In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

We fetched the csv file.

In [2]:
```
!wget https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?163
```

```
--2024-03-21 15:37:07--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofi
t_treadmill.csv?1639992749
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 13.224.9.129, 13.224.9.181, 13.224.9
.103, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|13.224.9.129|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7279 (7.1K) [text/plain]
Saving to: 'aerofit_treadmill.csv?1639992749'

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

2024-03-21 15:37:07 (1.30 GB/s) - 'aerofit_treadmill.csv?1639992749' saved [7279/7279]
```

In [6]:
```python
data = pd.read_csv('aerofit_treadmill.csv?1639992749')
data
```

Out[6]:

	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
...	...	...	...	...	...	...	...	...	...
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

180 rows × 9 columns

Exploring the Dataset

In [10]:
```python
data.head()
```

Out[10]:

	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

In [14]:
```python
print ("Number of rows and cols: ")
data.shape
```

```
Number of rows and cols:
```

Out[14]:
```
(180, 9)
```

In [16]:
```python
print("First 4 rows of the dataset: ")

data.head()
```

```
First few rows of the dataset :
```

Out[16]:

	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

In [19]:

```
print("List of Parameters available: ")
data.info()
```

```
List of Parameters available:
<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 [22]:

```
print("Description of all the parameters: ")
data.describe(include = "all")
```

```
Description of all the parameters:
```

Out[22]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

Analysis

- Treadmills are employed for exercise purposes at least twice in a week by their owners.
- The minimum weekly mileage covered on a treadmill is 21 miles.
- Typically, individuals utilize treadmills approximately 3.45 times per week.
- On average, a treadmill accumulates a distance of 103 miles in a week's usage.
- Individuals who own treadmills exhibit a fitness score of at least 1, according to the measurement criteria.
- The average fitness level or score reported by treadmill owners is 3.3 on a given scale.

Visual Analysis

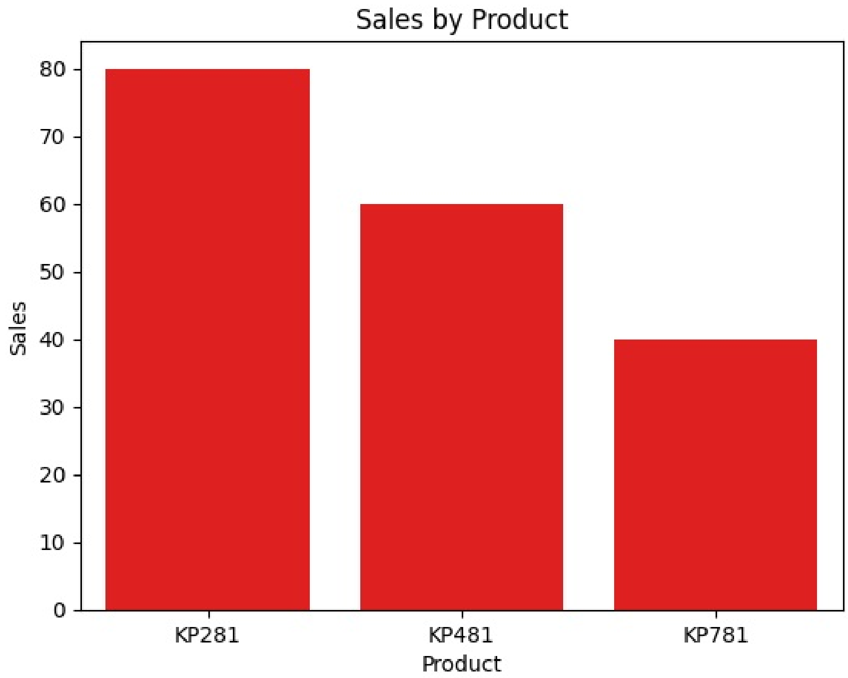
Analizing the sale of products categorizing:

In [24]:

```
# Create a bar plot to visualize sales count for each product
product_sales_visualization = sns.countplot(x=data["Product"], color='red')

# Set a descriptive title for the plot
product_sales_visualization.set_title('Sales by Product')

# Label the x-axis of the plot with 'Product'
product_sales_visualization.xaxis.set_label_text('Product')

# Label the y-axis of the plot with 'Sales'
```

```
product_sales_visualization.yaxis.set_label_text('Sales')
```

Out[24]: Text(0, 0.5, 'Sales')



In [25]:
```
# Create a contingency table (crosstab) with 'Product' as rows and 'count' as columns
product_count_table = pd.crosstab(index=data['Product'], columns='count')

# Calculate the marginal probabilities by dividing the counts by the total sum
marginal_probabilities = product_count_table / product_count_table.sum()

# Rename the column of the marginal probabilities DataFrame to 'Probability'
marginal_probabilities.columns = ['Probability']

# Display the marginal probabilities DataFrame
print(marginal_probabilities)
```

```
         Probability
Product
KP281       0.444444
KP481       0.333333
KP781       0.222222
```

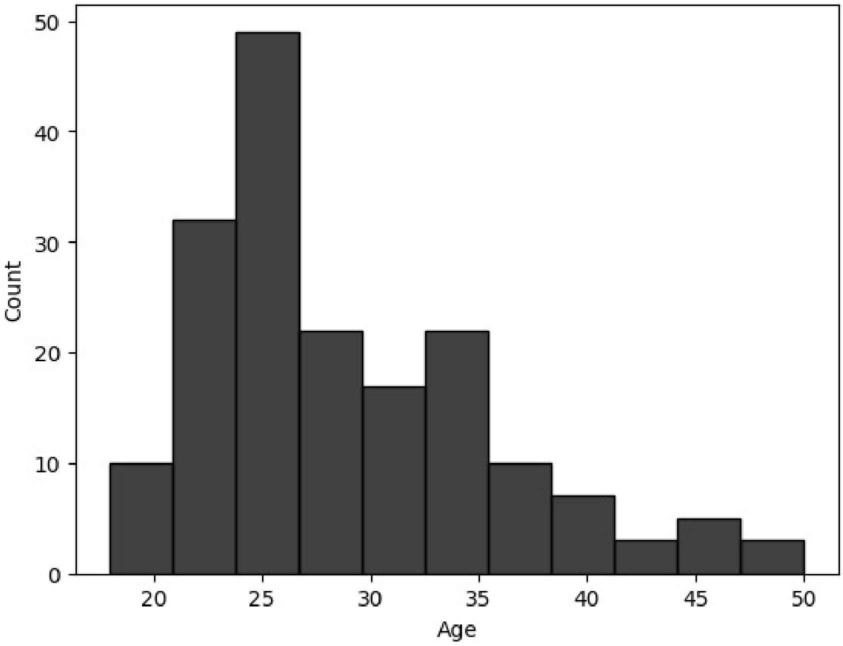
Inference:

- We can figure out, KP281(Base Model) is the most sold product.
- Followed by KP481(Mid-Level Model) and then the least sold product is KP781(Top Model).

Age Wise Data Inferening

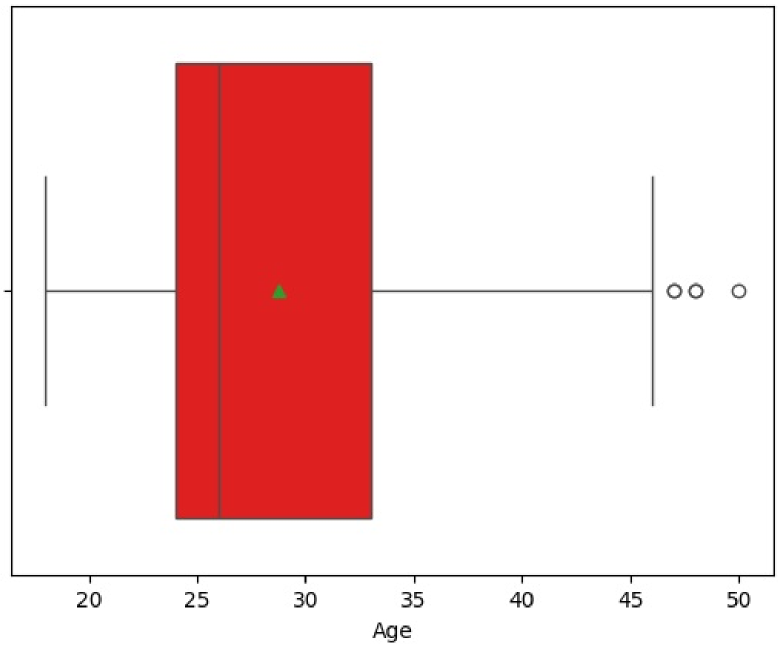
In [31]:
```
sns.histplot(data=data['Age'], kde=False, color='black')
```

Out[31]: <Axes: xlabel='Age', ylabel='Count'>



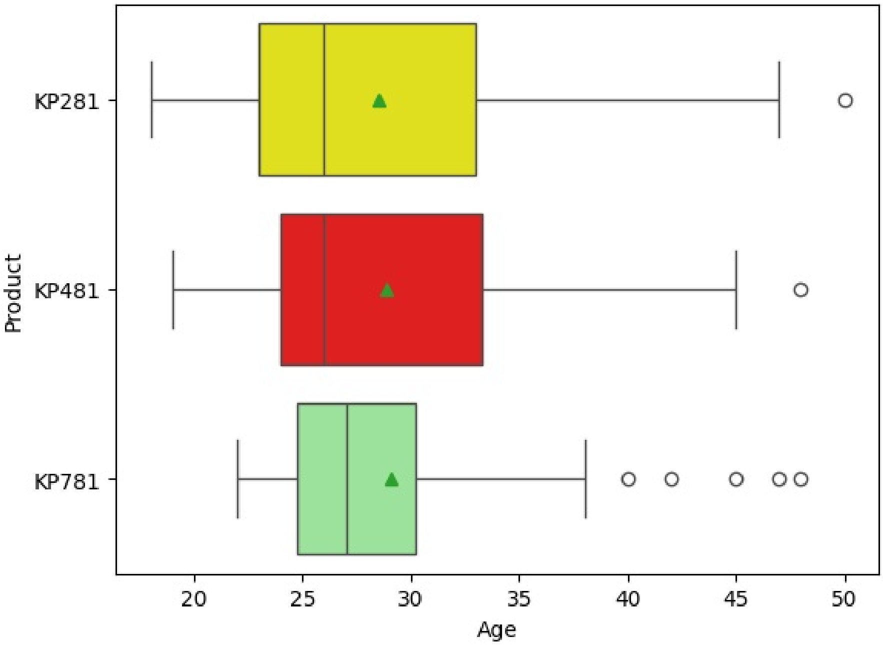
In [29]:
```
sns.boxplot(x=data["Age"], showmeans=True, color='red')
```

Out[29]: <Axes: xlabel='Age'>



In [33]:
```
sns.boxplot(x=data["Age"], y=data["Product"],showmeans=True, hue=data["Product"], palette=['yellow', 'red', 'li
```

Out[33]: <Axes: xlabel='Age', ylabel='Product'>



In [34]:
```
data["Age"].describe()
```

Out[34]:
```
count    180.000000
mean      28.788889
std        6.943498
min       18.000000
25%       24.000000
50%       26.000000
75%       33.000000
max       50.000000
Name: Age, dtype: float64
```

In [44]:
```
crosstab_age = pd.crosstab(index=data['Product'], columns='MaritalStatus')

marginal_prob_age_group = crosstab_age / crosstab_age.sum()
marginal_prob_age_group.columns = ['Probability']

print("Marginal probabilities for Product Models:")
marginal_prob_age_group
```

```
Marginal probabilities for age groups:
```

Out[44]:

	Probability
Product	
KP281	0.444444
KP481	0.333333
KP781	0.222222

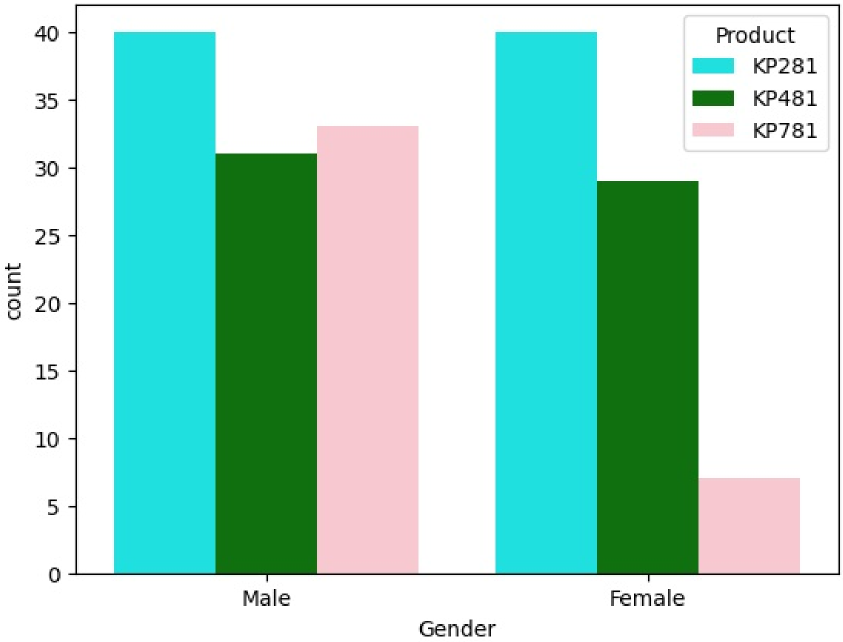
Analysis

Product KP281 is sold the highest, followed by KP481 and KP781.

Analysis based on the gender:

In [50]:
```
sns.countplot(x=data["Gender"], hue=data["Product"], palette=['cyan', 'green', 'pink'])
```

Out[50]:
```
<Axes: xlabel='Gender', ylabel='count'>
```



In [54]:
```
crosstab_gender = pd.crosstab(index=data['Gender'], columns='count')
crosstab_age = data['Age']
marginal_prob_gender = crosstab_gender / crosstab_age.sum()
marginal_prob_gender.columns = ['Probability']

print("Marginal probabilities based on gender:")
marginal_prob_gender
```

```
Marginal probabilities based on gender:
```

Out[54]:

	Probability
Gender	
Female	0.014666
Male	0.020069

Analysis:

- Female customers purchased significantly fewer units of the premium KP781 model compared to males.
- Male customers bought more premium KP781 units than the mid-range model.
- For the base KP281 model, sales were highest and nearly equal across both genders.
- Product positioning and marketing may need to be adjusted to better appeal to gender preferences, especially for premium and entry-level offerings.
- The visualization identified potential areas for targeted strategies to address observed gender-specific sales patterns.

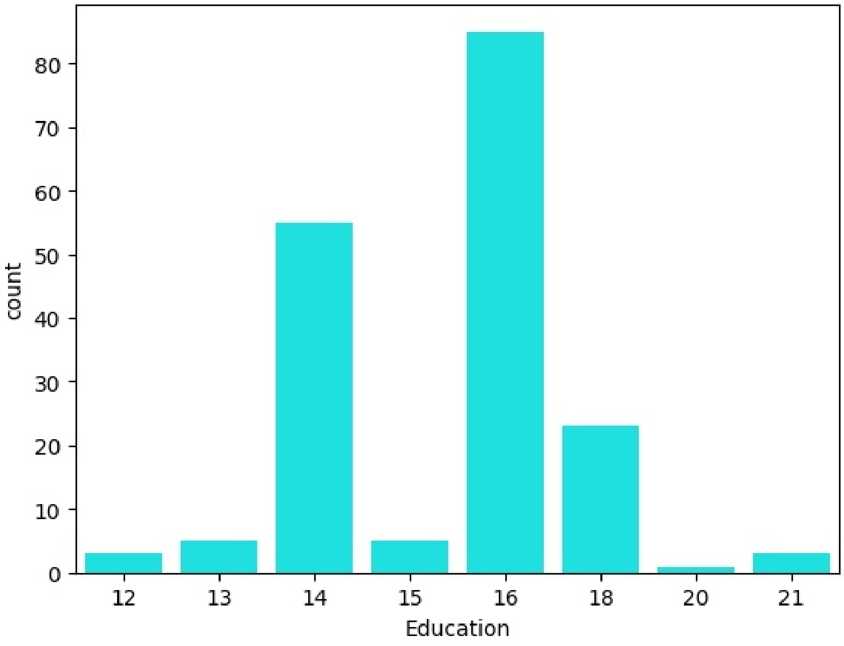
Analysis based on Education level of customers:

In [56]:
```
sns.countplot(x=data["Education"], palette = ['cyan'])
```

```
<ipython-input-56-29973b8b0f5c>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable
to `hue` and set `legend=False` for the same effect.

  sns.countplot(x=data["Education"], palette = ['cyan'])
<ipython-input-56-29973b8b0f5c>:1: UserWarning:
The palette list has fewer values (1) than needed (8) and will cycle, which may produce an uninterpretable plot
.
  sns.countplot(x=data["Education"], palette = ['cyan'])
```

<Axes: xlabel='Education', ylabel='count'>

Out[56]:



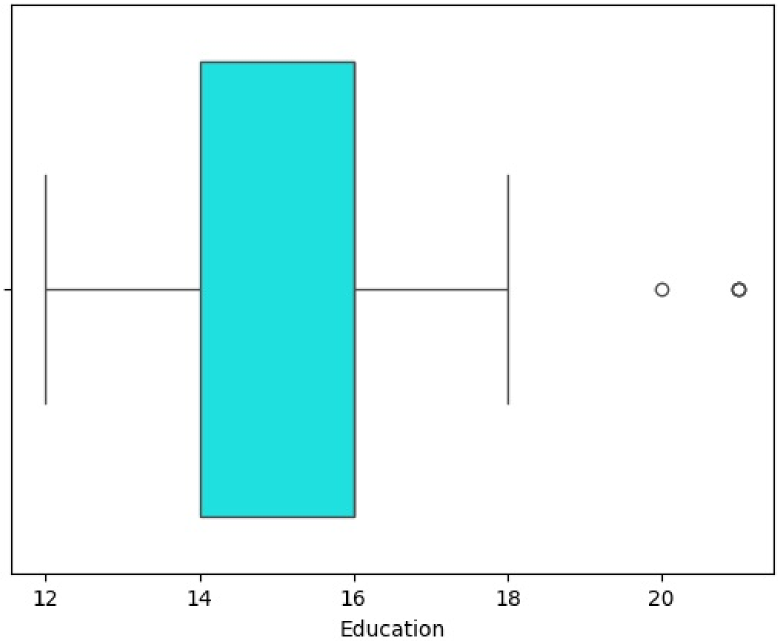
In [59]:

```
sns.boxplot(data["Education"], orient='h', palette = ['cyan'])
```

```
<ipython-input-59-7920da415607>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable
to `hue` and set `legend=False` for the same effect.

  sns.boxplot(data["Education"], orient='h', palette = ['cyan'])
```

Out[59]:

<Axes: xlabel='Education'>

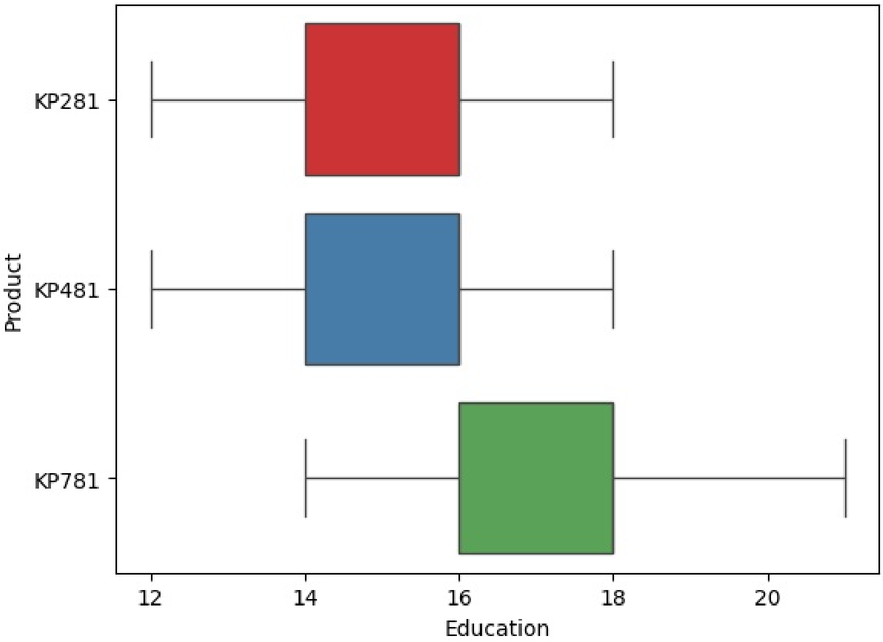


In [64]:

```
sns.boxplot(x=data["Education"],y=data["Product"], orient='h',hue=data["Product"], palette='Set1')
```

Out[64]:

<Axes: xlabel='Education', ylabel='Product'>

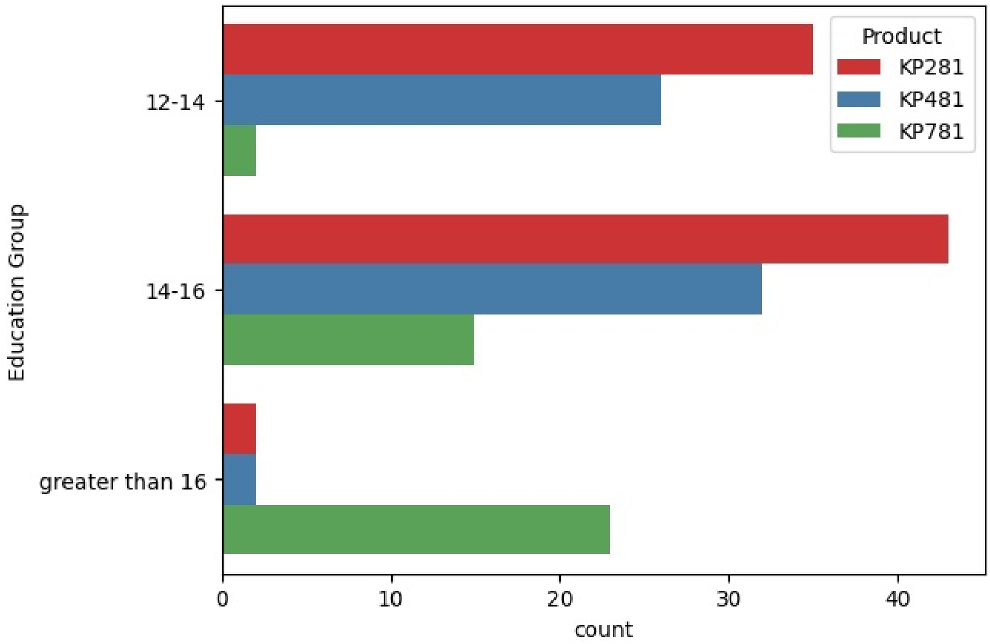


In [65]:
```
data["Education"].describe()
```

Out[65]:
```
count    180.000000
mean      15.572222
std        1.617055
min       12.000000
25%       14.000000
50%       16.000000
75%       16.000000
max       21.000000
Name: Education, dtype: float64
```

In [71]:
```
def education_group(education):
    if education <= 14:
        return '12-14'
    elif education <= 16:
        return '14-16'
    else:
        return 'greater than 16'
data['Education Group'] = data['Education'].apply(education_group)
sns.countplot(y=data["Education Group"], hue=data["Product"], palette='Set1')
```

Out[71]:
```
<Axes: xlabel='count', ylabel='Education Group'>
```



In [70]:
```
def education_group(education):
    if education <= 14:
        return '12-14'
    elif education <= 16:
        return '14-16'
    else:
        return 'greater than 16'
data['Education Group'] = data['Education'].apply(education_group)

crosstab_education = pd.crosstab(index=data['Education Group'], columns='count')

marginal_prob_education_group = crosstab_education / crosstab_education.sum()
marginal_prob_education_group.columns = ['Probability']

print("Marginal probabilities for education groups:")
marginal_prob_education_group
```

```
Marginal probabilities for education groups:
```

Out[70]:

Education Group	Probability
12-14	0.35
14-16	0.50
greater than 16	0.15

Inference:

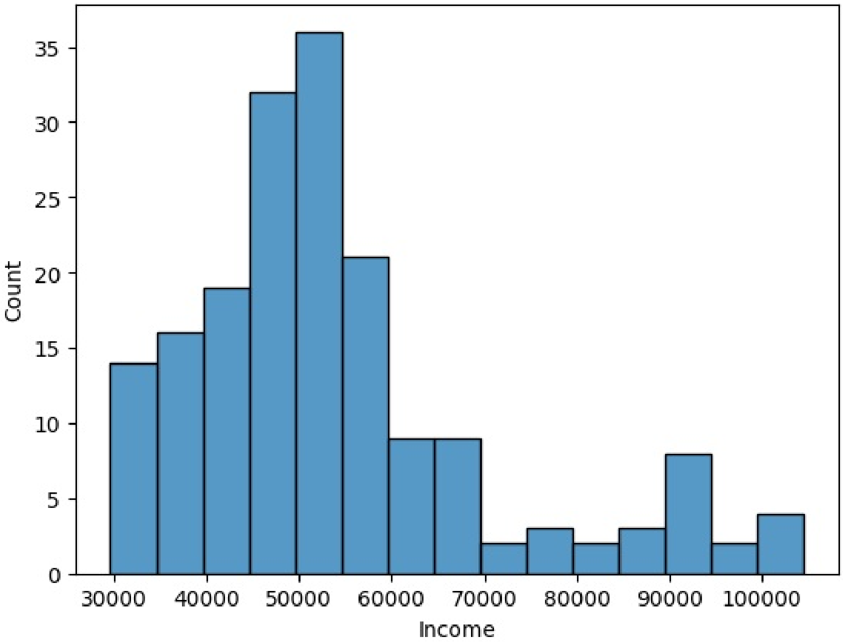
- The visualizations suggest that the average customer base for these products predominantly falls within the education level range of 14-16 years, closely followed by the 12-14 years education group.
- Additionally, there is a noticeable trend where customers with higher levels of educational attainment exhibit a greater propensity to purchase the premium or high-end model, KP781, compared to the other product offerings.
- The plots reveal a positive correlation between the customers' educational background and their likelihood of opting for the top-tier,

feature-rich model, indicating that product positioning and marketing strategies may need to be tailored to cater to the preferences and purchasing power of more educated consumer segments.

Income Wise Analysis:

In [72]:
```
sns.histplot(data=data, x="Income")
```

Out[72]:
```
<Axes: xlabel='Income', ylabel='Count'>
```



In [73]:
```
crosstab = pd.crosstab(index=data[data["Income"]> data["Income"].mean() ]["Product"], columns='count')
marginal_probability = crosstab / crosstab.sum()

marginal_probability_percent = marginal_probability * 100
marginal_probability_percent.columns = ['Percentage']
print(marginal_probability_percent.round())
```

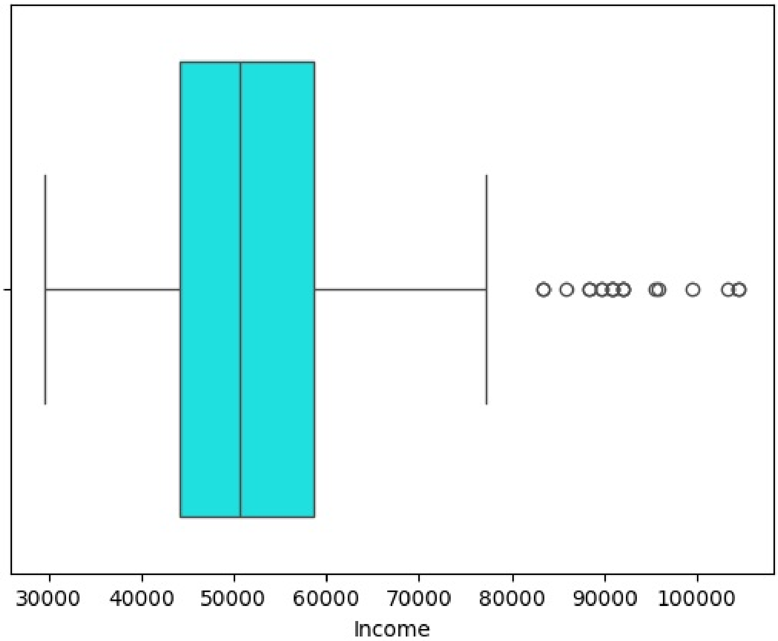
```
         Percentage
Product
KP281          29.0
KP481          21.0
KP781          51.0
```

In [ ]:

In [75]:
```
sns.boxplot(data=data, x="Income", palette = ['cyan'])
```

```
<ipython-input-75-65a038122864>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.boxplot(data=data, x="Income", palette = ['cyan'])
```

Out[75]:
```
<Axes: xlabel='Income'>
```



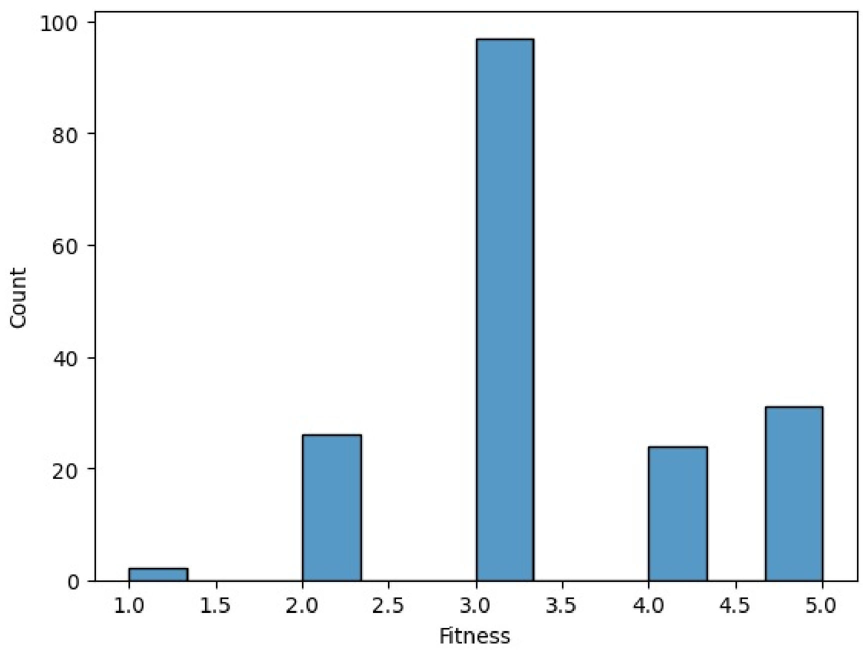
Analysis

- Majority of customers have an income range between 40,000 and 60,000.
- Higher income groups tend to have less time for fitness due to work commitments.
- The median income of most buyers is around 50,000.
- Presence of outliers indicates a segment of high-income individuals investing in health and fitness.
- Despite high-income customers purchasing advanced treadmills, their numbers are lower compared to those buying intermediate and beginner models.
- Revenue distribution is relatively equal across different product tiers.

Analysis based on user's Fitness Levels

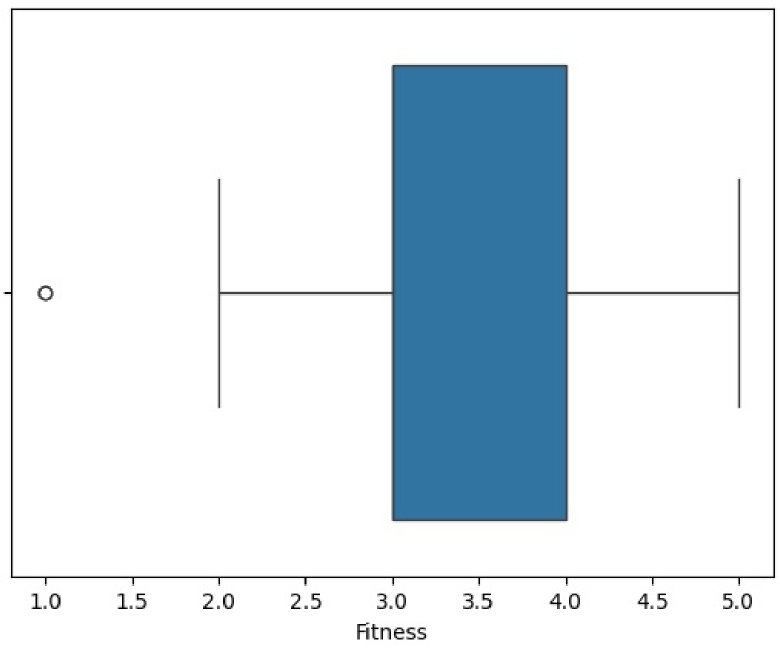
In [78]:
```
sns.histplot( data, x="Fitness")
```

Out[78]: <Axes: xlabel='Fitness', ylabel='Count'>




```
In [79]: sns.boxplot(data, x="Fitness")
```

```
Out[79]: <Axes: xlabel='Fitness'>
```

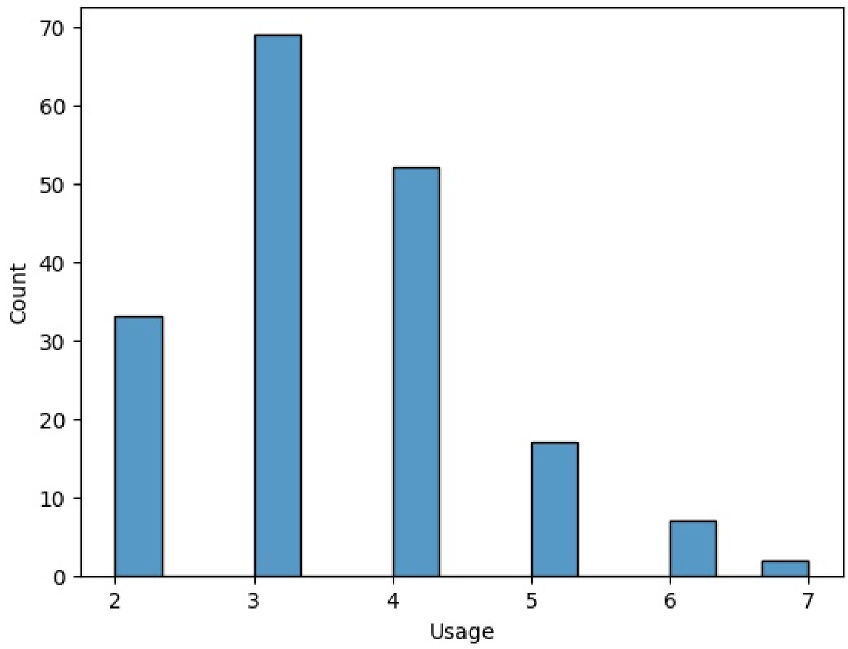


Analysis

- Individuals with very low fitness levels are less likely to purchase a treadmill.
- People with median or above-median fitness levels are more inclined to buy a new treadmill.
- The majority of treadmill buyers have a fitness score of 3 or higher.
- Conclusion: Most treadmill customers have a fitness level of 3 or above, indicating a preference among individuals with moderate to high fitness levels.

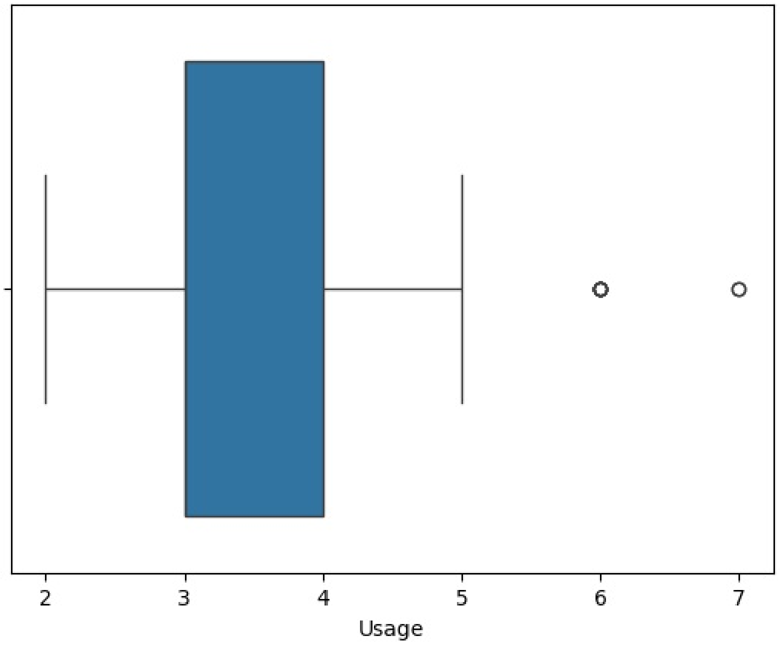
```
In [82]: sns.histplot(data, x="Usage")
```

```
Out[82]: <Axes: xlabel='Usage', ylabel='Count'>
```



In [83]:
```
sns.boxplot(data, x="Usage")
```

Out[83]:
```
<Axes: xlabel='Usage'>
```



Analysis

- Majority of treadmill buyers use it for 2-4 days per week.
- Most buyers utilize their treadmills for approximately 3 days a week.
- Conclusion: The typical treadmill customer is a beginner or intermediate user, exercising on their treadmill around 3 days per week.

- There are 5 outliers observed in the data.

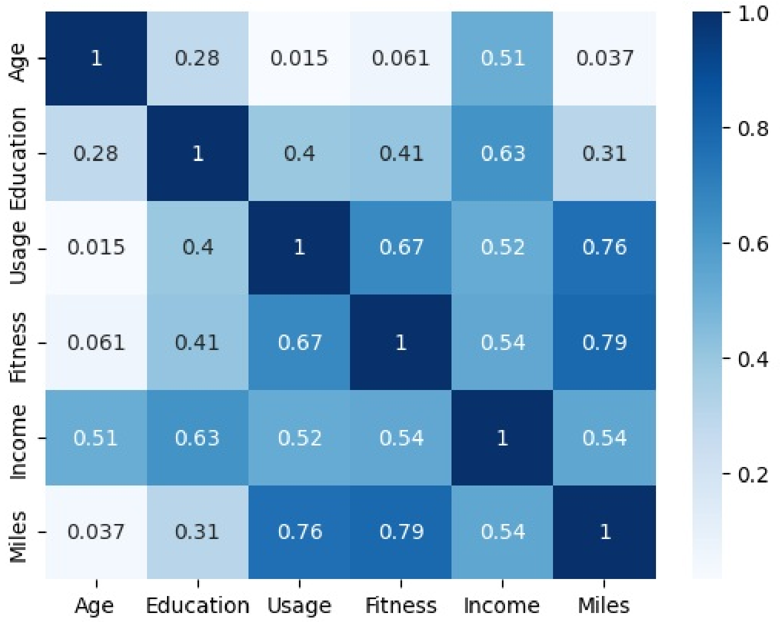
Bivariate Analysis

In [84]:
```
print("Correlation between numerical (continuous) variables : \n")

numeric_data = data.select_dtypes(include=['int'])
correlation_matrix = numeric_data.corr()

correlation_matrix
```

```
Correlation between numerical (continuous) variables :
```

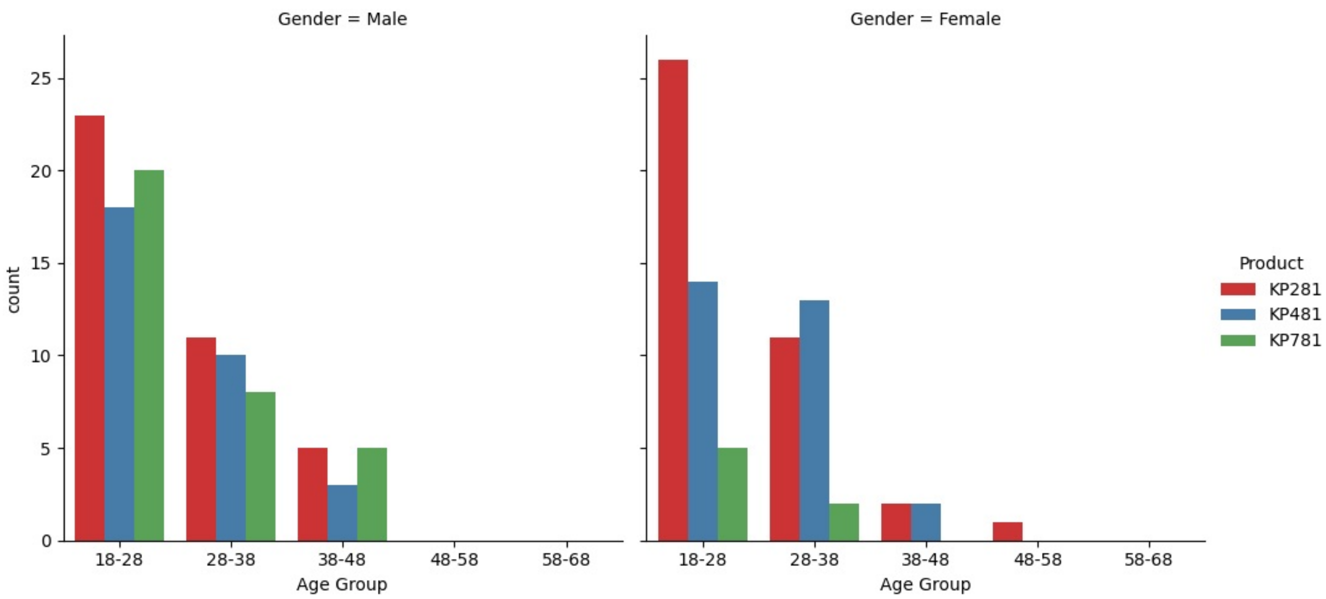
Out[84]:

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

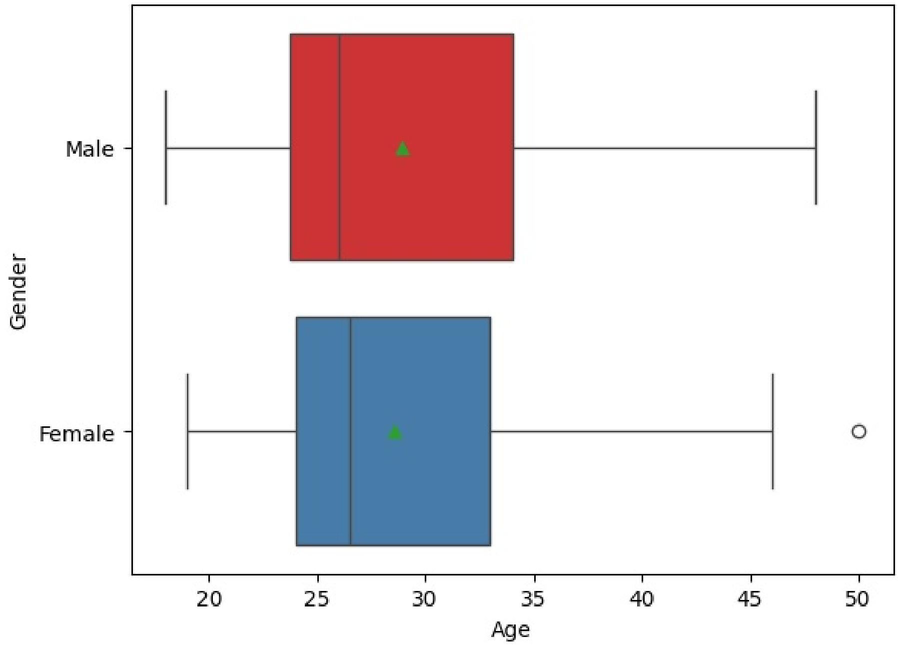
In [85]:
```
sns.heatmap(correlation_matrix, annot=True, cmap='Blues')
```
Out[85]:
```
<Axes: >
```


Age vs Gender

In [ ]:

In [90]:
```
df_copy = data
df_copy['Age Group'] = pd.cut(data['Age'],bins=[18,28,38,48,58,68],labels=['18-28','28-38','38-48','48-58','58-

sns.catplot(x='Age Group', col='Gender', hue='Product', data=data, kind='count', palette='Set1')
```
Out[90]:
```
<seaborn.axisgrid.FacetGrid at 0x7eb6e9da12a0>
```


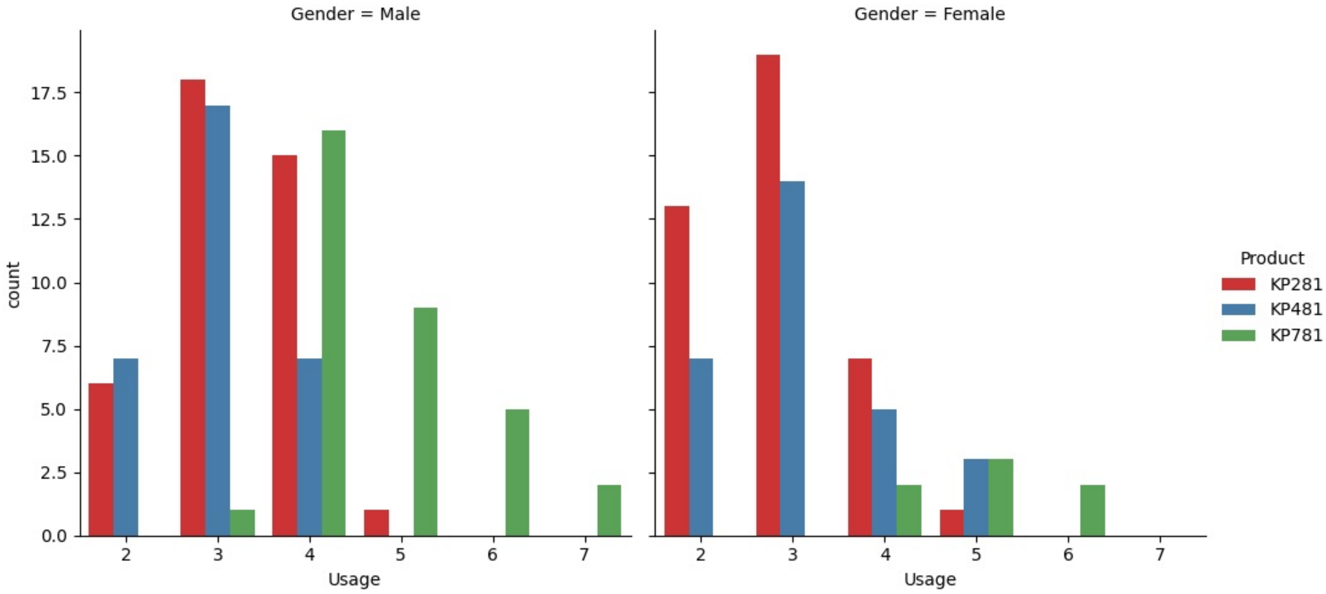
In [92]:
```
sns.boxplot(x=data["Age"], y=data["Gender"], hue=data["Gender"], showmeans=True, palette='Set1')
```
Out[92]:
```
<Axes: xlabel='Age', ylabel='Gender'>
```



Gender vs Usage

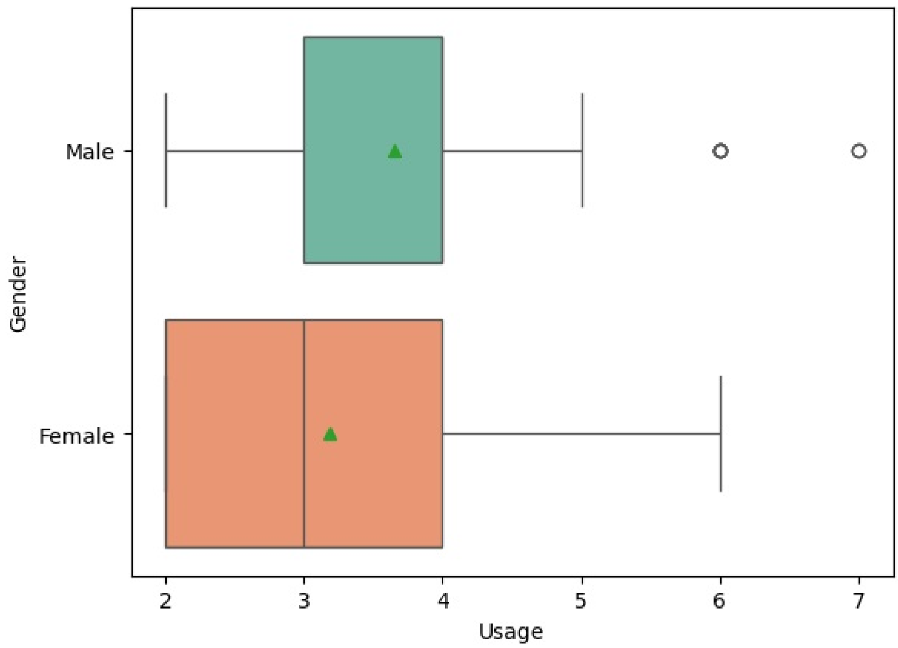
In [95]:
```
sns.catplot(x='Usage', col='Gender', hue='Product', data=data, kind='count', palette='Set1')
```

Out[95]: <seaborn.axisgrid.FacetGrid at 0x7eb6e9cbbc10>



In [97]:
```
sns.boxplot(y=data["Gender"], x=data["Usage"], showmeans=True, hue=data["Gender"], palette='Set2')
```

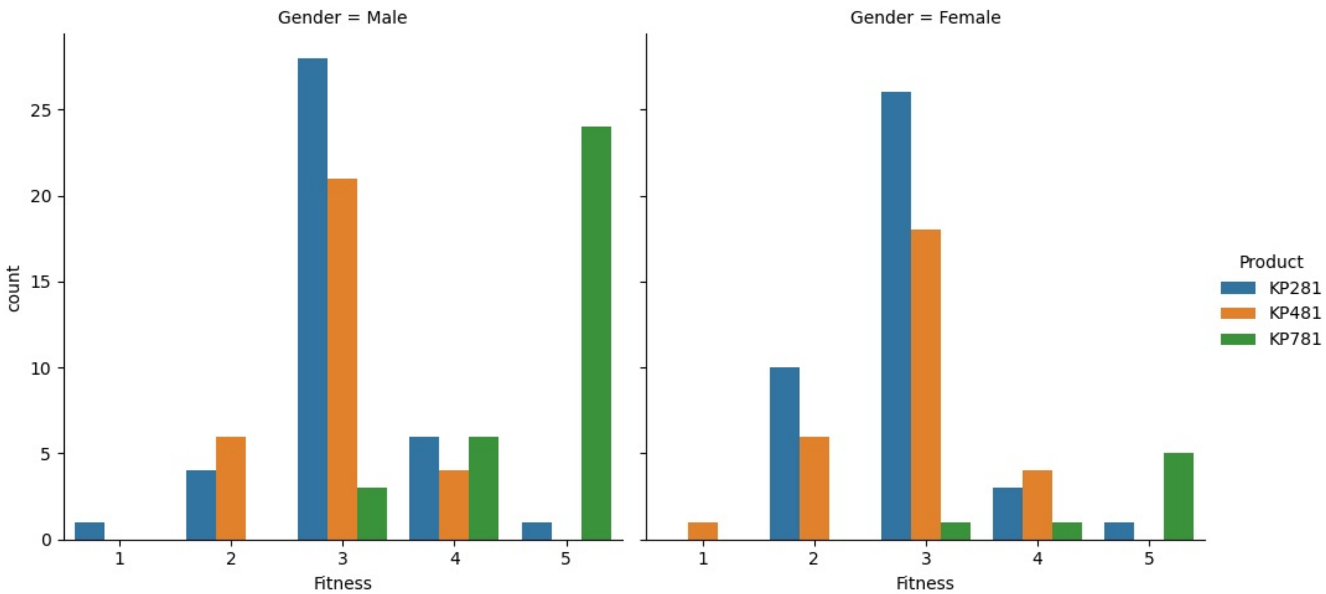
Out[97]: <Axes: xlabel='Usage', ylabel='Gender'>



Gender vs Fitness

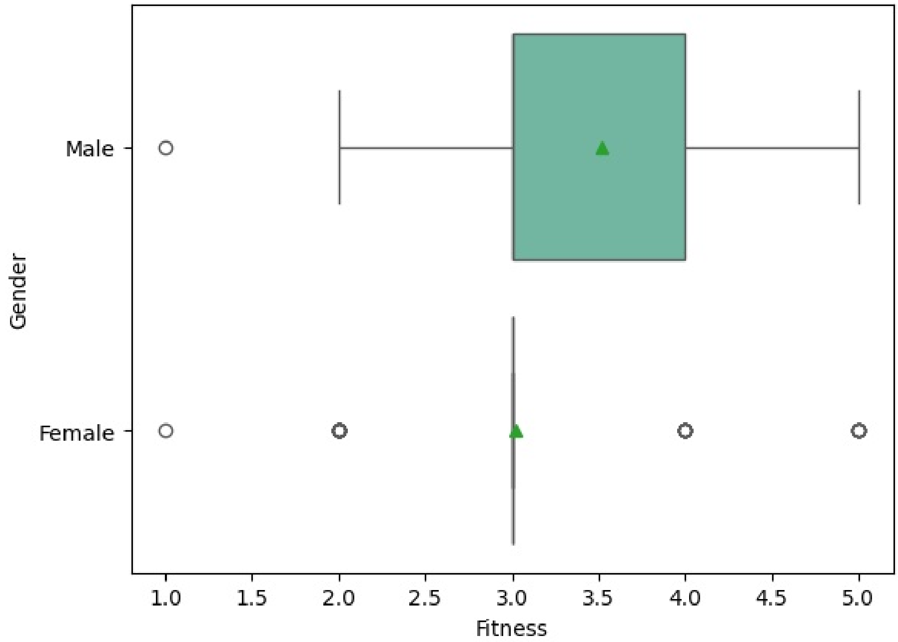
In [101…
```
sns.catplot(data=data,x="Fitness",col="Gender",kind="count",hue="Product")
```

Out[101]: <seaborn.axisgrid.FacetGrid at 0x7eb6e98cb6a0>



In [104…
```
sns.boxplot(y=data["Gender"], x=data["Fitness"], showmeans=True, hue=data["Gender"], palette='Set2')
```

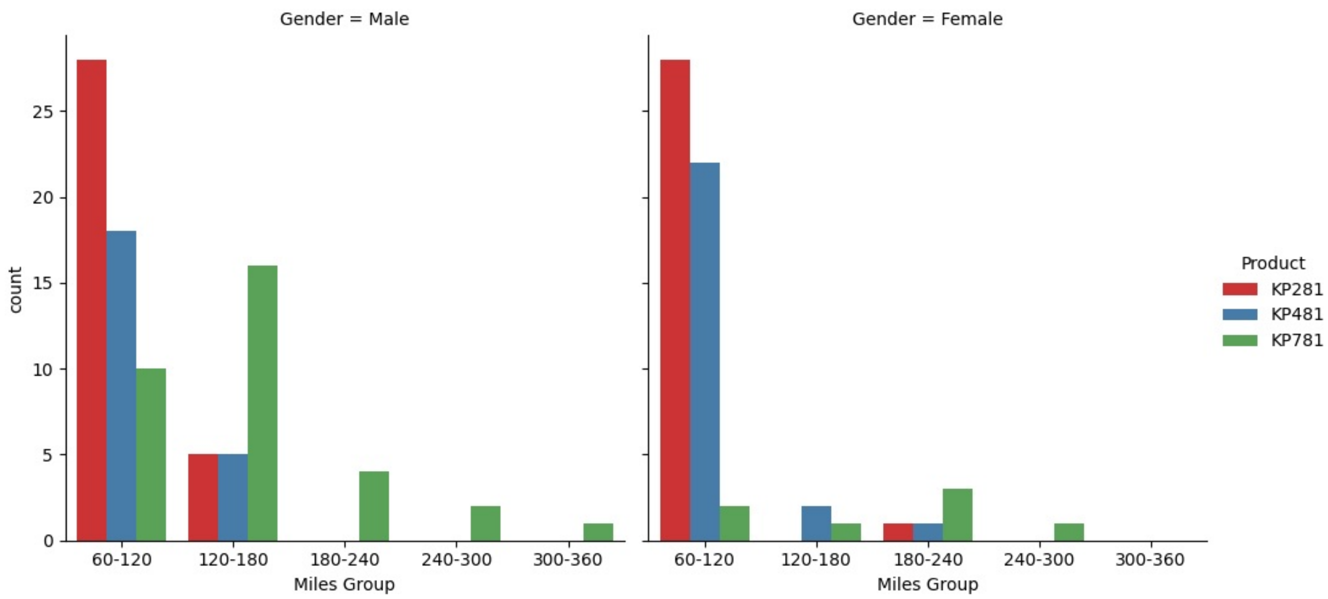
Out[104]: <Axes: xlabel='Fitness', ylabel='Gender'>



Gender vs Miles

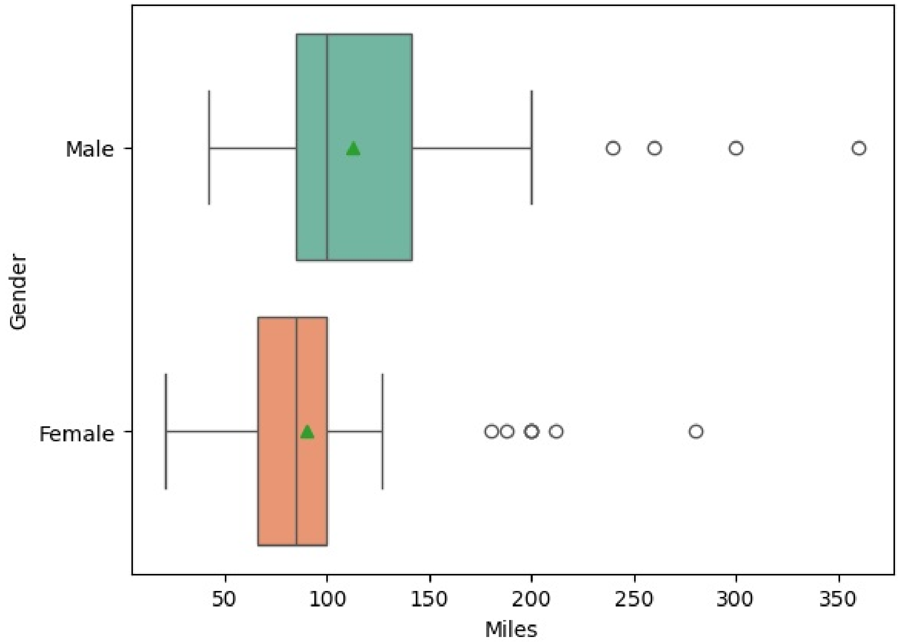
In [105…
```
df_copy['Miles Group'] = pd.cut(data['Miles'],bins=[60,120,180,240,300,360],labels=['60-120','120-180','180-240
sns.catplot(x='Miles Group', col='Gender', hue='Product', data=data, kind='count', palette='Set1')
```

Out[105]: <seaborn.axisgrid.FacetGrid at 0x7eb6e96cf670>



In [106…
```
sns.boxplot(y=data["Gender"], x=data["Miles"], showmeans=True, hue=data["Gender"], palette='Set2')
```

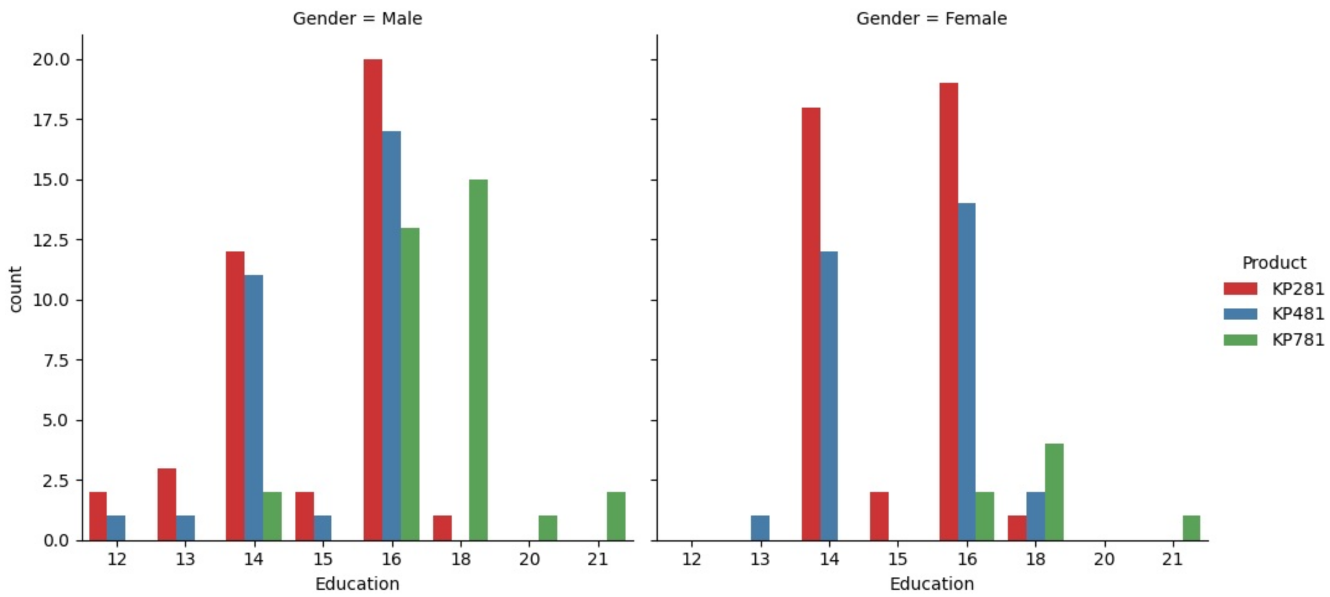
Out[106]: <Axes: xlabel='Miles', ylabel='Gender'>



Gender vs Education

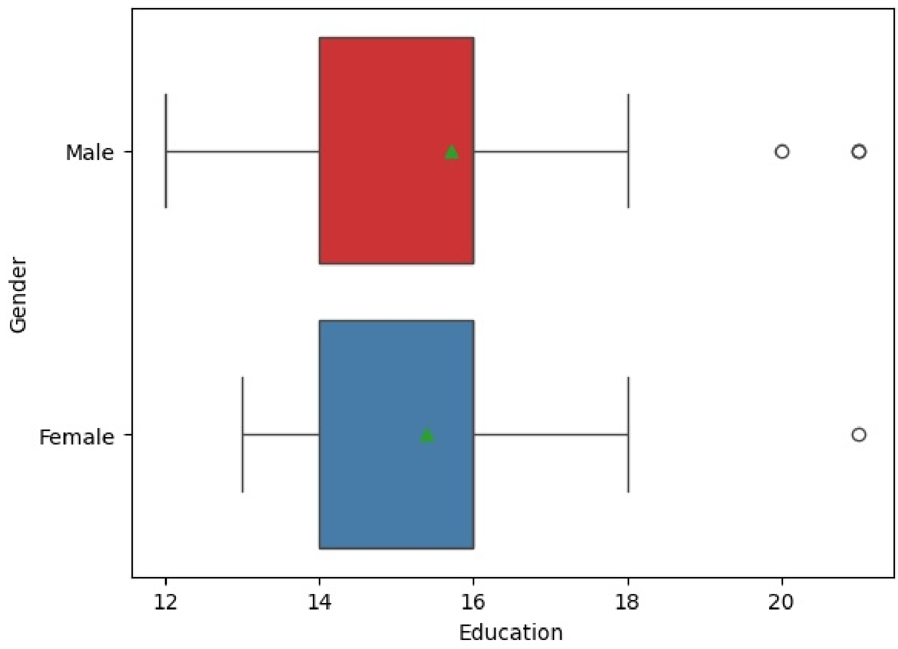
In [107…
```python
sns.catplot(x='Education', col='Gender', hue='Product', data=data, kind='count', palette='Set1')
```

Out[107]: <seaborn.axisgrid.FacetGrid at 0x7eb6f05df580>



In [109…
```python
sns.boxplot(y=data["Gender"], x=data["Education"], showmeans=True, hue=data["Gender"], palette='Set1')
```

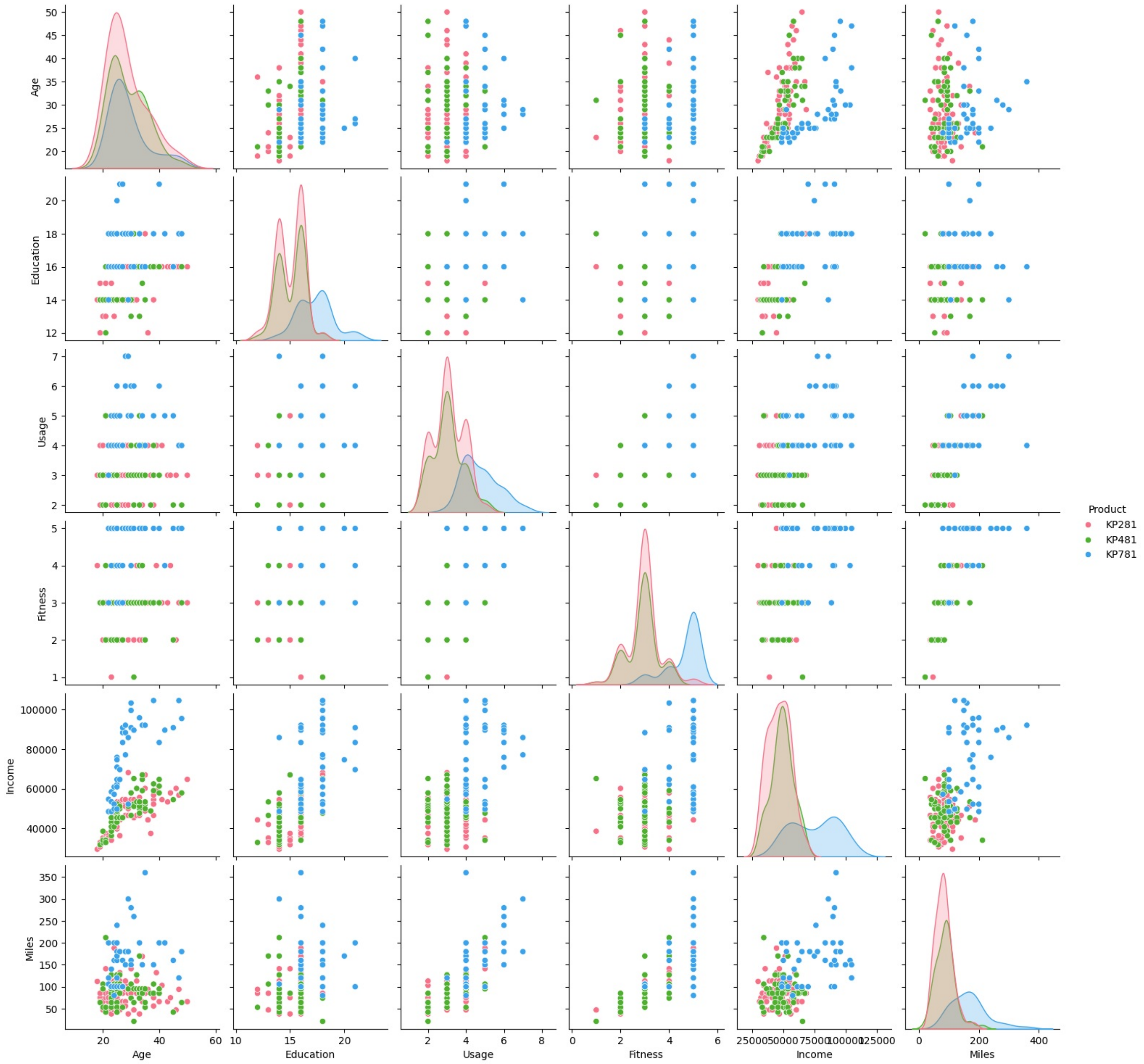
Out[109]: <Axes: xlabel='Education', ylabel='Gender'>



PairPlot

In [116…
```
sns.pairplot(data,hue="Product",palette="husl")
```

Out[116]: <seaborn.axisgrid.PairGrid at 0x7eb6e88046a0>



- Strong positive correlations observed between Miles-Fitness, Miles-Usage, and Fitness-Usage.
- High correlation between Education and Income levels, as expected.
- The identified correlations provide insights into relationships between physical activity, fitness, and socioeconomic factors.

Conditional Probabilities

Probabilities of Purchasing a Product with respect to Age Group.

In [117…

```
pd.crosstab(index=df_copy["Product"],columns=data["Age Group"],margins=True,normalize=True).round(2) * 100
```

Out[117]:

Age Group	18-28	28-38	38-48	48-58	All
Product					
KP281	27.0	12.0	4.0	1.0	44.0
KP481	18.0	13.0	3.0	0.0	34.0
KP781	14.0	6.0	3.0	0.0	22.0
All	59.0	31.0	9.0	1.0	100.0

Probability of buying:

- For KP281:
  - Age 18-28: 27%
  - Age 28-38: 12%
  - Age 38-48: 4%
- For KP481:

- Age 18-28: 17%
    - Age 28-38: 12%
    - Age 38-48: 2%
- For KP781:
    - Age 18-28: 13%
    - Age 28-38: 5%
    - Age 38-48: 2%

Probabilities of Purchasing a Product with respect to Education group.

In [118…
```
df_copy["Education Group"] = pd.cut(df_copy["Education"],bins=[12,14,16,18,20,22],labels=["12-14","14-16","16-1

pd.crosstab(index=df_copy["Product"],columns=data["Education Group"],margins=True,normalize=True).round(2) * 10
```

Out[118]:

Education Group / Product	12-14	14-16	16-18	18-20	20-22	All
KP281	19.0	24.0	1.0	0.0	0.0	44.0
KP481	14.0	18.0	1.0	0.0	0.0	33.0
KP781	1.0	8.0	11.0	1.0	2.0	23.0
All	34.0	51.0	13.0	1.0	2.0	100.0

Here are the probabilities of a person who has purchased each product being in different education groups:

- For KP281:
    - Education group 12-14: 18%
    - Education group 14-16: 24%
    - Education group 16-18: 1%
- For KP481:
    - Education group 12-14: 14%
    - Education group 14-16: 18%
    - Education group 16-18: 1%
- For KP781:
    - Education group 12-14: 1%
    - Education group 14-16: 8%
    - Education group 16-18: 10%

Probabilities of Purchasing a Product with respect to Usage.

In [119…
```
pd.crosstab(index=df_copy["Product"],columns=data["Usage"],margins=True,normalize=True).round(2) * 100
```

Out[119]:

Usage / Product	2	3	4	5	6	7	All
KP281	11.0	21.0	12.0	1.0	0.0	0.0	44.0
KP481	8.0	17.0	7.0	2.0	0.0	0.0	33.0
KP781	0.0	1.0	10.0	7.0	4.0	1.0	22.0
All	18.0	38.0	29.0	9.0	4.0	1.0	100.0

For KP281:

Usage per week group 2: 10% Usage per week group 3: 20% Usage per week group 4: 12% Usage per week group 5: 1% Usage per week group 6: 0% For KP481:

Usage per week group 2: 7% Usage per week group 3: 17% Usage per week group 4: 6% Usage per week group 5: 1% Usage per week group 6: 0% For KP781:

Usage per week group 2: 0% Usage per week group 3: 0% Usage per week group 4: 10% Usage per week group 5: 6% Usage per week group 6: 4%

Probabilities of Purchasing a Product with respect to Income Group.

In [121…
```
pd.crosstab(columns=df_copy["Product"],index=data["IG"],margins=True,normalize=True).round(2) * 100
```

Out[121]:

Product	KP281	KP481	KP781	All
IG				
2X̄-3X̄	1.0	0.0	0.0	1.0
3X̄-4X̄	12.0	5.0	0.0	17.0
4X̄-5X̄	14.0	12.0	3.0	28.0
5X̄-6X̄	14.0	13.0	3.0	31.0
6X̄-7X̄	3.0	4.0	3.0	11.0
7X̄-8X̄	0.0	0.0	2.0	2.0
<8X̄	0.0	0.0	11.0	11.0
All	44.0	33.0	22.0	100.0

Here are the probabilities of a person who has purchased each product being in different income groups:

- For KP281:
  - Income group 20K-30K: 0%
  - Income group 30K-40K: 12%
  - Income group 40K-50K: 13%
  - Income group 50K-60K: 14%
  - Income group 60K-70K: 3%
  - Income group 70K-80K: 0%
  - Income group <80K: 0%
- For KP481:
  - Income group 20K-30K: 0%
  - Income group 30K-40K: 5%
  - Income group 40K-50K: 11%
  - Income group 50K-60K: 12%
  - Income group 60K-70K: 3%
  - Income group 70K-80K: 0%
  - Income group <80K: 0%
- For KP781:
  - Income group 20K-30K: 0%
  - Income group 30K-40K: 0%
  - Income group 40K-50K: 2%
  - Income group 50K-60K: 3%
  - Income group 60K-70K: 3%
  - Income group 70K-80K: 2%
  - Income group <80K: 10%

Probabilities of Purchasing a Product with respect to Fitness Level.

In [122…

```
pd.crosstab(index=df_copy["Product"],columns=data["Fitness"],margins=True,normalize=True).round(2) * 100
```

Out[122]:

Fitness	1	2	3	4	5	All
Product						
KP281	1.0	8.0	30.0	5.0	1.0	44.0
KP481	1.0	7.0	22.0	4.0	0.0	33.0
KP781	0.0	0.0	2.0	4.0	16.0	22.0
All	1.0	14.0	54.0	13.0	17.0	100.0

In [ ]:

```
Here are the probabilities of a person who has purchased each product being in different fitness level groups:

- For KP281:
  - Fitness Level group 1: 0%
  - Fitness Level group 2: 7%
  - Fitness Level group 3: 30%
  - Fitness Level group 4: 5%
  - Fitness Level group 5: 1%

- For KP481:
  - Fitness Level group 1: 0%
  - Fitness Level group 2: 6%
  - Fitness Level group 3: 21%
  - Fitness Level group 4: 4%
  - Fitness Level group 5: 0%
```

```
- For KP781:
  - Fitness Level group 1: 0%
  - Fitness Level group 2: 0%
  - Fitness Level group 3: 2%
  - Fitness Level group 4: 3%
  - Fitness Level group 5: 16%
```

Probabilities of Purchasing a Product with respect to MaritalStatus.

In [123…
```
pd.crosstab(index=df_copy["Product"],columns=data["MaritalStatus"],margins=True,normalize=True).round(2) * 100
```

Out[123]:

MaritalStatus	Partnered	Single	All
Product			
KP281	27.0	18.0	44.0
KP481	20.0	13.0	33.0
KP781	13.0	9.0	22.0
All	59.0	41.0	100.0

Here are the probabilities of a person who has purchased each product having different marital statuses:

- For KP281:
  - Marital Status Single: 26%
  - Marital Status Partnered: 17%
- For KP481:
  - Marital Status Single: 20%
  - Marital Status Partnered: 13%
- For KP781:
  - Marital Status Single: 12%
  - Marital Status Partnered: 9%

Probabilities of Purchasing a Product with respect to Gender.

In [124…
```
pd.crosstab(index=df_copy["Product"],columns=data["Gender"],margins=True,normalize=True,).round(2) * 100
```

Out[124]:

Gender	Female	Male	All
Product			
KP281	22.0	22.0	44.0
KP481	16.0	17.0	33.0
KP781	4.0	18.0	22.0
All	42.0	58.0	100.0

Here are the probabilities of a person who has purchased each product being male or female:

- For KP281:
  - Male: 22%
  - Female: 22%
- For KP481:
  - Male: 17%
  - Female: 16%
- For KP781:
  - Male: 18%
  - Female: 3%

Customer Profiling:

- The likelihood of purchasing KP281 stands at 44%.
- KP481 has a purchase probability of 33%.
- For KP781, the probability of purchase is 22%.

Customer Profile for KP281:

- Age Demographics: Customers primarily fall within the 18-28 age range, with some individuals extending into the 28-38 age bracket,

reflecting a youthful customer base.
- Fitness Status: Customers typically exhibit a fitness level ranging from 2 to 4, indicating a moderate level of physical activity.
- Income Range: The majority of customers belong to the income bracket of 30K-60K, indicating a middle-income segment.
- Educational Background: Customers generally have an education level between 12-16 years, indicating a moderately educated demographic.
- Marital Status: Most customers are in a partnered relationship, reflecting a stable family-oriented customer base.
- Weekly Usage: Customers use the product 2-4 days per week, indicating regular but not intensive usage.
- Mileage Range: Customers cover a weekly mileage of 60-120 miles, indicating moderate usage patterns.
- Gender Representation: The customer base comprises a balanced mix of male and female buyers.

Customer Profile for KP481:

- Age Demographics: Customers are predominantly in the 18-28 age range, with some extending to the 28-48 age range.
- Fitness Status: Customers exhibit a slightly higher fitness level of 3-4 compared to KP281, indicating a more active user base.
- Income Range: Customers' income levels range from 40K-70K, suggesting a slightly higher disposable income compared to KP281 customers.
- Educational Background: Customers typically have an education level between 12-16 years, reflecting a similar educational profile to KP281.
- Marital Status: The customer base is predominantly partnered, but there is also a significant number of single customers.
- Weekly Usage: Customers use the product 2-4 days per week, similar to KP281.
- Mileage Range: Customers cover a weekly mileage of 60-180 miles, indicating a range of usage from moderate to more intensive.
- Gender Representation: Both male and female customers contribute equally to the customer base.

Customer Profile for KP781:

- Age Demographics: Customers are primarily in the 18-28 age range, with some extending to the 28-48 age range.
- Fitness Status: Customers exhibit a higher fitness level of 3-5, indicating a more active and fitness-conscious user base.
- Income Range: There is a mix of income levels, with some customers in the 40K-80K range and a majority above 80K, indicating a higher purchasing power among certain segments.
- Educational Background: Customers generally have an education level between 14-18 years, indicating a relatively higher level of education compared to KP281 and KP481 customers.
- Marital Status: Similar to KP481, the customer base is predominantly partnered, but there are also single customers.
- Weekly Usage: Customers use the product 4-6 days per week, indicating a more frequent usage pattern compared to KP281 and KP481.
- Mileage Range: Customers cover a weekly mileage of 120-300 miles, indicating a more intensive usage compared to the other products.
- Gender Representation: The customer base is predominantly male, indicating a bias towards male buyers for this product.

Concluding Remarks and Recommendations:

- The majority of buyers across all products are young adults, suggesting that marketing strategies should primarily target this demographic.
- Female buyers are slightly underrepresented for KP781, presenting an opportunity for tailored advertising and fitness initiatives to attract this segment.
- Adjusting the price of KP281 and organizing fitness events can attract more buyers from lower-income brackets and boost overall product usage.
- Initiating fitness activities such as marathons can enhance customer engagement, foster loyalty, and increase usage, ultimately leading to improved revenue generation.