Business Case: Aerofit - Descriptive Statistics & Probability

1.Problem Statement:

AeroFit wants to understand the customer characteristics for each treadmill model — KP281 (entry-level), KP481 (mid-level), and KP781 (advanced) — to better recommend products and target marketing efforts.

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 : 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

2. Load Data and Initial Exploration

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import copy
```

In [ ]:
```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/ori
```

```
--2025-06-02 12:57:43--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 108.157.172.10, 108.157.172.173, 108.157.172.183, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|108.157.172.10|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7279 (7.1K) [text/plain]
Saving to: 'aerofit_treadmill.csv?1639992749.1'


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

2025-06-02 12:57:43 (1.28 GB/s) - 'aerofit_treadmill.csv?1639992749.1' saved [7279/7279]
```

In [ ]:
```
!mv 'aerofit_treadmill.csv?1639992749' aerofit_treadmill.csv
```

In [ ]:
```
df = pd.read_csv('aerofit_treadmill.csv')
```

In [ ]:
```
df .head()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-9681a06601ea> in <cell line: 0>()
----> 1 df .head()

NameError: name 'df' is not defined
```

3.Check data shape and types

In [ ]:
```
df.shape
```

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

In [ ]:
```
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
```

In [ ]:
```
df.describe()
```

Out[ ]:

	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

4.

Check unique values for categorical columns

In [ ]:

```
df['Product'].value_counts()
```

Out[ ]:

Product	count
KP281	80
KP481	60
KP781	40

dtype: int64

In [ ]:

```
df['Gender'].value_counts()
```

Out[ ]:

Gender	count
Male	104
Female	76

dtype: int64

In [ ]:

```
df['MaritalStatus'].value_counts()
```

Out[ ]:

MaritalStatus	count
Partnered	107
Single	73

dtype: int64

Check for nulls

In [ ]:
```
df.isnull().sum()
```

Out[ ]:

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

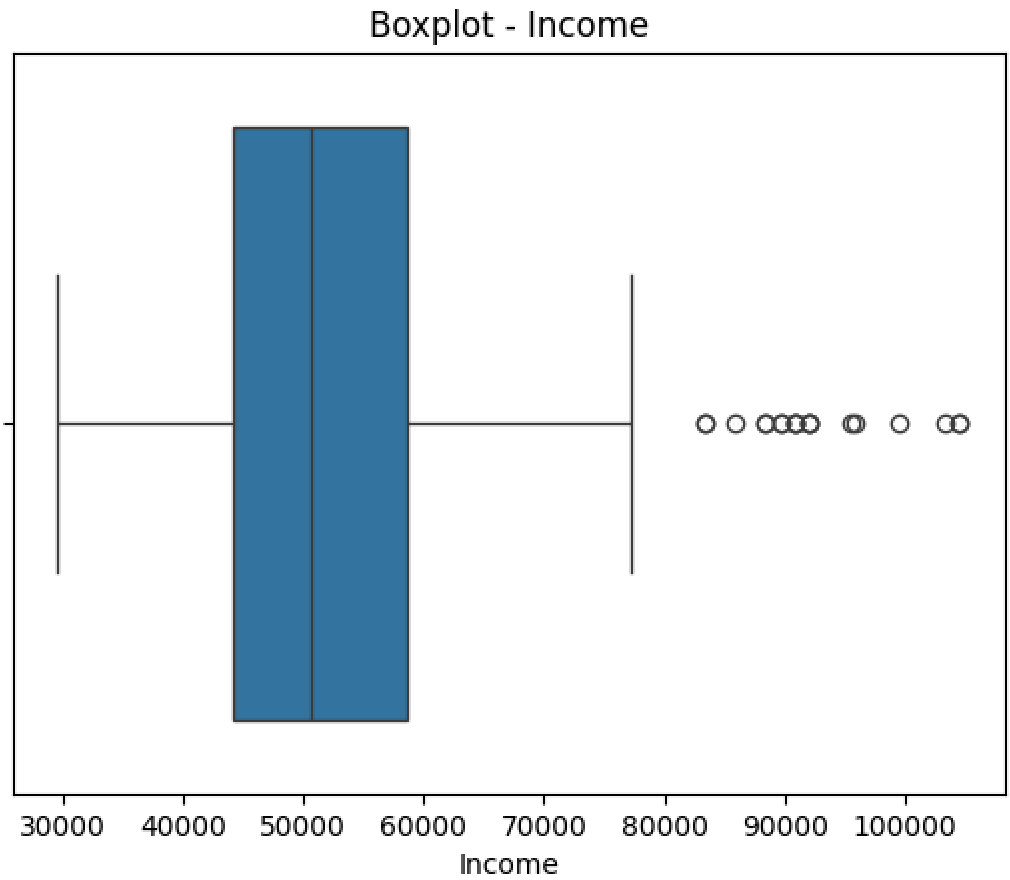
dtype: int64

Convert to categorical

In [ ]:
```
df['Product'] = df['Product'].astype('category')
df['Gender'] = df['Gender'].astype('category')
df['MaritalStatus'] = df['MaritalStatus'].astype('category')
```

Outlier Detection

In [ ]:
```
sns.boxplot(x=df['Income'])
plt.title("Boxplot - Income")
plt.show()
```



Duplicate Detection

In [ ]:
```
df.duplicated().value_counts()
```

Out[ ]:

	count
False	180

dtype: int64

Unique Values per Column

In [ ]:
```
# Number of unique values per column
df.nunique()
```

Out[ ]:

	0
Product	3
Age	32
Gender	2
Education	8
MaritalStatus	2
Usage	6
Fitness	5
Income	62
Miles	37

dtype: int64

Mode for Categorical Attributes

In [ ]:

```
df.mode().iloc[0]
```

Out[ ]:

	0
Product	KP281
Age	25
Gender	Male
Education	16
MaritalStatus	Partnered
Usage	3
Fitness	3
Income	45480
Miles	85

dtype: object

In [ ]:

```
df.head()
```

Out[ ]:

	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

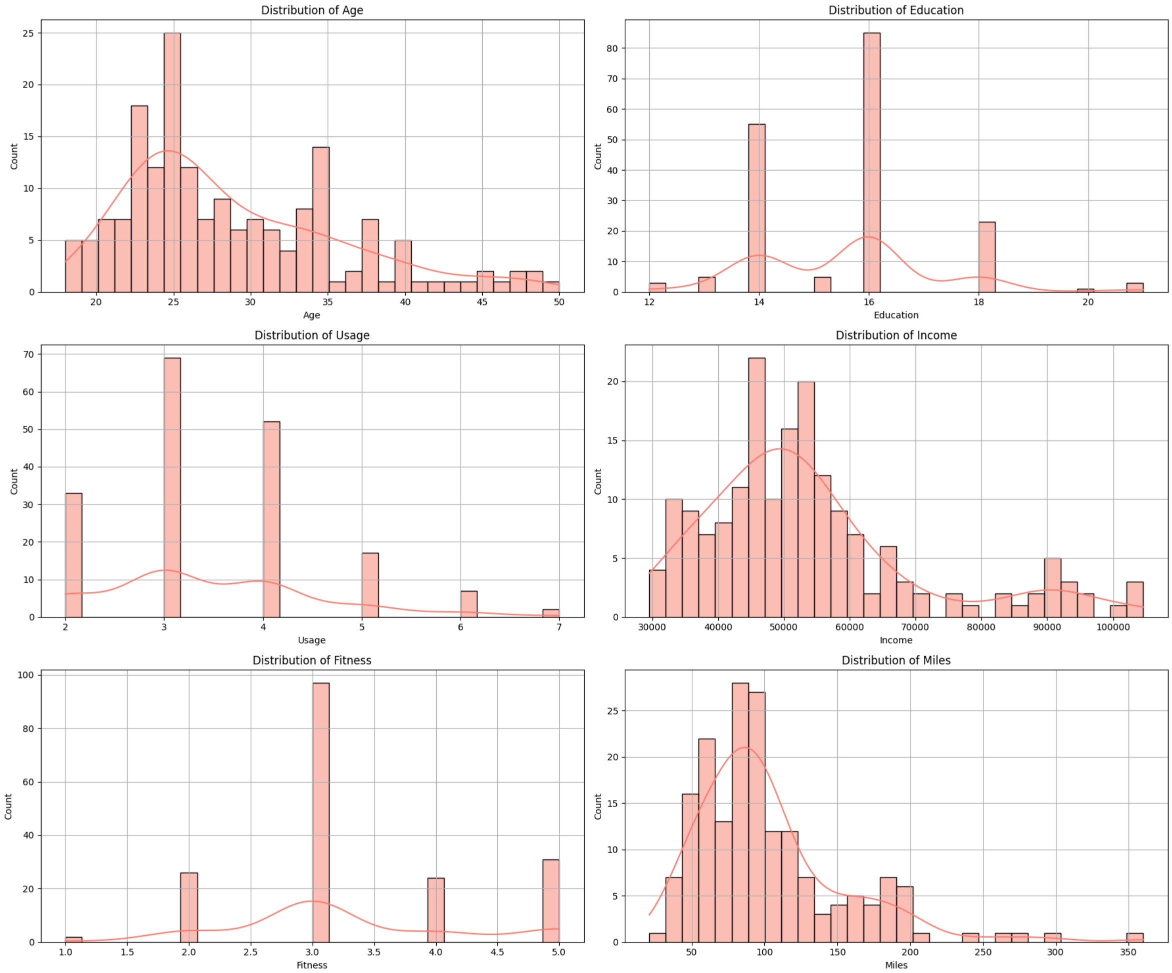
5.Visual Analysis - Univariate & Bivariate

Univariate Analysis

Continuous Variables

In [ ]:

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

plt.figure(figsize=(18, 15))

for i, col in enumerate(continuous_cols, 1):
    plt.subplot(3, 2, i)
    sns.histplot(data=df, x=col, kde=True, bins=30, color='salmon')
    plt.title(f'Distribution of {col}', fontsize=12)
    plt.xlabel(col)
    plt.ylabel("Count")
    plt.grid(True)

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



Business Insights:

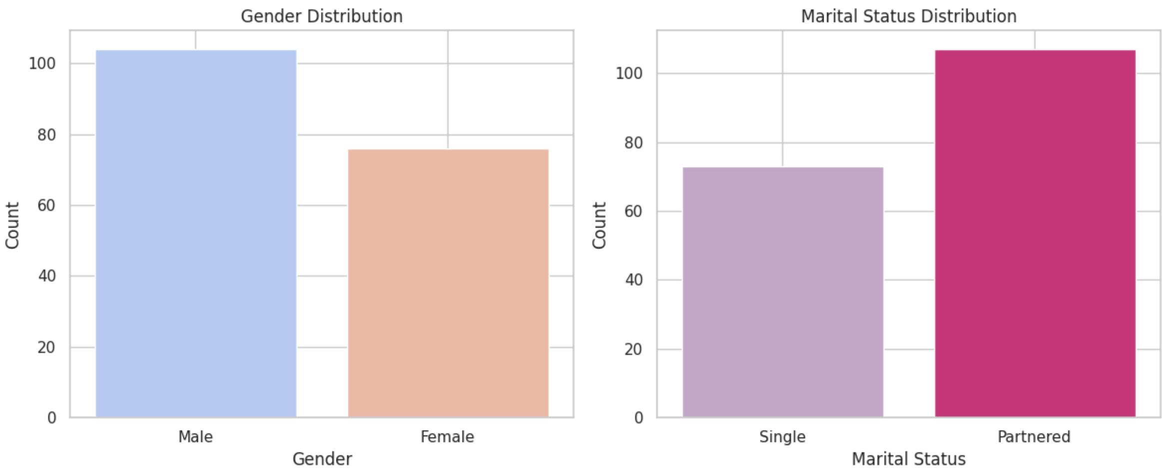
The analysis of continuous variables reveals clear customer segments tied to product preference. Income, weekly usage, fitness level, and expected mileage are strong indicators of which treadmill a customer will choose, with higher values in these areas pointing towards the premium KP781 model. Age and education levels also correlate positively with income, indirectly influencing the purchase of more expensive equipment. This suggests that the entry-level KP281 appeals primarily to more casual users and those with lower income, while the advanced KP781 is favored by dedicated fitness enthusiasts who have higher incomes and plan for more intense workouts. Understanding these relationships allows for more effective targeting based on a customer's financial status and fitness commitment.

Gender and Marital Status distributions

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Plotting Gender and Marital Status
plt.figure(figsize=(12, 5))

# Gender
plt.subplot(1, 2, 1)
```

```
sns.countplot(data=df, x='Gender', palette='coolwarm')
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.grid(True)

# Marital Status
plt.subplot(1, 2, 2)
sns.countplot(data=df, x='MaritalStatus', palette='PuRd')
plt.title('Marital Status Distribution')
plt.xlabel('Marital Status')
plt.ylabel('Count')
plt.grid(True)

# Percentage distribution
gender_dist = df['Gender'].value_counts(normalize=True) * 100
marital_dist = df['MaritalStatus'].value_counts(normalize=True) * 100

# Display as tables
print("Gender Distribution (%):")
print(gender_dist)
print("\nMarital Status Distribution (%):")
print(marital_dist)


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

```
Gender Distribution (%):
Gender
Male      57.777778
Female    42.222222
Name: proportion, dtype: float64

Marital Status Distribution (%):
MaritalStatus
Partnered    59.444444
Single       40.555556
Name: proportion, dtype: float64
```

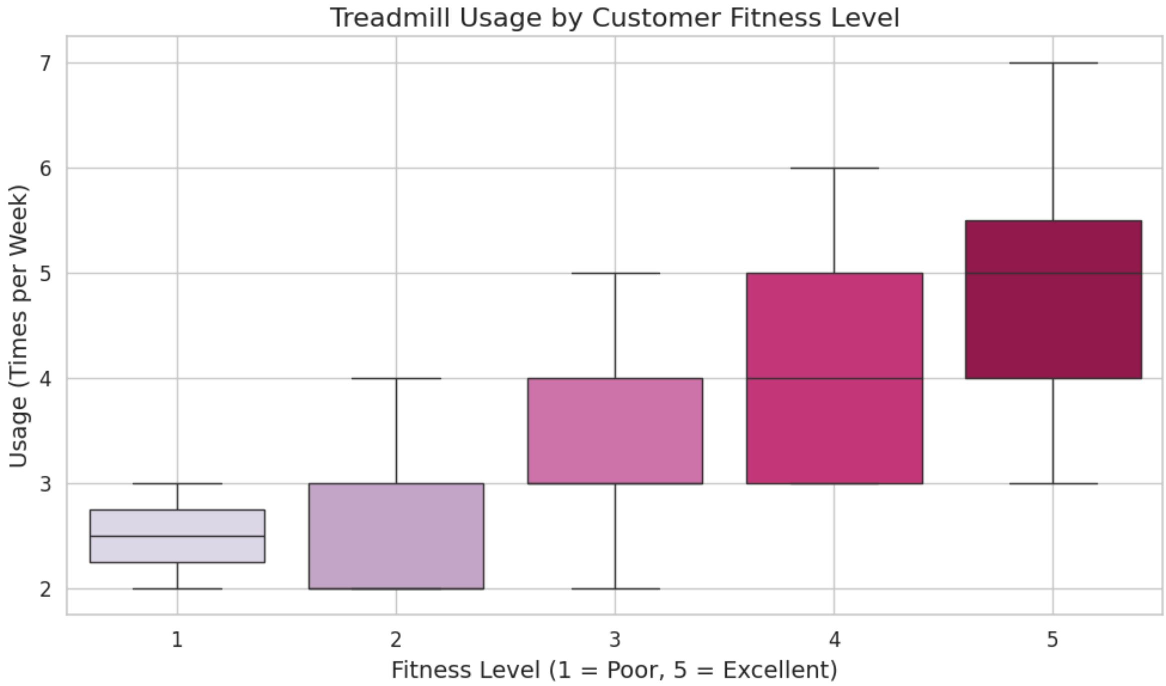


Business insights

:The analysis of customer demographics by gender and marital status reveals key insights for targeting. The customer base is relatively balanced between genders, with a slightly higher proportion of males. More notably, a majority of customers are partnered, indicating that marketing strategies could benefit from considering household fitness needs or appeals to couples. Understanding these distributions provides a foundational layer for more targeted marketing efforts based on these demographic factors.

Fitness vs Usage (Boxplot + Stats)

In [ ]:
```
# Set plot style
sns.set(style="whitegrid")

# Create a detailed boxplot
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='Fitness', y='Usage', palette='PuRd')

# Title and labels
plt.title('Treadmill Usage by Customer Fitness Level', fontsize=16)
plt.xlabel('Fitness Level (1 = Poor, 5 = Excellent)', fontsize=14)
plt.ylabel('Usage (Times per Week)', fontsize=14)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```



Business insights

Looking at how fit people say they are compared to how often they plan to use the treadmill shows a clear pattern: the fitter someone feels, the more days a week they plan to use it. This tells us that people who are already serious about fitness are the ones who will use the treadmill more often. Their planned usage directly reflects how committed they are to working out.

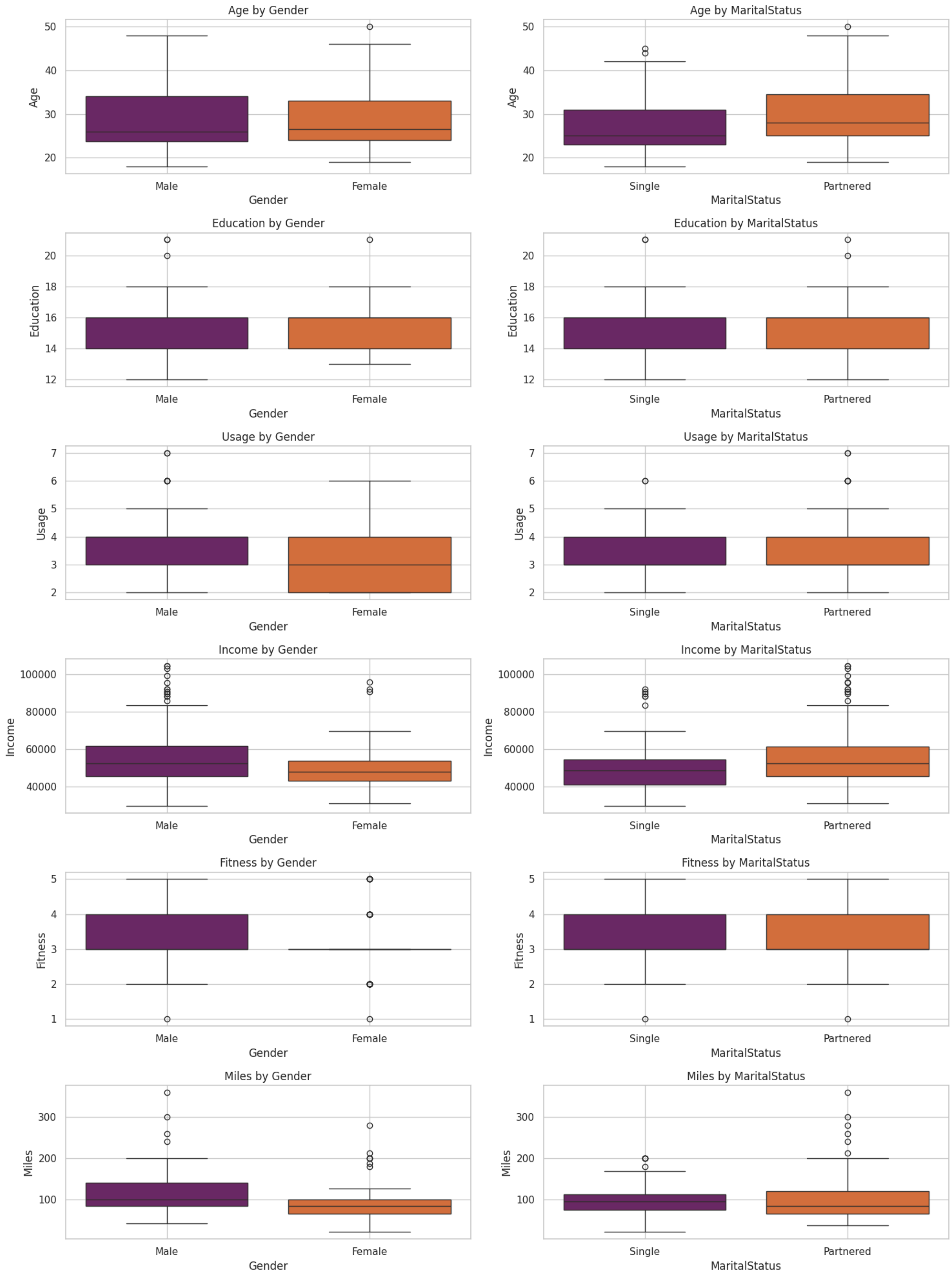
In [ ]:
```
# Grouped statistics
usage_stats = df.groupby('Fitness')['Usage'].describe()
print(usage_stats)
```

```
         count      mean       std  min   25%  50%   75%  max
Fitness
1          2.0  2.500000  0.707107  2.0  2.25  2.5  2.75  3.0
2         26.0  2.538462  0.646886  2.0  2.00  2.0  3.00  4.0
3         97.0  3.164948  0.745548  2.0  3.00  3.0  4.00  5.0
4         24.0  3.916667  0.928611  3.0  3.00  4.0  5.00  6.0
5         31.0  4.838710  1.003221  3.0  4.00  5.0  5.50  7.0
```

categorical variables

boxplots for categorical variables like Gender and MaritalStatus

In [ ]:
```
# Set plot style
sns.set(style="whitegrid")

# Define categorical and continuous variables
categorical_vars = ['Gender', 'MaritalStatus']
continuous_vars = ['Age', 'Education', 'Usage', 'Income', 'Fitness', 'Miles']

# Create subplots
fig, axes = plt.subplots(len(continuous_vars), len(categorical_vars), figsize=(1

# Plot each boxplot
for i, cont in enumerate(continuous_vars):
    for j, cat in enumerate(categorical_vars):
        sns.boxplot(data=df, x=cat, y=cont, ax=axes[i, j], palette='inferno')
        axes[i, j].set_title(f'{cont} by {cat}', fontsize=12)
        axes[i, j].set_xlabel(cat)
        axes[i, j].set_ylabel(cont)
        axes[i, j].grid(True)

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



Business Insights

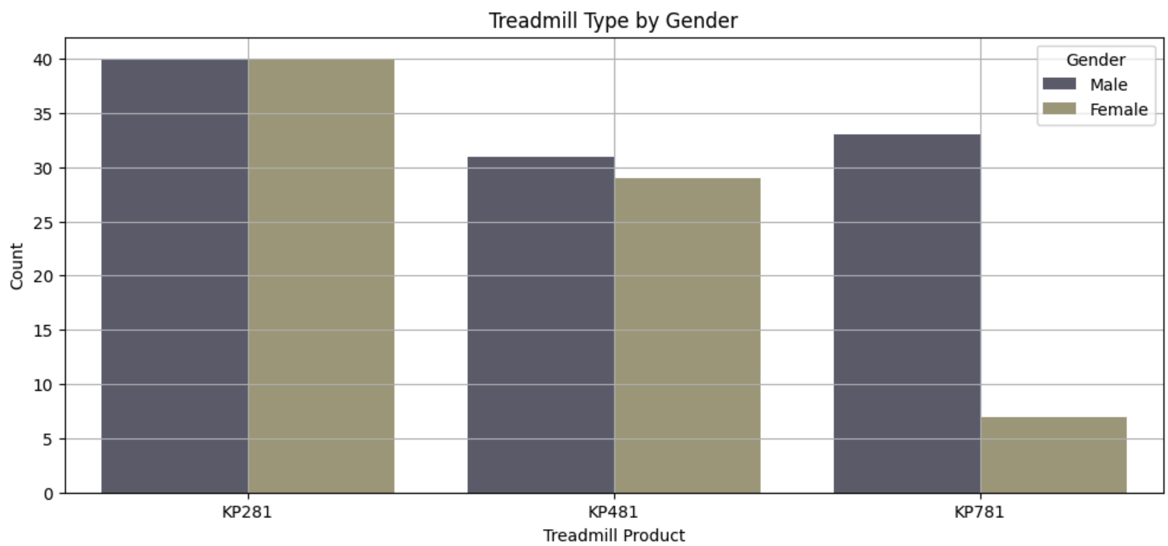
Analyzing the relationship between customer fitness level and planned weekly usage reveals a strong positive correlation: as customers rate their fitness higher, they consistently plan to use their treadmill more times per week. This clearly shows that individuals who are already more dedicated to fitness are the ones who will utilize the equipment more intensely, indicating that usage frequency is a direct reflection of a customer's commitment to their fitness journey.

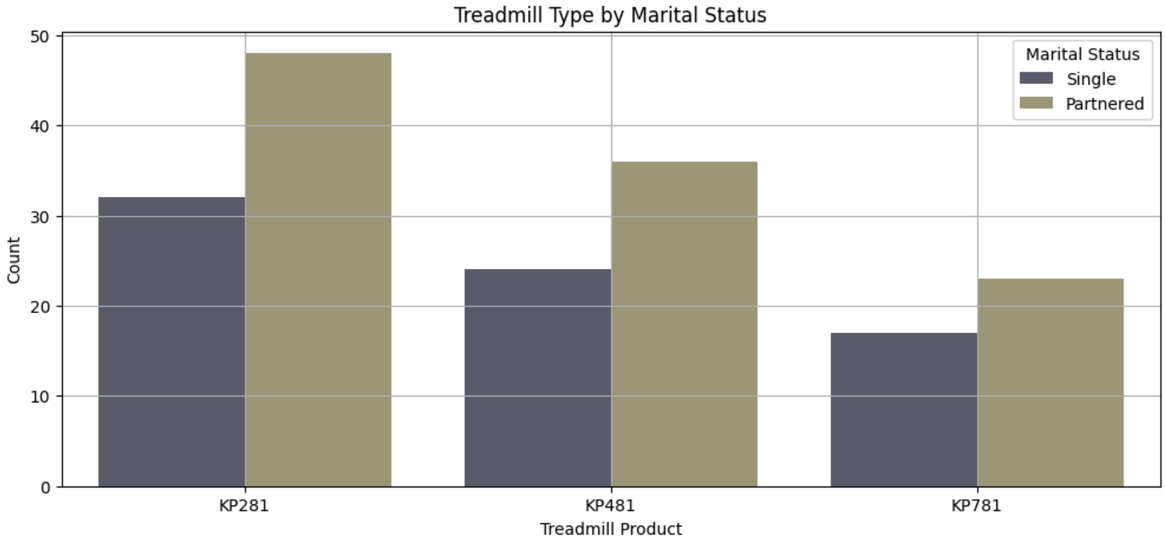
Bivariate Analysis

Bivariate Analysis of Product Type (KP281, KP481, KP781)

Categorical vs Product: Countplots

In [ ]:
```
# Gender vs Product
plt.figure(figsize=(12, 5))
sns.countplot(data=df, x='Product', hue='Gender', palette='cividis')
plt.title('Treadmill Type by Gender')
plt.xlabel('Treadmill Product')
plt.ylabel('Count')
plt.legend(title='Gender')
plt.grid(True)
plt.show()

# Marital Status vs Product
plt.figure(figsize=(12, 5))
sns.countplot(data=df, x='Product', hue='MaritalStatus', palette='cividis')
plt.title('Treadmill Type by Marital Status')
plt.xlabel('Treadmill Product')
plt.ylabel('Count')
plt.legend(title='Marital Status')
plt.grid(True)
plt.show()
```



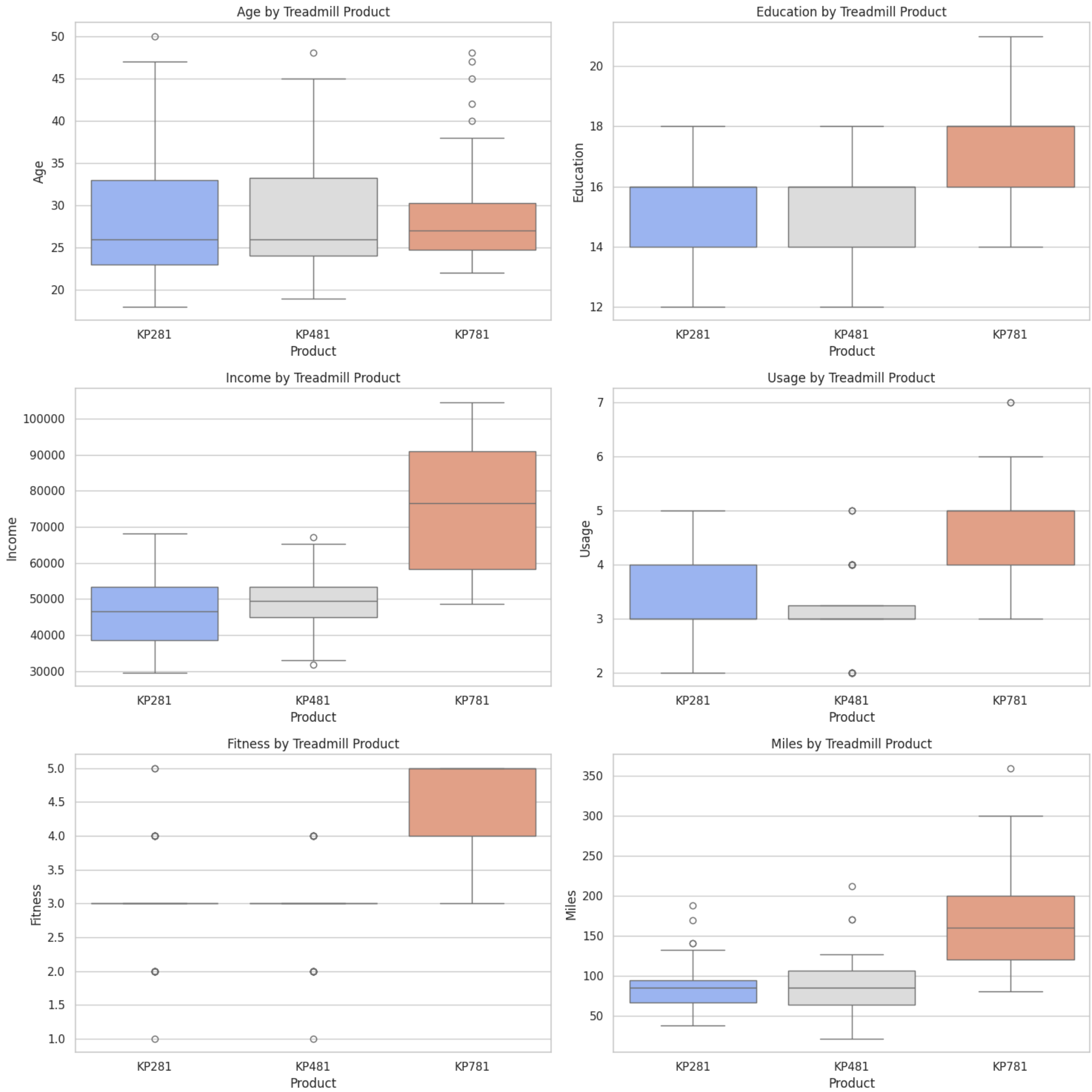


Business Insights

Analyzing categorical variables against product choice reveals distinct patterns. While gender and marital status show some variation in product preference (e.g., a slightly higher proportion of males buying the KP781 and partnered individuals slightly favoring higher-end models), these demographic factors are less strongly correlated with product type compared to fitness-related variables. This suggests that a customer's fitness goals, income, and planned usage are more dominant drivers of their treadmill purchase decision than just their gender or marital status alone.

Continuous vs Product

In [ ]:
```
# Continuous variables to analyze
cont_vars = ['Age', 'Education', 'Income', 'Usage', 'Fitness', 'Miles']

# Plot each continuous variable against Product
import matplotlib.pyplot as plt
fig, axes = plt.subplots(3, 2, figsize=(15, 15))

for i, var in enumerate(cont_vars):
    sns.boxplot(data=df, x='Product', y=var, palette='coolwarm', ax=axes[i//2, i
    axes[i//2, i%2].set_title(f'{var} by Treadmill Product')
    axes[i//2, i%2].set_xlabel('Product')
    axes[i//2, i%2].set_ylabel(var)

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



Business Insights

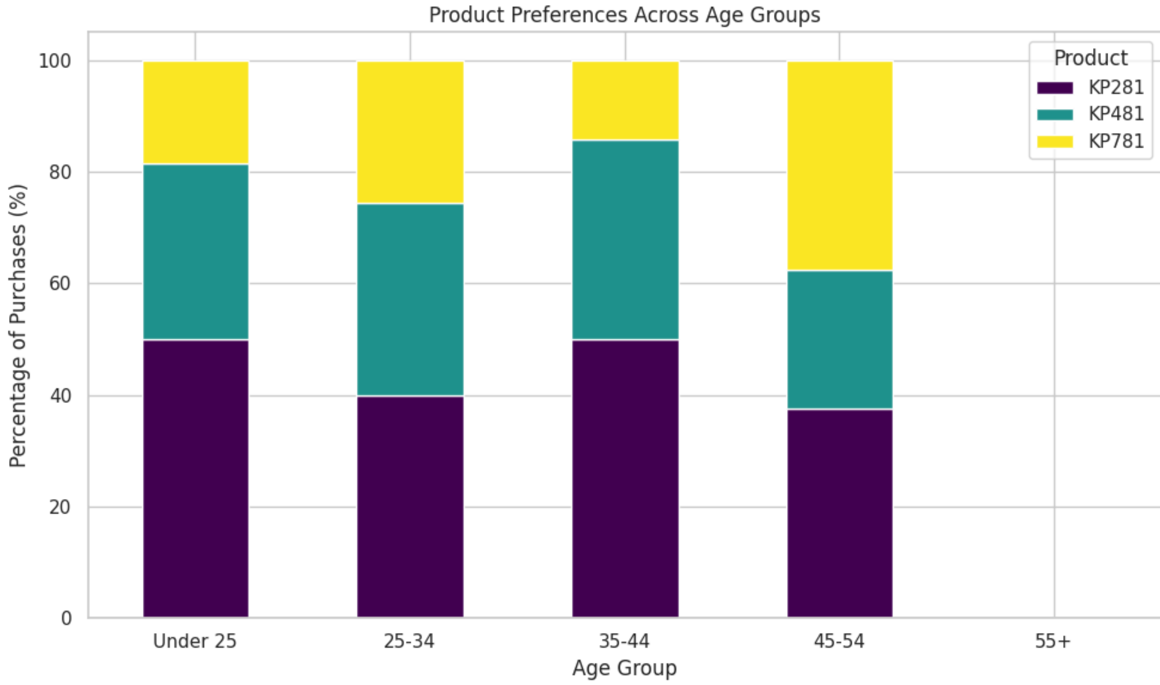
Analyzing continuous variables like Age, Education, Income, Usage, Fitness, and Miles against product choice reveals significant distinctions. Customers purchasing the entry-level KP281 tend to be younger with lower income, education, usage, fitness, and mileage expectations. The mid-range KP481 buyers fall in the middle across these variables, representing a moderate fitness commitment. The premium KP781 attracts older, higher-income, more educated individuals with high fitness levels and plans for frequent, intense workouts, highlighting that financial capacity and serious fitness goals are key drivers for higher-end purchases.

Product Preferences Across Age

In [ ]:
```
# Check the key columns
print(df.columns)  # Look for columns like 'Age', 'Product' or similar

# Step 1: Define age groups
bins = [0, 24, 34, 44, 54, 100]  # Adjust based on data
```

```
labels = ['Under 25', '25-34', '35-44', '45-54', '55+']
df['AgeGroup'] = pd.cut(df['Age'], bins=bins, labels=labels)

# Step 2: Count purchases by age group and product
product_counts = df.groupby(['AgeGroup', 'Product']).size().unstack(fill_value=0

# Step 3: Calculate percentage preference within each age group
product_pct = product_counts.div(product_counts.sum(axis=1), axis=0) * 100

# Step 4: Plotting preference percentages
product_pct.plot(kind='bar', stacked=True, figsize=(10,6), colormap='viridis')

plt.title('Product Preferences Across Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Percentage of Purchases (%)')
plt.legend(title='Product')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```



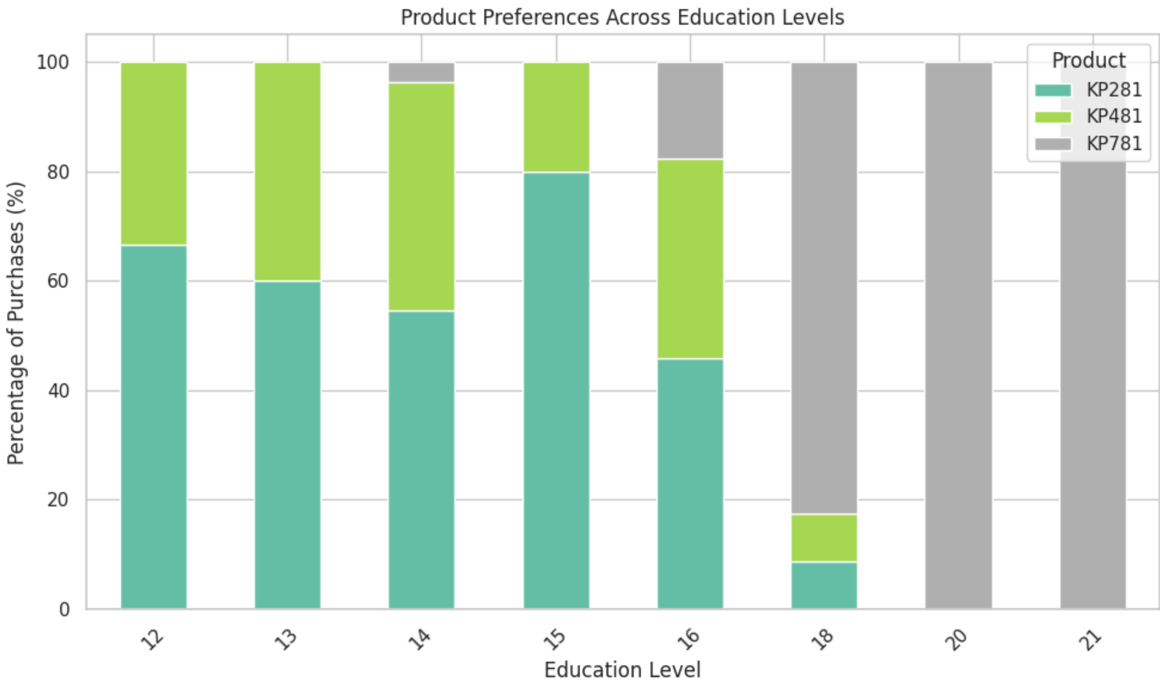
Business Insights

Younger buyers often choose the basic treadmills, while older customers, especially those in their mid-30s to mid-50s, prefer the more advanced models. This shows that as people get older, they're more likely to spend on higher-quality fitness equipment, likely because they have more money and are more serious about fitness. This helps AeroFit know who to market each treadmill to based on their age.

Product Preferences Across Education

In [ ]:
```
# Check columns for education level and product
print(df.columns)  # Look for something like 'Education' and 'Product'

# Step 2: Count purchases by education level and product
edu_product_counts = df.groupby(['Education', 'Product']).size().unstack(fill_va

# Step 3: Calculate percentage preferences
edu_product_pct = edu_product_counts.div(edu_product_counts.sum(axis=1), axis=0)

# Step 4: Plotting
edu_product_pct.plot(kind='bar', stacked=True, figsize=(10,6), colormap='Set2')

plt.title('Product Preferences Across Education Levels')
plt.xlabel('Education Level')
plt.ylabel('Percentage of Purchases (%)')
plt.legend(title='Product')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles', 'AgeGroup'],
      dtype='object')
```



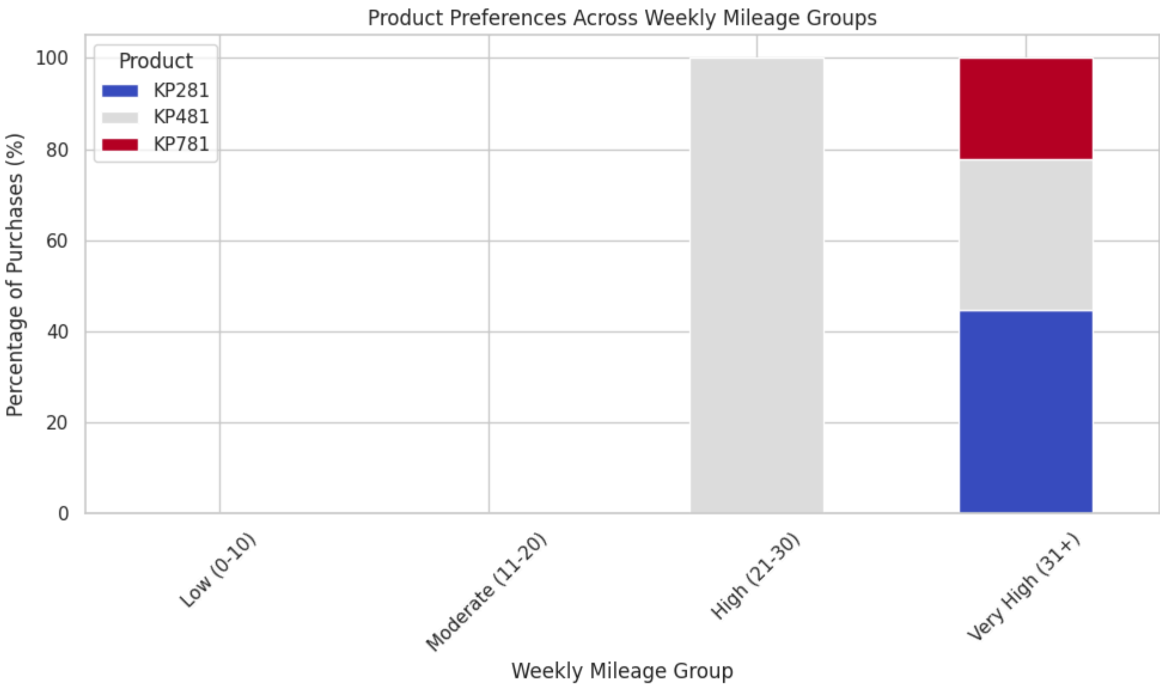
Business Insights

People with more education tend to buy the more expensive, advanced treadmills. Those with less education usually pick the basic models. This shows that education is linked to buying higher-end fitness gear, likely because of higher income and focusing more on health. AeroFit can use this to target their marketing based on how much schooling someone has.

Product preference across customer weekly mileage

In [ ]:
```
# Inspect column names
print(df.columns)  # Look for 'WeeklyMileage' and 'Product'

# Step 1: Define mileage groups (adjust bins as needed)
bins = [0, 10, 20, 30, df['Miles'].max()]
labels = ['Low (0-10)', 'Moderate (11-20)', 'High (21-30)', 'Very High (31+)']
df['MileageGroup'] = pd.cut(df['Miles'], bins=bins, labels=labels, include_lowes

# Step 2: Count purchases by mileage group and product
mileage_product_counts = df.groupby(['MileageGroup', 'Product']).size().unstack(

# Step 3: Calculate percentage preferences
mileage_product_pct = mileage_product_counts.div(mileage_product_counts.sum(axis

# Step 4: Plotting
mileage_product_pct.plot(kind='bar', stacked=True, figsize=(10,6), colormap='coc

plt.title('Product Preferences Across Weekly Mileage Groups')
plt.xlabel('Weekly Mileage Group')
plt.ylabel('Percentage of Purchases (%)')
plt.legend(title='Product')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles', 'AgeGroup'],
      dtype='object')
```

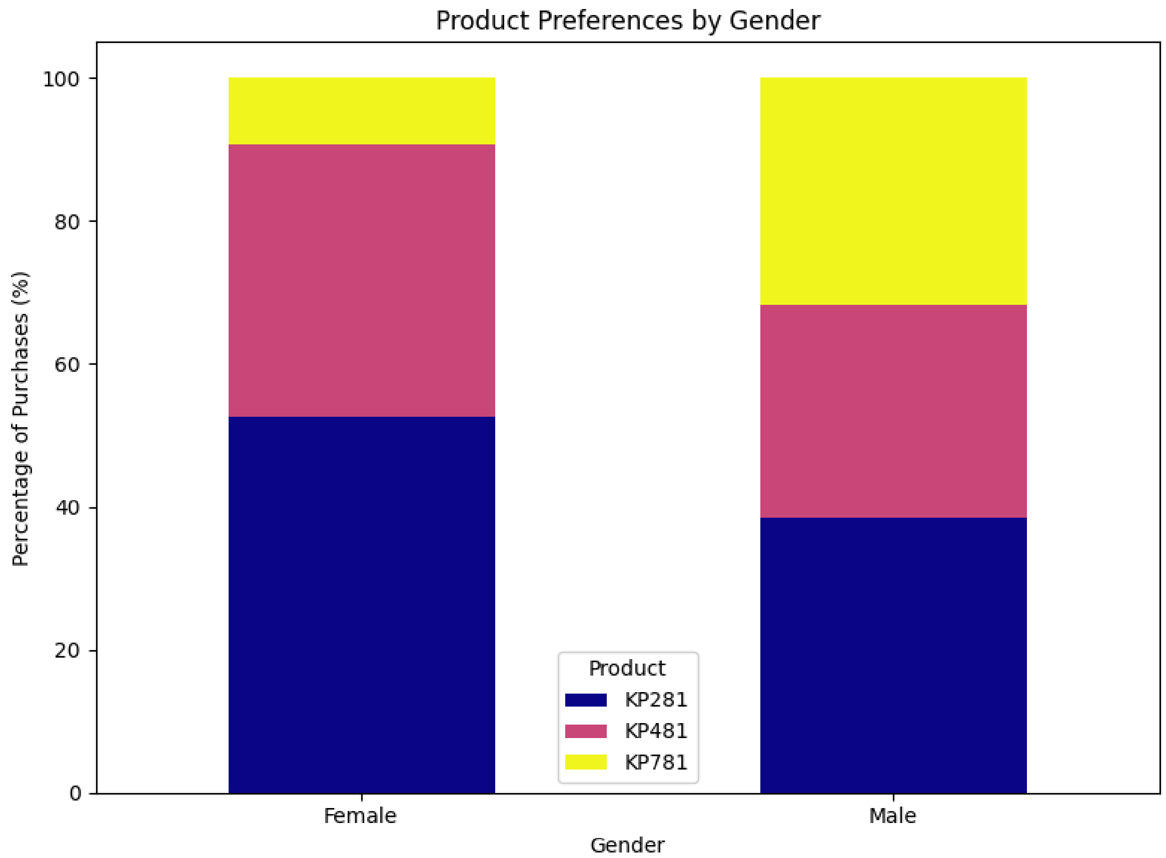


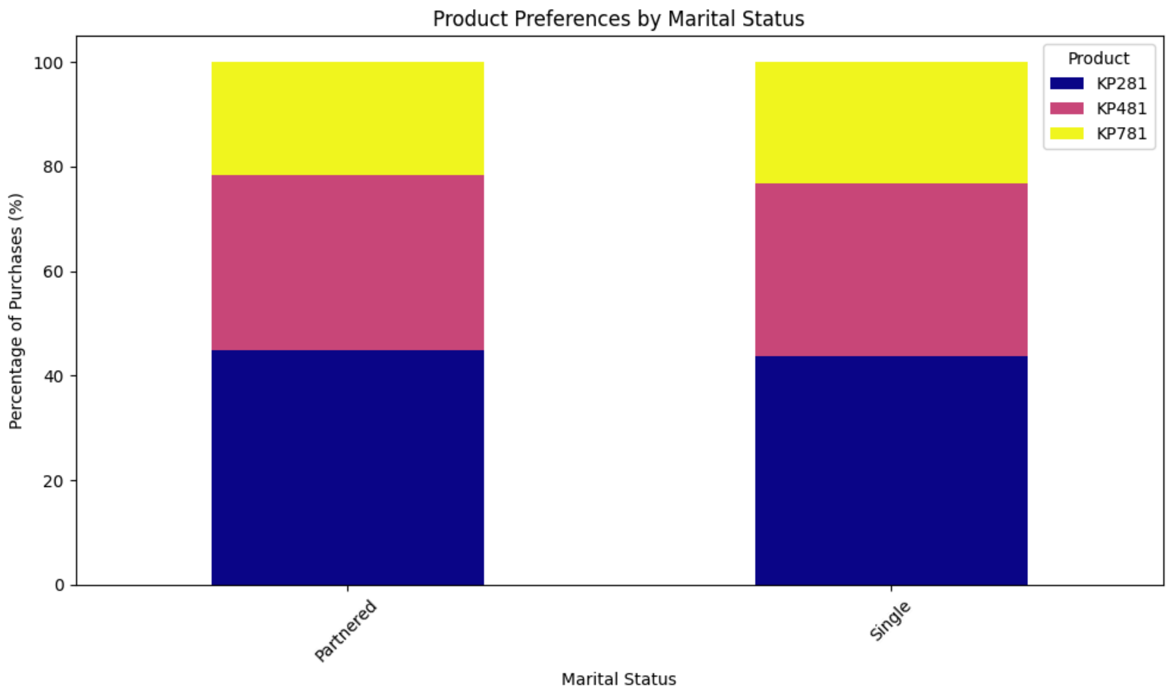
Business Insights

Customers who plan to run more miles each week prefer the advanced treadmills, while those expecting fewer miles opt for basic or mid-range models. This means serious runners who plan for longer workouts buy the more expensive machines, likely needing their durability and features. AeroFit should target customers with high mileage goals for premium treadmills.

Product Preference across Gender and Marital Status

In [ ]:
```
# Group by Gender and Product
gender_product_counts = df.groupby(['Gender', 'Product']).size().unstack(fill_va

# Percentage within each gender
gender_product_pct = gender_product_counts.div(gender_product_counts.sum(axis=1)

# Plot
gender_product_pct.plot(kind='bar', stacked=True, figsize=(8,6), colormap='plasm

plt.title('Product Preferences by Gender')
plt.xlabel('Gender')
plt.ylabel('Percentage of Purchases (%)')
plt.legend(title='Product')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()

# Group by Marital Status and Product
marital_product_counts = df.groupby(['MaritalStatus', 'Product']).size().unstack

# Percentage within each marital status group
marital_product_pct = marital_product_counts.div(marital_product_counts.sum(axis

# Plot
marital_product_pct.plot(kind='bar', stacked=True, figsize=(10,6), colormap='pla

plt.title('Product Preferences by Marital Status')
plt.xlabel('Marital Status')
plt.ylabel('Percentage of Purchases (%)')
plt.legend(title='Product')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```





Business Insights

While gender and marital status show slight differences in treadmill choice, they don't drive product preference as much as factors like income and fitness goals

6. For correlation: Heatmaps, Pairplots

Select Numeric Columns

In [ ]:
```
numeric_cols = ['Age', 'Education', 'Usage', 'Income', 'Fitness', 'Miles']  # Ad
data_num = df[numeric_cols]

correlation_matrix = data_num.corr()
print(correlation_matrix)
```

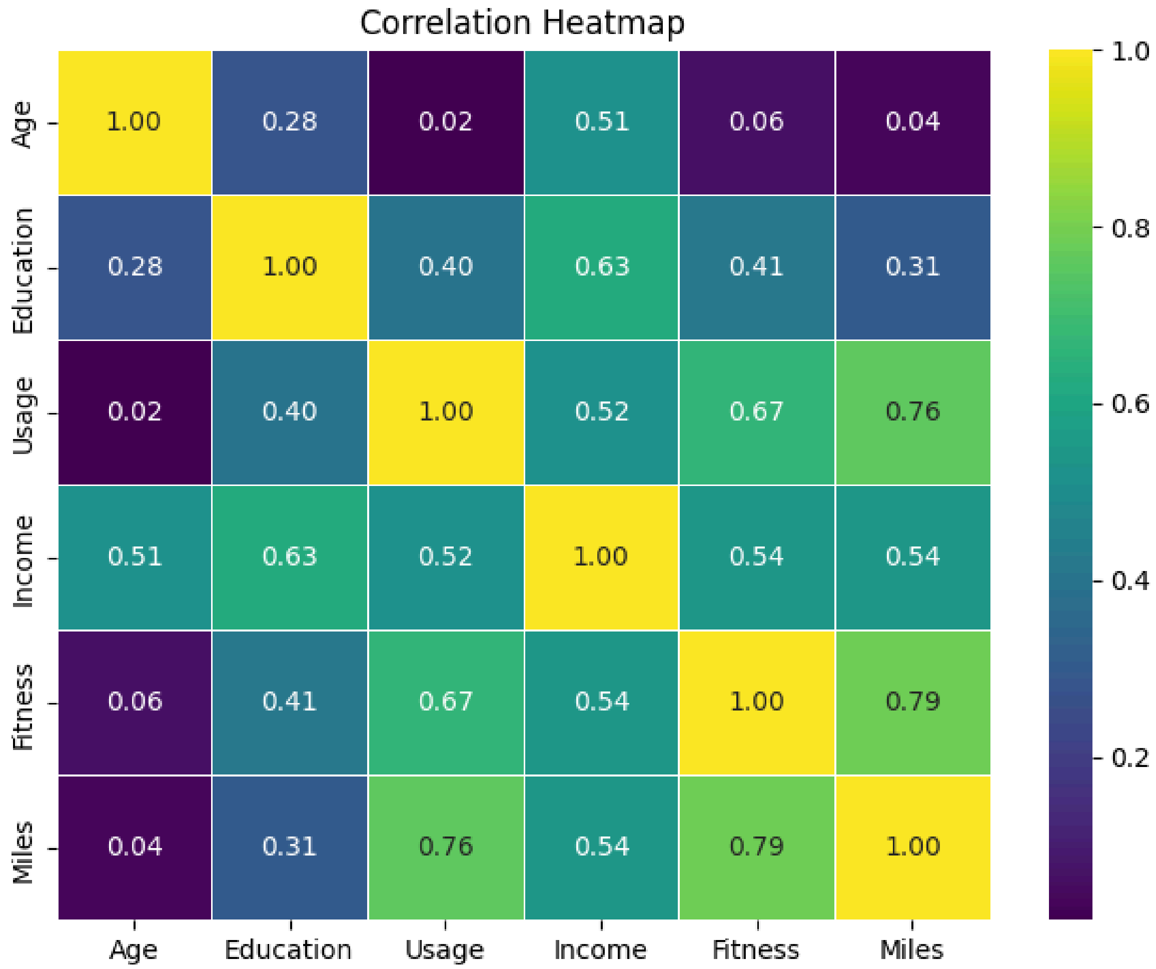
```
                Age  Education     Usage    Income   Fitness     Miles
Age        1.000000   0.280496  0.015064  0.513414  0.061105  0.036618
Education  0.280496   1.000000  0.395155  0.625827  0.410581  0.307284
Usage      0.015064   0.395155  1.000000  0.519537  0.668606  0.759130
Income     0.513414   0.625827  0.519537  1.000000  0.535005  0.543473
Fitness    0.061105   0.410581  0.668606  0.535005  1.000000  0.785702
Miles      0.036618   0.307284  0.759130  0.543473  0.785702  1.000000
```

Calculate Correlation Matrix

In [ ]:
```
corr_matrix = data_num.corr()
```

Plot Correlation Heatmap

In [ ]:
```
plt.figure(figsize=(8,6))
sns.heatmap(corr_matrix, annot=True, cmap='viridis', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```

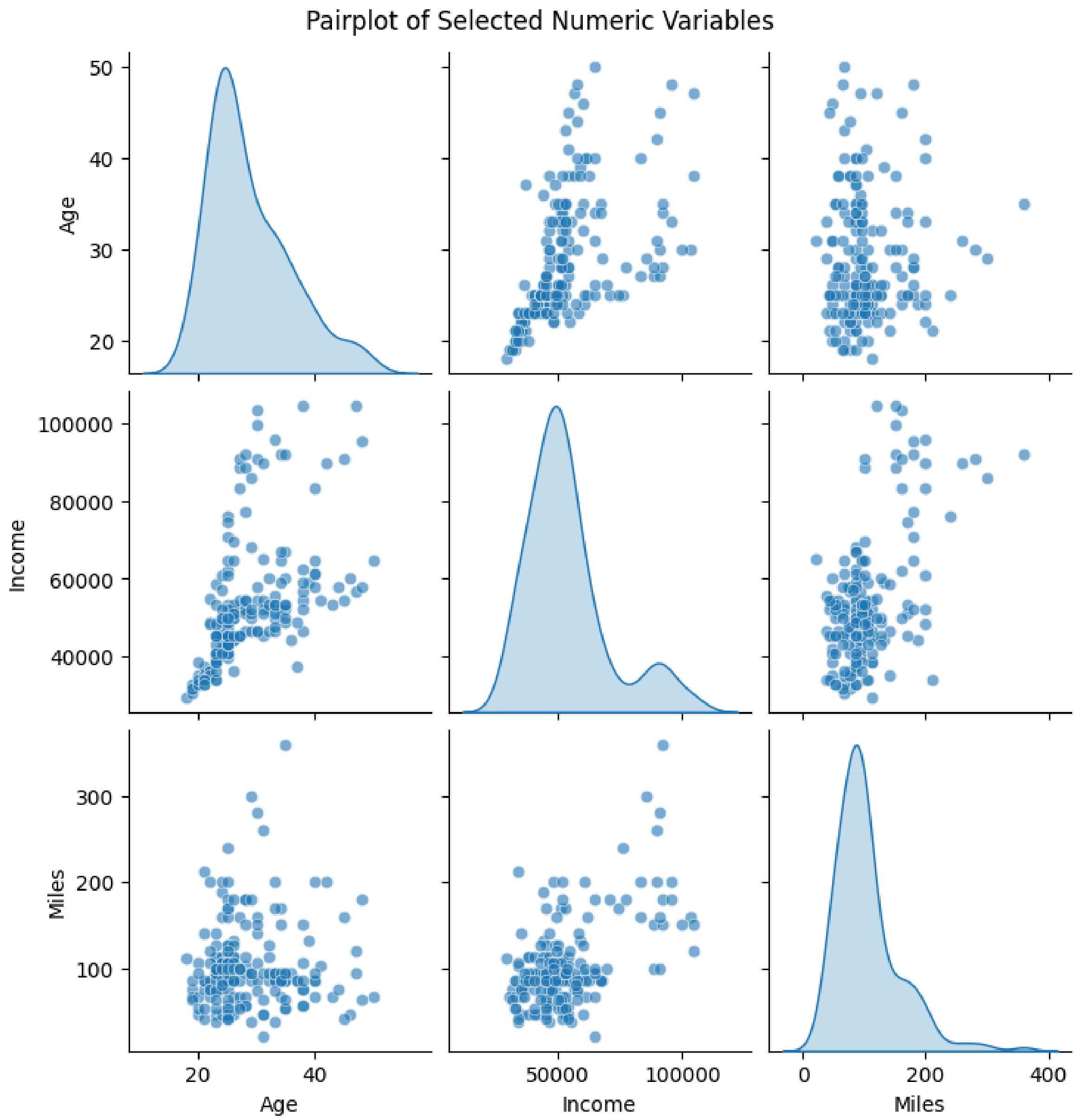


Business Insights

The heatmap shows that things like income, how often someone uses the treadmill, their fitness level, and how many miles they plan to walk/run are all connected. This means people with higher incomes and fitness goals tend to use the treadmills more.

Pairplot for Scatter Matrix

In [ ]:
```
# Select a subset of numeric columns
selected_cols = ['Age', 'Income', 'Miles']

sns.pairplot(data_num[selected_cols], diag_kind='kde', kind='scatter', plot_kws=
plt.suptitle('Pairplot of Selected Numeric Variables', y=1.02)
plt.show()
```



Business Insights

The pairplot helps visualize relationships between key factors like Age, Income, and Miles. It allows us to see if there are clear trends, such as higher income potentially correlating with more planned miles, indicating a link between financial capacity and fitness commitment.

7. Computing Probability - Marginal, Conditional Probability

Probability of product purchase w.r.t. gender

In [ ]:
```
pd.crosstab(index =df['Product'],columns = df['Gender'],margins = True,normalize
```

Out[ ]:

Gender Product	Female	Male	All
KP281	0.22	0.22	0.44
KP481	0.16	0.17	0.33
KP781	0.04	0.18	0.22
All	0.42	0.58	1.00

Probability of product purchase w.r.t. Age

In [ ]:
```
pd.crosstab(pd.cut(df['Age'], bins=[15,25,35,45,55,65], labels=['18-25','26-35',
```

Out[ ]:

Product Age	KP281	KP481	KP781
18-25	0.430380	0.354430	0.215190
26-35	0.438356	0.328767	0.232877
36-45	0.500000	0.318182	0.181818
46-55	0.500000	0.166667	0.333333

Probability of product purchase w.r.t. Education level

In [ ]:
```
pd.crosstab(df['Education'], df['Product'], normalize='index')
```

Out[ ]:

Product Education	KP281	KP481	KP781
12	0.666667	0.333333	0.000000
13	0.600000	0.400000	0.000000
14	0.545455	0.418182	0.036364
15	0.800000	0.200000	0.000000
16	0.458824	0.364706	0.176471
18	0.086957	0.086957	0.826087
20	0.000000	0.000000	1.000000
21	0.000000	0.000000	1.000000

Probability of product purchase w.r.t. Income

In [ ]:
```
pd.crosstab(df['Income'], df['Product'], normalize='index')
```

Out[ ]:

Product Income	KP281	KP481	KP781
29562	1.0	0.0	0.0
30699	1.0	0.0	0.0
31836	0.5	0.5	0.0
32973	0.6	0.4	0.0
34110	0.4	0.6	0.0
...	...	...	...
95508	0.0	0.0	1.0
95866	0.0	0.0	1.0
99601	0.0	0.0	1.0
103336	0.0	0.0	1.0
104581	0.0	0.0	1.0

62 rows × 3 columns

Probability of product purchase w.r.t. Marital Status

In [ ]:
```
pd.crosstab(df['MaritalStatus'], df['Product'], normalize='index')
```

Out[ ]:

Product MaritalStatus	KP281	KP481	KP781
Partnered	0.448598	0.336449	0.214953
Single	0.438356	0.328767	0.232877

Probability of product purchase w.r.t. Weekly Usage

In [ ]:
```
pd.crosstab(df['Usage'], df['Product'], normalize='index')
```

Out[ ]:

Product Usage	KP281	KP481	KP781
2	0.575758	0.424242	0.000000
3	0.536232	0.449275	0.014493
4	0.423077	0.230769	0.346154
5	0.117647	0.176471	0.705882
6	0.000000	0.000000	1.000000
7	0.000000	0.000000	1.000000

Probability of product purchase w.r.t. Customer Fitness

In [ ]:

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

Out[ ]:

Product Fitness	KP281	KP481	KP781
1	0.500000	0.500000	0.000000
2	0.538462	0.461538	0.000000
3	0.556701	0.402062	0.041237
4	0.375000	0.333333	0.291667
5	0.064516	0.000000	0.935484

8.AeroFit Treadmill Customer Profiles

1. KP281 (Entry-Level Model)

1. Age: Bought mainly by younger people in their 20s and early 30s. These are often students, early-career professionals, or fitness beginners.
2. Gender: More popular with females – around 60–70% of buyers.
3. Income: Buyers usually have a low to middle income, so they look for a product that fits their budget.
4. Weekly Usage: Used 2–3 times per week — light or casual exercise.

5. Fitness Level: Most rate their fitness as 2 or 3 out of 5, meaning they're just starting or moderately active.
6. Marital Status: More single or unmarried customers buy this model, possibly living alone or just starting adult life.

Business Insight:

KP281 is ideal for fitness beginners who want a basic and affordable treadmill. Marketing should focus on ease of use, value for money, and starting a fitness journey.

2. KP481 – Mid-Range Treadmill

1. Age: Buyers are generally in their 30s and 40s, meaning they may have more stability in life and career.
2. Gender: Balanced mix of males and females, showing the product appeals to everyone.
3. Income: Middle-income buyers who can afford something better than basic but not overly expensive.
4. Weekly Usage: Used about 3–4 times a week — by regular users who care about staying fit.
5. Fitness Level: Most users rate their fitness as 3 or 4 out of 5 – moderately active to fit.
6. Marital Status: Even mix of single and married people, or slightly more married individuals.

Business Insight:

KP481 attracts people who are serious about fitness but still want a good deal. Marketing should highlight the balance of features and affordability, along with reliability.

3. KP781 – Premium/Advanced Treadmill

1. Age: Popular with people aged 35 to 55. These are usually settled in life and value health highly.

2. Gender: More male buyers, who may be more performance-focused or advanced in their training.

3. Income: High-income group – they can afford premium, high-performance equipment.

4. Weekly Usage: Used 4 or more times per week – showing a strong workout routine.

5. Fitness Level: Most rate their fitness as 4 or 5 out of 5, so they are fit or very fit.

6. Marital Status: More likely to be married or in a relationship.

Business Insight:

KP781 is for serious fitness enthusiasts or runners who need a durable, high-performance machine. Marketing should focus on premium features, durability, advanced workouts, and performance.

9. Recommendations for AeroFit

KP281 – Entry-Level Model

- Target young adults and people with a limited budget.
- Target young adults and people with a limited budget.
- Highlight features like affordability and ease of use – perfect for beginners.
- Use social media platforms (Instagram, YouTube, etc.) to reach younger audiences.

🔹 **KP481 – Mid-Range Model**

- Aim at middle-income individuals with regular fitness routines.
- Promote it as a balanced option – good features at a fair price.
- Advertise on fitness blogs, YouTube fitness channels, and email campaigns.

🔹 **KP781 – Premium Model**

- Focus on high-income professionals and serious fitness enthusiasts.
- Emphasize advanced features, durability, and performance for intense workout
- Use premium marketing channels like LinkedIn ads, fitness forums, and sports magazines.