Import Dataset and check basic info

Import Dataset

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

Load the Dataset

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

Display first few rows

```
print(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
```

Data types of all columns

```
print(df.dtypes)
```

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

Shape of the dataset

```
print(df.shape)
```

```
(180, 9)
```

Missing values if any

```
print(df.isnull().sum())
```

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

```
dtype: int64
```

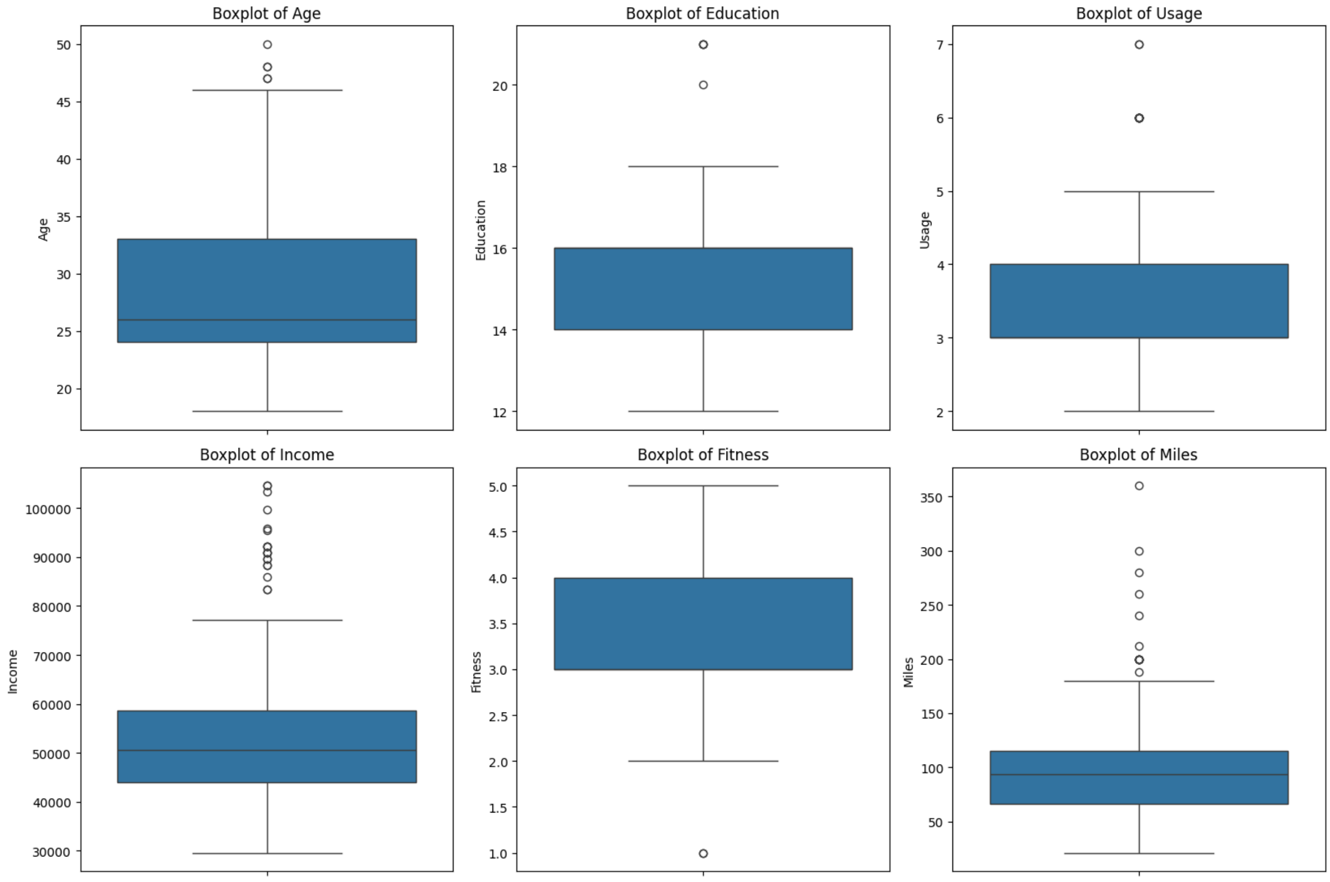
Detect Outliers

Continous variables to check for outliers

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

Plot boxplots to visualize outliers

```
plt.figure(figsize=(15, 10))
for i, col in enumerate(continuous_vars, 1):
    plt.subplot(2, 3, i)
    sns.boxplot(y=df[col])
    plt.title(f'Boxplot of {col}')
plt.tight_layout()
plt.show()
```



Clip data between 5th and 95th percentiles

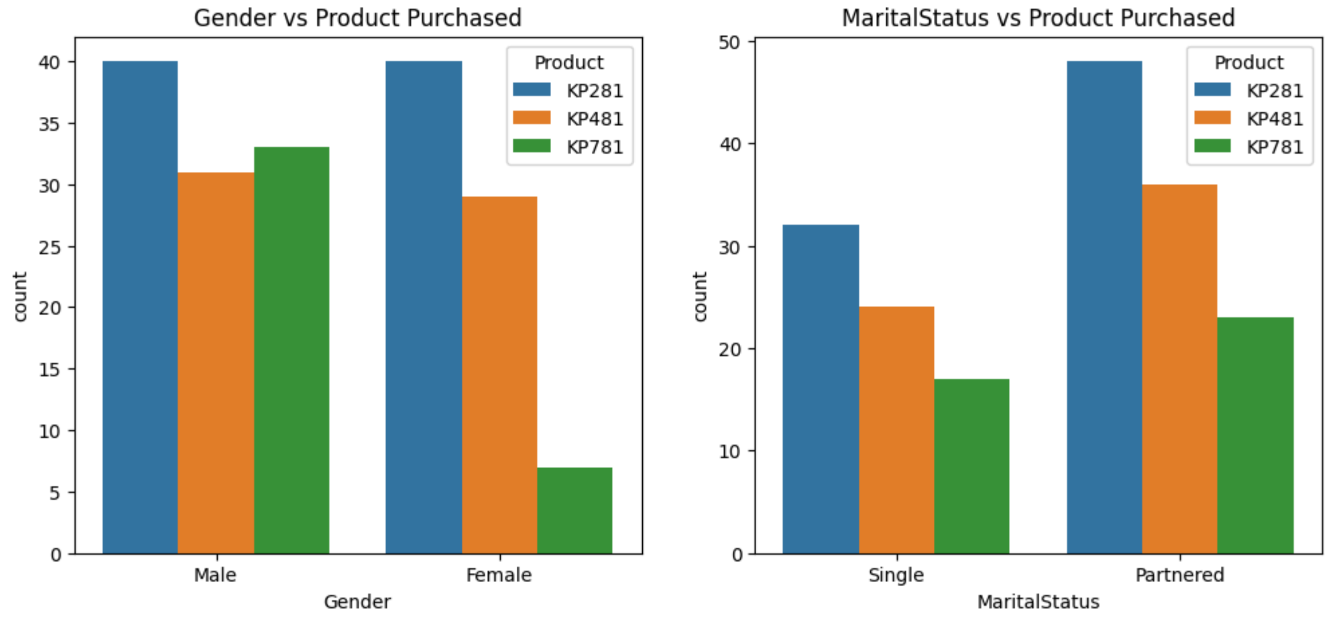
```
for col in continuous_vars:
    lower = df[col].quantile(0.05)
    upper = df[col].quantile(0.95)
    df[col] = np.clip(df[col], lower, upper)
```

Relationship Between Features and Product Purchased

Categorical Variables vs Product

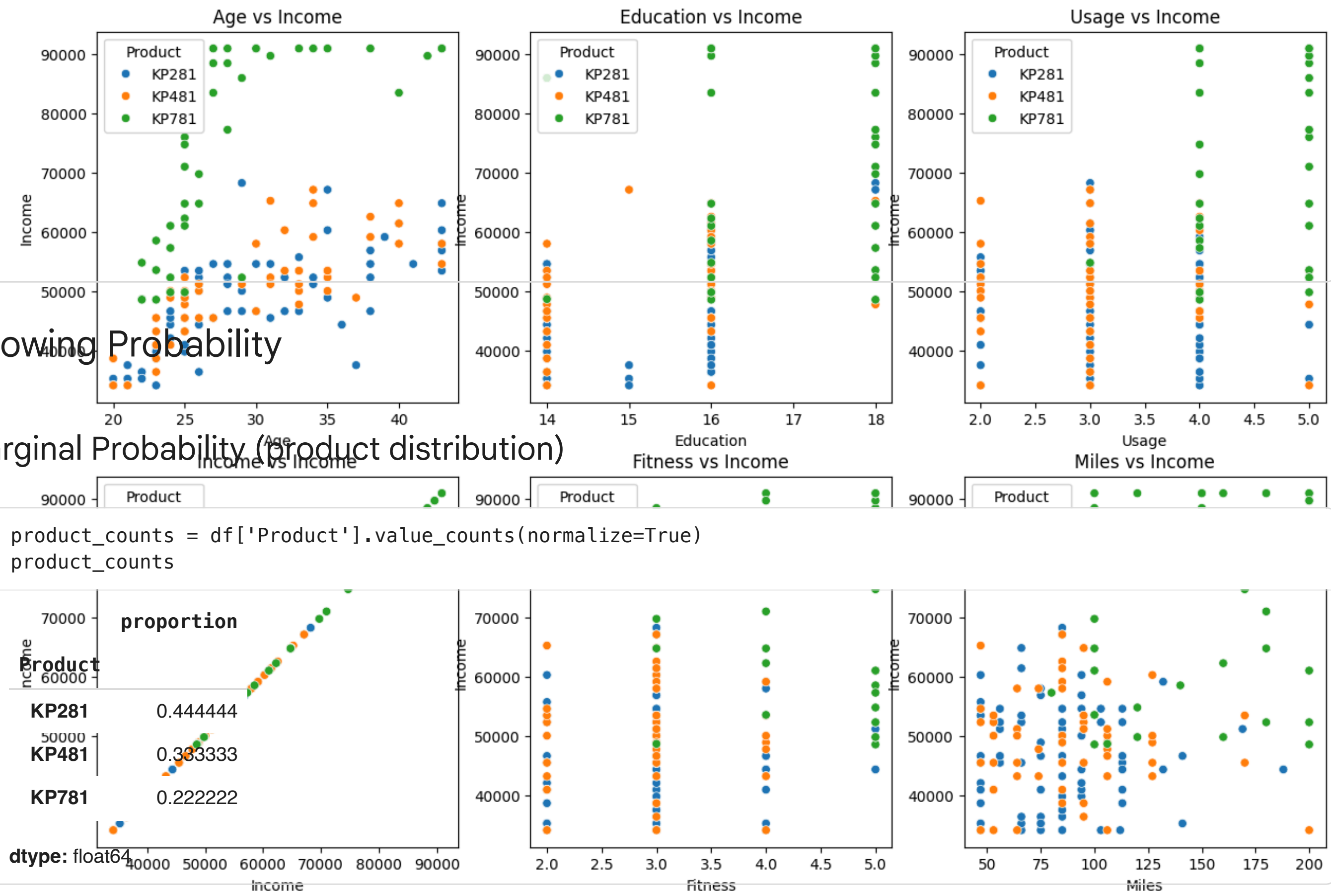
```
categorical_vars = ['Gender', 'MaritalStatus']

plt.figure(figsize=(12, 5))
for i, col in enumerate(categorical_vars, 1):
    plt.subplot(1, 2, i)
    sns.countplot(data=df, x=col, hue='Product')
    plt.title(f'{col} vs Product Purchased')
```



Continuous Variables vs Product (Scatterplots)

```
plt.figure(figsize=(15, 10))
for i, col in enumerate(continuous_vars, 1):
    plt.subplot(2, 3, i)
    sns.scatterplot(data=df, x=col, y='Income', hue='Product')
    plt.title(f'{col} vs Income')
```



Showing Probability

Marginal Probability (product distribution)

```
product_counts = df['Product'].value_counts(normalize=True)
product_counts
```

	proportion
Product	
KP281	0.444444
KP481	0.333333
KP781	0.222222

dtype: float64

Probability: Column

```
gender_prob = pd.crosstab(df['Gender'], df['Product'], normalize='index')
gender_prob
```

Product	KP281	KP481	KP781
Gender			
Female	0.526316	0.381579	0.092105
Male	0.384615	0.298077	0.317308

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

Conditional Probability

```
prob = gender_prob.loc['Female', 'KP481']
prob
```

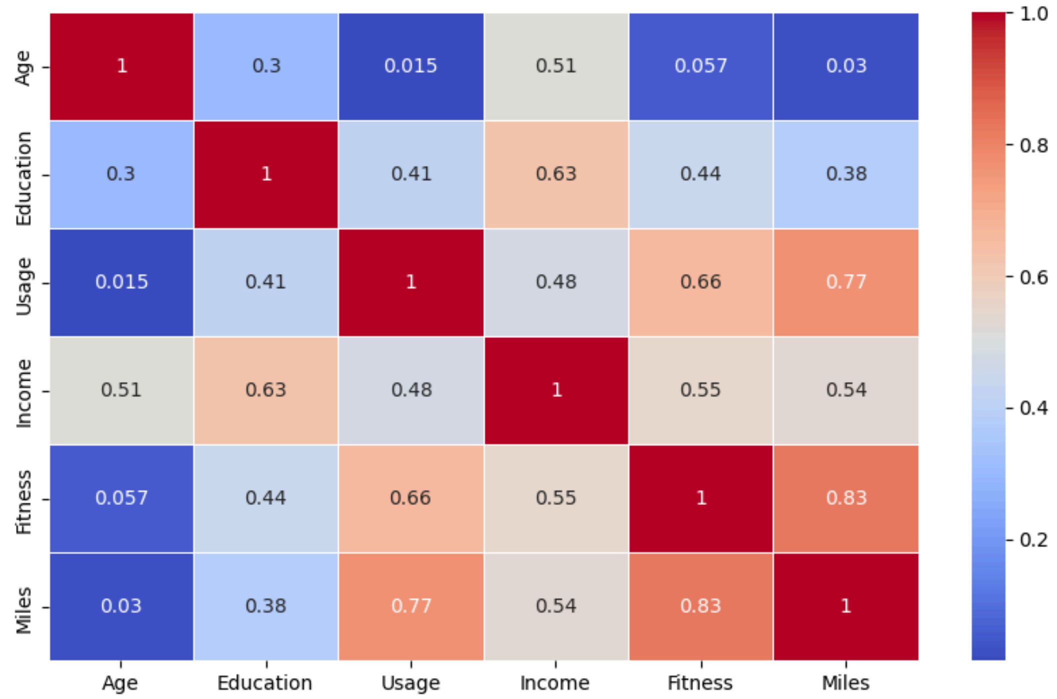
```
np.float64(0.3815789473684211)
```

Double-click (or enter) to edit

Corelation

```
plt.figure(figsize=(10, 6))
corr_matrix = df[continuous_vars].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
```

```
<Axes: >
```



	Age	Education	Usage	Income	Fitness	Miles
Age	1	0.3	0.015	0.51	0.057	0.03
Education	0.3	1	0.41	0.63	0.44	0.38
Usage	0.015	0.41	1	0.48	0.66	0.77
Income	0.51	0.63	0.48	1	0.55	0.54
Fitness	0.057	0.44	0.66	0.55	1	0.83
Miles	0.03	0.38	0.77	0.54	0.83	1

Customer profiling

```
product_profiles = df.groupby('Product')[['Age', 'Income', 'Usage', 'Fitness', 'Miles']].mean()
product_profiles
```

Product	Age	Income	Usage	Fitness	Miles
KP281	28.425006	46588.564341	3.087500	2.975000	83.125
KP481	28.800004	49049.442894	3.066667	2.916667	88.500
KP781	28.825009	73894.310016	4.500028	4.625000	155.900

Next steps: Generate code with `product_profiles` View recommended plots New interactive sheet