Business Case: Aerofit - Descriptive Statistics & Probability

🔮 Introduction

🏃 About Aerofit

Aerofit is a leading brand in the field of fitness equipment. Aerofit provides a product range including machines such as treadmills, exercise bikes, gym equipment, and fitness accessories to cater to the needs of all categories of people.

🎯 Objective

To conduct a comprehensive analysis of the Aerofit customer dataset to identify distinct customer segments for each treadmill product (KP281, KP481, and KP781) by employing descriptive analytics and probability techniques. This analysis will inform the development of targeted marketing strategies and optimized product recommendations.

📁 Dataset

The company collected the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months. The data is available in a csv file

Product Portfolio:

- The `KP281` is an entry-level treadmill that sells for `$1,500`.
- The `KP481` is for mid-level runners that sell for `$1,750`.
- The `KP781` treadmill is having advanced features that sell for `$2,500`.

🗃️ Features of the dataset.

Feature	Description
Product	Product Purchased: KP281, KP481, or KP781
Age	Age of buyer in years
Gender	Gender of buyer (Male/Female)
Education	Education of buyer in years
MaritalStatus	MaritalStatus of buyer (Single or partnered)
Usage	The average number of times the buyer plans to use the treadmill each week
Income	Annual income of the buyer (in $)
Fitness	Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 is the excellent shape
Miles	The average number of miles the buyer expects to walk/run each week

📚 Import Necessary Libraries:

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

📁 Loading the Dataset:

```
# Download the data
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749 -O aerofit_
```

```
--2024-08-14 18:17:54--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 18.64.229.71, 18.64.229.135, 18.64.229.91, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|18.64.229.71|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7279 (7.1K) [text/plain]
Saving to: ‘aerofit_treadmill.csv’

aerofit_treadmill.c 100%[===================>]   7.11K  --.-KB/s    in 0s

2024-08-14 18:17:54 (273 MB/s) - ‘aerofit_treadmill.csv’ saved [7279/7279]
```

```python
# Read the CSV file into a Pandas DataFrame
df = pd.read_csv("aerofit_treadmill.csv")

# Display the first few rows of the DataFrame
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 | New interactive sheet

1. 🔍 Exploratory Data Analysis (EDA):

```python
# The data type of all columns in the “customers” table.
df.dtypes
```

	0
Product	object
Age	int64
Gender	object
Education	int64
MaritalStatus	object
Usage	int64
Fitness	int64
Income	int64
Miles	int64

```python
# The number of rows and columns given in the dataset
df.shape
```

```
(180, 9)
```

```python
#number of dimensions
df.ndim
```

```
2
```

```python
# Check for the missing values and find the number of missing values in each column
df.isnull().sum()
```

	0
Product	0
Age	0
Gender	0
Education	0
MaritalStatus	0
Usage	0
Fitness	0
Income	0
Miles	0

```
# Checking for duplicate rows in the dataset
df.duplicated().sum()
```

```
0
```

💡 Insights:

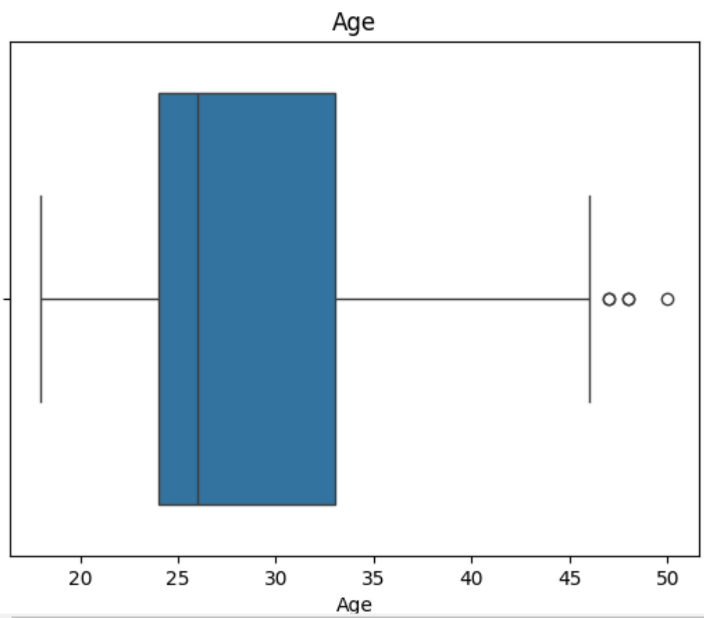
- The dataset contains **information about customers**, with columns for product, age, gender, education, marital status, usage, fitness, income, and miles.
- There are **no missing values** and **no duplicate found** in the dataset.
- The dataset consists of **180 customers** and **9 attributes**.
- All columns currently have data types that align with their content. However, for analysis purposes, the **'Usage'** and **'Fitness'** columns will be converted to **string** format.

🕵️ 2. Detect Outliers

👫 Age Column Outliers

```
# Ploting a boxplot
sns.boxplot(x=df["Age"])
plt.title("Age")
```

```
Text(0.5, 1.0, 'Age')
```



```
q1 = df['Age'].quantile(0.25)
q3 = df['Age'].quantile(0.75)
IQR = q3 - q1
df[(df['Age'] < (q1 - 1.5*IQR)) | (df['Age'] > (q3 + 1.5*IQR))]
```

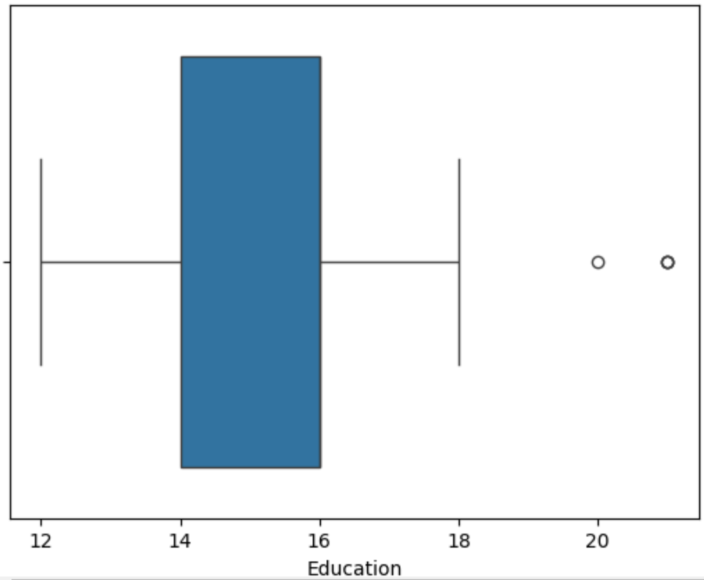
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
78	KP281	47	Male	16	Partnered	4	3	56850	94
79	KP281	50	Female	16	Partnered	3	3	64809	66
139	KP481	48	Male	16	Partnered	2	3	57987	64
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

🔍 Insights

- 85% of the customers fall in the age range of 18 to 35. with a median age of 26, suggesting young people showing more interest in the companies products
- **Outliers**
  - As we can see from the box plot, there are 3 outlier's present in the age data.

👨‍🎓 Education column outliers

```
sns.boxplot(data = df, x = 'Education')
plt.show()
```




```
q1 = df['Education'].quantile(0.25)
q3 = df['Education'].quantile(0.75)
IQR = q3 - q1
df[(df['Education'] < (q1 - 1.5*IQR)) | (df['Education'] > (q3 + 1.5*IQR))]
```

	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

🔍 Insights

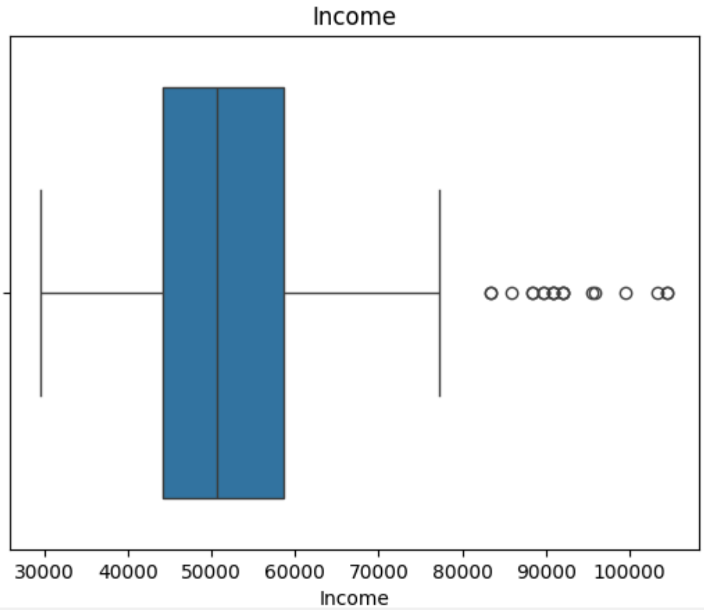
- 98% of the customers have education more than 13 years highlighting a strong inclination among well-educated individuals to purchase the products. It's plausible that health awareness driven by education could play a pivotal role in this trend.

- **Outliers**
  - As we can see from the box plot, there are `2 outlier's` present in the education data.

💰 Income column outliers

```
# Ploting a boxplot
sns.boxplot(x=df["Income"])
plt.title("Income")
```

```
Text(0.5, 1.0, 'Income')
```




```
q1 = df['Income'].quantile(0.25)
q3 = df['Income'].quantile(0.75)
IQR = q3 - q1
df[(df['Income'] < (q1 - 1.5*IQR)) | (df['Income'] > (q3 + 1.5*IQR))]
```

	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

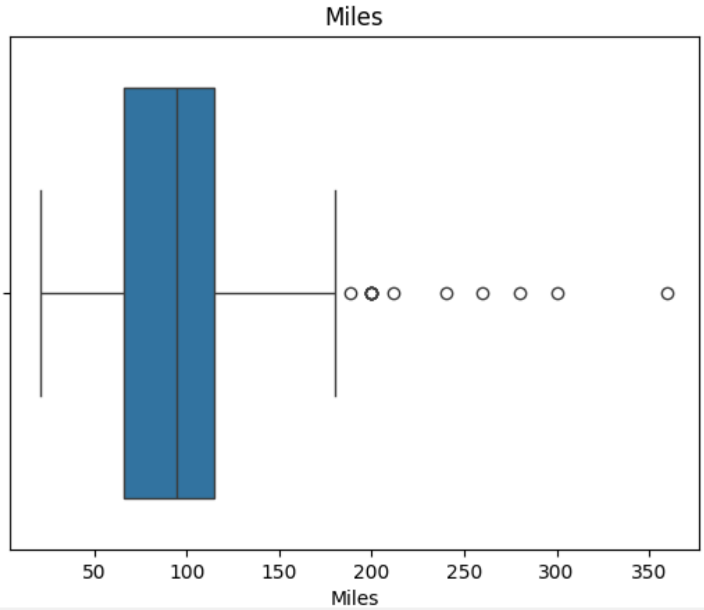
🔍 Insights

- Almost `60%` of the customers fall in the income group of (40k to 60k) dollars suggesting higher inclination of this income group people towards the products.
- Surprisingly `18%` of the customers fall in the income group of (<40) suggesting almost `77%` of the total customers fall in income group of below 60k and only `23%` of them falling in 60k and above income group
- **Outliers**
  - As we can see from the box plot, there are `many outlier's` present in the income data.

🗺️ Miles Column Outliers

```
# Ploting a boxplot
sns.boxplot(x=df["Miles"])
plt.title("Miles")
```

```
Text(0.5, 1.0, 'Miles')
```




```
q1 = df['Miles'].quantile(0.25)
q3 = df['Miles'].quantile(0.75)
IQR = q3 - q1
df[(df['Miles'] < (q1 - 1.5*IQR)) | (df['Miles'] > (q3 + 1.5*IQR))]
```

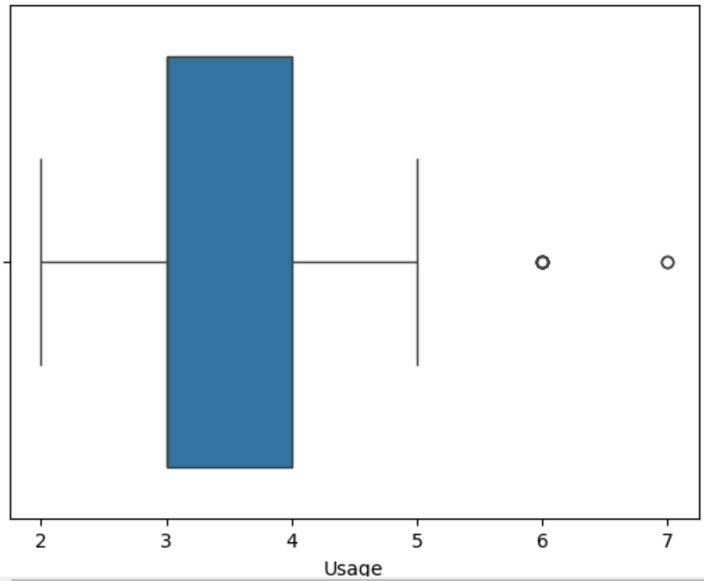
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
23	KP281	24	Female	16	Partnered	5	5	44343	188
84	KP481	21	Female	14	Partnered	5	4	34110	212
142	KP781	22	Male	18	Single	4	5	48556	200
148	KP781	24	Female	16	Single	5	5	52291	200
152	KP781	25	Female	18	Partnered	5	5	61006	200
155	KP781	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

🔍 Insights

- Almost `88%` of the customers plans to use the treadmill for `50 to 200 miles` per week with a median of `94 miles per week`.
- **Outliers**
  - As we can see from the box plot, there are `8 outlier's` present in the miles data

🏃 Usage Column Outliers

```
# Ploting a boxplot
sns.boxplot(data = df, x = 'Usage')
plt.show()
```




```
q1 = df['Usage'].quantile(0.25)
q3 = df['Usage'].quantile(0.75)
IQR = q3 - q1
df[(df['Usage'] < (q1 - 1.5*IQR)) | (df['Usage'] > (q3 + 1.5*IQR))]
```

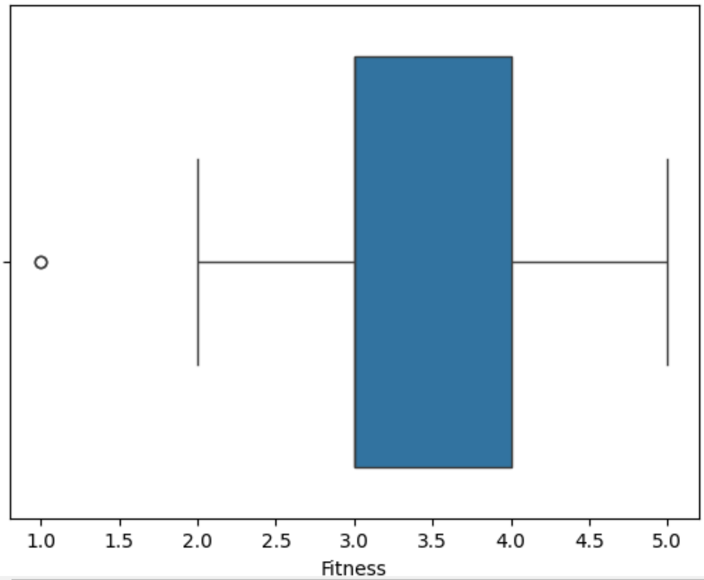
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
154	KP781	25	Male	18	Partnered	6	4	70966	180
155	KP781	25	Male	18	Partnered	6	5	75946	240
162	KP781	28	Female	18	Partnered	6	5	92131	180
163	KP781	28	Male	18	Partnered	7	5	77191	180
164	KP781	28	Male	18	Single	6	5	88396	150
166	KP781	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

🔍 Insights

- The dataset contains a cluster of outliers related to product usage (KP781), with these users reporting significantly higher usage frequencies compared to the rest of the population. These individuals, primarily male and with higher income levels, exhibit a strong affinity for the product and may represent a valuable target segment for further analysis and potentially tailored marketing strategies.
- It's essential to investigate the reasons behind this outlier group to understand if it represents genuine high usage or potential data anomalies.
- **Outliers**
  - As we can see from the box plot, there are `2 outlier's` present in the Usage data

🏋 Fitness Column Outliers

```
# Ploting a boxplot
sns.boxplot(data = df, x = 'Fitness')
plt.show()
```




```
q1 = df['Fitness'].quantile(0.25)
q3 = df['Fitness'].quantile(0.75)
IQR = q3 - q1
df[(df['Fitness'] < (q1 - 1.5*IQR)) | (df['Fitness'] > (q3 + 1.5*IQR))]
```

	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

🔍 Insights

The analysis identified two potential outliers in terms of fitness levels. it feels like its a Data entry errors, Incorrectly recorded fitness levels.

Outliers

- As we can see from the box plot, there are `2 outlier's` present in the Fitness data

✂️ Trim the Middle 90%

```
minn = np.percentile(df['Income'], 5)
maxx = np.percentile(df['Income'], 95)
df['Income'] = np.clip(df['Income'], minn, maxx)

minn1 = np.percentile(df['Age'], 5)
maxx1 = np.percentile(df['Age'], 95)
df['Age'] = np.clip(df['Age'], minn1, maxx1)

minn2 = np.percentile(df['Education'], 5)
maxx2 = np.percentile(df['Education'], 95)
df['Education'] = np.clip(df['Education'], minn2, maxx2)

minn3 = np.percentile(df['Fitness'], 5)
maxx3 = np.percentile(df['Fitness'], 95)
df['Fitness'] = np.clip(df['Fitness'], minn3, maxx3)

minn4 = np.percentile(df['Miles'], 5)
maxx4 = np.percentile(df['Miles'], 95)
df['Miles'] = np.clip(df['Miles'], minn4, maxx4)

minn5 = np.percentile(df['Usage'], 5)
maxx5 = np.percentile(df['Usage'], 95)
df['Usage'] = np.clip(df['Usage'], minn5, maxx5)
```

🔍 Insights

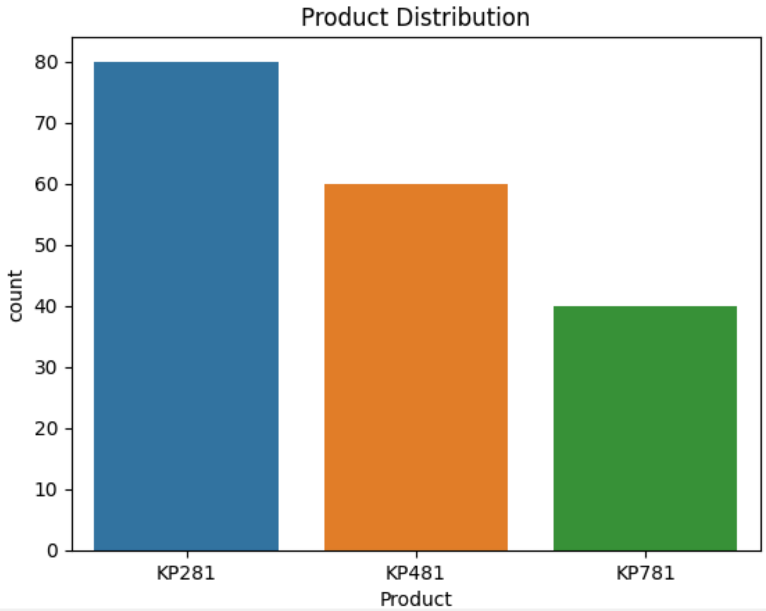
To enhance data accuracy and reliability, potential outliers in key numerical variables (Income, Age, Education, Fitness, Miles, Usage) have been capped at the 5th and 95th percentiles. This data cleaning step is crucial to mitigate the undue influence of extreme values on statistical analyses and modeling, ensuring a more robust representation of the underlying data distribution.

🛍️ 3. IntrodDemographic Impact on Product Choice

3.1 Find if there is any relationship between the categorical variables and the output variable in the data.

```
sns.countplot(data = df, x = 'Product', hue = 'Product')
plt.title('Product Distribution')
```

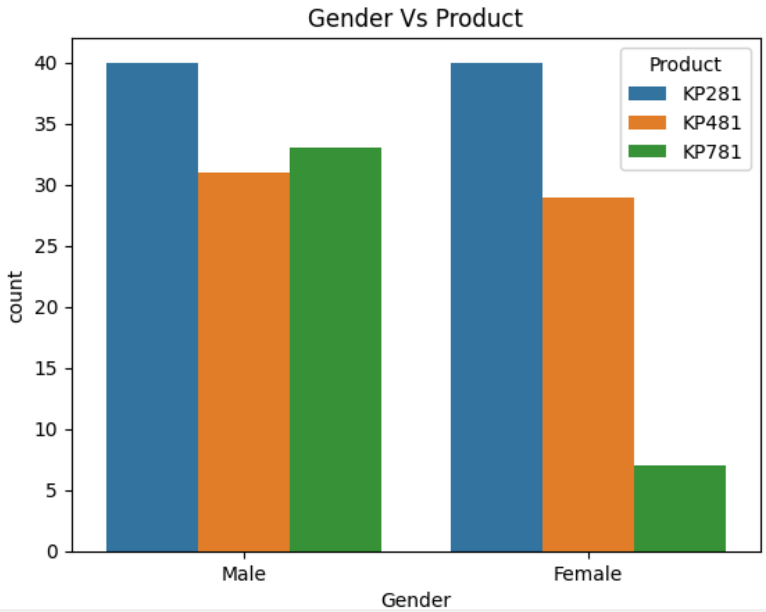
```
Text(0.5, 1.0, 'Product Distribution')
```



Double-click (or enter) to edit

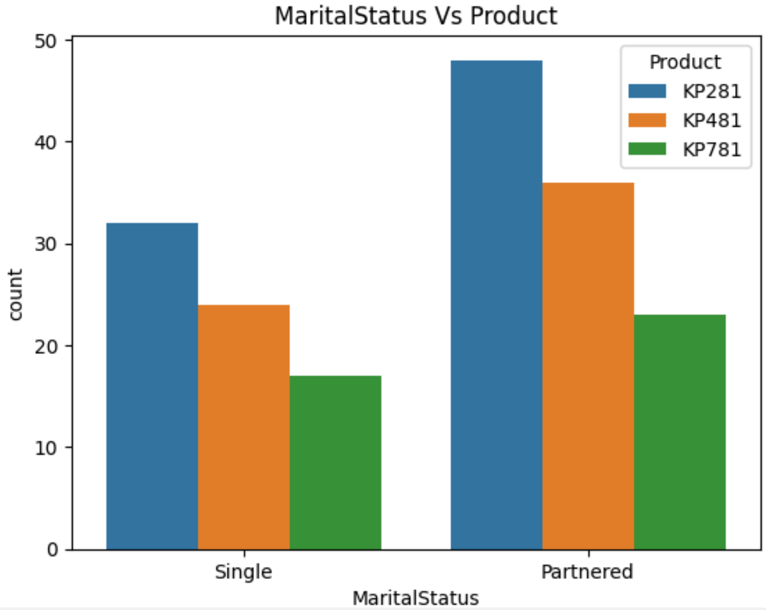
```
sns.countplot(data = df, x = 'Gender', hue = 'Product')
plt.title('Gender Vs Product')
```

```
Text(0.5, 1.0, 'Gender Vs Product')
```



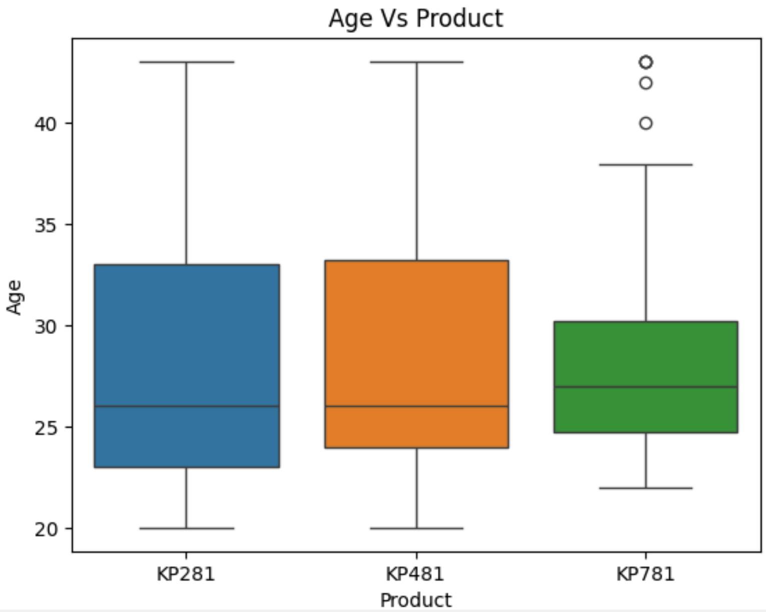
```
sns.countplot(data = df, x = 'MaritalStatus', hue = 'Product')
plt.title('MaritalStatus Vs Product')
```

```
Text(0.5, 1.0, 'MaritalStatus Vs Product')
```



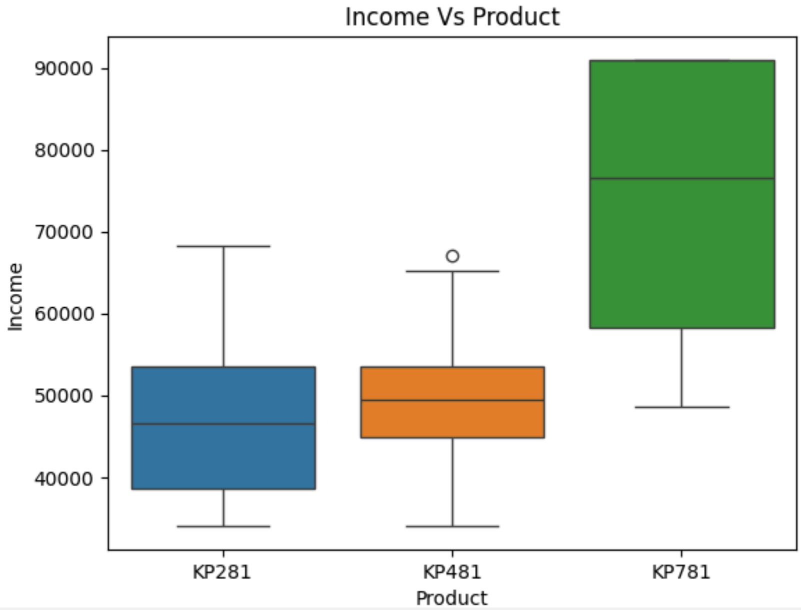

```
sns.boxplot(data = df, x = 'Product', y = 'Age', hue = 'Product')
plt.title('Age Vs Product')
```

```
Text(0.5, 1.0, 'Age Vs Product')
```



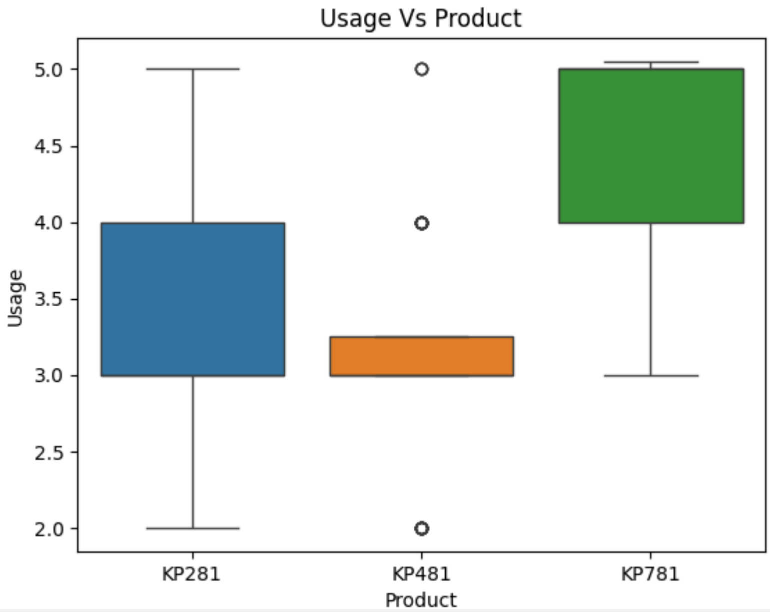

```
sns.boxplot(data = df, x = 'Product', y = 'Income', hue = 'Product')
plt.title('Income Vs Product')
```

```
Text(0.5, 1.0, 'Income Vs Product')
```



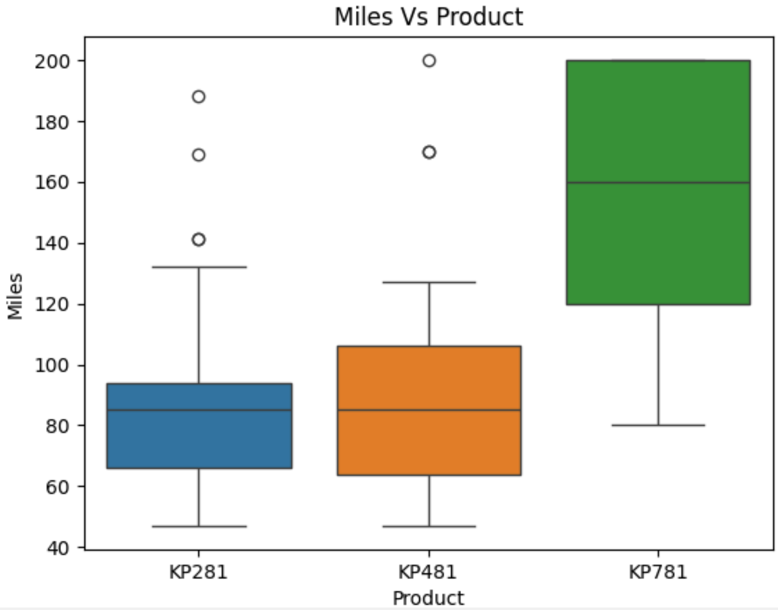

```
sns.boxplot(data = df, x = 'Product', y = 'Usage', hue = 'Product')
plt.title('Usage Vs Product')
```

```
Text(0.5, 1.0, 'Usage Vs Product')
```




```
sns.boxplot(data = df, x = 'Product', y = 'Miles', hue = 'Product')
plt.title('Miles Vs Product')
```

```
Text(0.5, 1.0, 'Miles Vs Product')
```

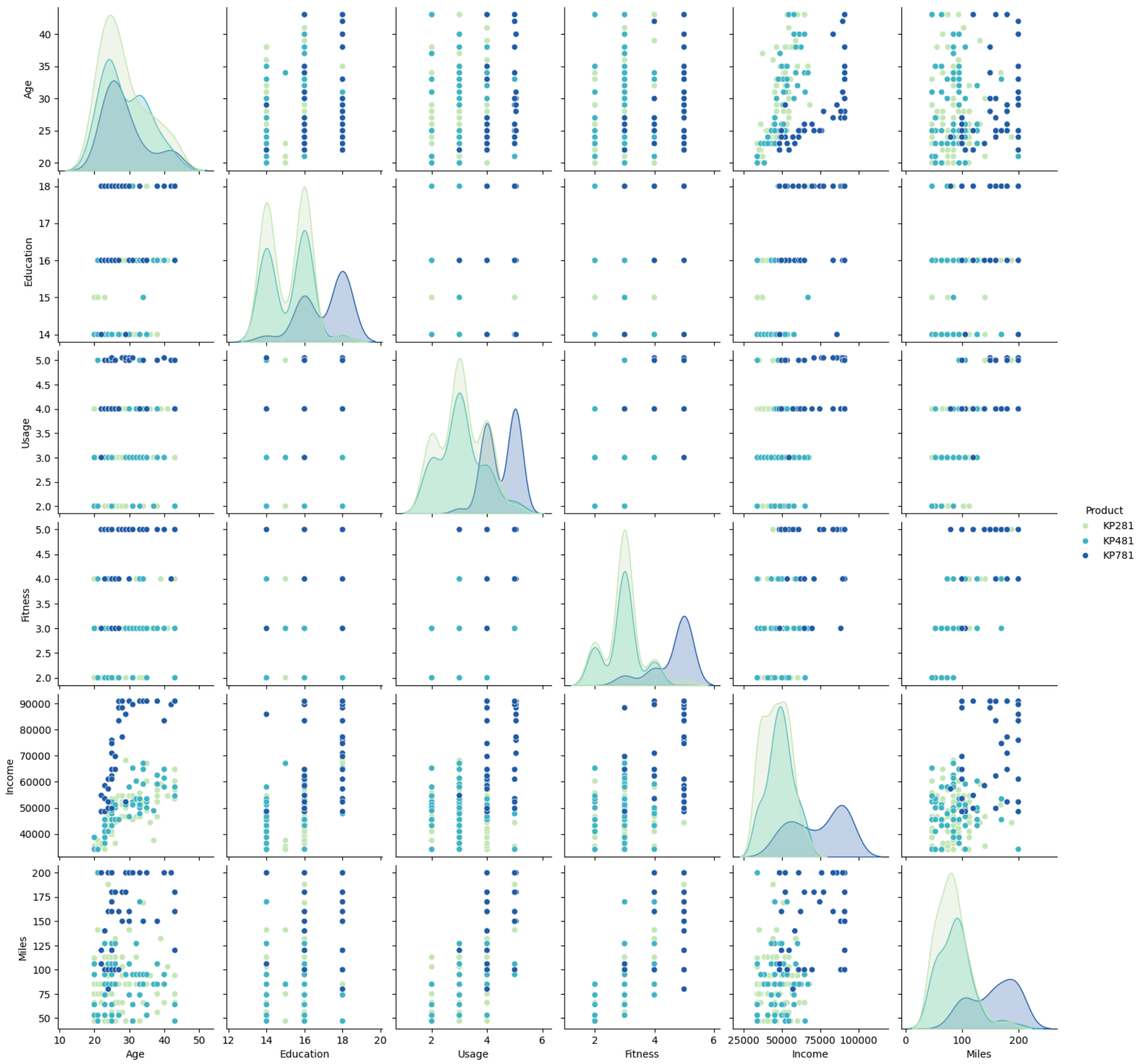


🔍 Insights

- The analysis presented above clearly indicates a strong preference for the treadmill model `KP781` among customers who possess **higher education, higher income levels, and intend to engage in running activities exceeding 150 miles per week.**

3.2 Find if there is any relationship between the continuous variables and the output variable in the data. Hint: We want you to use a scatter plot to find the relationship between continuous variables and output variables.

```
df_copy = df
sns.pairplot(df_copy, hue ='Product', palette= 'YlGnBu')
plt.show()
```



🔍 Insights

- From the pair plot we can see `Age and Income` are **positively correlated**.
- `Eductaion` and `Income` are highly correlated as its obvious. Eductation also has significatnt correlation between `Fitness rating and Usage of the treadmill.`
- `Usage` is highly correlated with `Fitness and Miles` as more the usage more the fitness and mileage.

🎲 4. Representing the Probability

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

```
pd.crosstab(index=df['Product'], columns='count', normalize=True)
```

col_0	count
Product	
KP281	0.444444
KP481	0.333333
KP781	0.222222

🔍 Insights

Product Popularity

The analysis reveals that **KP281** is the most popular treadmill model, accounting for **44.44%** of total purchases. Following closely is **KP481** with a market share of **33.33%**, while **KP781** constitutes **22.22%** of the market.

This data indicates a clear preference for the KP281 model among customers. Understanding the factors driving this preference can inform product development and marketing strategies.

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

```
# Probability of buying a product based on Gender
pd.crosstab(index=df['Product'], columns=df['Gender'], margins=True, normalize='columns').round(2)
```

Gender	Female	Male	All
Product			
KP281	0.53	0.38	0.44
KP481	0.38	0.30	0.33
KP781	0.09	0.32	0.22

```
# Probability of buying a product based on Marital Status
pd.crosstab(index=df['Product'], columns=df['MaritalStatus'], margins=True, normalize='columns').round(2)
```

MaritalStatus	Partnered	Single	All
Product			
KP281	0.45	0.44	0.44
KP481	0.34	0.33	0.33
KP781	0.21	0.23	0.22

```
# Probability of buying a product based on Usage
pd.crosstab(index=df['Product'], columns=df['Usage'], margins=True, normalize='columns').round(2)
```

Usage	2.0	3.0	4.0	5.0	5.049999999999983	All
Product						
KP281	0.58	0.54	0.42	0.12	0.0	0.44
KP481	0.42	0.45	0.23	0.18	0.0	0.33
KP781	0.00	0.01	0.35	0.71	1.0	0.22

```
# Probability of buying a product based on Fitness
pd.crosstab(index=df['Product'], columns=df['Fitness'], margins=True, normalize='columns').round(2)
```

Fitness	2	3	4	5	All
Product					
KP281	0.54	0.56	0.38	0.06	0.44
KP481	0.46	0.40	0.33	0.00	0.33
KP781	0.00	0.04	0.29	0.94	0.22

```
# Probability of buying a product based on Education
pd.crosstab(index=df['Product'], columns=df['Education'], margins=True, normalize='columns').round(2)
```

Education	14	15	16	18	All
Product					
KP281	0.56	0.8	0.46	0.07	0.44
KP481	0.41	0.2	0.36	0.07	0.33
KP781	0.03	0.0	0.18	0.85	0.22

```
# Probability of buying a product based on Income
pd.crosstab(index=df['Product'], columns=df['Income'], margins=True, normalize='columns').round(2)
```

Income	34053.15	34110.0	35247.0	36384.0	37521.0	38658.0	39795.0	40932.0	42069.0	43206.0	...	74701.0	75946.0	7719
Product														
KP281	0.67	0.4	1.0	0.75	1.0	0.6	1.0	0.67	1.0	0.2	...	0.0	0.0	
KP481	0.33	0.6	0.0	0.25	0.0	0.4	0.0	0.33	0.0	0.8	...	0.0	0.0	
KP781	0.00	0.0	0.0	0.00	0.0	0.0	0.0	0.00	0.0	0.0	...	1.0	1.0	

3 rows × 55 columns

```
# Probability of buying a product based on Miles
pd.crosstab(index=df['Product'], columns=df['Miles'], margins=True, normalize='columns').round(2)
```

Miles	47	53	56	64	66	74	75	80	85	94	...	140	141	150	160	169	170	180	188	200	All
Product																					
KP281	0.71	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.59	1.0	...	0.0	1.0	0.0	0.0	1.0	0.00	0.0	1.0	0.00	0.44
KP481	0.29	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.41	0.0	...	0.0	0.0	0.0	0.0	0.0	0.67	0.0	0.0	0.08	0.33
KP781	0.00	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.00	0.0	...	1.0	0.0	1.0	1.0	0.0	0.33	1.0	0.0	0.92	0.22

3 rows × 29 columns

🔍 Insights

Marital Status

- **No significant impact** of marital status on product choice. Purchase probabilities are relatively evenly distributed across product models for both partnered and single customers.

Usage

- **Higher usage** correlates with **KP781** preference. Customers with higher usage frequency are more likely to choose KP781.

Fitness

- **Fitness level** shows **minimal influence** on product selection. There's no clear pattern between fitness and product preference.

Education

- Customers with **higher education** tend to lean towards **KP281** and **KP481**. However, the difference is not substantial.

Income

- There's a **weak indication** that customers with **higher income** might prefer **KP281**. Further analysis with more granular income brackets could provide clearer insights.

Miles

- As previously mentioned, there's a **strong correlation** between **high mileage** and **KP781** preference. Customers running longer distances are more likely to choose KP781.

4.3 Find 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)

```
# Probability of a female customer purchasing KP481 given that the customer is female
female_customers = df[df['Gender'] == 'Female']
prob_female_kp481 = female_customers[female_customers['Product'] == 'KP481'].shape[0] / female_customers.shape[0]
print("Probability of female customer purchasing KP481:", prob_female_kp481)

# Probability of a male customer purchasing KP781 given that the customer is male
male_customers = df[df['Gender'] == 'Male']
prob_male_kp781 = male_customers[male_customers['Product'] == 'KP781'].shape[0] / male_customers.shape[0]
print("Probability of male customer purchasing KP781:", prob_male_kp781)

# Probability of purchasing KP781 given the customer is partnered
partnered_customers = df[df['MaritalStatus'] == 'Partnered']
prob_partnered_kp781 = partnered_customers[partnered_customers['Product'] == 'KP781'].shape[0] / partnered_customers.shape[0]
print("Probability of purchasing KP781 given the customer is partnered:", prob_partnered_kp781)

# Probability of purchasing KP281 given the customer plans to use the treadmill 3 times a week
usage_3_customers = df[df['Usage'] == 3]
prob_usage3_kp281 = usage_3_customers[usage_3_customers['Product'] == 'KP281'].shape[0] / usage_3_customers.shape[0]
print("Probability of purchasing KP281 given the customer plans to use the treadmill 3 times a week:", prob_usage3_kp281)

# Probability of a customer purchasing KP281 given that the customer's income is greater than 60000
high_income_customers = df[df['Income'] > 60000]
prob_high_income_kp281 = high_income_customers[high_income_customers['Product'] == 'KP281'].shape[0] / high_income_customers.shape
print("Probability of purchasing KP281 given the customer's income is greater than 60000:", prob_high_income_kp281)
```

```
Probability of female customer purchasing KP481: 0.3815789473684211
Probability of male customer purchasing KP781: 0.3173076923076923
Probability of purchasing KP781 given the customer is partnered: 0.21495327102803738
Probability of purchasing KP281 given the customer plans to use the treadmill 3 times a week: 0.5362318840579711
Probability of purchasing KP281 given the customer's income is greater than 60000: 0.14285714285714285
```

🔍 Insights

Gender and Product Choice

- **Women** show a higher propensity to purchase **KP481** compared to men.
- **Men** are more likely to opt for **KP781** than women.

Marital Status and Product Choice

- Customers in **partnerships** are slightly less likely to purchase **KP781** compared to the overall population.

Usage Frequency and Product Choice

- There's a moderate correlation between **higher usage frequency** and purchasing **KP281**.

Income and Product Choice

- Customers with **higher income levels** are less likely to choose **KP281**. There's a potential association with higher-end models for this income bracket.

These conditional probabilities provide valuable insights into customer behavior and preferences. They can be leveraged to tailor marketing strategies, product positioning, and customer segmentation efforts.

🔗 5. Understanding Variable Relationships: Correlation Analysis

```
df.head()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	20.0	Male	14	Single	3.0	4	34053.15	112
1	KP281	20.0	Male	15	Single	2.0	3	34053.15	75
2	KP281	20.0	Female	14	Partnered	4.0	3	34053.15	66
3	KP281	20.0	Male	14	Single	3.0	3	34053.15	85
4	KP281	20.0	Male	14	Partnered	4.0	2	35247.00	47

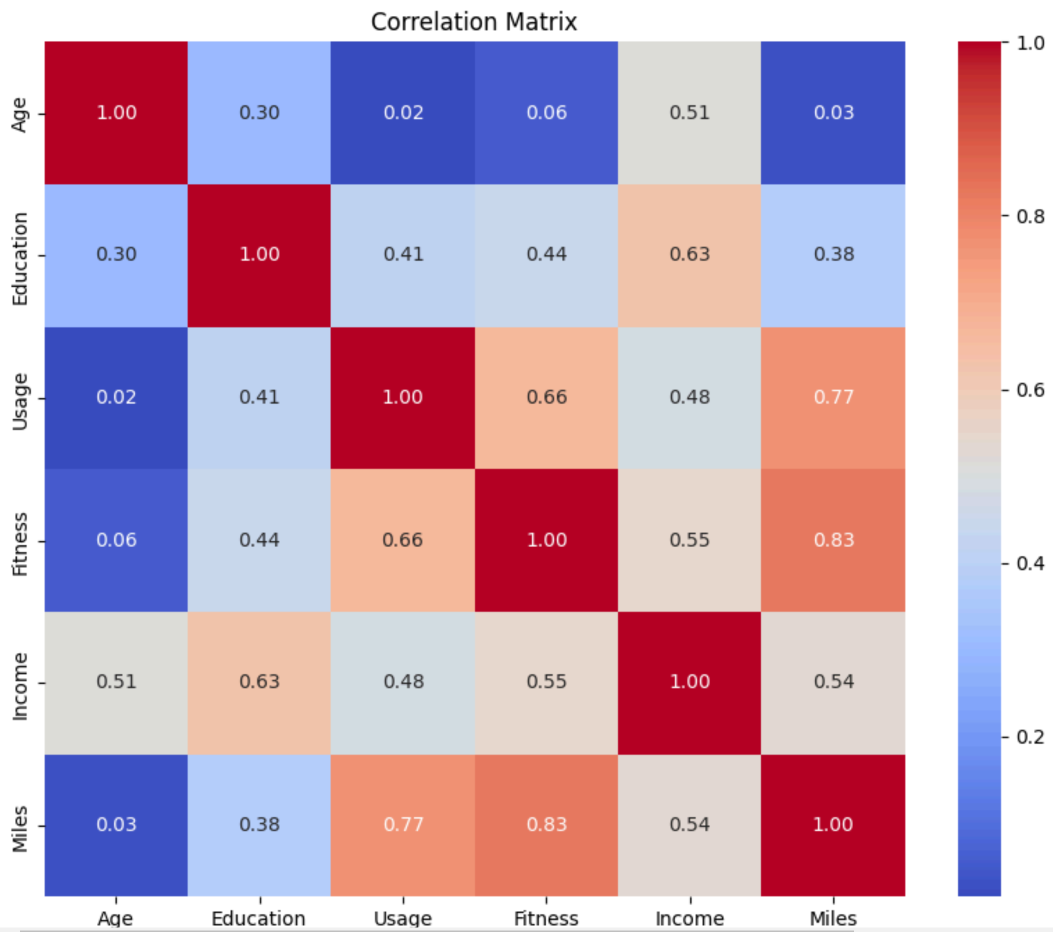
Next steps: Generate code with df | View recommended plots | New interactive sheet

```
correlation_matrix = df.corr(numeric_only=True)
correlation_matrix
```

	Age	Education	Usage	Fitness	Income	Miles
Age	1.000000	0.301971	0.015394	0.057361	0.514362	0.029636
Education	0.301971	1.000000	0.413600	0.441082	0.628597	0.377294
Usage	0.015394	0.413600	1.000000	0.661978	0.481608	0.771030
Fitness	0.057361	0.441082	0.661978	1.000000	0.546998	0.826307
Income	0.514362	0.628597	0.481608	0.546998	1.000000	0.537297
Miles	0.029636	0.377294	0.771030	0.826307	0.537297	1.000000

Next steps: Generate code with correlation_matrix | View recommended plots | New interactive sheet

```
# Create a heatmap to visualize the correlations
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Matrix")
plt.show()
```



🔍 Insights

The correlation matrix reveals several interesting relationships between the variables:

- **Strong Positive Correlations:**
  - There is a strong positive correlation between `Usage`, `Fitness`, and `Miles`. This suggests that individuals who exercise more tend to have higher usage and cover more miles.
  - `Income` and `Education` are also positively correlated, indicating that higher education levels often correspond to higher income.
- **Moderate Positive Correlations:**
  - `Age` and `Education` share a moderate positive correlation, suggesting older individuals tend to have higher education levels.
  - `Income` and `Fitness` also exhibit a moderate positive relationship, implying that individuals with higher incomes might be more likely to engage in fitness activities.
- **Weak or No Correlation:**
  - `Age` shows minimal correlation with other variables, indicating it might not be a strong predictor of the other factors.

Overall, the correlation matrix highlights the interconnectedness of lifestyle factors such as income, education, fitness, and physical activity levels. These insights can be valuable for targeted marketing, product development, and customer segmentation.

🎯 6. Customer profiling and recommendation

6.1 Make customer profilings for each and every product.

```
# KP281 Profiling
kp281_customers = df[df['Product'] == 'KP281']
print("KP281 Customer Profile:")
print("Age (Min - Max):", kp281_customers['Age'].min(), "-", kp281_customers['Age'].max())
print("Income (Min - Max):", kp281_customers['Income'].min(), "-", kp281_customers['Income'].max())
print("Usage (Mode):", kp281_customers['Usage'].mode()[0])
print("Marital Status (More Likely):", kp281_customers['MaritalStatus'].mode()[0])
print("Gender Distribution:\n", kp281_customers['Gender'].value_counts(normalize=True))

# KP481 Profiling
kp481_customers = df[df['Product'] == 'KP481']
print("\nKP481 Customer Profile:")
print("Age (Min - Max):", kp481_customers['Age'].min(), "-", kp481_customers['Age'].max())
print("Income (Min - Max):", kp481_customers['Income'].min(), "-", kp481_customers['Income'].max())
print("Usage (Mode):", kp481_customers['Usage'].mode()[0])
print("Marital Status (More Likely):", kp481_customers['MaritalStatus'].mode()[0])
print("Gender Distribution:\n", kp481_customers['Gender'].value_counts(normalize=True))

# KP781 Profiling
kp781_customers = df[df['Product'] == 'KP781']
print("\nKP781 Customer Profile:")
print("Age (Min - Max):", kp781_customers['Age'].min(), "-", kp781_customers['Age'].max())
print("Income (Min - Max):", kp781_customers['Income'].min(), "-", kp781_customers['Income'].max())
print("Usage (Mode):", kp781_customers['Usage'].mode()[0])
print("Marital Status (More Likely):", kp781_customers['MaritalStatus'].mode()[0])
print("Gender Distribution:\n", kp781_customers['Gender'].value_counts(normalize=True))
```