Business Problem

AeroFit, a leading fitness equipment brand, aims to improve its treadmill recommendations to new customers by better understanding the characteristics of its existing customers. The market research team wants to identify distinct customer profiles for each treadmill model and investigate whether customer traits like age, gender, income, fitness level, and treadmill usage vary significantly across products. This analysis will enable AeroFit to provide more tailored recommendations, optimize marketing strategies, and increase sales.

AeroFit Dataset

Product Purchased: The treadmill model purchased by the customer (KP281, KP481, or KP781).

Age: The age of the customer in years.

Gender: The gender of the customer (Male/Female).

Education: The number of years of formal education completed by the customer.

Marital Status: The marital status of the customer (Single or Partnered).

Usage: The average number of times per week the customer plans to use the treadmill.

Income: The customer's annual income (in dollars).

Fitness: The customer's self-rated fitness level on a scale from 1 (poor shape) to 5 (excellent shape).

Miles: The average number of miles the customer expects to walk or run on the treadmill each week.

Product Portfolio:

AeroFit offers a range of treadmills tailored to different customer needs and fitness levels, allowing the brand to target various segments of the market:

- 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.

By offering three distinct treadmill models, AeroFit is catering to a broad spectrum of customers, from casual users to advanced athletes, providing a product for every level of fitness and budget.

Libraries Utilized for Data Exploration and Analysis

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Loading the Dataset for Analysis

In [2]:
```
df = pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadı
```

Previewing the First Five Entries of the Dataset

In [3]:
```
df.head(5)
```

Out[3]:

	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

Analysing basic metrics

In [4]:
```
print(f"Total Number of rows: {df.shape[0]}")
print(f"Total Number of columns: {df.shape[1]}")
```

```
Total Number of rows: 180
Total Number of columns: 9
```

Summary of the DataFrame Structure

In [5]:
```
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
```

Type conversion of Catagorical data

In [6]:
```
categorical_columns = ['Product', 'MaritalStatus','Gender']
df[categorical_columns] = df[categorical_columns].astype('category')
```

Summary of the DataFrame Structure after Conversion

In [7]:
```
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    category
 1   Age            180 non-null    int64   
 2   Gender         180 non-null    category
 3   Education      180 non-null    int64   
 4   MaritalStatus  180 non-null    category
 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: category(3), int64(6)
memory usage: 9.5 KB
```

Statistical Summary

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

Out[8]:

	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

Observations

- Customer Age: Most customers are around 29 years old, with ages ranging from 18 to 50. This suggests that the treadmills appeal to a relatively young to middle-aged audience.
- Education: On average, customers have around 16 years of education, indicating that they are likely college-educated.
- Treadmill Usage: Customers plan to use the treadmill about 3 to 4 times per week on average. This shows moderate to regular usage, suggesting that the products are mostly used for consistent exercise routines.
- Fitness Levels: Most customers rate their fitness as average (3 on a scale of 1 to 5). This indicates that the majority are likely fitness-conscious but not necessarily advanced athletes.
- Income: The average annual income of customers is around 54,000. This shows that AeroFit's treadmills attract middle-income customers who have disposable income for fitness equipment.
- Miles: On average, customers expect to use the treadmill to walk or run about 103 miles per week, but this varies widely. This could indicate that some users are casual walkers, while others are more serious runners.

Data Sanitization

Examination of Missing Values by Column

In [9]:
```
df.isna().sum()
```

Out[9]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

The dataset does not contain any missing values

Exploring Data and Conducting Non-Graphical Analysis

Unique Values, Count of Unique Values and Frequency of Unique Values in the Dataset.

In [10]:
```
print(f"Total Number of Unique Product in the Dataset: {df['Product'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['Product'].unique()}")
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['Product'].value_counts().head(5)}")
```

```
Total Number of Unique Product in the Dataset: 3
------------------------------
Unique Values in the Dataset:
 ['KP281', 'KP481', 'KP781']
Categories (3, object): ['KP281', 'KP481', 'KP781']
------------------------------
Available frequency Values in the Dataset:
 Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
```

Observation

Unique Product Portfolio:

- The dataset features three treadmill models: KP281, KP481, and KP781, catering to diverse customer needs.
- Sales Performance:
  - KP281 leads with 80 units sold, indicating strong demand among budget-conscious consumers.
  - KP481 has 60 units sold, appealing to customers seeking value without compromising features.
  - KP781 records 40 units sold, targeting a niche market of serious fitness enthusiasts.
- Strategic Insights:The popularity of the KP281 presents
- a significant opportunity for marketing efforts focused on entry-level users.
- The sales distribution allows AeroFit to tailor promotions and inventory strategies for each product segment effectively.

In [11]:
```
print(f"Total Number of Unique Age in the Dataset: {df['Age'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['Age'].unique()}")
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['Age'].value_counts().head(5)}")
```

```
Total Number of Unique Age in the Dataset: 32
------------------------------
Unique Values in the Dataset:
 [18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
 43 44 46 47 50 45 48 42]
------------------------------
Available frequency Values in the Dataset:
 Age
25    25
23    18
24    12
26    12
28     9
Name: count, dtype: int64
```

In [12]:
```
print(f"Total Number of Unique Gender in the Dataset: {df['Gender'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['Gender'].unique()}")
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['Gender'].value_counts().head(5)}")
```

```
Total Number of Unique Gender in the Dataset: 2
------------------------------
Unique Values in the Dataset:
 ['Male', 'Female']
Categories (2, object): ['Female', 'Male']
------------------------------
Available frequency Values in the Dataset:
 Gender
Male      104
Female     76
Name: count, dtype: int64
```

The higher number of Male customers indicates a potential opportunity to tailor marketing strategies to engage more female customers.

In [13]:
```
print(f"Total Number of Unique Education in the Dataset: {df['Education'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['Education'].unique()}")
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['Education'].value_counts().head(5)}")
```

```
Total Number of Unique Education in the Dataset: 8
------------------------------
Unique Values in the Dataset:
 [14 15 12 13 16 18 20 21]
------------------------------
Available frequency Values in the Dataset:
 Education
16    85
14    55
18    23
15     5
13     5
Name: count, dtype: int64
```

In [14]:
```
print(f"Total Number of Unique MaritalStatus in the Dataset: {df['MaritalStatus'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['MaritalStatus'].unique()}")
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['MaritalStatus'].value_counts().head(5)}")
```

```
Total Number of Unique MaritalStatus in the Dataset: 2
------------------------------
Unique Values in the Dataset:
 ['Single', 'Partnered']
Categories (2, object): ['Partnered', 'Single']
------------------------------
Available frequency Values in the Dataset:
 MaritalStatus
Partnered    107
Single        73
Name: count, dtype: int64
```

The higher number of Partnered customers suggests opportunities for targeted marketing to couples.

In [15]:
```
print(f"Total Number of Unique Usage in the Dataset: {df['Usage'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['Usage'].unique()}")
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['Usage'].value_counts().head(5)}")
```

```
Total Number of Unique Usage in the Dataset: 6
------------------------------
Unique Values in the Dataset:
 [3 2 4 5 6 7]
------------------------------
Available frequency Values in the Dataset:
 Usage
3    69
4    52
2    33
5    17
6     7
Name: count, dtype: int64
```

The most frequent usage is 3 times per week, suggesting that marketing efforts can emphasize maintaining or increasing usage among customers.

In [16]:
```
print(f"Total Number of Unique Fitness in the Dataset: {df['Fitness'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['Fitness'].unique()}")
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['Fitness'].value_counts().head(5)}")
```

```
Total Number of Unique Fitness in the Dataset: 5
------------------------------
Unique Values in the Dataset:
 [4 3 2 1 5]
------------------------------
Available frequency Values in the Dataset:
 Fitness
3    97
5    31
2    26
4    24
1     2
Name: count, dtype: int64
```

The majority of customers rate their fitness as average (3), indicating potential for marketing strategies focused on improving fitness levels.

In [17]:
```
print(f"Total Number of Unique Income in the Dataset: {df['Income'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['Income'].unique()}")
```

```
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['Income'].value_counts().head(5)}")
```

```
Total Number of Unique Income in the Dataset: 62
------------------------------
Unique Values in the Dataset:
 [ 29562  31836  30699  32973  35247  37521  36384  38658  40932  34110
  39795  42069  44343  45480  46617  48891  53439  43206  52302  51165
  50028  54576  68220  55713  60261  67083  56850  59124  61398  57987
  64809  47754  65220  62535  48658  54781  48556  58516  53536  61006
  57271  52291  49801  62251  64741  70966  75946  74701  69721  83416
  88396  90886  92131  77191  52290  85906 103336  99601  89641  95866
 104581  95508]
------------------------------
Available frequency Values in the Dataset:
 Income
45480    14
52302     9
46617     8
54576     8
53439     8
Name: count, dtype: int64
```

In [18]:
```
print(f"Total Number of Unique Miles in the Dataset: {df['Miles'].nunique()}")
print("-"*30)
print(f"Unique Values in the Dataset:\n {df['Miles'].unique()}")
print("-"*30)
print(f"Available frequency Values in the Dataset:\n {df['Miles'].value_counts().head(5)}")
```

```
Total Number of Unique Miles in the Dataset: 37
------------------------------
Unique Values in the Dataset:
 [112  75  66  85  47 141 103  94 113  38 188  56 132 169  64  53 106  95
 212  42 127  74 170  21 120 200 140 100  80 160 180 240 150 300 280 260
 360]
------------------------------
Available frequency Values in the Dataset:
 Miles
85    27
95    12
66    10
75    10
47     9
Name: count, dtype: int64
```

In [19]:
```
print(f"Oldest consumer in the dataset: {df['Age'].max()}")
print(f"Youngest consumer in the dataset: {df['Age'].min()}")
```

```
Oldest consumer in the dataset: 50
Youngest consumer in the dataset: 18
```

In [20]:
```
print(f"Longest Run in the dataset: {df['Miles'].max()}")
print(f"Shortest Run in the dataset: {df['Miles'].min()}")
```

```
Longest Run in the dataset: 360
Shortest Run in the dataset: 21
```

In [21]:
```
print(f"Highest income in the dataset: {df['Income'].max()}")
print(f"Lowest income in the dataset: {df['Income'].min()}")
print(f"Average income in the dataset: {df['Income'].mean()}")
print(f"Income standard deviation in the dataset: {df['Income'].std()}")
```

```
Highest income in the dataset: 104581
Lowest income in the dataset: 29562
Average income in the dataset: 53719.57777777778
Income standard deviation in the dataset: 16506.68422623862
```

Treadmill Preferences for Highest and Lowest Income Consumer

In [22]:
```
df.sort_values('Income', ascending=True)[['Product', 'Income']].head(3)
```

Out[22]:

	Product	Income
0	KP281	29562
2	KP281	30699
1	KP281	31836

In [23]:
```
df.sort_values('Income', ascending=False)[['Product', 'Income']].head(3)
```

Out[23]:

	Product	Income
174	KP781	104581
178	KP781	104581
168	KP781	103336

Observations:

- **Lowest Income Users**: All three lowest income entries are associated with the KP281 treadmill, which is the entry-level model priced at $1,500. This indicates that budget-conscious consumers prioritize affordable options, likely due to financial constraints.
- **Highest Income Users**:The highest income entries are dominated by the KP781 treadmill, priced at $2,500, which offers advanced features. This suggests that affluent customers are willing to invest in premium fitness equipment, reflecting a preference for quality and advanced functionalities.

let's create new column before analys

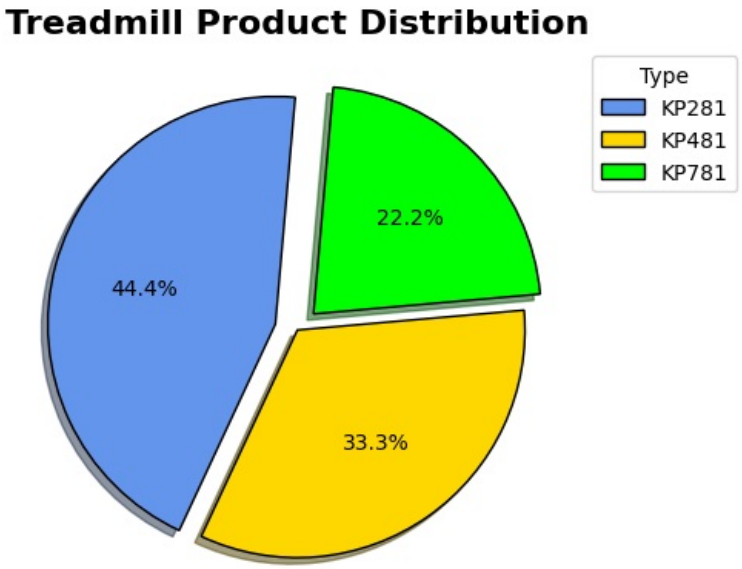
In [24]:
```
bin_range1 = [17,25,35,45,float('inf')]
bin_labels1 = ['Young Adults', 'Adults', 'Middle Aged Adults', 'Elder']
df['age_group'] = pd.cut(df['Age'],bins = bin_range1,labels = bin_labels1)

bin_range2 = [0,12,15,float('inf')]
bin_labels2 = ['Primary Education', 'Secondary Education', 'Higher Education']
df['edu_group'] = pd.cut(df['Education'],bins = bin_range2,labels = bin_labels2)

bin_range3 = [0,40000,60000,80000,float('inf')]
bin_labels3 = ['Low Income','Moderate Income','High Income','Very High Income']
df['income_group'] = pd.cut(df['Income'],bins = bin_range3,labels = bin_labels3)
```

Data Exploration and Conducting Graphical Analysis

In [25]:
```
types = df.Product.value_counts()
plt.pie(types,  autopct='%1.1f%%' , colors = ['cornflowerblue','gold','lime'],
          startangle=85,
    explode=(0.1, 0,0.1),
    shadow=True,
    wedgeprops={'edgecolor': 'black'})
plt.legend(types.index, title="Type", loc="upper left", bbox_to_anchor=(1, 0, 0.5, 1))
plt.title('Treadmill Product Distribution',fontsize=16, fontweight='bold')
plt.show()
```



Observations

KP281 is the most popular treadmill model, accounting for 44.4% of sales.

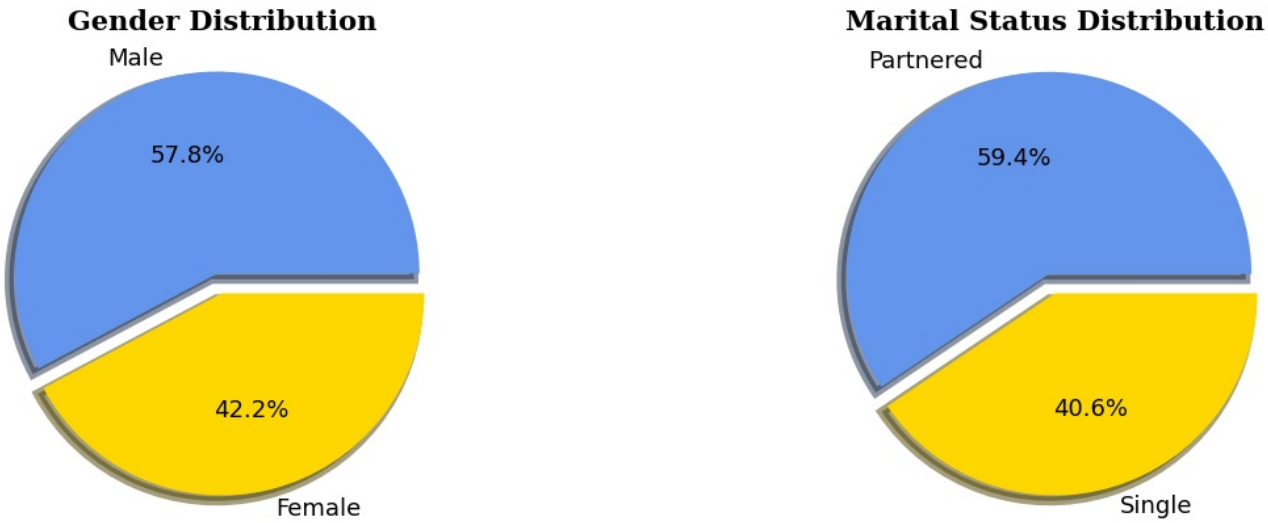
KP481 is the second most popular model, with 33.3% of sales.

KP781 is the least popular model, representing 22.2% of sales.

Overall, the pie chart indicates a clear preference for the KP281 model among customers. This could be due to various factors such as price, features, or perceived value.

In [26]:
```
fig, ax = plt.subplots(1, 2, figsize=(15, 5))
ax[0].pie(df['Gender'].value_counts().values,labels = df['Gender'].value_counts().index,autopct = '%.1f%%',
        shadow = True,colors = ['cornflowerblue','gold'],wedgeprops = {'linewidth': 5},textprops={'fontsize': 1
              explode=(0.1, 0),)
ax[0].set_title('Gender Distribution',{'font':'serif', 'size':15,'weight':'bold'})
ax[1].pie(df['MaritalStatus'].value_counts().values,labels = df['MaritalStatus'].value_counts().index,autopct =
        shadow = True,colors = ['cornflowerblue','gold'],wedgeprops = {'linewidth': 5},textprops={'fontsize': 1
ax[1].set_title('Marital Status Distribution',{'font':'serif', 'size':15,'weight':'bold'})

plt.show()
```



Observations:

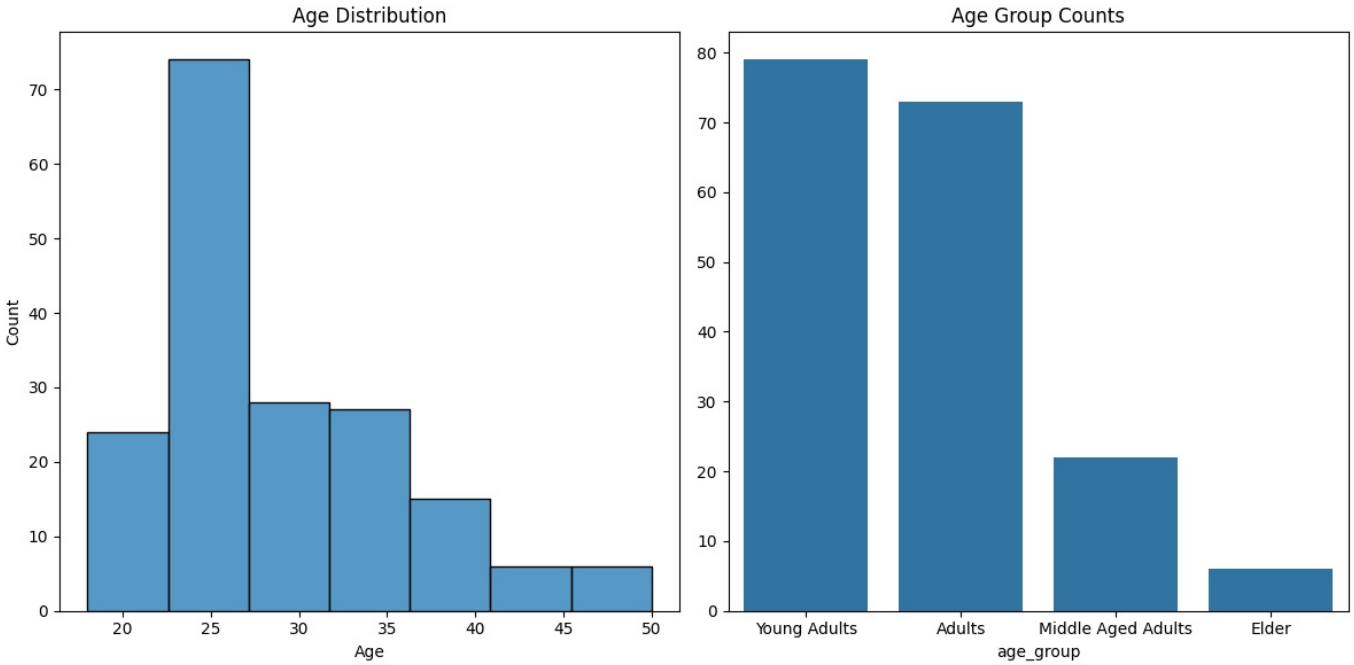
- Gender Distribution:
  - Male: 57.8% of customers are male.
  - Female: 42.2% of customers are female.
- Marital Status Distribution:
  - Partnered: 59.4% of customers are partnered.
  - Single: 40.6% of customers are single.

Overall, the pie charts indicate a slight majority of male customers and partnered individuals in the dataset. This information can be useful for understanding the target customer base and tailoring marketing strategies accordingly. For example, Aerofit could consider creating targeted campaigns specifically for male customers or partnered individuals.

In [27]:
```
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

sns.histplot(df['Age'], bins=7, ax=ax[0])
ax[0].set_title('Age Distribution')
sns.barplot(x=df['age_group'].value_counts().index,
            y=df['age_group'].value_counts().values,
            ax=ax[1])
ax[1].set_title('Age Group Counts')

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



Observations:

Age Distribution:

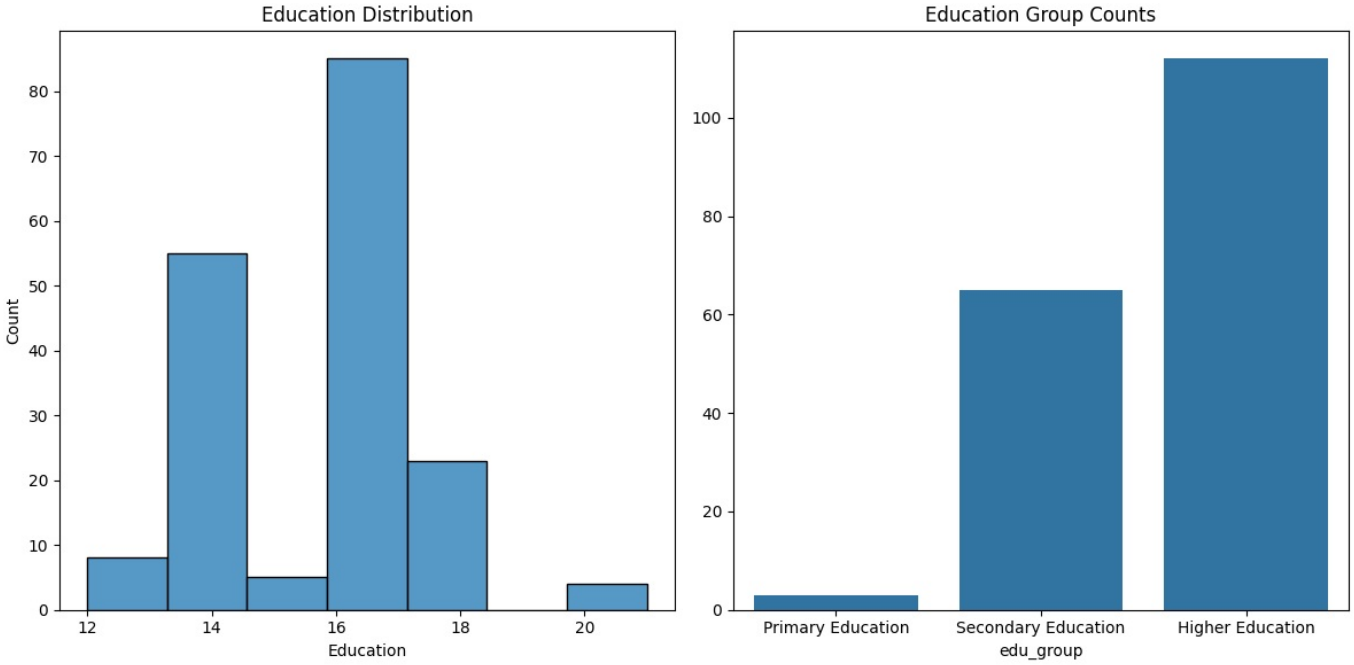
- The age distribution is skewed to the right, indicating that there are more younger customers compared to older customers.
- The majority of customers are between 25 and 30 years old.
- There is a smaller number of customers in the older age groups.

Age Group Counts:

- Young Adults: The largest group of customers is in the "Young Adults" category, followed by "Adults."
- Middle-Aged Adults: There is a smaller number of customers in the "Middle-Aged Adults" category.
- Elder: The smallest group of customers is in the "Elder" category.

Overall, the plots suggest that Aerofit's customer base is primarily made up of young adults. This information can be valuable for targeting marketing efforts and developing products that cater to the needs and preferences of this demographic.

In [28]:
```
fig, ax = plt.subplots(1, 2, figsize=(12, 6))
sns.histplot(df['Education'], bins=7, ax=ax[0])
ax[0].set_title('Education Distribution')
sns.barplot(x=df['edu_group'].value_counts().index,
            y=df['edu_group'].value_counts().values,
            ax=ax[1])
ax[1].set_title('Education Group Counts')
plt.tight_layout()
```



Observations

Education Distribution:

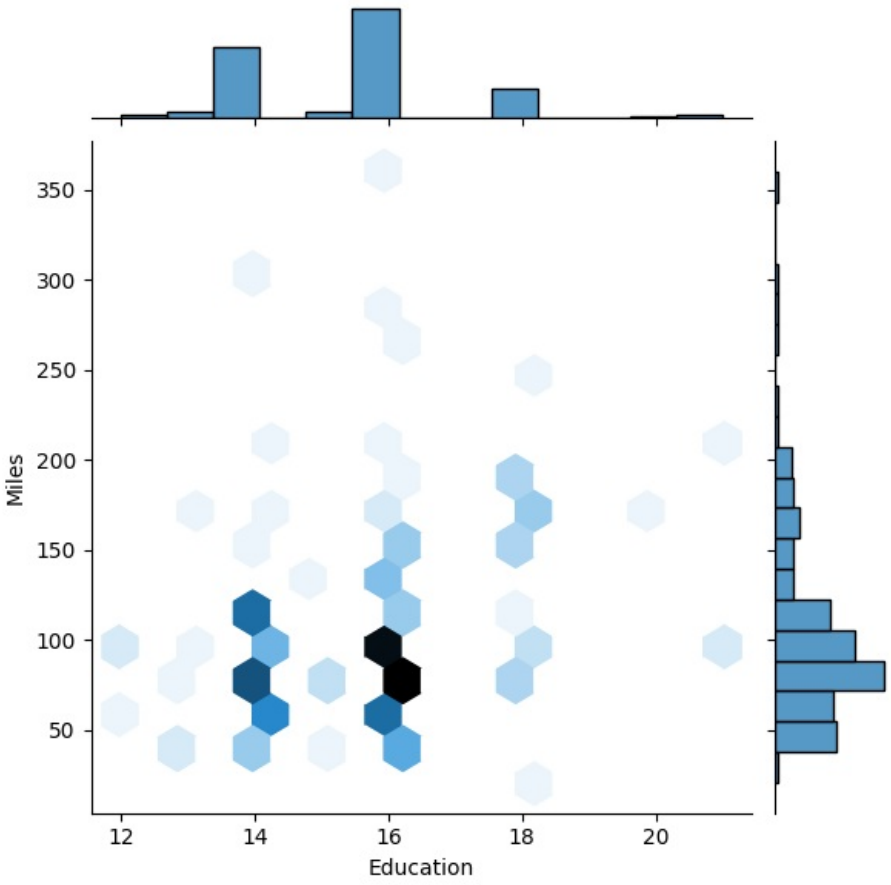
- The distribution of education levels is skewed to the right, indicating that there are more customers with higher education levels.
- The majority of customers have completed 16 years of education.
- There is a smaller number of customers with lower education levels.

Education Group Counts:

- Secondary Education: The largest group of customers has completed secondary education.
- Higher Education: There is a significant number of customers with higher education.
- Primary Education: The smallest group of customers has completed primary education.

Overall, the plots suggest that Aerofit's customer base is well-educated, with a majority of customers having completed secondary or higher education. This information can be valuable for targeting marketing efforts and developing products that cater to the needs of educated customers.

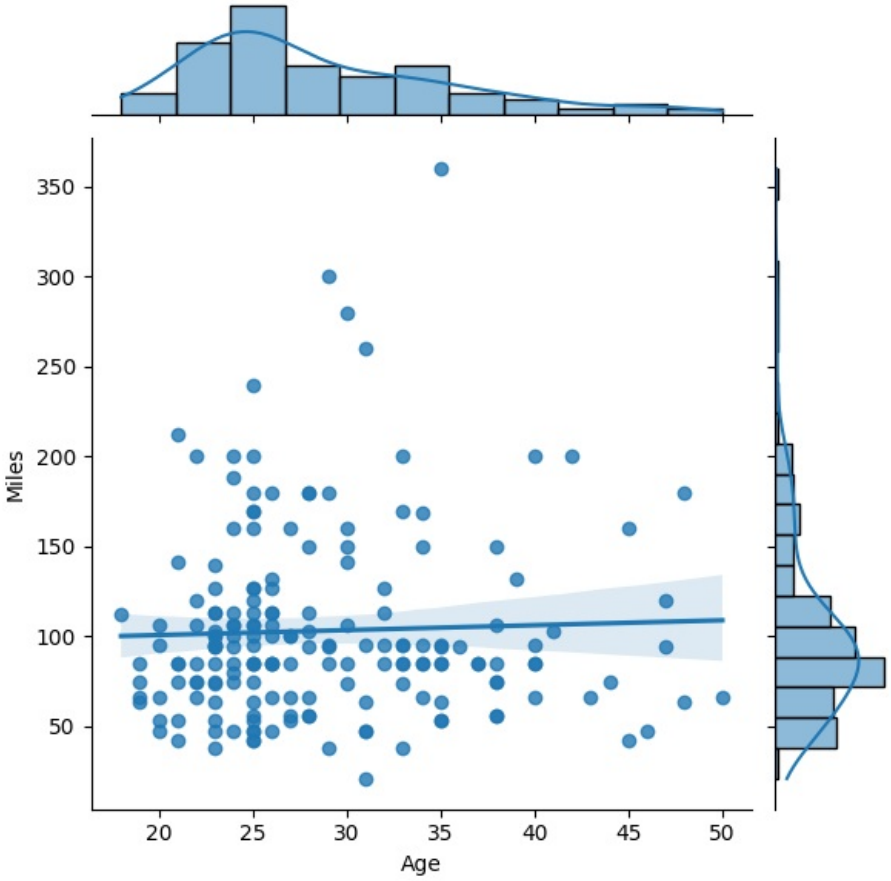
In [29]:
```
sns.jointplot(x="Education", y="Miles", data=df, kind='hex')
plt.show()
```



Observations:

- Concentration: There's a clear concentration of data points in the middle of the plot, indicating that the majority of customers have a moderate level of education and tend to walk or run a moderate number of miles per week.
- Positive Correlation: The hexagonal bins are slightly elongated in the upward direction, suggesting a weak positive correlation between education and miles walked. This means that, generally, customers with higher education levels tend to walk or run more miles.
- Outliers: A few outliers can be observed, particularly in the upper right corner of the plot, indicating that some customers with higher education levels walk or run significantly more miles than the average.

In [30]:
```
sns.jointplot(x="Age", y="Miles", data=df, kind='reg')
plt.show()
```



Observations:

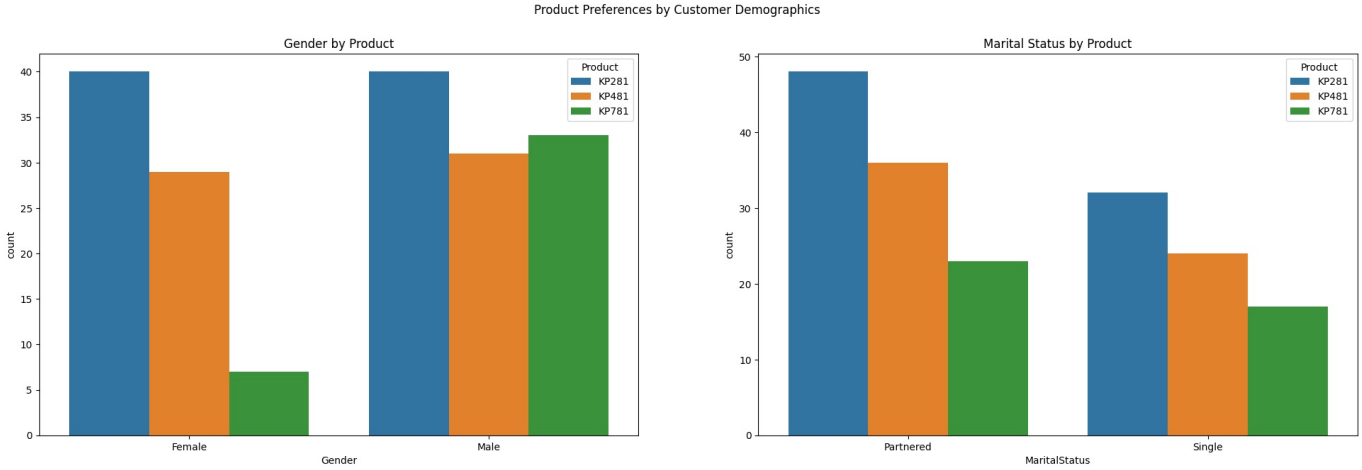
Key Observations:

- Scatterplot: The scatterplot shows a general positive correlation between age and miles walked. This suggests that, as customers get older, they tend to walk or run more miles. However, there is also a significant amount of variation around this trend, indicating that age is not the sole determinant of miles walked.
- Distributions: The marginal histograms show the distributions of age and miles. The age distribution is skewed to the right, indicating that there are more younger customers compared to older customers. The miles distribution is also skewed to the right, suggesting that a small number of customers walk or run significantly more miles than the average.
- Regression Line: The regression line shows the linear relationship between age and miles walked. While there is a positive correlation, the slope of the line is relatively shallow, indicating that the relationship is not very strong.

Business Insights:

- Target Market: Aerofit could focus on marketing their treadmills to customers in different age groups. For example, they could emphasize the benefits of treadmill use for improving fitness and overall health for older customers. Product Features: The company could consider adding features that appeal to customers at different stages of life. For example, they could offer programs specifically designed for older customers or for younger customers who are just starting their fitness journey.
- Pricing Strategy: Aerofit could adjust their pricing strategy based on the age of their target customers. For example, they could offer discounts to older customers or students.

In [31]:
```
fig, ax = plt.subplots(1, 2, figsize=(24, 7))
sns.countplot(data=df, x='Gender', ax=ax[0], hue='Product')
sns.countplot(data=df, x='MaritalStatus', ax=ax[1], hue='Product')
ax[0].set_title('Gender by Product')
ax[1].set_title('Marital Status by Product')
plt.suptitle('Product Preferences by Customer Demographics')
plt.show()
```



Observations

Gender by Product:

- KP281: Has a relatively equal distribution among males and females.
- KP481: Has a slightly higher proportion of male customers.
- KP781: Has a significantly higher proportion of male customers.

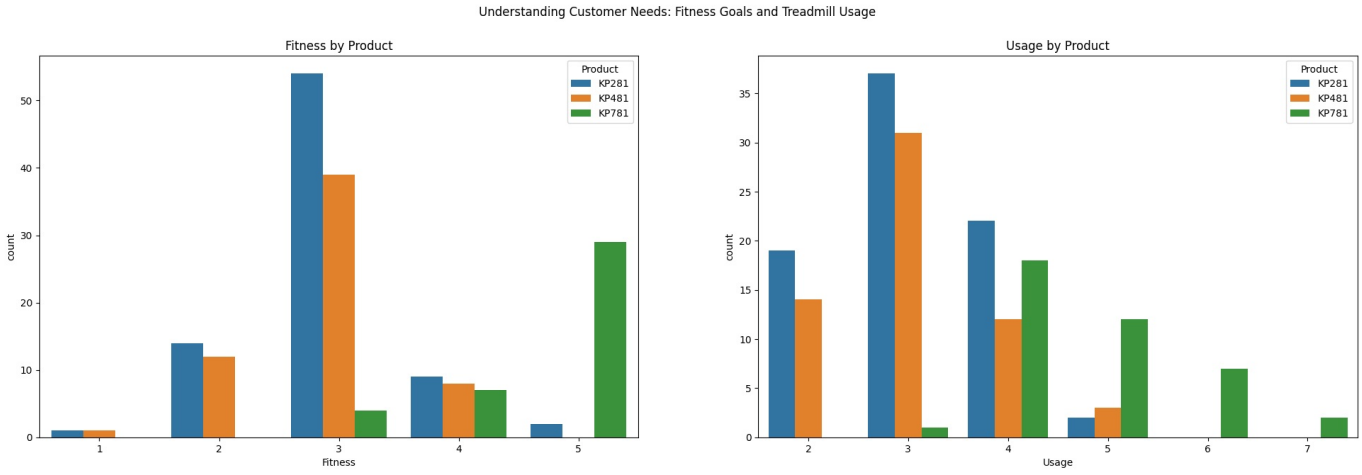
Marital Status by Product:

- KP281: Has a slightly higher proportion of single customers.
- KP481: Has a more balanced distribution between partnered and single customers.
- KP781: Has a slightly higher proportion of partnered customers.

Overall, the charts suggest that:

- Gender: There are some differences in gender preferences for the treadmill models, particularly for KP781.
- Marital Status: Marital status may also play a role in product choice, with KP281 being more popular among single customers and KP781 being more popular among partnered customers.

In [32]:
```
fig, ax = plt.subplots(1, 2, figsize=(24, 7))
sns.countplot(data=df, x='Fitness',hue='Product',ax=ax[0])
sns.countplot(data=df, x='Usage',hue='Product',ax=ax[1])
ax[0].set_title('Fitness by Product')
ax[1].set_title('Usage by Product')
plt.suptitle('Understanding Customer Needs: Fitness Goals and Treadmill Usage')
plt.show()
```



Observations

Fitness by Product:

- KP281: Has a relatively balanced distribution across fitness levels, with a slight concentration in the middle range (3-4).
- KP481: Has a higher proportion of customers with a fitness level of 3, followed by 4.
- KP781: Has a higher proportion of customers with a fitness level of 5, indicating it may be more popular among customers with higher fitness goals.

Usage by Product:

- KP281: Has a higher proportion of customers with a usage level of 3, followed by 4.
- KP481: Has a relatively balanced distribution across usage levels.
- KP781: Has a higher proportion of customers with a usage level of 2.

Overall, the charts suggest that:

- Product-Fitness Fit: KP781 may be more suitable for customers with higher fitness goals, while KP281 and KP481 may be more versatile for customers with varying fitness levels.
- Usage Patterns: Customers using KP281 and KP481 tend to use the treadmill more frequently than those using KP781.

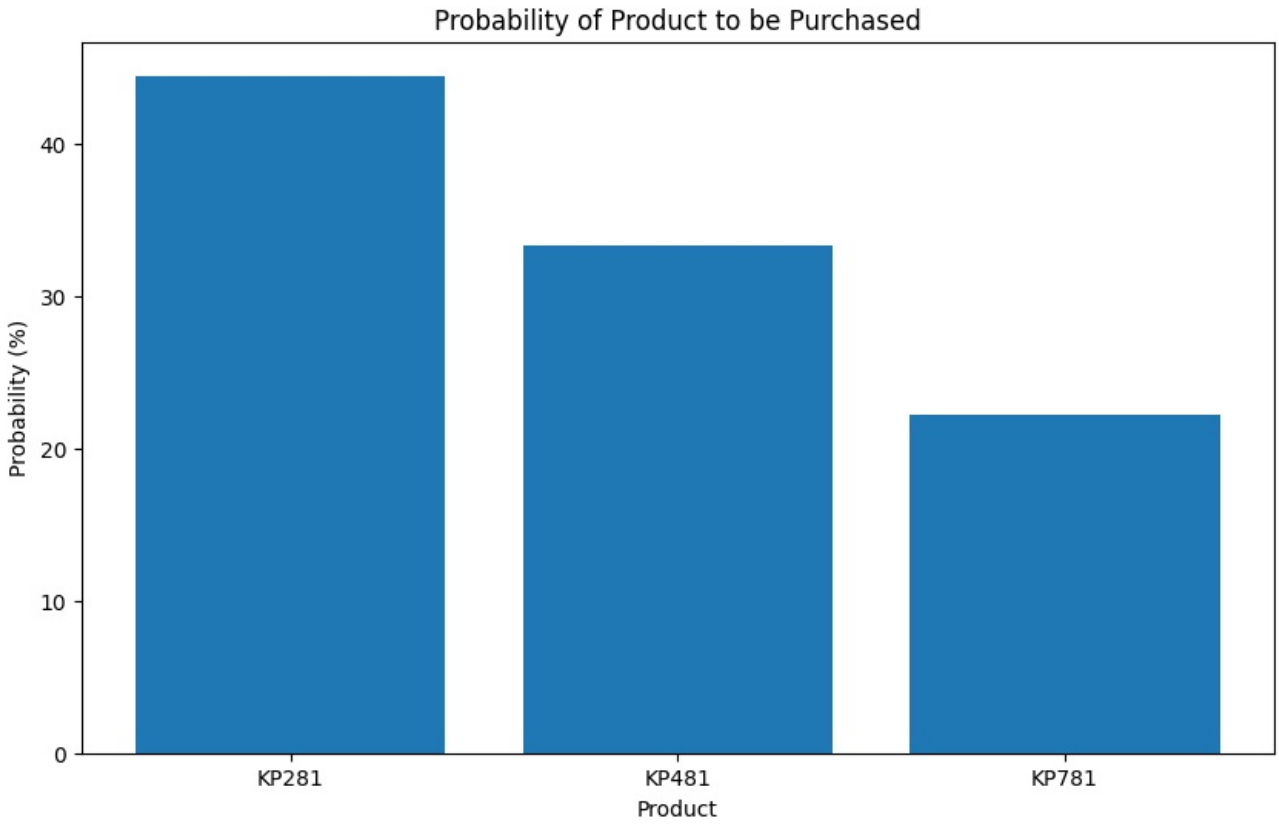
In [33]:
```
pd.crosstab(df['Gender'],
            df['Product'],
            margins=True,)
```

Out[33]:

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

In [34]:
```
crosstab=pd.crosstab(df['Gender'],
            df['Product'],
            margins=True,
            normalize='index')
x=crosstab.columns.tolist()
y=crosstab.loc['All']*100
y=y.tolist()
```

In [35]:
```
plt.figure(figsize=(10, 6))
plt.title('Probability of Product to be Purchased')
plt.bar(x, y)
plt.xlabel('Product')
plt.ylabel('Probability (%)')
plt.show()
```



Observation

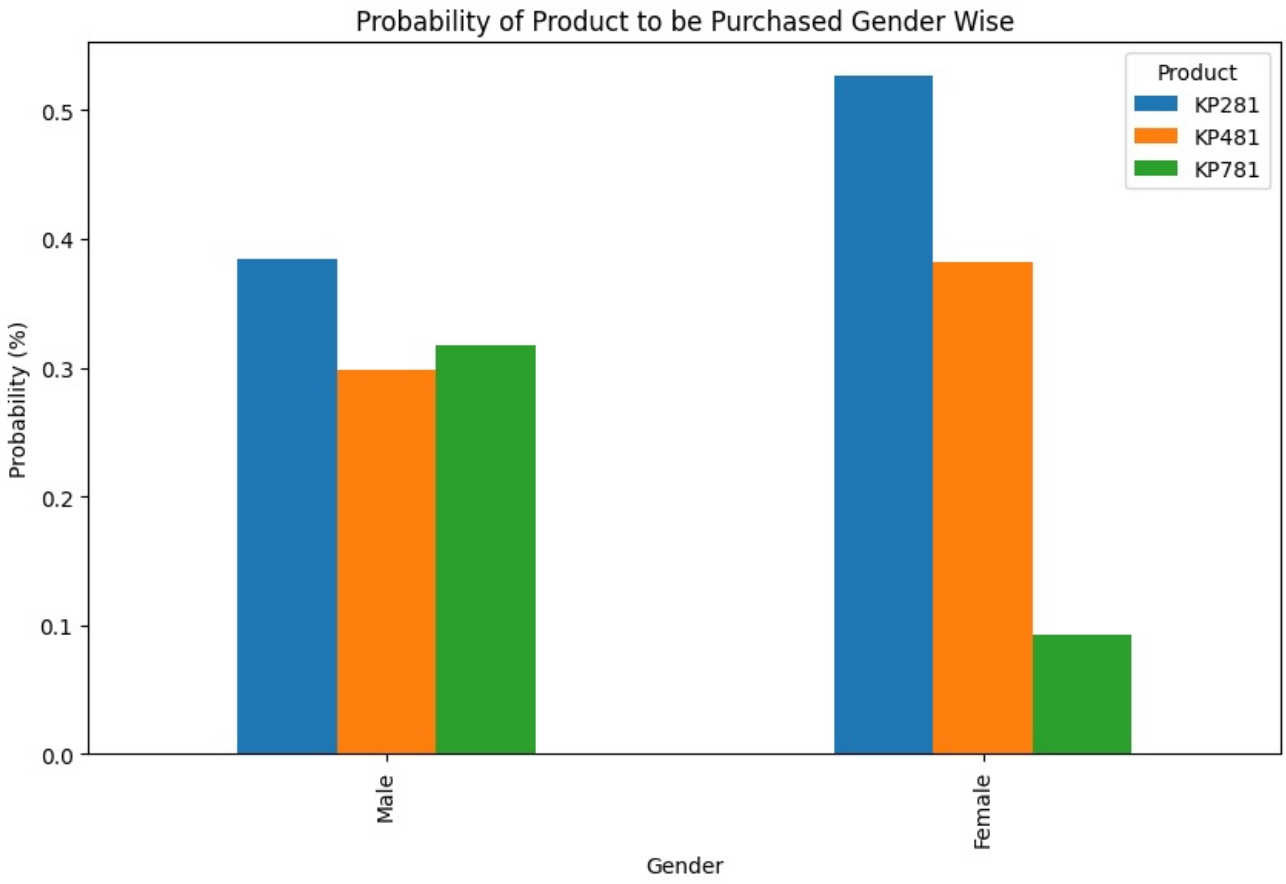
KP281 has the highest probability of being purchased by customers, followed by KP481 and then KP781.

KP781 has the lowest probability of being purchased.

All this could be due to several Factors:

- Price: KP281 may be the most affordable option, making it more attractive to budget-conscious customers.
- Durability: KP281 may be perceived as being more durable or reliable than the other models, which could influence customer purchasing decisions.
- Features: KP781 may have advanced features that are not as important to many customers, or it may be priced higher than KP281 and KP481.

In [36]:
```
crosstab.loc[['Male','Female']].plot(kind='bar', figsize=(10, 6))
plt.title('Probability of Product to be Purchased Gender Wise')
plt.xlabel('Gender')
plt.ylabel('Probability (%)')
plt.show()
```



Observations

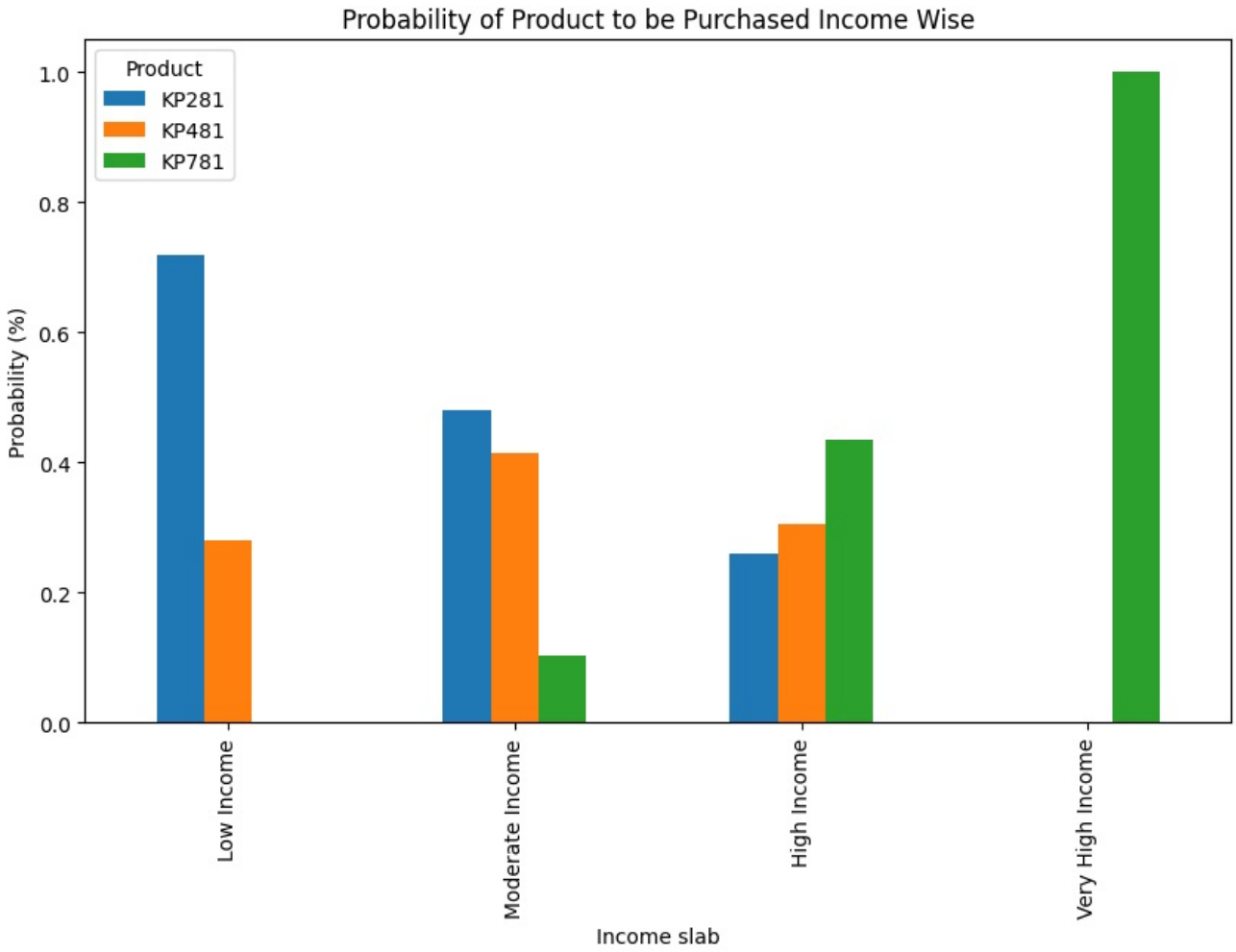
- KP281: Has a relatively equal probability of being purchased by both males and females.
- KP481: Has a slightly higher probability of being purchased by males than females.
- KP781: Has a significantly higher probability of being purchased by males compared to females.
- Gender: There are some differences in gender preferences for the treadmill
- models. KP781 is particularly popular among male customers.

In [37]:
```
crosstab=pd.crosstab(df['income_group'],
            df['Product'],
            margins=True,
            normalize='index')
x=crosstab.columns.tolist()
y=crosstab.loc['All']*100
y=y.tolist()
crosstab
```

Out[37]:

Product	KP281	KP481	KP781
income_group			
Low Income	0.718750	0.281250	0.000000
Moderate Income	0.481132	0.415094	0.103774
High Income	0.260870	0.304348	0.434783
Very High Income	0.000000	0.000000	1.000000
All	0.444444	0.333333	0.222222

In [38]:
```
crosstab.loc[['Low Income','Moderate Income','High Income','Very High Income']].plot(kind='bar', figsize=(10, 6
plt.title('Probability of Product to be Purchased Income Wise')
plt.xlabel('Income slab')
plt.ylabel('Probability (%)')
plt.show()
```



Observations:

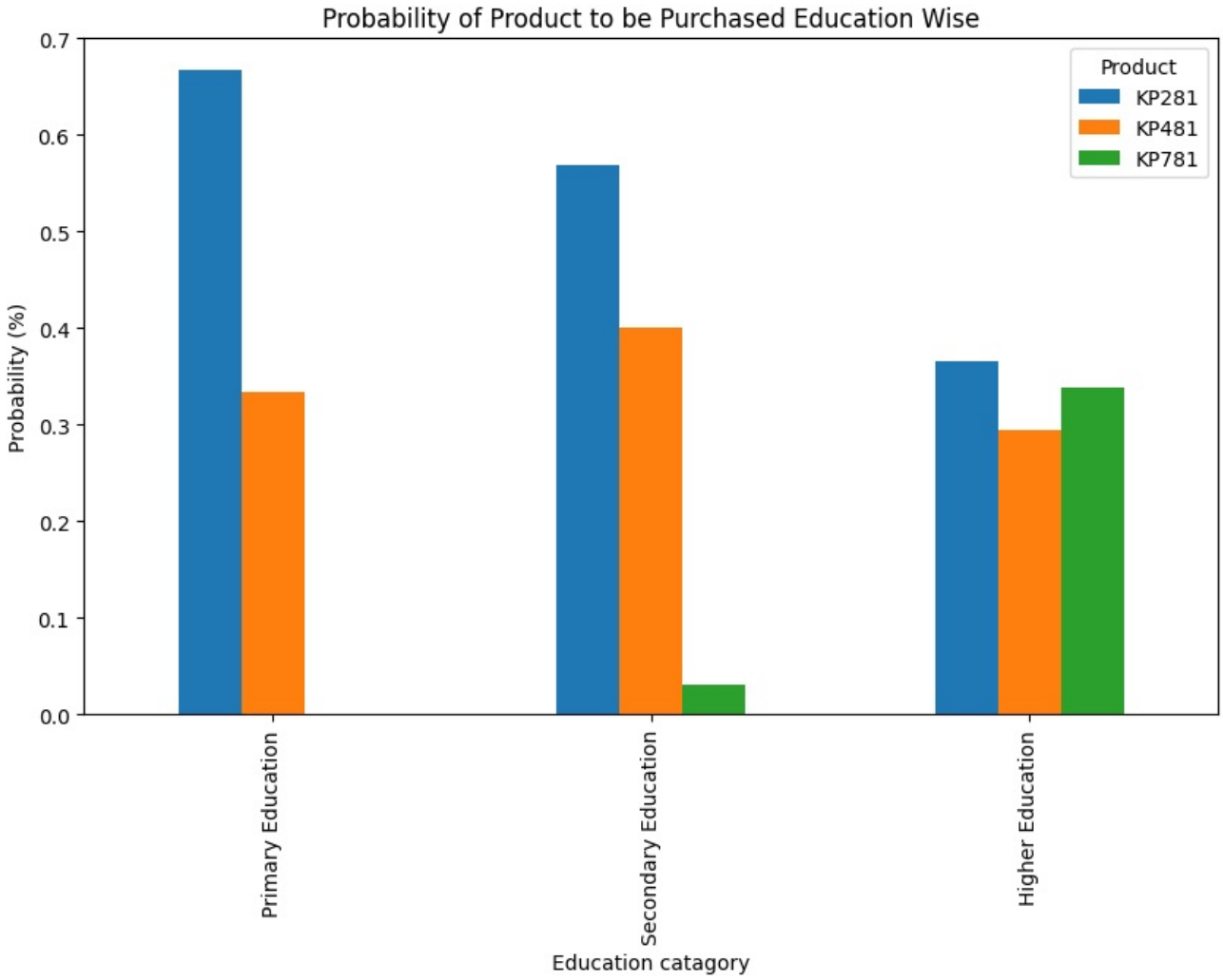
- KP281: Has a relatively high probability of being purchased across all income slabs, suggesting it may be a popular choice for customers with varying income levels.
- KP481: Has a higher probability of being purchased by customers in the moderate and high-income brackets, indicating it may be more appealing to customers with higher purchasing power.
- KP781: Has the highest probability of being purchased by customers in the very high-income bracket, suggesting it may be considered a premium product.

In [39]:
```
crosstab=pd.crosstab(df['edu_group'],
            df['Product'],
            margins=True,
            normalize='index')
x=crosstab.columns.tolist()
y=crosstab.loc['All']*100
y=y.tolist()
crosstab
```

Out[39]:

Product	KP281	KP481	KP781
edu_group			
Primary Education	0.666667	0.333333	0.000000
Secondary Education	0.569231	0.400000	0.030769
Higher Education	0.366071	0.294643	0.339286
All	0.444444	0.333333	0.222222

In [40]:
```
crosstab.loc[['Primary Education','Secondary Education','Higher Education']].plot(kind='bar', figsize=(10, 6))
plt.title('Probability of Product to be Purchased Education Wise')
plt.xlabel('Education catagory')
plt.ylabel('Probability (%)')
plt.show()
```



Observations:

- KP281: Has a relatively high probability of being purchased across all education levels, suggesting it may be a popular choice for customers with varying educational backgrounds.
- KP481: Has a higher probability of being purchased by customers with secondary or higher education, indicating it may be more appealing to customers with higher education levels.
- KP781: Has a very low probability of being purchased by customers with primary education, but a moderate probability for those with secondary or higher education.

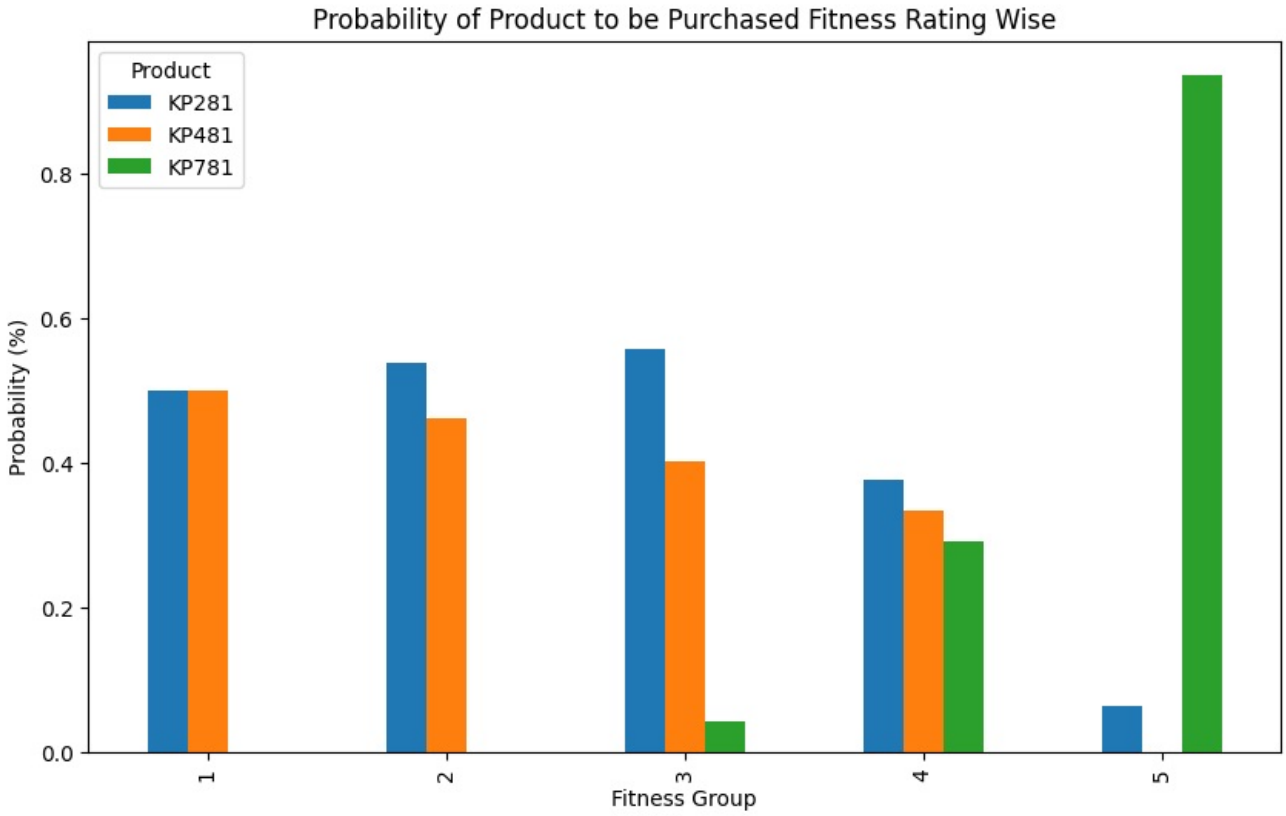
In [41]:
```
crosstab=pd.crosstab(df['Fitness'],
            df['Product'],
            margins=True,
            normalize='index')
x=crosstab.columns.tolist()
y=crosstab.loc['All']*100
y=y.tolist()
crosstab
```

Out[41]:

Product	KP281	KP481	KP781
Fitness			
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
All	0.444444	0.333333	0.222222

In [42]:

```
crosstab.iloc[0:5].plot(kind='bar', figsize=(10, 6))
plt.title('Probability of Product to be Purchased Fitness Rating Wise')
plt.xlabel('Fitness Group')
plt.ylabel('Probability (%)')
plt.show()
```

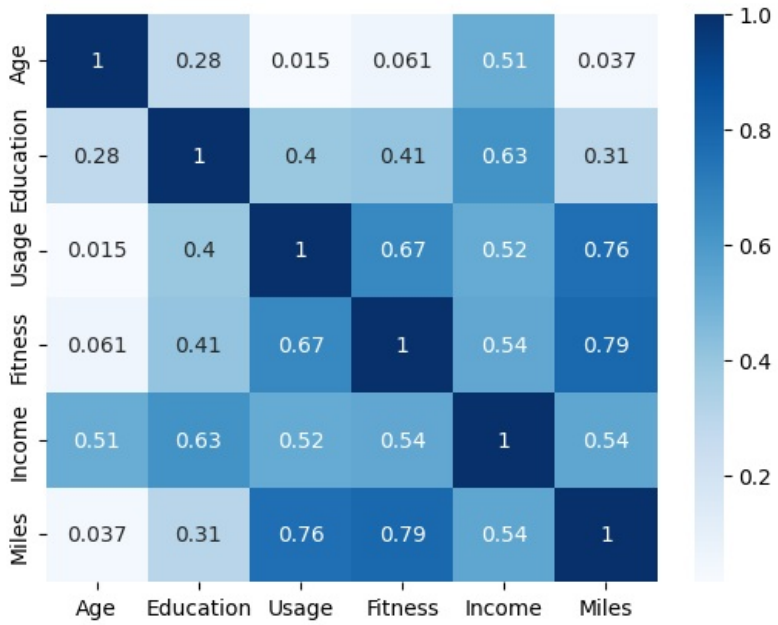


Observations:

- KP281: Has a relatively high probability of being purchased across all fitness ratings, suggesting it may be a popular choice for customers with varying fitness levels.
- KP481: Has a higher probability of being purchased by customers with a fitness rating of 3 or 4, indicating it may be more suitable for customers with moderate fitness levels.
- KP781: Has a significantly higher probability of being purchased by customers with a fitness rating of 5, suggesting it may be particularly attractive to customers with high fitness goals.

In [43]:

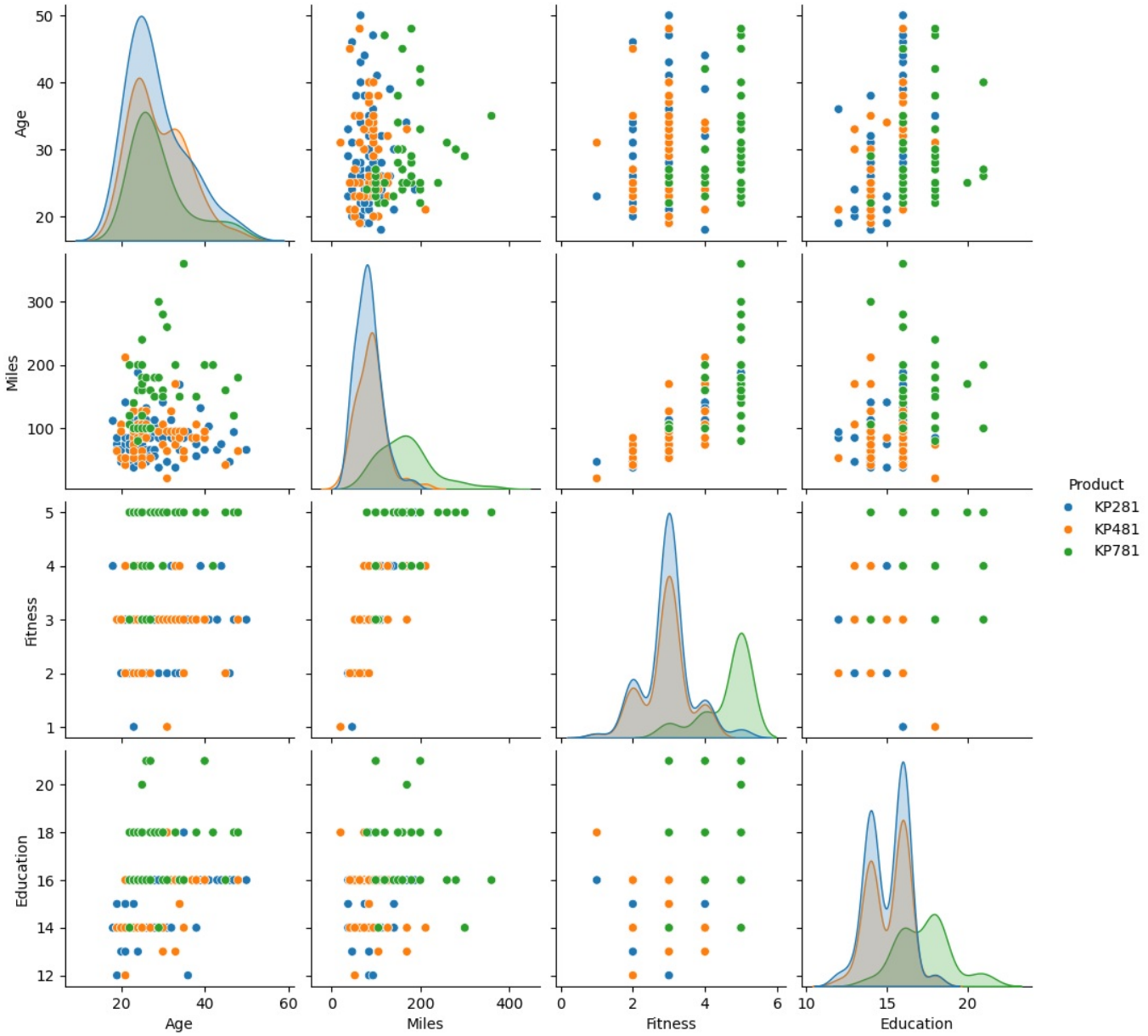
```
df_corr = df.select_dtypes(include=[np.number])
sns.heatmap(df_corr.corr(), cmap="Blues", annot=True)
plt.show()
```



Observations

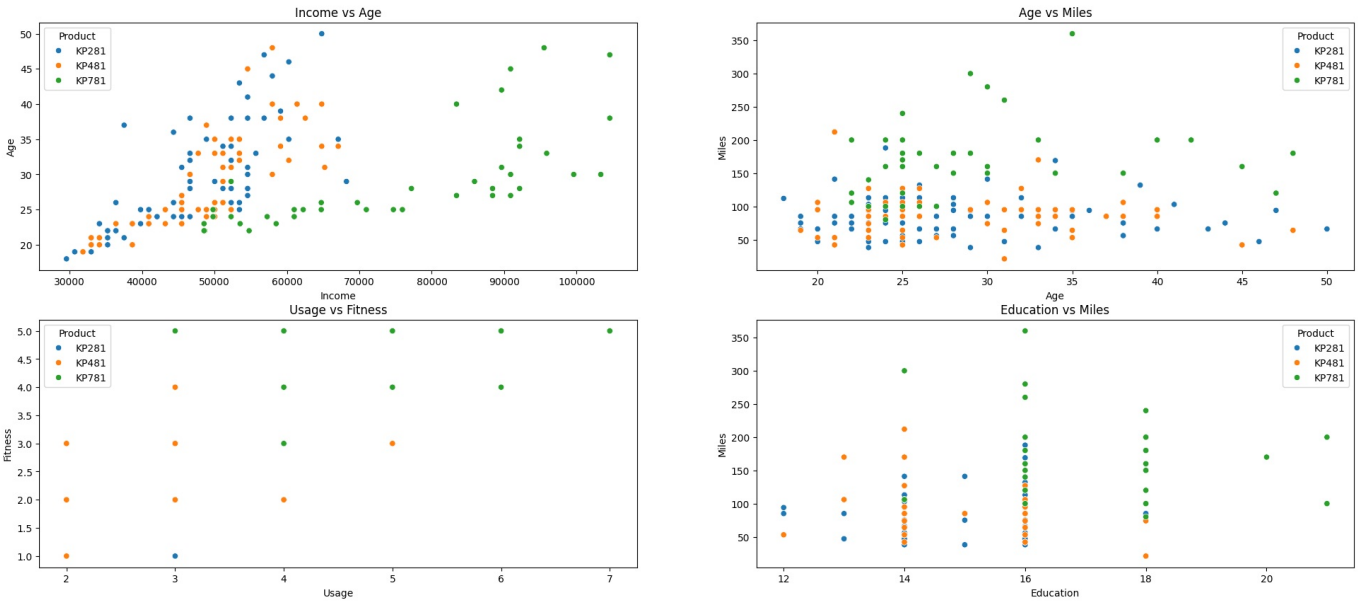
- Age and Education: There is a weak positive correlation between age and education (0.28), suggesting that older customers tend to have higher education levels.
- Usage and Fitness: There is a strong positive correlation between usage and fitness (0.67), indicating that customers who use the treadmill more frequently tend to have higher self-rated fitness levels.
- Income and Miles: There is a moderate positive correlation between income and miles (0.54), suggesting that customers with higher incomes tend to walk or run more miles per week.
- Usage and Miles: There is a strong positive correlation between usage and miles (0.79), indicating that customers who use the treadmill more frequently also tend to walk or run more miles.

In [44]:
```
sns.pairplot(data=df,hue='Product',vars=['Age', 'Miles', 'Fitness','Education'])
plt.show()
```



In [45]:

```
fig, ax = plt.subplots(2, 2, figsize=(24, 10))
sns.scatterplot(x= 'Income', y= 'Age', data = df,ax=ax[0,0],hue='Product')
ax[0,0].set_title('Income vs Age')
sns.scatterplot(x= 'Age', y= 'Miles', data = df,ax=ax[0,1],hue='Product')
ax[0,1].set_title('Age vs Miles')
sns.scatterplot(x= 'Usage', y= 'Fitness', data = df,ax=ax[1,0],hue='Product')
ax[1,0].set_title('Usage vs Fitness')
sns.scatterplot(x= 'Education', y= 'Miles', data = df,ax=ax[1,1],hue='Product')
ax[1,1].set_title('Education vs Miles')
plt.show()
```



Observation

Income vs. Age:

There is a weak positive correlation between income and age, suggesting that older customers tend to have higher incomes. However, there is also a significant amount of variation around this trend, indicating that income is not solely determined by age.

Age vs. Miles:

There is a weak positive correlation between age and miles walked, suggesting that older customers tend to walk or run more miles. However, this relationship is not very strong, and there is a significant amount of variation around the trend.

Usage vs. Fitness:

There is a moderate positive correlation between usage and fitness, suggesting that customers who use the treadmill more frequently tend to have higher self-rated fitness levels. However, there are also some outliers, indicating that there may be other factors influencing fitness levels.

Education vs. Miles:

There is a weak positive correlation between education and miles walked, suggesting that customers with higher education levels tend to walk or run more miles. However, this relationship is not very strong, and there is a significant amount of variation around the trend.

Understanding Distribution and Identifying Outliers

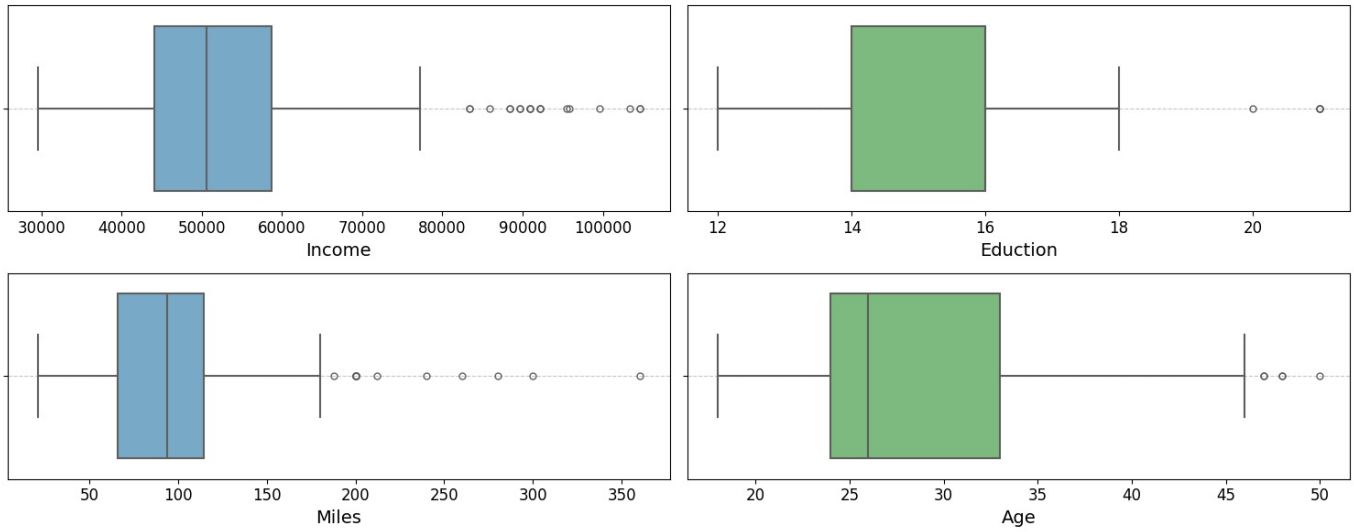
Box plot-Visiual Analysis Before Outlier treament

In [46]:
```
fig, ax = plt.subplots(2, 2, figsize=(15, 6))
sns.boxplot(data=df, x='Income', ax=ax[0,0], palette='Blues', fliersize=5, linewidth=1.5)
ax[0,0].set_xlabel('Income', fontsize=14)
ax[0,0].tick_params(axis='x', labelsize=12)
ax[0,0].tick_params(axis='y', labelsize=12)
ax[0,0].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=df, x='Education', ax=ax[0,1], palette='Greens', fliersize=5, linewidth=1.5)
ax[0,1].set_xlabel('Eduction', fontsize=14)
ax[0,1].tick_params(axis='x', labelsize=12)
ax[0,1].tick_params(axis='y', labelsize=12)
ax[0,1].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=df, x='Miles', ax=ax[1,0], palette='Blues', fliersize=5, linewidth=1.5)
ax[1,0].set_xlabel('Miles', fontsize=14)
ax[1,0].tick_params(axis='x', labelsize=12)
ax[1,0].tick_params(axis='y', labelsize=12)
ax[1,0].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=df, x='Age', ax=ax[1,1], palette='Greens', fliersize=5, linewidth=1.5)
ax[1,1].set_xlabel('Age', fontsize=14)
ax[1,1].tick_params(axis='x', labelsize=12)
ax[1,1].tick_params(axis='y', labelsize=12)
ax[1,1].grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



Observation

Income:

- The distribution of income is skewed to the right, indicating that there are a smaller number of customers with higher incomes compared to lower incomes.
- There are several outliers on the right side of the box plot, suggesting that a few customers have significantly higher incomes than the majority of the sample.

Education:

- The distribution of education is relatively symmetrical, with a slight skew to the right.
- There are a few outliers on the right side of the box plot, indicating that a small number of customers have higher education levels than the majority of the sample.

Miles:

- The distribution of miles is skewed to the right, indicating that there are a smaller number of customers who walk or run significantly more miles than the average.
- There are several outliers on the right side of the box plot, suggesting that a few customers are significantly more active than the majority of the sample.

Age

- The distribution of age is skewed to the right, indicating that there are a smaller number of older customers compared to younger customers.

In [47]:
```
Q1 = df['Income'].quantile(0.25)
Q3 = df['Income'].quantile(0.75)
IQR = Q3 - Q1
Income_lower_bound = Q1 - 1.5 * IQR
Income_upper_bound = Q3 + 1.5 * IQR
print(f"Income Upper Bound: {Income_upper_bound}, Income Lower Bound: {Income_lower_bound}")
is_outlier_upper = Income_upper_bound < df['Income'].max()
is_outlier_lower = Income_lower_bound > df['Income'].min()
print(f"Is the value exceeding the upper bound limit? {is_outlier_upper}")
print(f"Is the value falling below the lower bound limit? {is_outlier_lower}.")
print('---'*30)
Q1 = df['Education'].quantile(0.25)
Q3 = df['Education'].quantile(0.75)
IQR = Q3 - Q1
Education_lower_bound = Q1 - 1.5 * IQR
Education_upper_bound = Q3 + 1.5 * IQR
print(f"Education Upper Bound: {Education_upper_bound}, Education Lower Bound: {Education_lower_bound}")
is_outlier_upper = Education_upper_bound < df['Education'].max()
is_outlier_lower = Education_lower_bound > df['Education'].min()
print(f"Is the value exceeding the upper bound limit? {is_outlier_upper}")
print(f"Is the value falling below the lower bound limit? {is_outlier_lower}.")
print('---'*30)
Q1 = df['Miles'].quantile(0.25)
Q3 = df['Miles'].quantile(0.75)
IQR = Q3 - Q1
Miles_lower_bound = Q1 - 1.5 * IQR
Miles_upper_bound = Q3 + 1.5 * IQR
print(f"Miles Upper Bound: {Miles_upper_bound}, Miles Lower Bound: {Miles_lower_bound}")
is_outlier_upper = Miles_upper_bound < df['Miles'].max()
is_outlier_lower = Miles_lower_bound > df['Miles'].min()
print(f"Is the value exceeding the upper bound limit? {is_outlier_upper}")
print(f"Is the value falling below the lower bound limit? {is_outlier_lower}.")
print('---'*30)
Q1 = df['Age'].quantile(0.25)
Q3 = df['Age'].quantile(0.75)
IQR = Q3 - Q1
Age_lower_bound = Q1 - 1.5 * IQR
Age_upper_bound = Q3 + 1.5 * IQR
print(f"Age Upper Bound: {Age_upper_bound}, Age Lower Bound: {Age_lower_bound}")
is_outlier_upper = Age_upper_bound < df['Age'].max()
is_outlier_lower = Age_lower_bound > df['Age'].min()
print(f"Is the value exceeding the upper bound limit? {is_outlier_upper}")
print(f"Is the value falling below the lower bound limit? {is_outlier_lower}.")
```

```
Income Upper Bound: 80581.875, Income Lower Bound: 22144.875
Is the value exceeding the upper bound limit? True
Is the value falling below the lower bound limit? False.
--------------------------------------------------------------------------------
Education Upper Bound: 19.0, Education Lower Bound: 11.0
Is the value exceeding the upper bound limit? True
Is the value falling below the lower bound limit? False.
--------------------------------------------------------------------------------
Miles Upper Bound: 187.875, Miles Lower Bound: -7.125
Is the value exceeding the upper bound limit? True
Is the value falling below the lower bound limit? False.
--------------------------------------------------------------------------------
Age Upper Bound: 46.5, Age Lower Bound: 10.5
Is the value exceeding the upper bound limit? True
Is the value falling below the lower bound limit? False.
```

Observations:

Income:

- Upper Bound Exceeded: The maximum income value (99,581.875) exceeds the upper bound of 80,581.875, indicating that there is at least one outlier in the income data.
- Lower Bound Not Exceeded: The minimum income value is above the lower bound, indicating that there are no outliers below the lower bound.

Education:

- Upper Bound Exceeded: The maximum education value (21) exceeds the upper bound of 19.0, indicating that there is at least one outlier in the education data.
- Lower Bound Not Exceeded: The minimum education value is above the lower bound, indicating that there are no outliers below the lower bound.

Miles:

- Upper Bound Exceeded: The maximum miles value (360) exceeds the upper bound of 187.875, indicating that there is at least one outlier in the miles data.
- Lower Bound Not Exceeded: The minimum miles value is above the lower bound, indicating that there are no outliers below the lower bound.

Age:

- Upper Bound Exceeded: The maximum age value (50) exceeds the upper bound of 46.5, indicating that there is at least one outlier in the age data.
- Lower Bound Not Exceeded: The minimum age value is above the lower bound, indicating that there are no outliers below the lower bound.

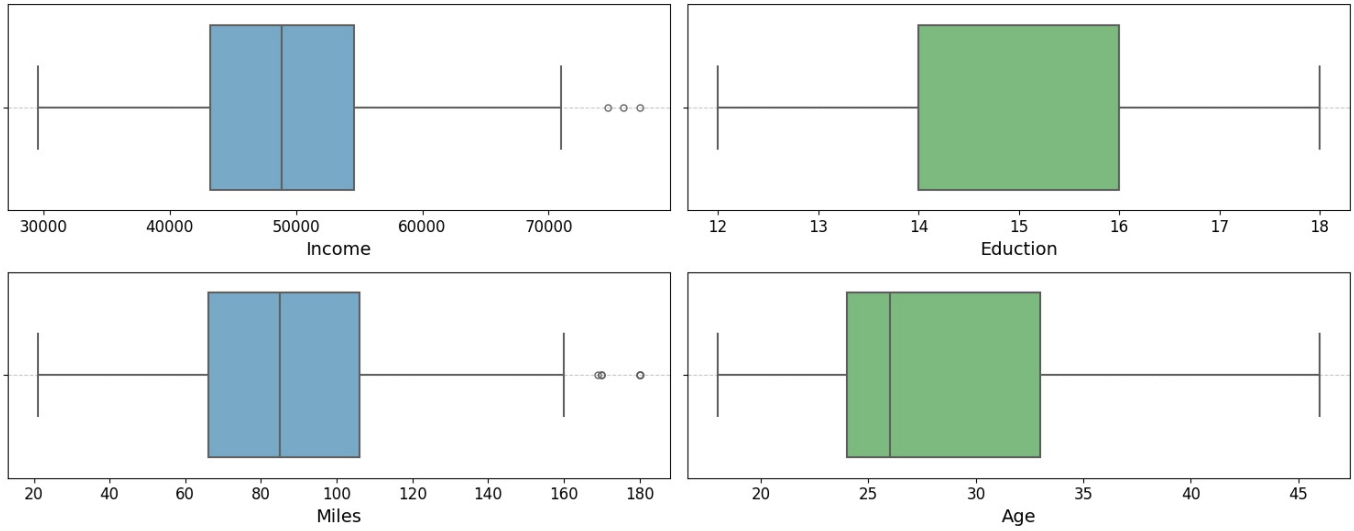
In [48]:
```
clean_Income_data = df[(df['Income'] >= Income_lower_bound) & (df['Income'] <= Income_upper_bound)]
clean_Education_data = df[(df['Education'] >= Education_lower_bound) & (df['Education'] <= Education_upper_boun
clean_Miles_data = df[(df['Miles'] >= Miles_lower_bound) & (df['Miles'] <= Miles_upper_bound)]
clean_Age_shows_data = df[(df['Age'] >= Age_lower_bound) & (df['Age'] <= Age_upper_bound)]
```

In [49]:
```
fig, ax = plt.subplots(2, 2, figsize=(15, 6))
sns.boxplot(data=clean_Income_data, x='Income', ax=ax[0,0], palette='Blues', fliersize=5, linewidth=1.5)
ax[0,0].set_xlabel('Income', fontsize=14)
ax[0,0].tick_params(axis='x', labelsize=12)
ax[0,0].tick_params(axis='y', labelsize=12)
ax[0,0].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=clean_Education_data, x='Education', ax=ax[0,1], palette='Greens', fliersize=5, linewidth=1.5)
ax[0,1].set_xlabel('Eduction', fontsize=14)
ax[0,1].tick_params(axis='x', labelsize=12)
ax[0,1].tick_params(axis='y', labelsize=12)
ax[0,1].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=clean_Miles_data, x='Miles', ax=ax[1,0], palette='Blues', fliersize=5, linewidth=1.5)
ax[1,0].set_xlabel('Miles', fontsize=14)
ax[1,0].tick_params(axis='x', labelsize=12)
ax[1,0].tick_params(axis='y', labelsize=12)
ax[1,0].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=clean_Age_shows_data, x='Age', ax=ax[1,1], palette='Greens', fliersize=5, linewidth=1.5)
ax[1,1].set_xlabel('Age', fontsize=14)
ax[1,1].tick_params(axis='x', labelsize=12)
ax[1,1].tick_params(axis='y', labelsize=12)
ax[1,1].grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

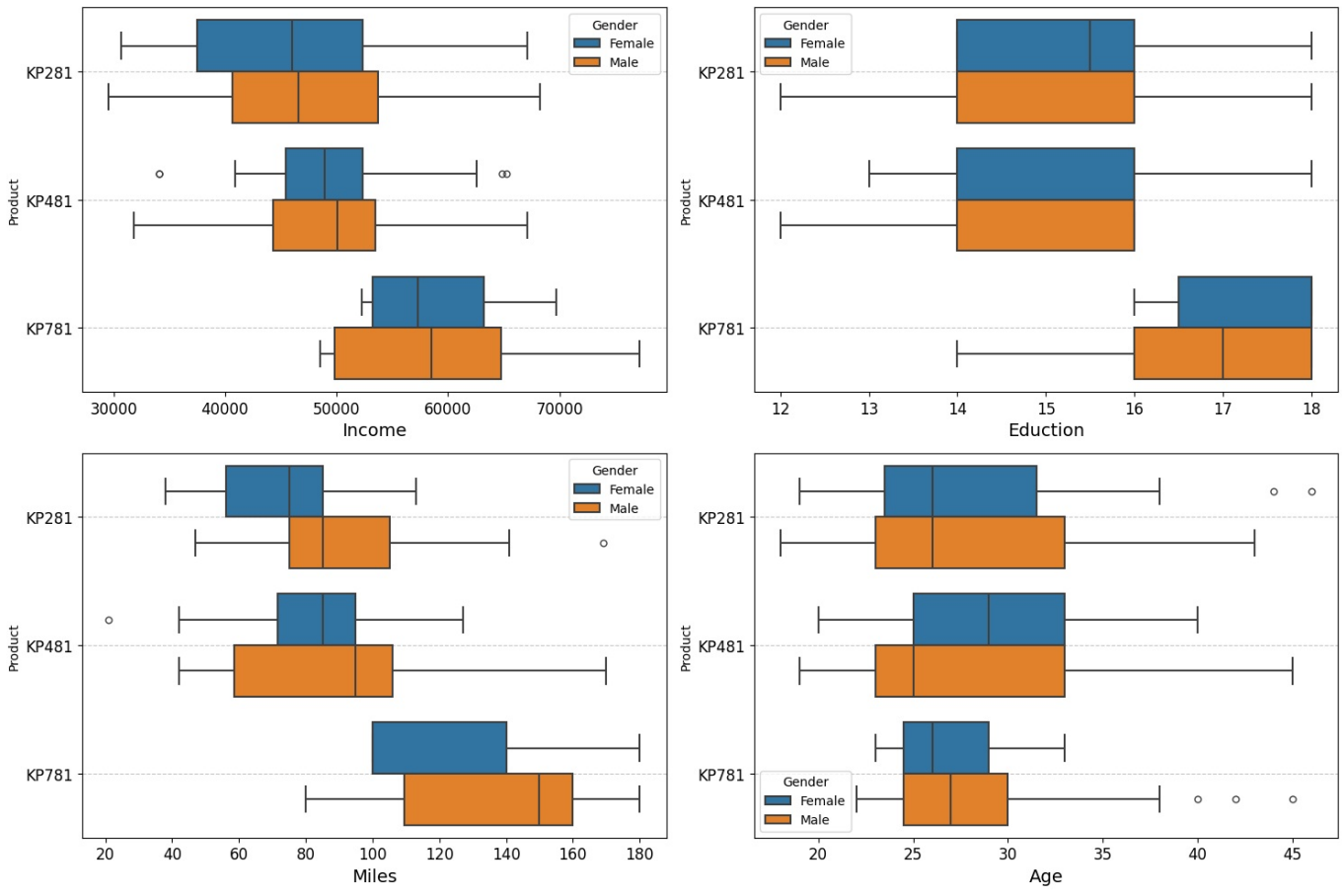


In [50]:
```
fig, ax = plt.subplots(2, 2, figsize=(15, 10))
sns.boxplot(data=clean_Income_data, x='Income', ax=ax[0,0], fliersize=5, linewidth=1.5,y='Product',hue='Gender'
ax[0,0].set_xlabel('Income', fontsize=14)
ax[0,0].tick_params(axis='x', labelsize=12)
ax[0,0].tick_params(axis='y', labelsize=12)
ax[0,0].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=clean_Education_data, x='Education', ax=ax[0,1], fliersize=5, linewidth=1.5,y='Product',hue='G
ax[0,1].set_xlabel('Eduction', fontsize=14)
ax[0,1].tick_params(axis='x', labelsize=12)
ax[0,1].tick_params(axis='y', labelsize=12)
ax[0,1].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=clean_Miles_data, x='Miles', ax=ax[1,0], fliersize=5, linewidth=1.5,y='Product',hue='Gender')
ax[1,0].set_xlabel('Miles', fontsize=14)
ax[1,0].tick_params(axis='x', labelsize=12)
ax[1,0].tick_params(axis='y', labelsize=12)
ax[1,0].grid(axis='y', linestyle='--', alpha=0.7)

sns.boxplot(data=clean_Age_shows_data, x='Age', ax=ax[1,1], fliersize=5, linewidth=1.5,y='Product',hue='Gender'
ax[1,1].set_xlabel('Age', fontsize=14)
ax[1,1].tick_params(axis='x', labelsize=12)
ax[1,1].tick_params(axis='y', labelsize=12)
ax[1,1].grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```



Observations:

Income by Product:

- KP281: Customers who purchased KP281 have a slightly lower median income compared to those who purchased KP481 and KP781.
- KP481: Customers who purchased KP481 have a higher median income compared to those who purchased KP281, but lower than those who purchased KP781.
- KP781: Customers who purchased KP781 have the highest median income, suggesting it may be a more premium product.

Education by Product:

- KP281: Customers who purchased KP281 have a slightly lower median education level compared to those who purchased KP481 and KP781.
- KP481: Customers who purchased KP481 have a slightly higher median education level compared to those who purchased KP281, but lower than those who purchased KP781.
- KP781: Customers who purchased KP781 have the highest median education level, suggesting it may be more popular among customers with higher education.

Miles by Product:

- KP281: Female customers tend to walk or run slightly more miles compared to male customers.
- KP481: There is no significant difference in miles walked between male and female customers.
- KP781: Male customers tend to walk or run slightly more miles compared to female customers.

Age by Product:

- KP281: There is no significant difference in age between male and female customers.
- KP481: Female customers tend to be slightly younger than male customers.
- KP781: Male customers tend to be slightly younger than female customers.

Overall, the box plots suggest that:

- Income: Customers who purchase KP781 tend to have higher incomes compared to those who purchase KP281 or KP481.
- Education: Customers who purchase KP781 tend to have higher education levels compared to those who purchase KP281 or KP481.
- Miles: There are some gender-specific differences in treadmill usage. Female customers using KP281 tend to walk or run more miles compared to male customers, while the opposite is true for KP781.
- Age: There are some subtle differences in age between male and female customers for certain products. However, these differences are not as pronounced as the differences in miles walked.

Customer Profiling

Demographic Segmentation:

- Age: Majority are young adults (25-30 years), with a smaller middle-aged and elderly segment. This suggests a focus on digital, youthful marketing campaigns .
- Education: The highly educated customer base means that they likely respond well to data-driven fitness plans and detailed product specifications .
- Income: Focus on middle-income customers with products like KP281, while premium options like KP781 should be marketed to individuals with higher disposable incomes .

Business Insights

Product Purchase Trends:

- KP281 is the most popular product, likely due to its affordability and durability, while KP781 is less favored, potentially due to its higher price and advanced features .

Customer Demographics:

- Majority of customers are males (57.8%) and partnered individuals (59.4%) .
- The largest age group consists of young adults between 25 and 30 years old .
- Customers generally have 16 years of education, indicating that they are well-educated .

Usage and Fitness Level:

- Customers plan to use treadmills 3-4 times per week, indicating moderate usage, with an average fitness level of 3 (average fitness) .

- KP281 is likely favored by budget-conscious or younger customers, while KP781 could be more appealing to advanced athletes .

1 Income Level:

- The average income of customers is approximately $54,000, which implies that the Aerofit brand attracts middle-income individuals .

Business Recommendations

Product Marketing:

- Focus marketing on young adults and middle-income individuals. Highlight KP281 for budget-conscious customers, and market KP781 with features appealing to advanced athletes or tech-savvy users .
- Consider bundling KP781 with additional services like virtual training or fitness programs to justify its higher price.

Custom Product Offers:

- Offer targeted product recommendations based on demographic profiles. For instance, younger customers might prefer simpler, cost-effective models, while older customers may value features that support joint protection or injury prevention .

Discount Strategies:

- Introduce pricing tiers or discounts for students and older customers to capture diverse market segments .
- Loyalty programs and periodic discounts for high-frequency users (based on usage patterns) could drive customer retention .

Targeted Campaigns:

- Develop gender-specific campaigns as a majority of customers are male, but also address the needs of female customers who form 42.2% of the customer base .
- Create marketing efforts directed toward partnered individuals, focusing on fitness as a shared experience .

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js