Business Case: Aerofit - Descriptive Statistics & Probability

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

Problem Statement:

Analyze customer characteristics for each Aerofit treadmill product to provide better recommendations and understand differences across products.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
data=pd.read_csv("C:/Users/saher/Downloads/aerofit.csv")
data.head()
```

Out[2]:

	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

In [3]:
```
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
```

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

In [4]:
```
data.shape
```

Out[4]:
```
(180, 9)
```

The data has 180 rows and 9 columns

Data type of attributes:-

In [5]:
```
data.dtypes
```

Out[5]:
```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

Conversion of categorical attributes to 'category'

In [6]:
```
# Convert categorical attributes to 'category' data type
categorical_attributes = [ 'Product','Gender','MaritalStatus','Usage','Fitness
data[categorical_attributes] = data[categorical_attributes].astype('category')
print(data.dtypes)
```

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

Missing Value Detection:-

In [7]:
```
data.isna().sum()
```

Out[7]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

There are no any missing values

Statistical Summary

In [8]:
```
data.describe()
```

Out[8]:

	Age	Education	Income	Miles
count	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	53719.577778	103.194444
std	6.943498	1.617055	16506.684226	51.863605
min	18.000000	12.000000	29562.000000	21.000000
25%	24.000000	14.000000	44058.750000	66.000000
50%	26.000000	16.000000	50596.500000	94.000000
75%	33.000000	16.000000	58668.000000	114.750000
max	50.000000	21.000000	104581.000000	360.000000

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

Value Counts and Unique Attributes of Column:- **Product**

In [9]:
```
data["Product"].value_counts()
```

Out[9]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

In [10]:
```
data["Product"].unique()
```

Out[10]:
```
['KP281', 'KP481', 'KP781']
Categories (3, object): ['KP281', 'KP481', 'KP781']
```

Value Counts and Unique Attributes of Column:- **Gender**

In [11]:
```
data["Gender"].value_counts()
```

Out[11]:
```
Male      104
Female     76
Name: Gender, dtype: int64
```

In [12]:
```
data["Gender"].unique()
```

Out[12]:
```
['Male', 'Female']
Categories (2, object): ['Female', 'Male']
```

Value Counts and Unique Attributes of Column:- **Marital Status**

In [13]:
```
data["MaritalStatus"].value_counts()
```

Out[13]:
```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

In [14]:
```
data["MaritalStatus"].unique()
```

Out[14]:
```
['Single', 'Partnered']
Categories (2, object): ['Partnered', 'Single']
```

Value Counts and Unique Attributes of Column:- **Usage**

In [15]:
```
data["Usage"].value_counts()
```

Out[15]:
```
3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

In [16]:
```
data["Usage"].unique()
```

Out[16]:
```
[3, 2, 4, 5, 6, 7]
Categories (6, int64): [2, 3, 4, 5, 6, 7]
```

Value Counts and Unique Attributes of Column:- **Fitness**

In [17]:
```
data["Fitness"].value_counts()
```

Out[17]:
```
3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

In [18]:
```
data["Fitness"].unique()
```

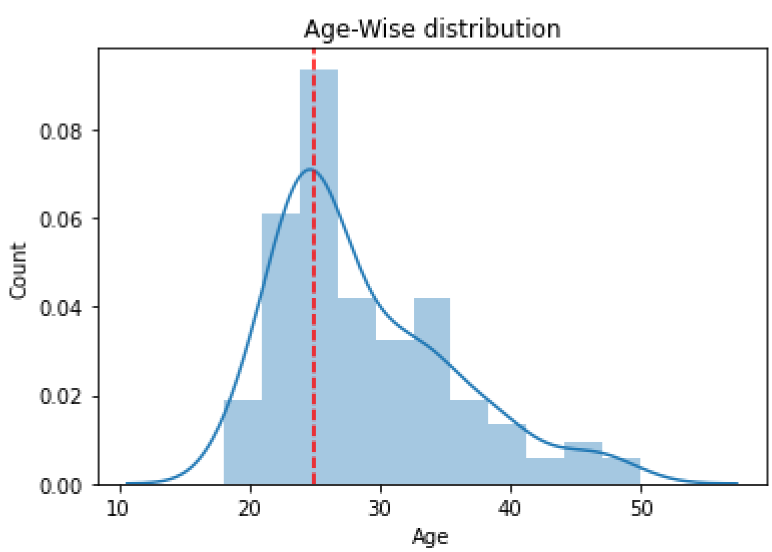
Out[18]:
```
[4, 3, 2, 1, 5]
Categories (5, int64): [1, 2, 3, 4, 5]
```

3. Visual Analysis - Univariate & Bivariate

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

Gender for Distplot

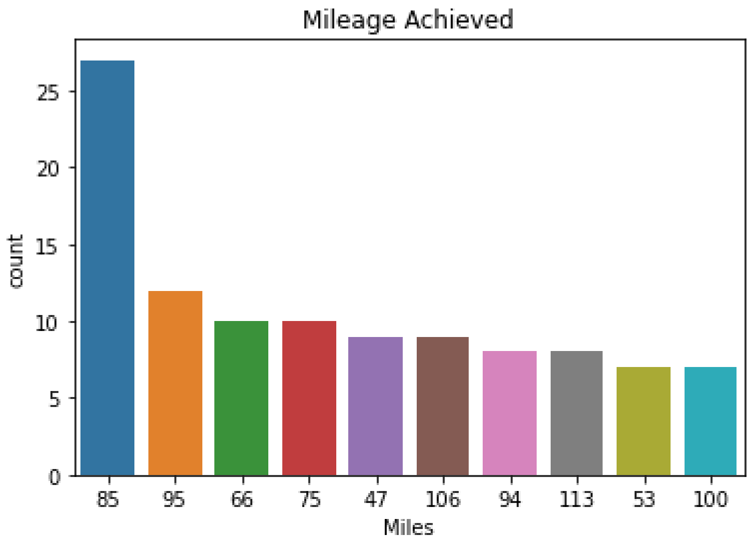
In [19]:
```
import warnings
warnings.filterwarnings('ignore')
sns.distplot(data["Age"], kde=True, hist=True)
plt.title("Age-Wise distribution")
value=25
plt.axvline(value, color='red', linestyle='--')
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
```



Maximum number of treadmill users are around *25* years of age

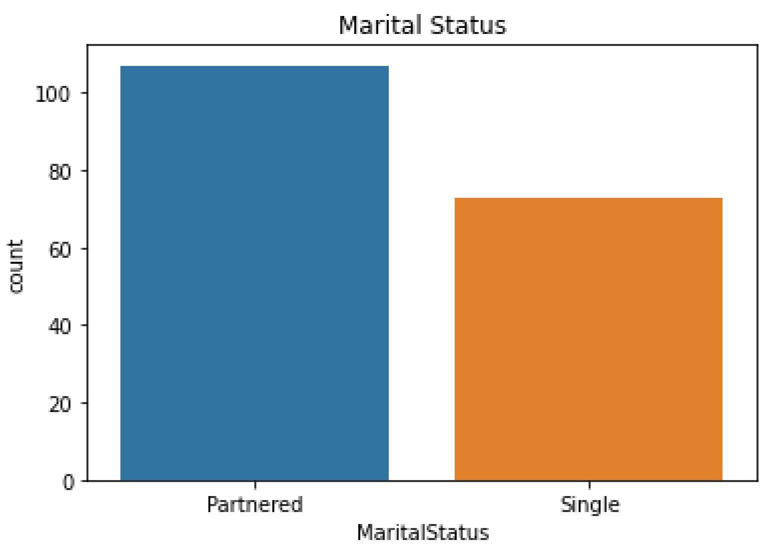
Miles for Countplot

In [20]:
```
sns.countplot(x=data['Miles'],order=data['Miles'].value_counts().index[0:10],c
plt.title("Mileage Achieved")
plt.show()
```



Maximum Mileage Achieved: *85* Miles

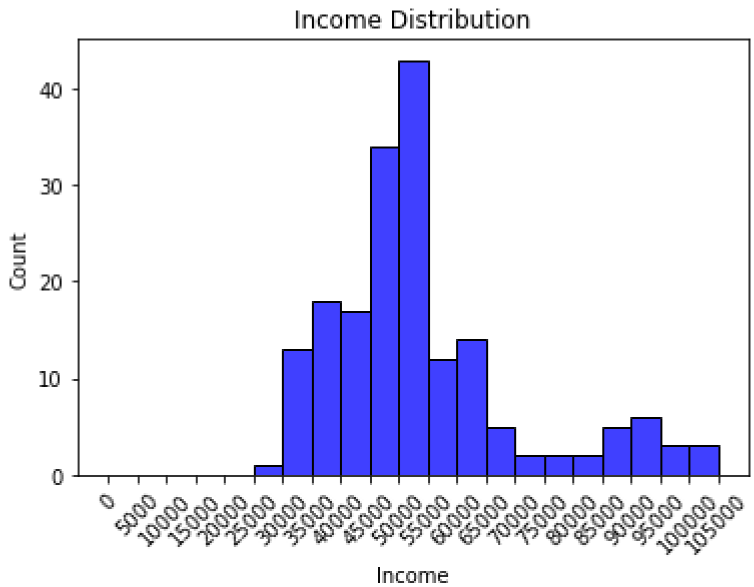
In [21]:
```
sns.countplot(x=data['MaritalStatus'],data=data)
plt.title("Marital Status")
plt.show()
```



The Majority of Treadmill Users Are Partnered

Income for Histogram

In [22]:
```
bin_edges = np.arange(0, data['Income'].max() + 5000, 5000)
sns.histplot(data=data, x='Income', bins=bin_edges, color='blue')
plt.xticks(bin_edges,rotation=45)
plt.xlabel('Income')
plt.ylabel('Count')
plt.title('Income Distribution')
plt.show()
```

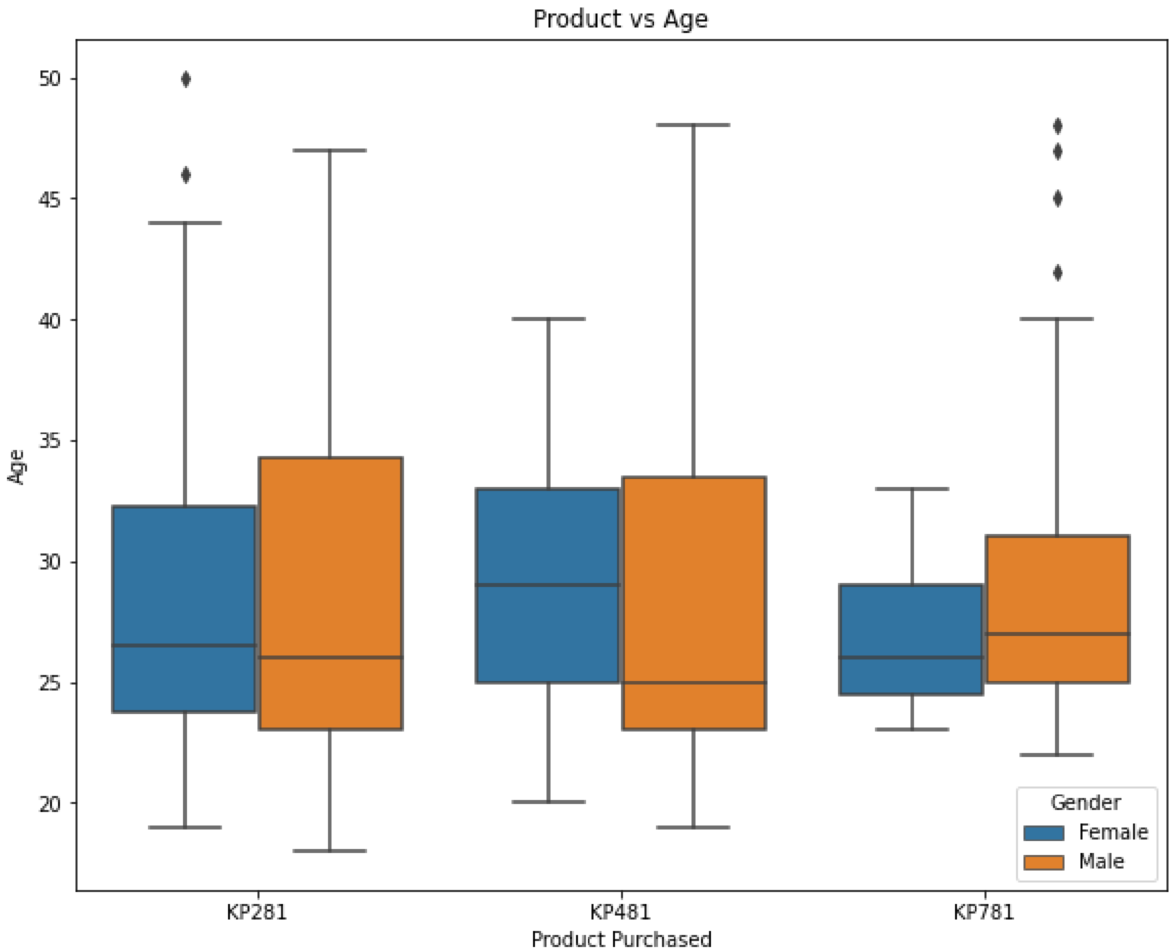


We can observe that the majority of treadmill users fall within the income range of around 45,000 to 50,000 dollars

2. For categorical variable(s): Boxplot

1. Product and Age

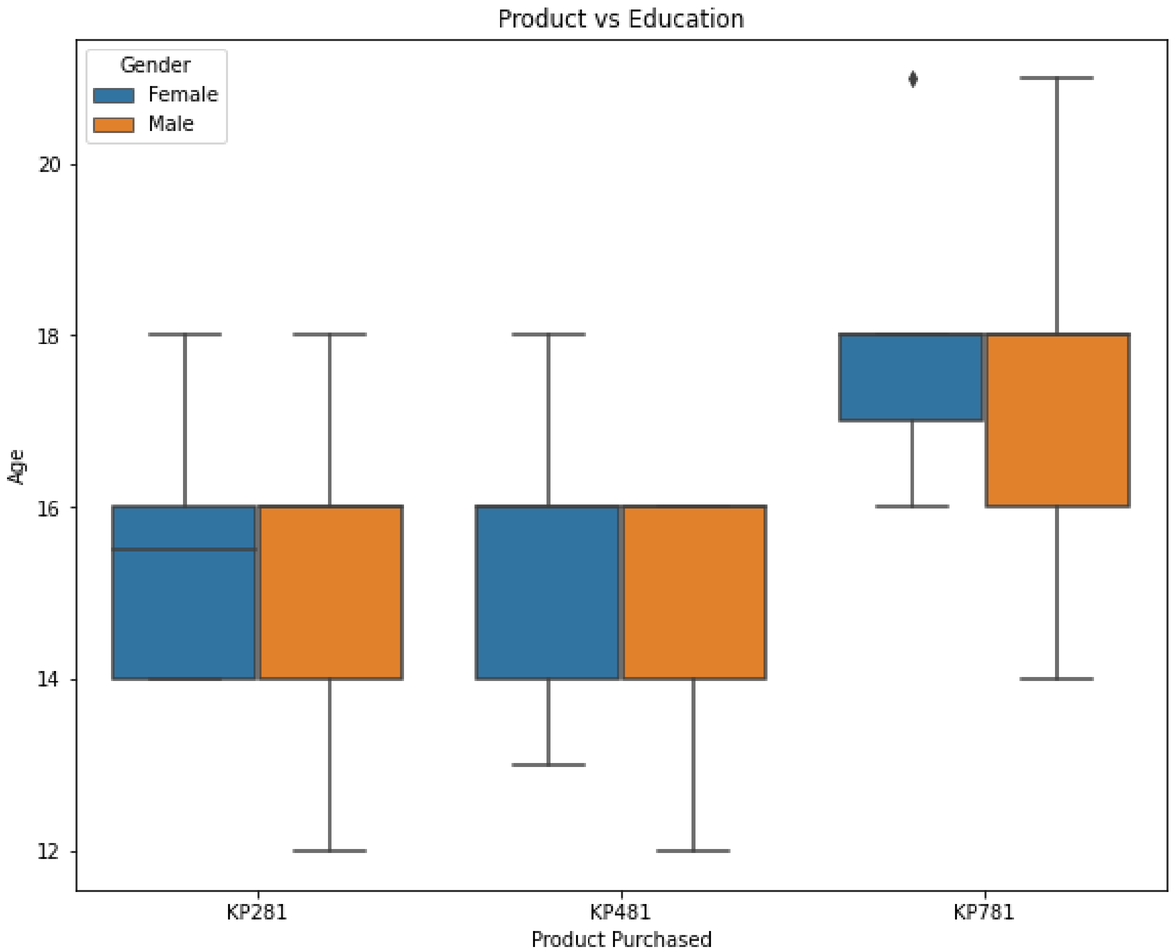
In [23]:
```
plt.figure(figsize=(10, 8))
sns.boxplot(data=data, x='Product', y='Age',hue="Gender")
plt.title('Product vs Age')
plt.xlabel('Product Purchased')
plt.ylabel('Age')
plt.show()
```



- We can see that KP281 is the most purchased product
- Customers purchasing products KP281 & KP481 are having same Age median value.
- Customers whose age lies between 25-30, are more likely to buy KP781 product

2. Product and Education

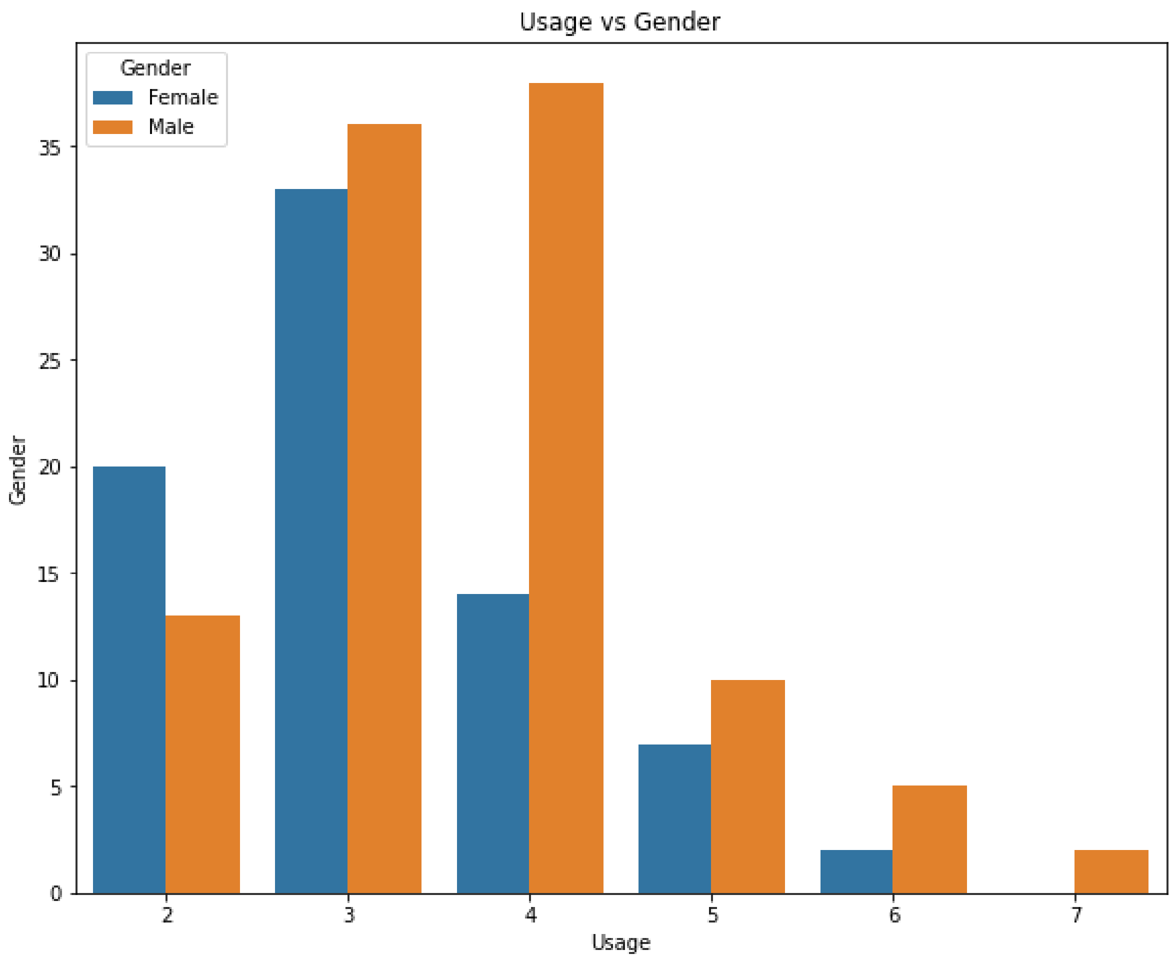
In [24]:
```
plt.figure(figsize=(10, 8))
sns.boxplot(data=data, x=data['Product'], y=data['Education'],hue="Gender")
plt.title('Product vs Education')
plt.xlabel('Product Purchased')
plt.ylabel('Age')
plt.show()
```



- Customers whose Education is greater than 16, have more chances to purchase the KP781 product.
- While the customers with Education less than 16 have equal chances of purchasing KP281 or KP481.

3. Usage and Gender

In [25]:
```
data_duplicate=data.copy()
data_duplicate["Usage"]=data_duplicate["Usage"].astype("int")
plt.figure(figsize=(10, 8))
sns.countplot(data=data_duplicate, x=data_duplicate['Usage'],hue="Gender")
plt.title('Usage vs Gender')
plt.xlabel('Usage')
plt.ylabel('Gender')
plt.show()
```

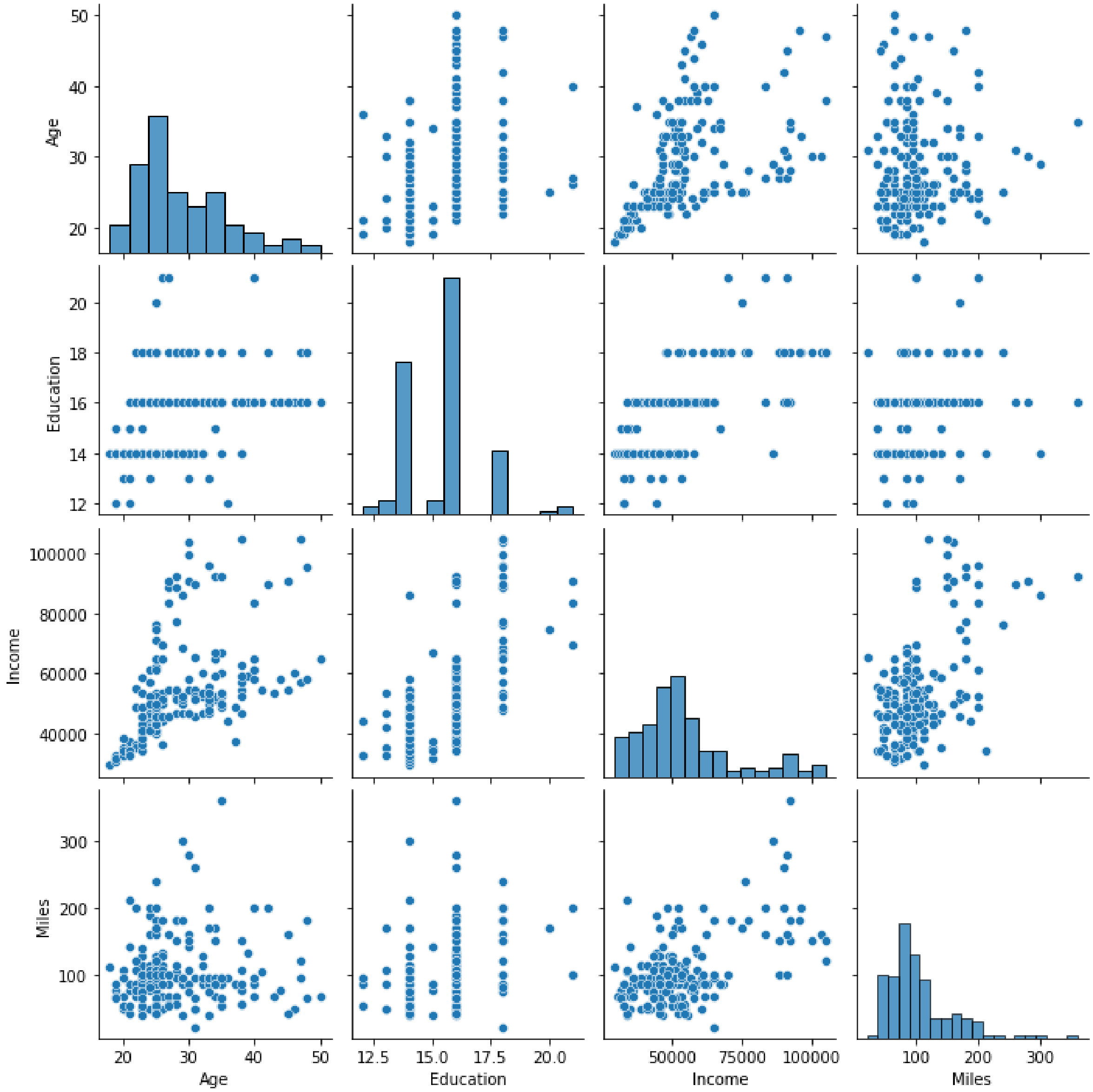


- Among Male and Female genders, Male's usage is 4 days per week
- Female customers mostly use 3 days per week
- Only few Male customers use 7 days per week whereas female customer's maximum usage is only 6 days per week

3. For correlation: Heatmaps, Pairplots

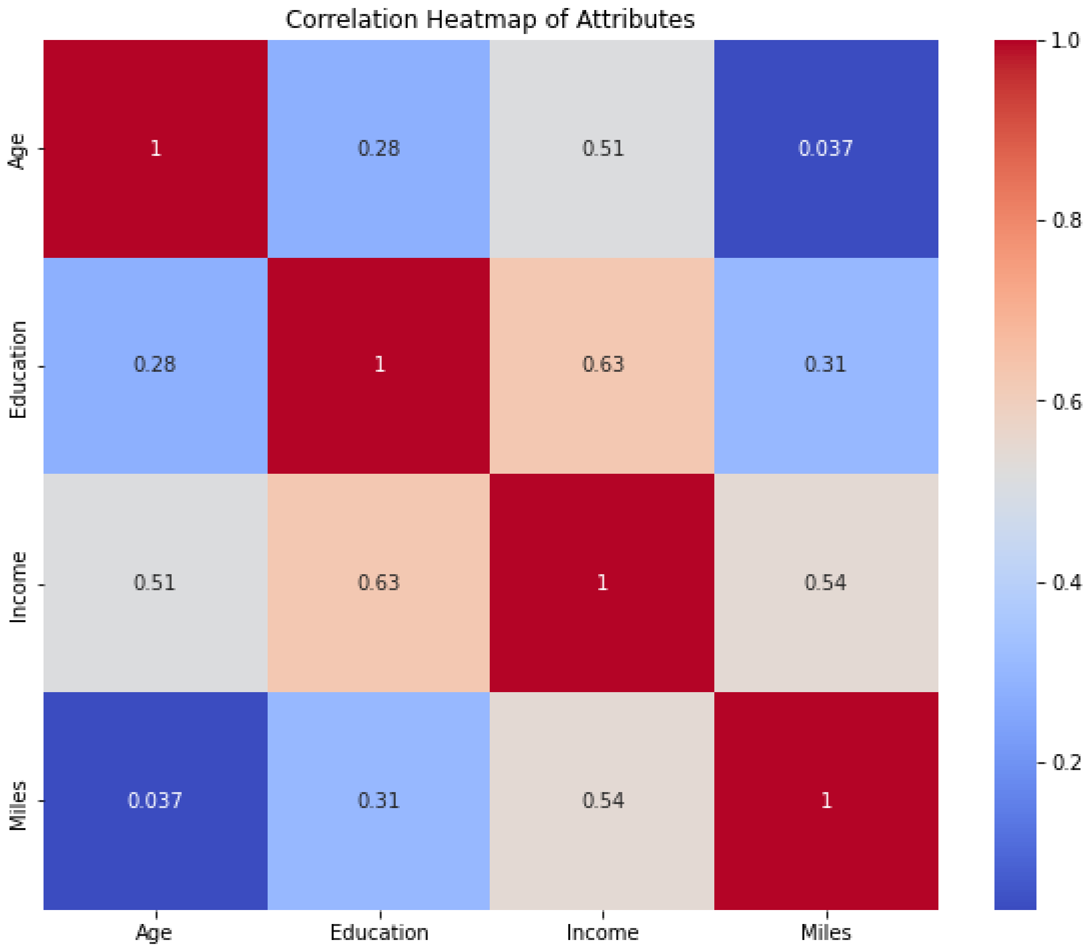
Pairplots

In [26]:
```
attrs = ['Age', 'Education', 'Income', 'Miles']
sns.pairplot(data=data[attrs])
plt.show()
```



Heatmaps

In [27]:
```
attrs = ['Age', 'Education', 'Income', 'Miles']
plt.figure(figsize=(10, 8))
correlation_matrix = data[attrs].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap of Attributes")
plt.show()
```



- Correlation between Age and Miles is 0.03
- Correlation between Education and Income is 0.62
- Correlation between Miles and Age is 0.03

4. Missing Value & Outlier check

MISSING VALUES CHECK

In [28]:
```
data.isna().sum()
```

Out[28]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

There are no any missing values in the dataset

OUTLIER CHECK

1. First, we need to calculate the IQR (Interquartile Range).
2. Next, we calculate the lower whisker as it is required for the calculation.
3. Finally, we determine the number of outlier values.

1. Age

In [29]:
```
q1_age = data['Age'].quantile(0.25)
q3_age = data['Age'].quantile(0.75)
iqr_age = q3_age - q1_age
lower_bound_age = q1_age - 1.5 * iqr_age
upper_bound_age = q3_age + 1.5 * iqr_age
outliers_age = data[(data['Age'] < lower_bound_age) | (data['Age'] > upper_bou
num_outliers_age = len(outliers_age)
print(f"Number of outliers in Age: {num_outliers_age}")
```

```
Number of outliers in Age: 5
```

2. Education

In [30]:
```
q1_education = data['Education'].quantile(0.25)
q3_education = data['Education'].quantile(0.75)
iqr_education = q3_education - q1_education
lower_bound_education = q1_education - 1.5 * iqr_education
upper_bound_education = q3_education + 1.5 * iqr_education
outliers_education = data[(data['Education'] < lower_bound_education) | (data[
num_outliers_education = len(outliers_education)
print(f"Number of outliers in Education: {num_outliers_education}")
```

```
Number of outliers in Education: 4
```

5. Business Insights based on Non-Graphical

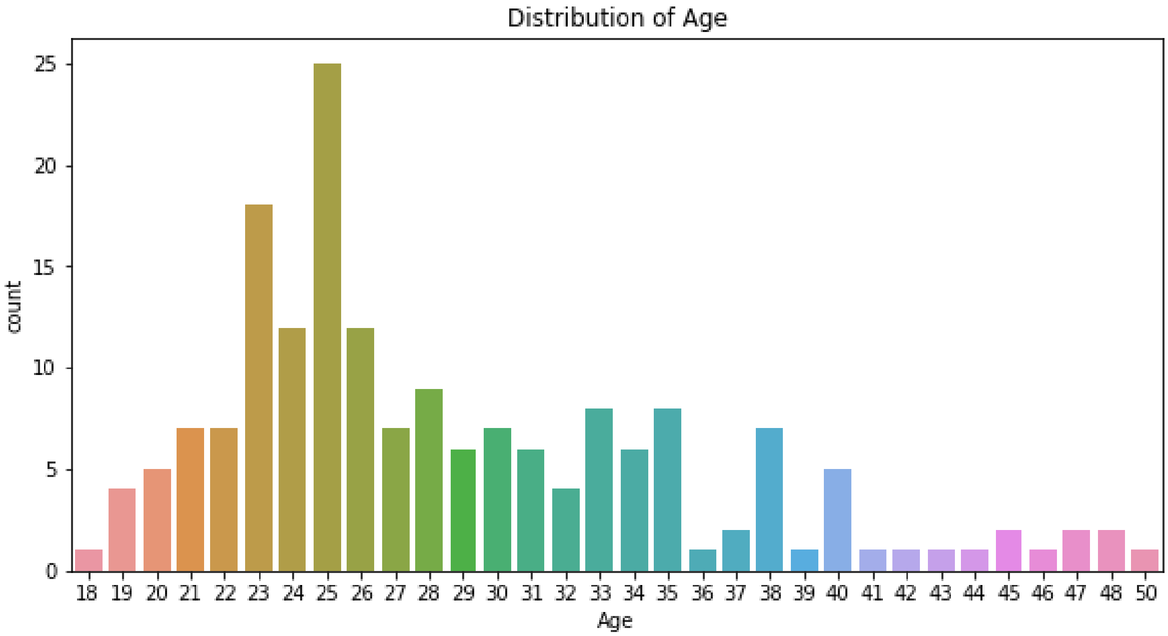
1. Comments on the range of attributes

1. Product Purchased: This attribute specifies the specific models (KP281, KP481, KP781) of the purchased product. It helps in making inventory management decisions.
2. Age: measured in years.
3. Gender: This attribute captures the customer's gender as male or female. It is important for understanding gender-based preferences and designing gender-specific product features or marketing campaigns.
4. Education: Measured in years, education level provides information about the customer's educational background.
5. MaritalStatus: This attribute captures the customer's marital status as either single or partnered. It can provide insights into lifestyle preferences for products or services catering to specific marital status groups.
6. Usage: The average number of times a customer plans to use the treadmill per week helpsin analysing usage patterns. It informs product design, features, and marketing messages to align with customer expectations.
7. Income: This attribute represents the customer's annual income in dollars. It provides valuable information for segmenting customers based on their financial capacity.
8. Fitness: Self-rated fitness on a 1-to-5 scale allows customers to indicate their perceived fitness level where 1 is the poor shape and 5 is the excellent shape,
9. Miles: The average number of miles a customer expects to walk or run per week provides insights into their anticipated exercise intensity or goals.

2. Comments on the distribution of the variables and relationship between them

1.Age

In [31]:
```
plt.figure(figsize=(10,5))
sns.countplot(data=data, x=data['Age'])
plt.title('Distribution of Age')
plt.show()
```

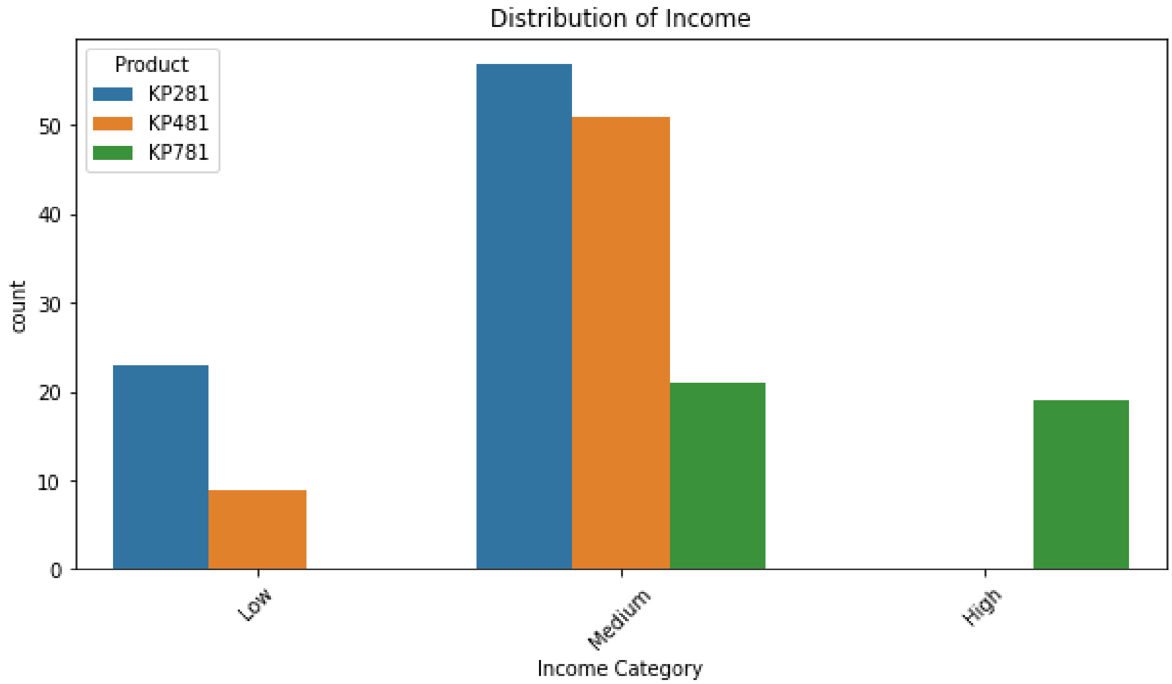


People who are 25 years of age typically uses the treadmill the most.

2. Income

In [32]:
```
data_duplicate=data.copy()
income_bins = [0, 40000, 80000, float('inf')]
income_labels = ['Low', 'Medium', 'High']
# Creating a new column 'Income Category' based on income levels
data_duplicate['Income Category'] = pd.cut(data_duplicate['Income'], bins=inco
```

In [33]:
```
plt.figure(figsize=(10,5))
sns.countplot(data=data_duplicate, x=data_duplicate['Income Category'],hue=dat
plt.title('Distribution of Income')
plt.xticks(rotation=45)
plt.show()
```



Income around 40,000 USD and 80,000 USD are prone to purchasing treadmills and KP281 is purchased the most.

6.3 Comments for each univariate and bivariate plot

UNIVARIATE PLOTS

1. Bar Plot: Bar plots are useful for visualizing the distribution or count of categorical variables, such as the distribution of age.I have used Bar plot for categorizing the age group around various products.
2. Histogram: Histograms are used to display the distribution of numeric variables.Used the income column to determine the income range of individuals who use treadmills.
3. Count Plot: Count plots are similar to bar plots but specifically designed for counting occurrences of each category in a categorical variable.I have used Count plot for categorizing maritial status.

BIVARIATE PLOTS

1. Heatmaps:-Heatmap is a graphical representation of data where values are displayed as colors in a grid-like format. I have used Heatmaps to find correlation between the variables and derived the following:-
   Correlation between Age and Miles is 0.03

Correlation between Education and Income is 0.62
Correlation between Miles and Age is 0.03

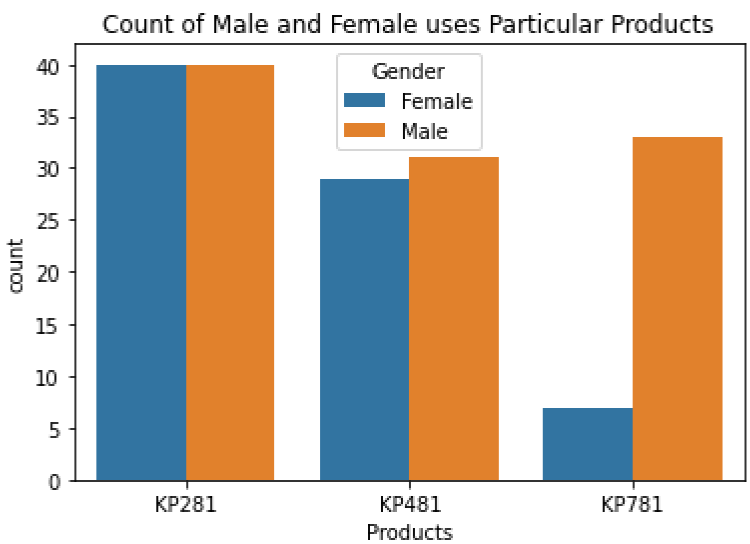
2. Pairplots:- It visualizes the relationships between variables using a pair plot.

Analysis using two - way Contingency Tables to Calculate Probabilities

(Marginal Probabilities ,Conditional Probabilities)

1. Marginal Probabilities

In [34]:
```
sns.countplot(x = "Product", data= data, hue = "Gender")
plt.xlabel("Products")
plt.title("Count of Male and Female uses Particular Products")
plt.show()
```



In [35]:
```
pd.crosstab([data.Product],data.Gender,margins=True)
```

Out[35]:

Gender	Female	Male	All
Product			
KP281	40	40	80
KP481	29	31	60
KP781	7	33	40
All	76	104	180

In [36]:
```
np.round(((pd.crosstab(data.Product,data.Gender,margins=True))/180)*100,2)
```

Out[36]:

Product \ Gender	Female	Male	All
KP281	22.22	22.22	44.44
KP481	16.11	17.22	33.33
KP781	3.89	18.33	22.22
All	42.22	57.78	100.00

Marginal Probability

- Probability of Male Customer Purchasing any product is : 57.77 %
- Probability of Female Customer Purchasing any product is : 42.22 %-

Marginal Probability of any customer buying Products

- Product KP281 is : 44.44 % (entry level product)
- Product KP481 is : 33.33 % (intermediate user level product)
- Product KP781 is : 22.22 % (Advanced product)

2. Conditional probabilities

In [37]:
```
np.round((pd.crosstab([data.Product],data.Gender,margins=True,normalize="colum
```

Out[37]:

Product \ Gender	Female	Male	All
KP281	52.63	38.46	44.44
KP481	38.16	29.81	33.33
KP781	9.21	31.73	22.22

- KP281 | Female = 52 %
- KP481 | Female = 38 %
- KP781 | Female = 10 %
- KP281 | male = 38 %
- KP481 | male = 30 %
- KP781 | male = 32 %

- Probability of Female customer buying KP281(52.63%) is more than male(38.46%).
- KP281 is more recommended for female customers.
- Probability of Male customer buying Product KP781(31.73%) is way more than female(9.21%).

- Probability of Female customer buying Product KP481(38.15%) is significantly higher than male (29.80%.)
- KP481 product is specifically recommended for Female customers who are intermediate user.

6. Recommendations

CUSTOMER PROFILING FOR EACH PRODUCT

1.KP281

- Easily affordable entry level product, which is also the maximum selling product.
- KP281 is the most popular product among the entry level customers.
- This product is easily afforded by both Male and Female customers.
- Income range between 40K to 80K have preferred this product.

2. KP481

- This is an Intermediate level Product.
- KP481 is the second most popular product among the customers.
- More Female customers prefer this product than males.
- Probability of Female customer buying KP481 is significantly higher than male.
- KP481 product is specifically recommended for Female customers who are intermediate user.

3. KP781

- Due to the High Price & being the advanced type, customer prefers less of this product.
- Customers use this product mainly to cover more distance.
- Customers who use this product have rated excelled shape as fitness rating.
- Probability of Male customer buying Product KP781(31.73%) is way more than female(9.21%).
- Average Income of KP781 buyers are over 80K per annum
- Customers who have more experience with previous aerofit products tend to buy this product

Recommendations

- Female who prefer exercising equipments are very low here. Hence, we should run a marketing campaign on to encourage women to exercise more
- KP281 & KP481 treadmills are preferred by the customers whose annual income lies in the range of 40K - 80K Dollars. These models should promoted as budget treadmills.
- As KP781 provides more features and functionalities, the treadmill should be marketed for professionals and athletes.
- KP781 product should be promoted by influencers and other international atheletes.

- To provide customer support and offer recommendations for users who are confused about what type of treadmill to purchase, taking into consideration their income and age.
- Can recommend KP781 model for female customers who engage in rigorous exercise and prefer advanced features. This treadmill offers easy usage guidance to ensure a seamless experience
- Implement customer feedback like surveys or reviews, to gather insights and suggestions from customers who have purchased the KP281, KP481, and KP781 products.
- Boost sales and maximize profit by implementing discounts on KP481 and KP781 treadmills, creating incentives and urgency for customers to purchase.
- Generate advertising campaigns to drive higher product sales and increase market demand.