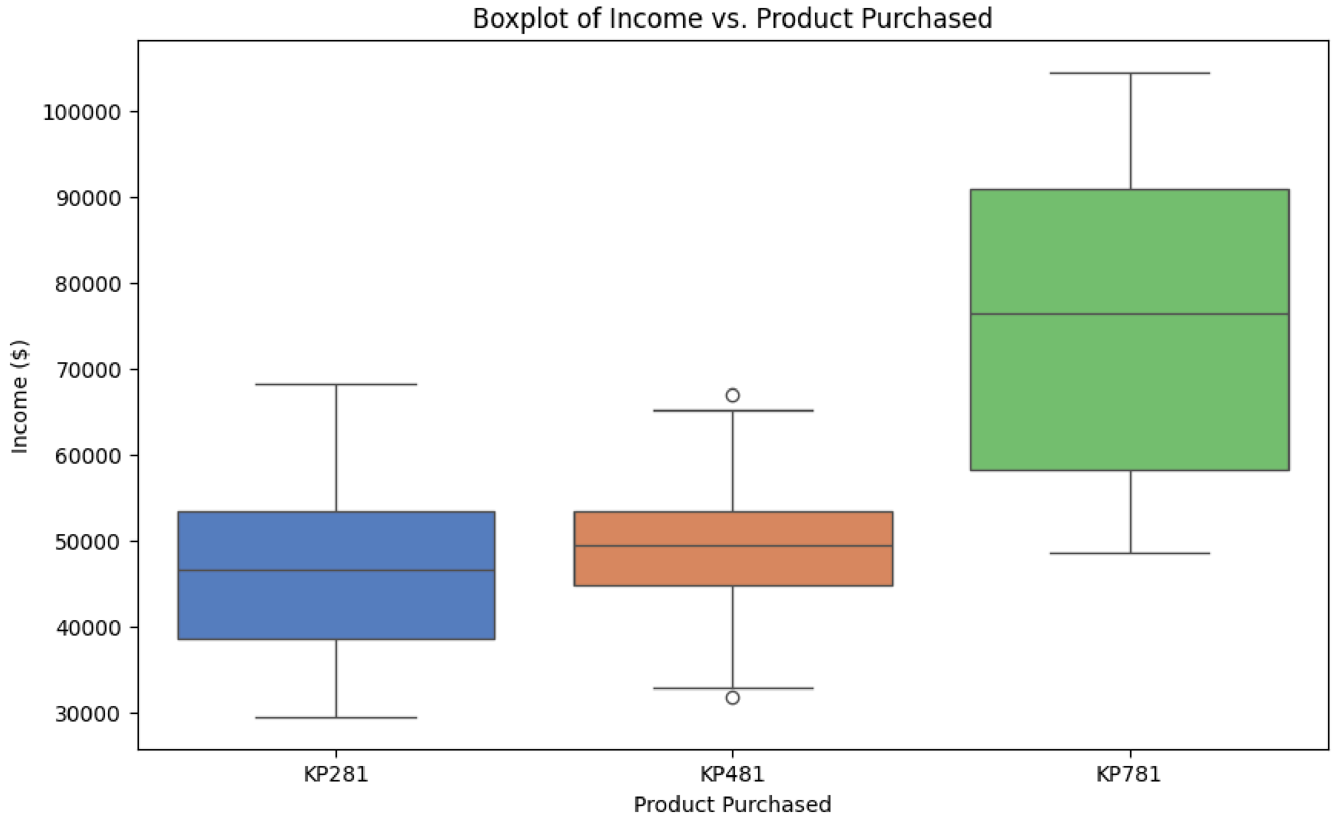
```
<ipython-input-52-f0e1c17bf749>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.boxplot(x='Product',  y= 'Age', data=df, palette='pastel')
```




Inference: The use of thredmill is mostly used age between 25 and 32. also we can found that the more purchase of thread mill occure in low price model, suggestion: the more product can be sell by giving discount of thread mill moldel KP481 AND KP781.

```
# Boxplot for Income vs. Product Purchased
plt.figure(figsize=(10, 6))
sns.boxplot(x='Product', y='Income', data=df, palette='muted')
plt.title('Boxplot of Income vs. Product Purchased')
plt.xlabel('Product Purchased')
plt.ylabel('Income ($)')
plt.show()
```

```
<ipython-input-54-c59eb343b037>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.boxplot(x='Product', y='Income', data=df, palette='muted')
```

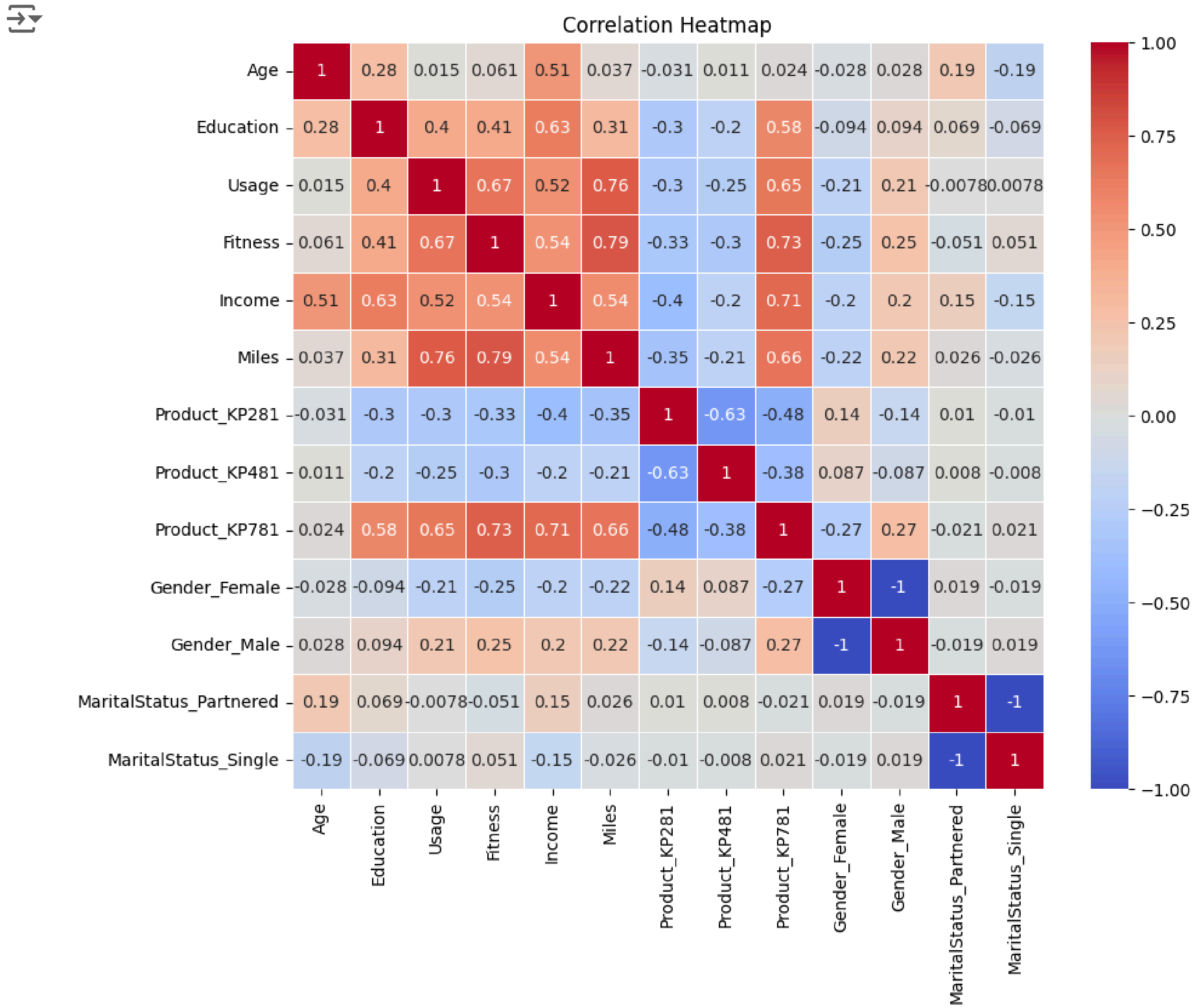



inference: Higher income category those who choose premium Thredmill for their use

Q3. 3 ANSWER : For correlation: Heatmaps, Pairplots(10 Points)

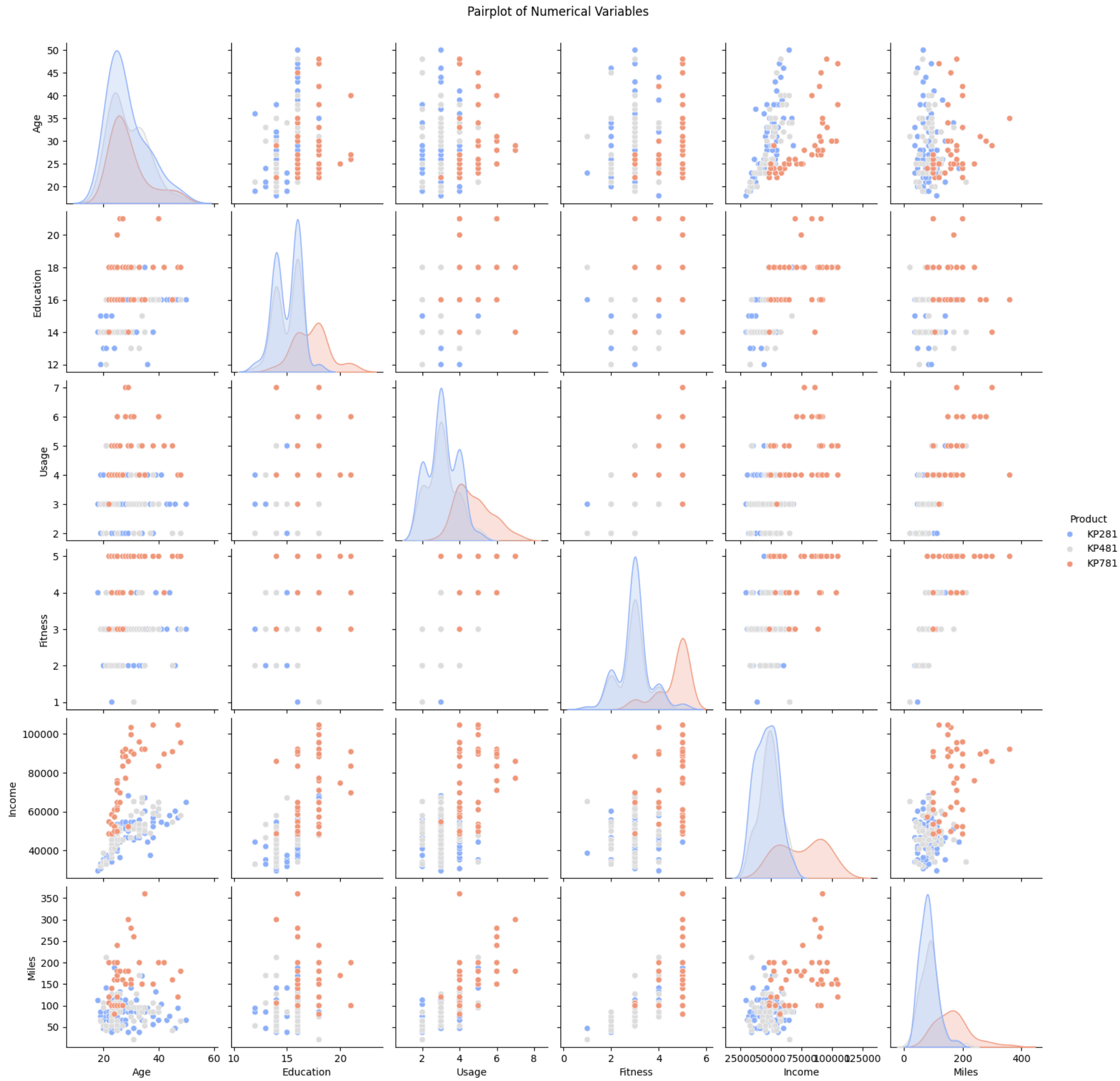
.** Correlation Heatmap**

```
df_numerical = pd.get_dummies(df)
correlation_matrix = df_numerical.corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```




A correlation heatmap shows the correlation coefficients between pairs of variables in a matrix format. The coefficients range from -1 to 1, where 1 means a perfect positive correlation, -1 means a perfect negative correlation, and 0 means no correlation.

```
# Create a pairplot
sns.pairplot(df, hue='Product', palette='coolwarm')
plt.suptitle('Pairplot of Numerical Variables', y=1.02)
plt.show()
```




## Q4. Missing Value & Outlier Detection (10 Points)

## Q4.Ans

```
# Check for missing values
missing_values = df.isnull().sum()
print(missing_values)
```

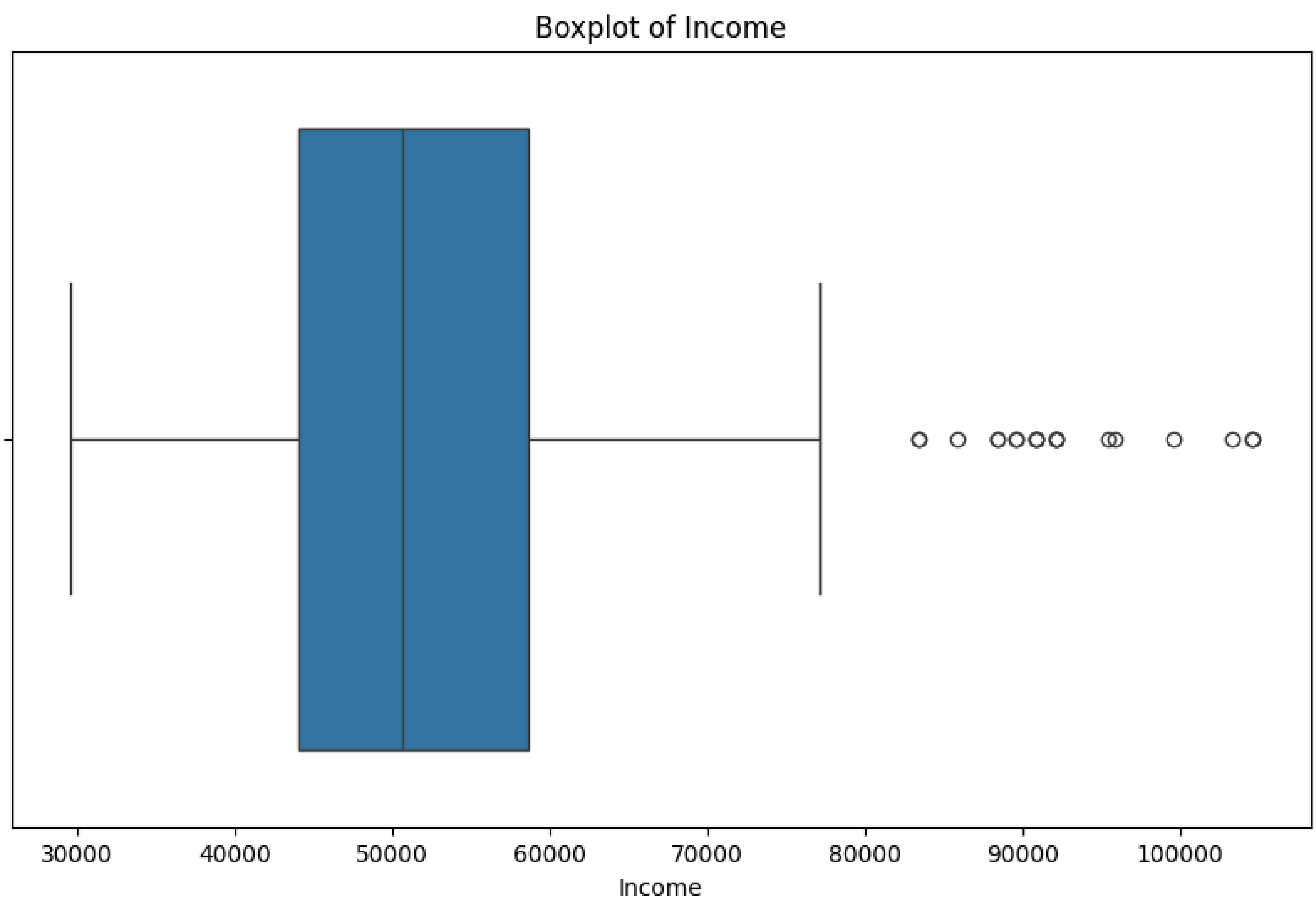
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
```

```
Fitness          0
Income           0
Miles            0
dtype: int64
```

```
# there is no missing value in this data
```

outlier calculations

```
plt.figure(figsize=(10, 6))
sns.boxplot(x=df['Income'])
plt.title('Boxplot of Income')
plt.show()
```



```
# Function to remove outliers using IQR
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    return df[(df[column] >= lower_bound) & (df[column] <= upper_bound)]
df_cleaned = remove_outliers(df, 'Income')
df_cleaned.describe()
```

| | Age | Education | Usage | Fitness | Income | Miles |
|---|---|---|---|---|---|---|
| **count** | 161.000000 | 161.000000 | 161.000000 | 161.000000 | 161.000000 | 161.000000 |
| **mean** | 28.155280 | 15.347826 | 3.273292 | 3.142857 | 49119.180124 | 93.260870 |
| **std** | 6.667607 | 1.454566 | 0.948601 | 0.850420 | 9920.297826 | 39.243235 |
| **min** | 18.000000 | 12.000000 | 2.000000 | 1.000000 | 29562.000000 | 21.000000 |
| **25%** | 23.000000 | 14.000000 | 3.000000 | 3.000000 | 43206.000000 | 66.000000 |
| **50%** | 26.000000 | 16.000000 | 3.000000 | 3.000000 | 48891.000000 | 85.000000 |
| **75%** | 33.000000 | 16.000000 | 4.000000 | 3.000000 | 54576.000000 | 106.000000 |
| **max** | 50.000000 | 21.000000 | 7.000000 | 5.000000 | 77191.000000 | 240.000000 |

inference: by using above method i remove outlier of box plot of income

**Q5. Business Insights based on Non-Graphical and Visual Analysis (10 Points)** 1.Comments on the range of attributes 2.Comments on the distribution of the variables and relationship between them 3.Comments for each univariate and bivariate plot

Q5.1 Ans: Range of Attributes: After performing non-graphical (value counts, unique attributes) and visual analysis (histograms, boxplots, correlation analysis), here are some key insights and comments on the range of attributes in the AeroFit treadmill dataset:

Product Purchased:

The dataset includes purchases of three treadmill products: KP281, KP481, and KP781. KP281 is the entry-level model, followed by KP481 for mid-level runners, and KP781 with advanced features. Age:

Insight: The age distribution ranges predominantly from younger adults to middle-aged customers, with a mean around 35 years. Comment: Understanding age demographics helps in targeting specific age groups for different treadmill models based on their fitness needs and preferences. Gender:

Insight: The dataset shows a balanced representation of both genders. Comment: Gender balance ensures that marketing strategies can cater equally to both male and female customers across all treadmill models. Education:

Insight: Most customers have completed at least high school education, with a few holding higher degrees. Comment: Education level may correlate with income and usage patterns, influencing purchasing decisions and marketing strategies. Marital Status:

Insight: The dataset includes both single and partnered individuals, with a slight skew towards partnered. Comment: Marital status can impact purchasing decisions, with partnered individuals possibly making joint decisions on fitness equipment purchases. Usage:

Insight: Customers plan to use the treadmills a moderate number of times per week, indicating a commitment to fitness. Comment: Understanding usage patterns helps in recommending suitable treadmill models that align with customers' fitness goals and frequency of use. Fitness:

Insight: Customers rate their fitness levels mostly between average to good shape. Comment: Fitness ratings guide product recommendations, ensuring customers find treadmills that match their current fitness levels and aspirations. Income:

Insight: Income levels vary, with a significant proportion falling in middle to upper-middle income brackets. Comment: Income influences affordability and willingness to invest in higher-end treadmill models like KP781, impacting marketing and pricing strategies.

Q5.2 ANs: Age Distribution:

The age distribution of customers ranges predominantly from young adults to middle-aged individuals, with a mean around 35 years. This distribution suggests that AeroFit's customer base is primarily adults who are likely concerned about maintaining or improving their fitness levels. Gender Representation:

There is a balanced representation of both male and female customers in the dataset. This balance indicates that AeroFit's treadmills appeal to a diverse gender demographic, allowing for targeted marketing strategies that cater to both segments equally. Education Levels:

Most customers have completed at least a high school education, with some holding higher degrees. This distribution suggests that education might influence income levels and possibly the willingness to invest in higher-end treadmill models.

Relationships Between Variables: Age and Usage:

There might be a positive correlation between age and treadmill usage. Older customers might prioritize fitness as a part of healthy aging, leading to more frequent treadmill use. Education and Income:

Higher education levels might correlate with higher income levels, influencing purchasing power and the ability to invest in higher-end treadmill models. Marital Status and Purchase Decision:

Partnered individuals might engage in joint decision-making when purchasing fitness equipment like treadmills, impacting the choice of model based on shared fitness goals.

Q5.3 Ans: Univariate Plots Histograms (Age, Income):

Age: The histogram shows a distribution skewed towards younger adults and middle-aged customers, with a mean age around 35 years. This age distribution suggests that AeroFit attracts a broad demographic concerned about fitness. Income: The income histogram displays a right-skewed distribution, with most customers falling into middle to upper-middle income brackets. This indicates that AeroFit's treadmills are affordable to a diverse income range.

Countplots (Gender, Marital Status):

Gender: The countplot shows an equal representation of male and female customers, suggesting that AeroFit's treadmills appeal equally to both genders. Marital Status: There are slightly more partnered customers than singles, indicating potential joint purchasing decisions among couples.

Bivariate Plots Scatter Plot (Age vs. Income):

The scatter plot shows a weak positive correlation between age and income. Older customers tend to have higher incomes, influencing their purchasing power for higher-end treadmill models.

**Q6. Recommendations (10 Points) - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand**

```
Q6.Ans: Based on the analysis of AeroFit's
treadmill dataset, here are actionable
recommendations for the business:

Targeted Marketing Campaigns:

Action: Develop targeted campaigns based on
age demographics to appeal to both younger
adults and middle-aged customers.
Why: This approach aligns marketing efforts
```

Q6.Ans: Based on the analysis of AeroFit's treadmill dataset, here are actionable recommendations for the business:

Targeted Marketing Campaigns:

Action: Develop targeted campaigns based on age demographics to appeal to both younger

with the age groups most likely to purchase
treadmills, optimizing advertising spend.
Gender-Specific Promotions:

adults and middle-aged customers. Why: This
approach aligns marketing efforts with the age