Aerofit case study. Question 1 :- Import the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset

```
# importing libraries -
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
from sklearn.cluster import KMeans

# reading the data file -
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

```
from sklearn.cluster import KMeans
# getting initial info about data -
df.info()
## 9 Columns + 180 Rows
```

```
<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
```

```
# getting all the descriptive stats of data -

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

```
# Checking for Null Values
df.isnull().any()
```

```
## Found no null values

     Product          False
     Age              False
     Gender           False
     Education        False
     MaritalStatus    False
     Usage            False
     Fitness          False
     Income           False
     Miles            False
     dtype: bool
```

Observations:

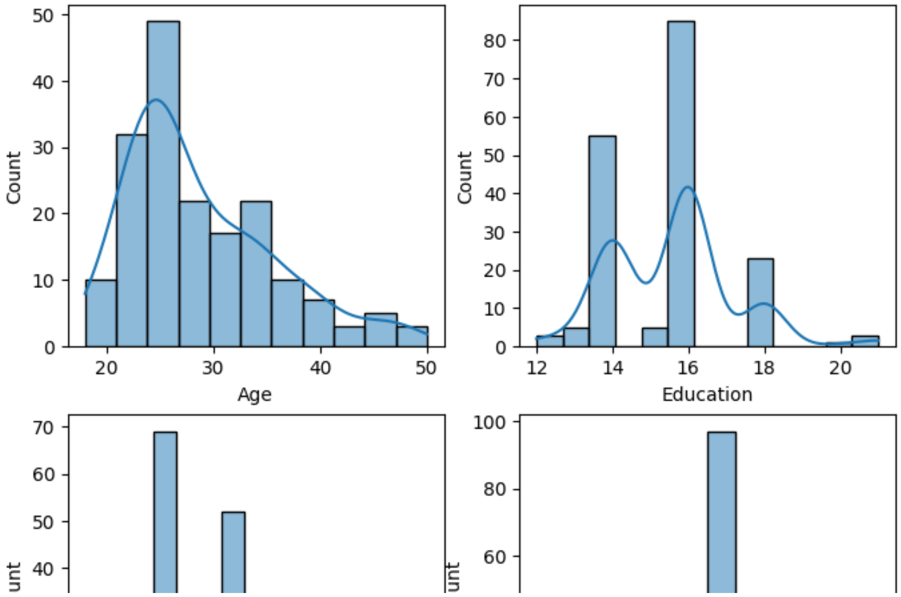
1. There are no missing values in the data.
2. There are 3 unique products in the dataset.
3. KP281 = most sold product.
4. Minimum & Maximum age of the person is 18 & 50, mean is 28.79 and 75% of persons have age less than or equal to 33.
5. Most of the people are having 16 years of education i.e., 75% of persons are having education <= 16 years.
6. Out of 180 data points, 104's gender is Male and rest are the female.

Question 2 :- Detect Outliers (using boxplot, “describe” method by checking the difference between mean and median)

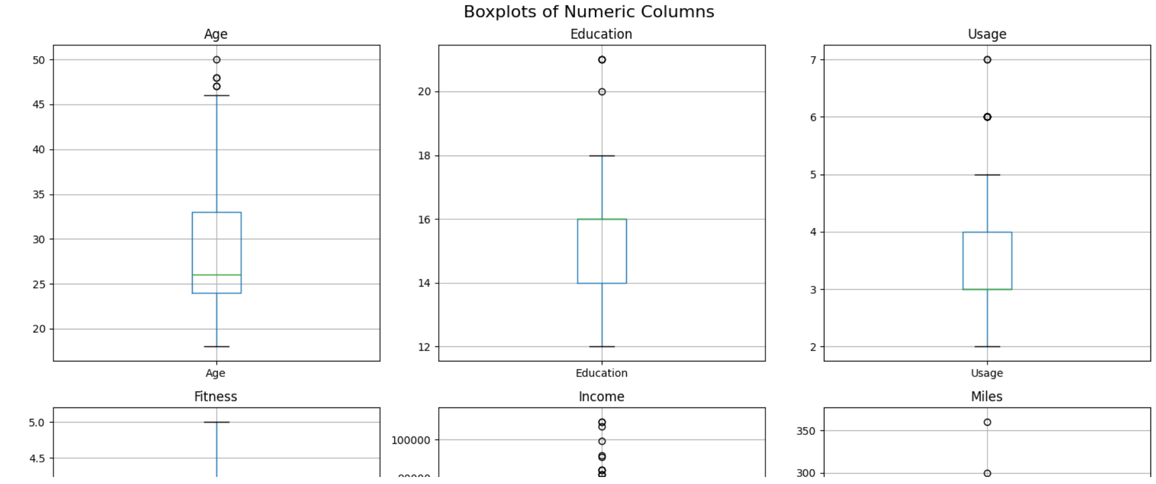
Understanding the distribution of the data for the

1. Age
2. Education
3. Usage
4. Fitness
5. Income
6. Miles

```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(8,8))
fig.subplots_adjust(top=1.2)

sns.histplot(data=df, x="Age", kde=True, ax=axis[0,0])
sns.histplot(data=df, x="Education", kde=True, ax=axis[0,1])
sns.histplot(data=df, x="Usage", kde=True, ax=axis[1,0])
sns.histplot(data=df, x="Fitness", kde=True, ax=axis[1,1])
sns.histplot(data=df, x="Income", kde=True, ax=axis[2,0])
sns.histplot(data=df, x="Miles", kde=True, ax=axis[2,1])
plt.show()
```



```
## checking for Outliers using box plot
numeric_columns = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']

fig, axes = plt.subplots(nrows=2, ncols=3, figsize=(15, 10))
fig.suptitle('Boxplots of Numeric Columns', fontsize=16)

for i, column in enumerate(numeric_columns):
    row = i // 3
    col = i % 3
    df.boxplot(column=[column], ax=axes[row, col])
    axes[row, col].set_title(column)

plt.tight_layout()
plt.show()

# Difference between Mean and Median
for column in numeric_columns:
    diff_mean_median = df[column].mean() - df[column].median()
    print(f"Column: {column}, Mean-Median Difference: {diff_mean_median}")
```



Observations: From the above box plot and difference between mean and median we can conclude

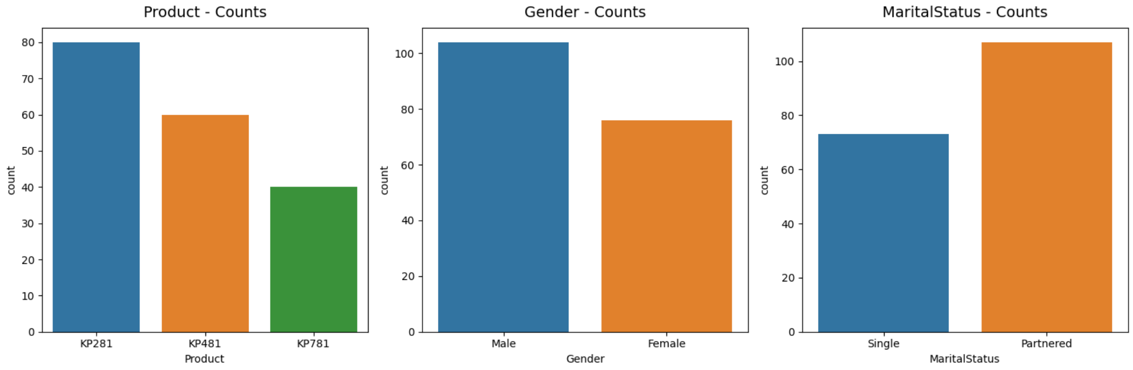
1. Income have the max outliers.
2. Miles have outliers but less then Income and more then other parameters/columns
3. Age, Education,Usage and fittness have the least outliers.

Question 3 :- Check if features like marital status, age have any effect on the product purchased (using countplot, histplots, boxplots etc)7

Understanding the distribution of the data for the qualitative attributes:

1. Product
2. Gender
3. MaritalStatus

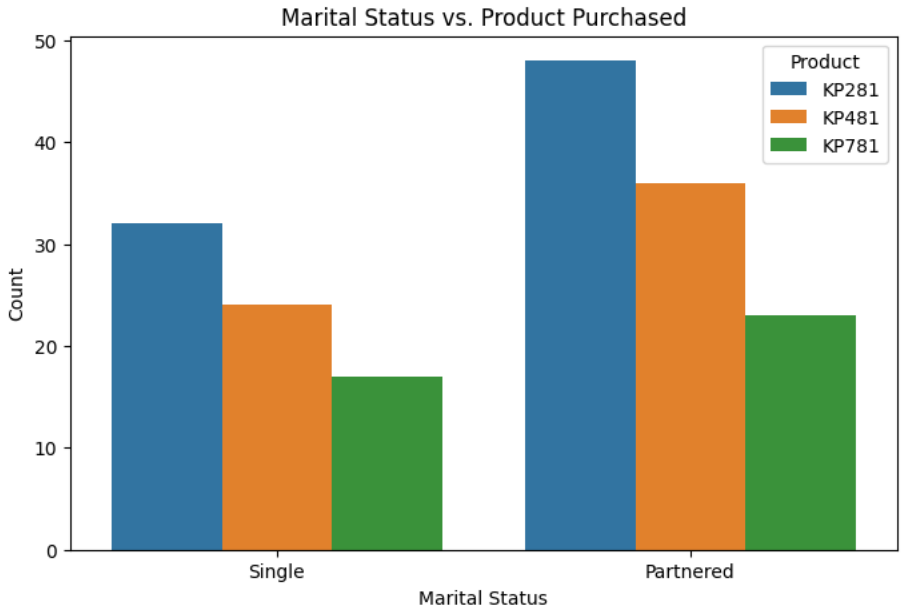
```
fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(15, 5))

sns.countplot(data=df, x='Product', ax=axs[0])
sns.countplot(data=df, x='Gender', ax=axs[1])
sns.countplot(data=df, x='MaritalStatus', ax=axs[2])

titles = ["Product - Counts", "Gender - Counts", "MaritalStatus - Counts"]
for ax, title in zip(axs, titles):
    ax.set_title(title, pad=10, fontsize=14)

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

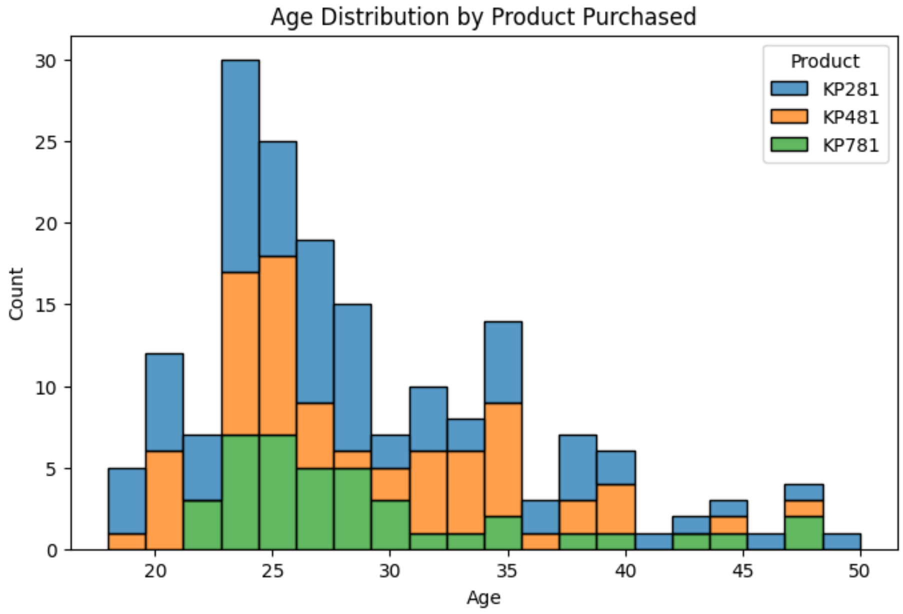


Observations:-

1. KP281 is the most sold product.
2. Aerofit have more male customers.
3. Patnered people have more purchase as compared to singles.

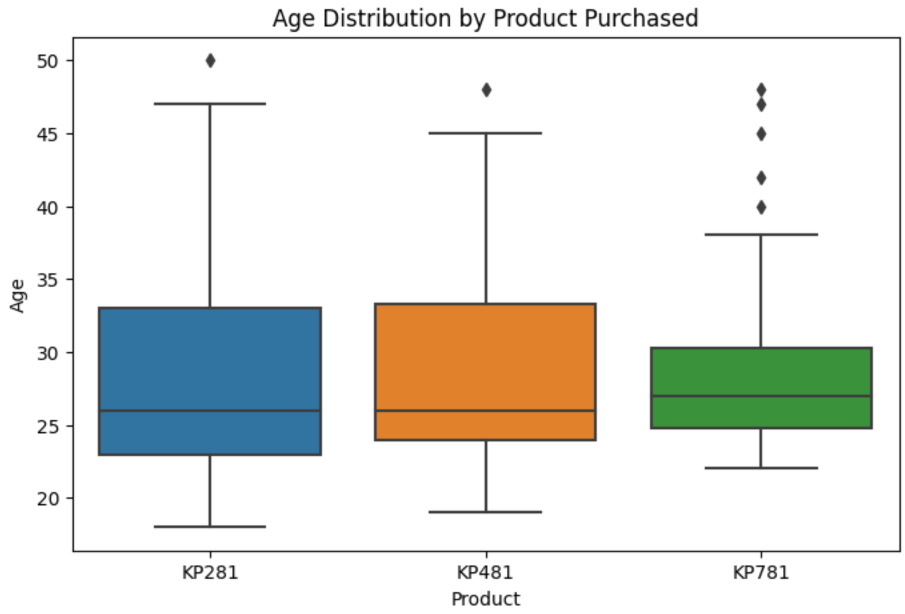
```
# Countplot of Marital Status vs. Product
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='MaritalStatus', hue='Product')
plt.title('Marital Status vs. Product Purchased')
plt.xlabel('Marital Status')
plt.ylabel('Count')
plt.show()

# Histogram of Age for each Product
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='Age', hue='Product', multiple='stack', bins=20)
plt.title('Age Distribution by Product Purchased')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```





```
# Boxplot of Age vs. Product
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x='Product', y='Age')
```

```
plt.title('Age Distribution by Product Purchased')
plt.xlabel('Product')
plt.ylabel('Age')
plt.show()
```



Observation:-

1. Patnered consumers are more likely to buy a product from aerofit
2. Maxiumn customer lies between range of 23(25%tile) to 33(75%tile)

Question 4 :- Representing the marginal probability like - what percent of customers have purchased KP281, KP481, or KP781 in a table (can use pandas.crosstab here)

```
marginal_probabilities = pd.crosstab(index=df['Product'], columns='count', normalize='columns') * 100
marginal_probabilities.columns = ['Marginal Probability (%)']
print(marginal_probabilities)
```

```
         Marginal Probability (%)
Product
KP281                   44.444444
KP481                   33.333333
KP781                   22.222222
```

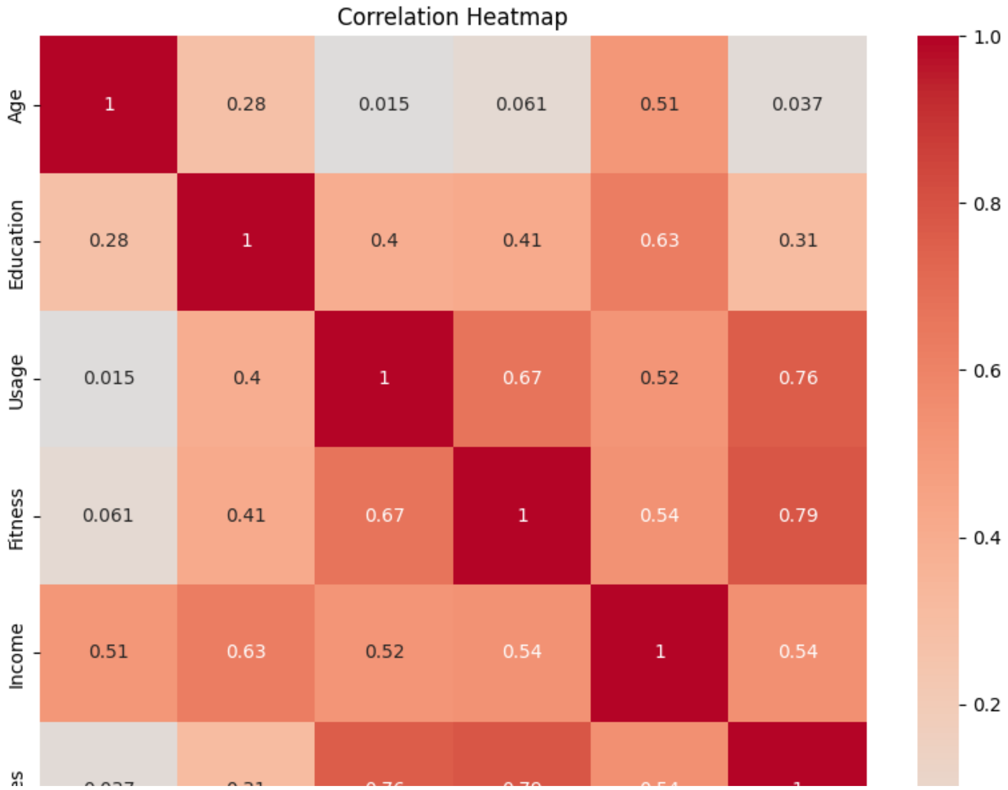
Observation:- The above crosstab- clearly shows

1. KP781 have least Marginal Probability = 22.22%
2. KP281 have highest Marginal Probability = 44.44%

Question 5 :- Check correlation among different factors using heat maps or pair plots.

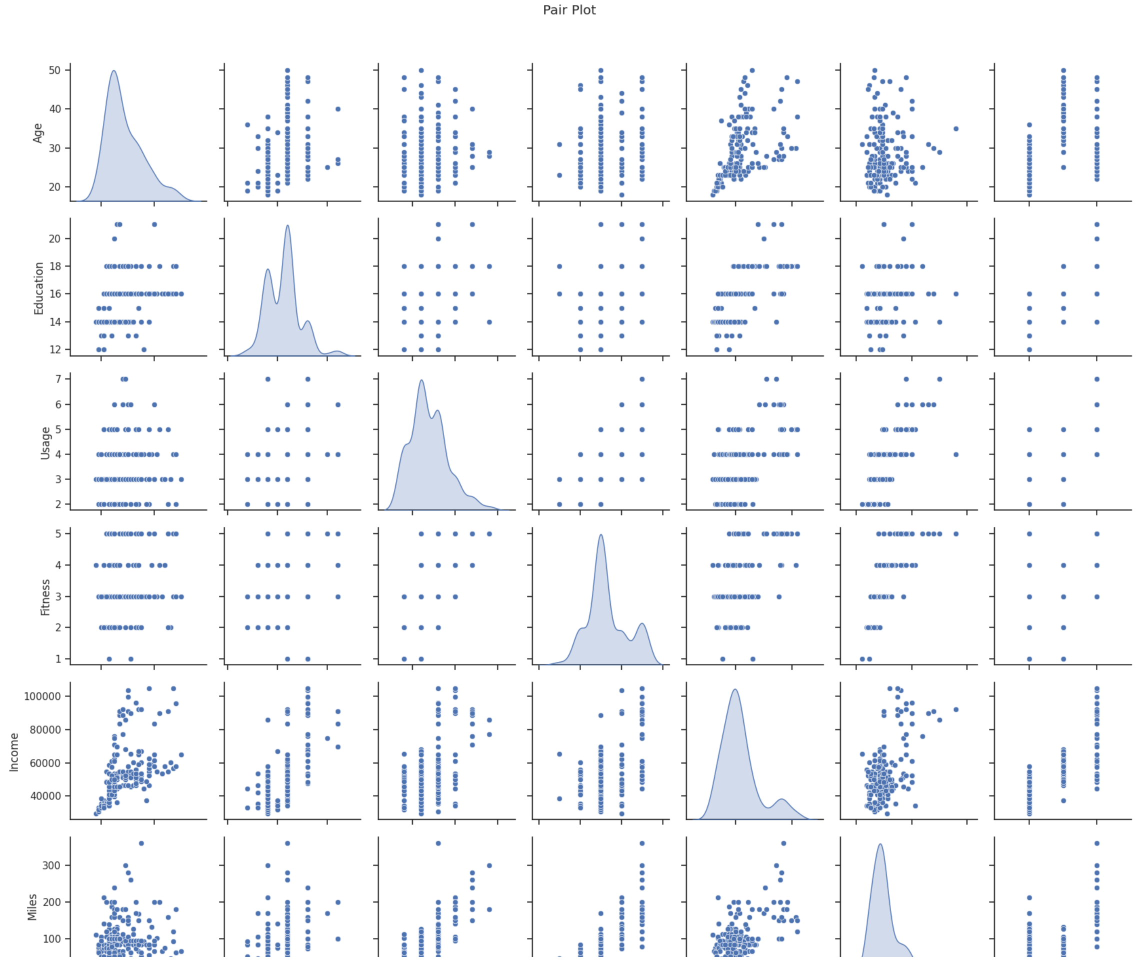
```
correlation_matrix = df.corr()

# Heatmap of correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Heatmap')
plt.show()
```

```
<ipython-input-33-54d01250af47>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprec
  correlation_matrix = df.corr()
```

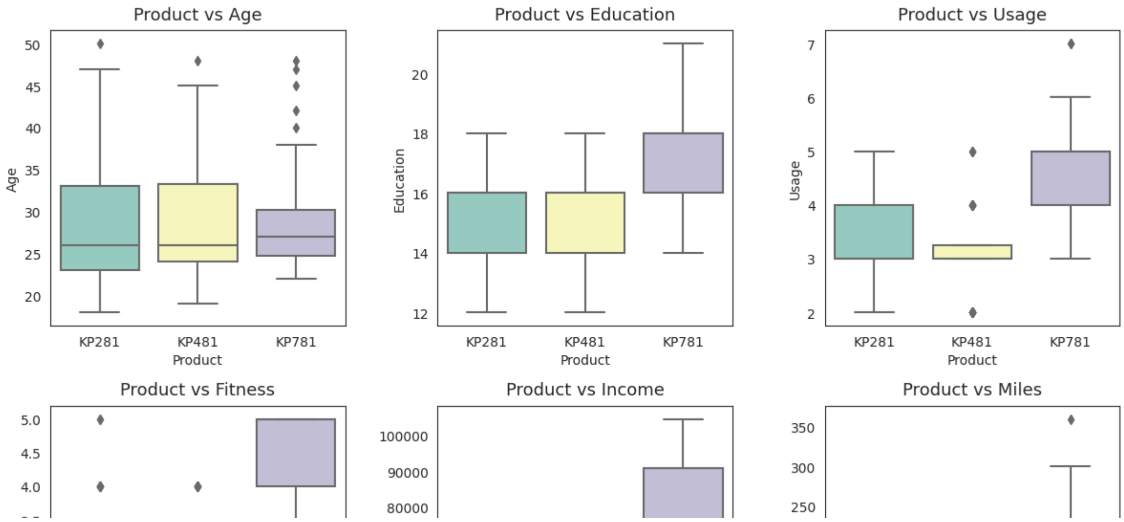


```
sns.set(style="ticks")
plt.figure(figsize=(6, 6))
sns.pairplot(df, diag_kind='kde')

plt.suptitle('Pair Plot', y=1.02)
plt.tight_layout()
plt.show()
```

```
<Figure size 600x600 with 0 Axes>
```



```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")

fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(12, 8))
fig.subplots_adjust(top=1.2)
count = 0

for i in range(2):
    for j in range(3):
        sns.boxplot(data=df, x='Product', y=attrs[count], ax=axs[i, j], palette='Set3')
        axs[i, j].set_title(f"Product vs {attrs[count]}", pad=8, fontsize=13)
        count += 1

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



Observations

1. **Product vs Age**

- 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 vs Education**

- 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. Product vs Usage

- Customers who are planning to use the treadmill greater than 4 times a week, are more likely to purchase the KP781 product.
- While the other customers are likely to purchasing KP281 or KP481.

4. Product vs Fitness

- The more the customer is fit (fitness >= 3), higher the chances of the customer to purchase the KP781 product.

5. Product vs Income

- Higher the Income of the customer (Income >= 60000), higher the chances of the customer to purchase the KP781 product.

6. Product vs Miles

- If the customer expects to walk/run greater than 120 Miles per week, it is more likely that the customer will buy KP781 product.

Question 6 :- With all the above steps you can answer questions like: What is the probability of a male customer buying a KP781 treadmill?

```
total_male_customers = len(df[df['Gender'] == 'Male'])
male_customers_with_KP781 = len(df[(df['Gender'] == 'Male') & (df['Product'] == 'KP781')])
probability_male_buying_KP781 = male_customers_with_KP781 / total_male_customers

print("Probability of a male customer buying a KP781 treadmill:", probability_male_buying_KP781)
```

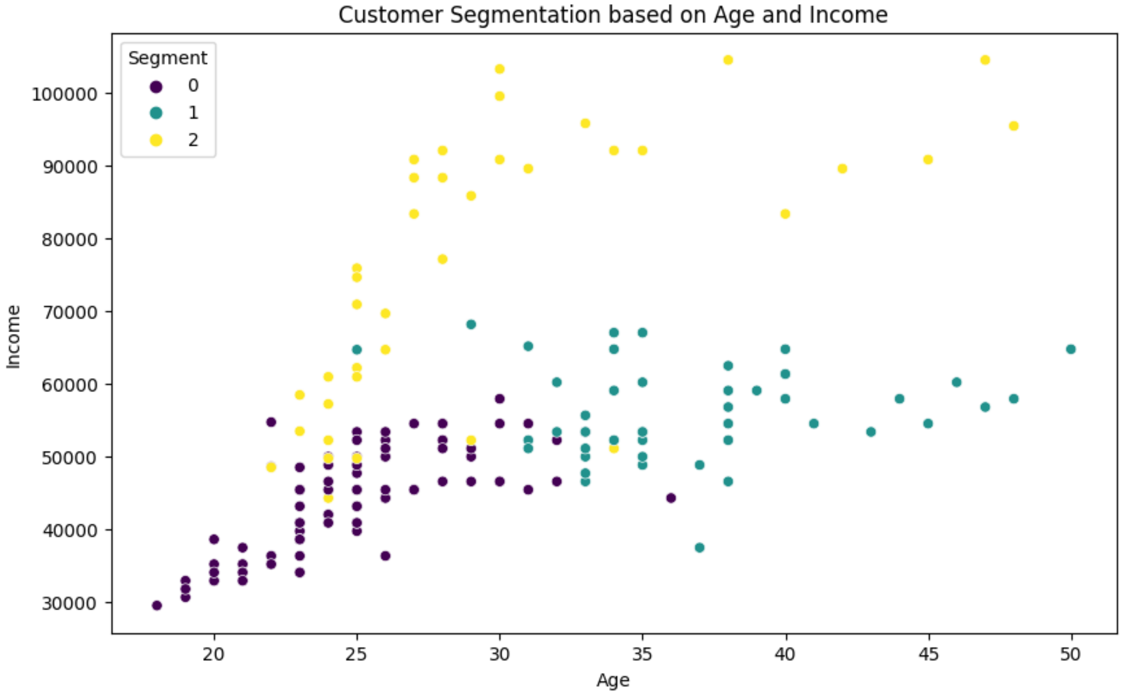
```
    Probability of a male customer buying a KP781 treadmill: 0.3173076923076923
```

Question 7:- Customer Profiling - Categorization of users.

```
# Selecting relevant features for customer profiling
selected_features = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']

# Normalizing the selected features
normalized_data = (df[selected_features] - df[selected_features].mean()) / df[selected_features].std()

# Using K-means clustering to segment customers
num_clusters = 3
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
df['Segment'] = kmeans.fit_predict(normalized_data)
```

```
# Visualize the distribution of segments based on age and income
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='Age', y='Income', hue='Segment', palette='viridis')
plt.title('Customer Segmentation based on Age and Income')
plt.show()

# Display segment details
segment_details = df.groupby('Segment')[selected_features].mean()
print(segment_details)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_
  warnings.warn(
```



```
               Age  Education     Usage   Fitness        Income       Miles
Segment
0        24.711340  14.628866  3.113402  2.958763  43718.010309   86.659794
1        36.777778  16.066667  3.000000  2.888889  56276.488889   79.155556
2        29.736842  17.394737  4.868421  4.710526  76221.973684  173.868421
```

Observation:-

Segment 0: 1.Age: Customers in this segment are relatively younger, with an average age of around 25 years.

2. Education: Education level is around 14-15 years, which could correspond to high school or some college education.
3. Usage: These customers use the products around 3 times a week on average.
4. Fitness: Fitness level is moderate, around 3 on average.
5. Income: The average income is lower, around $43,718.
6. Miles: These customers tend to cover fewer miles, with an average of 87 miles.

Segment 1:

1. Age: Customers in this segment are older, with an average age of around 37 years.
2. Education: Education level is higher, around 16 years or more, indicating more advanced education.
3. Usage: These customers also use the products around 3 times a week on average.
4. Fitness: Fitness level is slightly higher than in Segment 0, around 2.89 on average.
5. Income: The average income is higher, around $56,276.
6. Miles: These customers cover fewer miles, with an average of 79 miles.

Segment 2:

1. Age: Customers in this segment have an average age of around 30 years.
2. Education: Education level is higher, around 17 years, indicating a more educated group.

3. Usage: These customers use the products more frequently, around 4-5 times a week on average.
4. Fitness: Fitness level is higher, around 4.71 on average, indicating a health-conscious group.
5. Income: The average income is the highest among the segments, around $76,222.
6. Miles: These customers cover a relatively larger distance, with an average of 174 miles.

```
# Calculate total number of customers for each product
total_customers_KP281 = len(df[df['Product'] == 'KP281'])
total_customers_KP481 = len(df[df['Product'] == 'KP481'])
total_customers_KP781 = len(df[df['Product'] == 'KP781'])

# Probability of Male customers for each product
marginal_prob_male_KP281 = len(df[(df['Product'] == 'KP281') & (df['Gender'] == 'Male')]) / total_customers_KP281
marginal_prob_male_KP481 = len(df[(df['Product'] == 'KP481') & (df['Gender'] == 'Male')]) / total_customers_KP481
marginal_prob_male_KP781 = len(df[(df['Product'] == 'KP781') & (df['Gender'] == 'Male')]) / total_customers_KP781

# Probability of Female customers for each product
marginal_prob_female_KP281 = len(df[(df['Product'] == 'KP281') & (df['Gender'] == 'Female')]) / total_customers_KP281
marginal_prob_female_KP481 = len(df[(df['Product'] == 'KP481') & (df['Gender'] == 'Female')]) / total_customers_KP481
marginal_prob_female_KP781 = len(df[(df['Product'] == 'KP781') & (df['Gender'] == 'Female')]) / total_customers_KP781

# Probability of Partnered customers for each product
marginal_prob_partnered_KP281 = len(df[(df['Product'] == 'KP281') & (df['MaritalStatus'] == 'Partnered')]) / total_customers_KP281
marginal_prob_partnered_KP481 = len(df[(df['Product'] == 'KP481') & (df['MaritalStatus'] == 'Partnered')]) / total_customers_KP481
marginal_prob_partnered_KP781 = len(df[(df['Product'] == 'KP781') & (df['MaritalStatus'] == 'Partnered')]) / total_customers_KP781

# Probability of Single customers for each product
marginal_prob_single_KP281 = len(df[(df['Product'] == 'KP281') & (df['MaritalStatus'] == 'Single')]) / total_customers_KP281
marginal_prob_single_KP481 = len(df[(df['Product'] == 'KP481') & (df['MaritalStatus'] == 'Single')]) / total_customers_KP481
marginal_prob_single_KP781 = len(df[(df['Product'] == 'KP781') & (df['MaritalStatus'] == 'Single')]) / total_customers_KP781

print("Probability of Male customers for KP281:", marginal_prob_male_KP281)
print("Probability of Male customers for KP481:", marginal_prob_male_KP481)
print("Probability of Male customers for KP781:", marginal_prob_male_KP781)

print("Probability of Female customers for KP281:", marginal_prob_female_KP281)
print("Probability of Female customers for KP481:", marginal_prob_female_KP481)
print("Probability of Female customers for KP781:", marginal_prob_female_KP781)

print("Probability of Partnered customers for KP281:", marginal_prob_partnered_KP281)
print("Probability of Partnered customers for KP481:", marginal_prob_partnered_KP481)
print("Probability of Partnered customers for KP781:", marginal_prob_partnered_KP781)

print("Probability of Single customers for KP281:", marginal_prob_single_KP281)
print("Probability of Single customers for KP481:", marginal_prob_single_KP481)
print("Probability of Single customers for KP781:", marginal_prob_single_KP781)
```

```
Probability of Male customers for KP281: 0.5
Probability of Male customers for KP481: 0.5166666666666667
Probability of Male customers for KP781: 0.825
Probability of Female customers for KP281: 0.5
Probability of Female customers for KP481: 0.48333333333333334
Probability of Female customers for KP781: 0.175
Probability of Partnered customers for KP281: 0.6
Probability of Partnered customers for KP481: 0.6
Probability of Partnered customers for KP781: 0.575
Probability of Single customers for KP281: 0.4
Probability of Single customers for KP481: 0.4
Probability of Single customers for KP781: 0.425
```

Question - **Gender Distribution for Each Product:**

1. For both KP281 and KP781 treadmills, the gender distribution is almost equal, with a marginal probability of 0.5 for both Male and Female customers.
2. For KP481 treadmill, the marginal probability of Male customers (0.516) is slightly higher than Female customers (0.483), indicating a slightly higher preference for this product among Male customers.

Marital Status Distribution for Each Product:

1. The distribution of marital status varies among the products.
2. For KP281, the marginal probability of Partnered customers is 0.6, while that of Single customers is 0.4. This suggests a higher preference for KP281 among Partnered customers.

3. For KP481, the distribution is even, with a marginal probability of 0.6 for both Partnered and Single customers.
4. For KP781, the marginal probability of Partnered customers is 0.575, and that of Single customers is 0.425. This indicates a relatively higher preference for KP781 among Partnered customers.

Question 9:- Some recommendations and actionable insights, based on the inferences.

Targeted Marketing and Product Promotion:

- Since KP281 and KP781 treadmills are preferred by both Male and Female customers equally, consider creating gender-neutral marketing campaigns to reach a wider audience.
- For KP481 treadmill, focus on promoting features that specifically appeal to Male customers, as they show a slightly higher preference for this product.

Product Development and Improvement:

- Given that customers in Segment 2 have a higher fitness level and usage frequency, consider developing advanced treadmill models with features that cater to their fitness-conscious preferences.

Pricing Strategy:

- For KP781 treadmill, customers with higher incomes have a greater preference. Consider offering premium features or packages for this product to align with the higher income levels.

Customer Experience Enhancement:

- Based on the marital status distribution, you could offer family-oriented promotions or group packages for customers in Segment 0 who are more likely to be single.

Retention Strategies:

- Develop loyalty programs that offer discounts or rewards for customers who purchase multiple products from the company's range.

Cross-Selling Opportunities:

- If a customer purchases a certain product, recommend related accessories or complementary products to enhance their fitness experience.

Feedback Collection:

- Engage with customers to gather feedback on product features, quality, and overall satisfaction. Use this feedback to continuously improve products and services.

Segment-Specific Campaigns:

- Create specialized campaigns targeting each segment's preferences. For example, focus on fitness features for Segment 2 and convenience features for others.

Product Differentiation:

- Ensure that each product has distinct features and benefits to cater to various customer preferences. Highlight these differences in marketing materials.