aerofitcasestudy

May 22, 2025

[95]:
```
import pandas as pd
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/
↪original/aerofit_treadmill.csv?1639992749
```

```
Downloading…
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/ori
ginal/aerofit_treadmill.csv?1639992749
To: /content/aerofit_treadmill.csv?1639992749
100% 7.28k/7.28k [00:00<00:00, 30.8MB/s]
```

[96]:
```
df = pd.read_csv('aerofit_treadmill.csv?1639992749')
df
```

[96]:
```
    Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
0     KP281   18    Male         14        Single      3        4   29562
1     KP281   19    Male         15        Single      2        3   31836
2     KP281   19  Female         14     Partnered      4        3   30699
3     KP281   19    Male         12        Single      3        3   32973
4     KP281   20    Male         13     Partnered      4        2   35247
..      …    …       …          …           …      …        …       …
175   KP781   40    Male         21        Single      6        5   83416
176   KP781   42    Male         18        Single      5        4   89641
177   KP781   45    Male         16        Single      5        5   90886
178   KP781   47    Male         18     Partnered      4        5  104581
179   KP781   48    Male         18     Partnered      4        5   95508

     Miles
0      112
1       75
2       66
3       85
4       47
..     …
175    200
176    200
177    160
178    120
179    180
```

[180 rows x 9 columns]

[97]:
```
df.shape
```

[97]:
```
(180, 9)
```

[98]:
```
#number of unique values in our data
for i in df.columns:
  print(i,':',df[i].nunique())
```

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

[99]:
```
#checking null values in every column of our data
df.isnull().sum()
```

[99]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

[100]:
```
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
```

[101]:
```
df.columns
```

[101]:
```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

[102]:
```
df.isna()
```

[102]:
```
     Product    Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  \
0      False  False   False      False          False  False    False   False
1      False  False   False      False          False  False    False   False
2      False  False   False      False          False  False    False   False
3      False  False   False      False          False  False    False   False
4      False  False   False      False          False  False    False   False
..       …      …      …          …              …      …       …       …
175    False  False   False      False          False  False    False   False
176    False  False   False      False          False  False    False   False
177    False  False   False      False          False  False    False   False
178    False  False   False      False          False  False    False   False
179    False  False   False      False          False  False    False   False

     Miles
0    False
1    False
2    False
3    False
4    False
..     …
175  False
176  False
177  False
178  False
179  False

[180 rows x 9 columns]
```

[103]:
```
df.duplicated()
```

[103]:
```
0      False
1      False
2      False
```

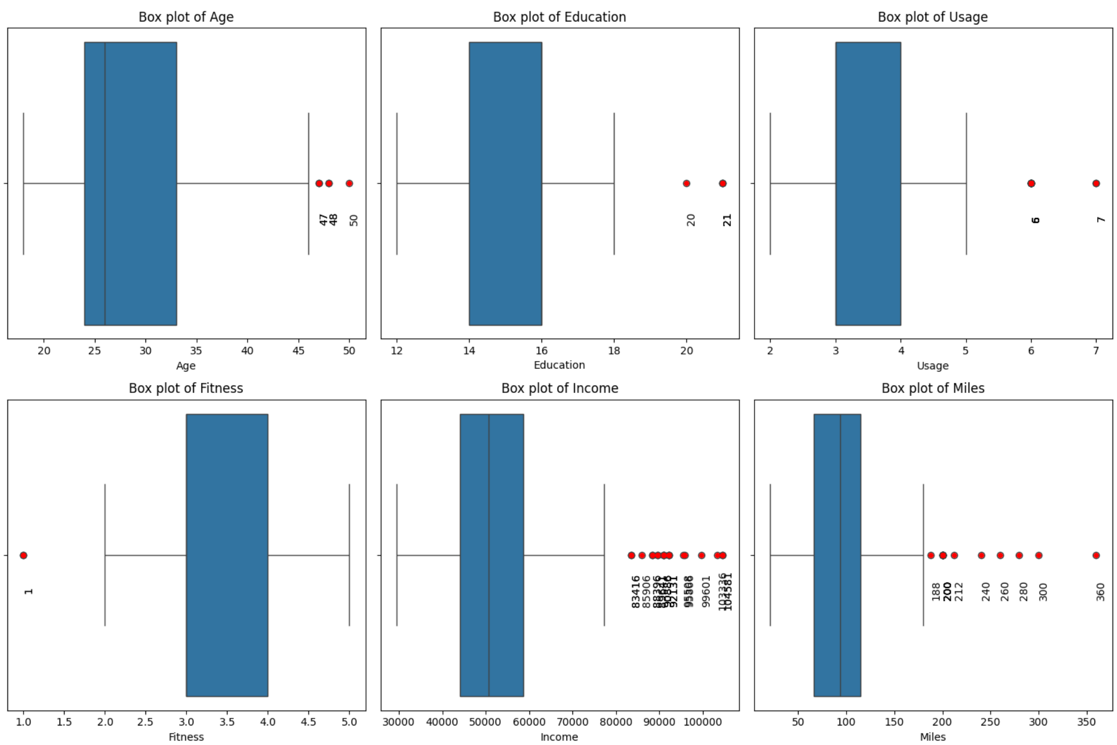
```
3      False
4      False
       …
175    False
176    False
177    False
178    False
179    False
Length: 180, dtype: bool
```

From the above analysis, it is clear that, data has total of 9 features with mixed alpha numeric data. Also we can see that there is no missing data in the columns.

[104]:

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

# Assuming 'df' is your pandas DataFrame

# Select the columns you want to plot
columns_to_plot = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles'] #␣
 ↪Replace with your actual column names

# Determine the number of rows and columns for the subplots
n_cols = 3
n_rows = 2

# Create a figure and subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(15, n_rows * 5))
axes = axes.flatten() # Flatten the axes array for easy iteration

# Plot box plots for each column and label outliers
for i, col in enumerate(columns_to_plot):
    if i < len(axes): # Ensure we don't go out of bounds
        sns.boxplot(x=df[col], ax=axes[i])
        axes[i].set_title(f'Box plot of {col}')

        # Calculate outlier boundaries
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        # Identify outliers
        outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]

        # Plot and label outliers
```

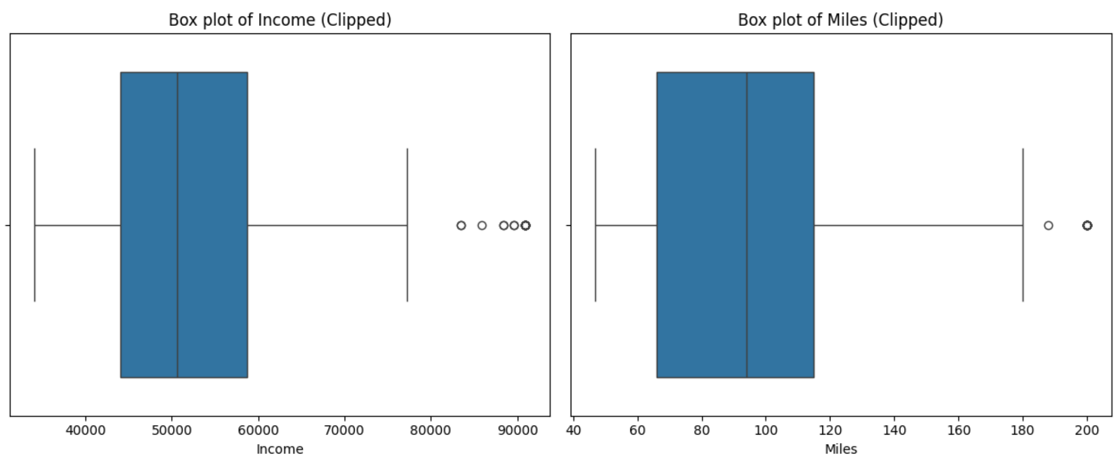
```
        for outlier in outliers[col]:
            axes[i].plot(outlier, 0, 'o', color='red', markersize=4) # Plot outlier as a red dot
            axes[i].text(outlier, 0.1, str(round(outlier, 2)), verticalalignment='center',rotation=90) # Label outlier value

# Remove any unused subplots
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout() # Adjust layout to prevent overlapping titles
plt.show()
```



It is observed that more outliers in Income and Miles data

[105]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df_clipped=df.copy()

# Calculate the 5th and 95th percentiles for the 'Income' column
lower_bound_income = df_clipped['Income'].quantile(0.05)
```

```
upper_bound_income = df_clipped['Income'].quantile(0.95)

# Apply np.clip() to the 'Income' column
df_clipped['Income'] = np.clip(df_clipped['Income'], lower_bound_income,␣
 ↪upper_bound_income)

# Calculate the 5th and 95th percentiles for the 'Miles' column
lower_bound_miles = df_clipped['Miles'].quantile(0.05)
upper_bound_miles = df_clipped['Miles'].quantile(0.95)

# Apply np.clip() to the 'Miles' column
df_clipped['Miles'] = np.clip(df['Miles'], lower_bound_miles, upper_bound_miles)

# Visualize with box plots
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

sns.boxplot(x=df_clipped['Income'], ax=axes[0])
axes[0].set_title('Box plot of Income (Clipped)')

sns.boxplot(x=df_clipped['Miles'], ax=axes[1])
axes[1].set_title('Box plot of Miles (Clipped)')

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



On cliping the data inbetween 5 and 95 percentile for vast outlier columns income and miles,the filtered df has reduced or clipped outliers

[106]:
```
df_clipped
```

[106]:

```
     Product  Age  Gender  Education MaritalStatus  Usage  Fitness    Income  \
0      KP281   18    Male         14        Single      3        4  34053.15
1      KP281   19    Male         15        Single      2        3  34053.15
2      KP281   19  Female         14     Partnered      4        3  34053.15
3      KP281   19    Male         12        Single      3        3  34053.15
4      KP281   20    Male         13     Partnered      4        2  35247.00
..       …   …     …          …           …    …      …       …
175    KP781   40    Male         21        Single      6        5  83416.00
176    KP781   42    Male         18        Single      5        4  89641.00
177    KP781   45    Male         16        Single      5        5  90886.00
178    KP781   47    Male         18     Partnered      4        5  90948.25
179    KP781   48    Male         18     Partnered      4        5  90948.25

     Miles
0      112
1       75
2       66
3       85
4       47
..     …
175    200
176    200
177    160
178    120
179    180

[180 rows x 9 columns]
```

[107]:

```python
df_clipped.describe(include=object)
```

[107]:

```
       Product Gender MaritalStatus
count      180    180           180
unique       3      2             2
top      KP281   Male     Partnered
freq        80    104           107
```

1. Product - The KP281 product demonstrated the highest sales performance among the three products
2. Gender -Around 58% of the buyers were Male and 42% were female
3. Marital Status -Around 60% of the buyers were Married and 40% were single

[108]:

```python
df_clipped.describe()
```

[108]:

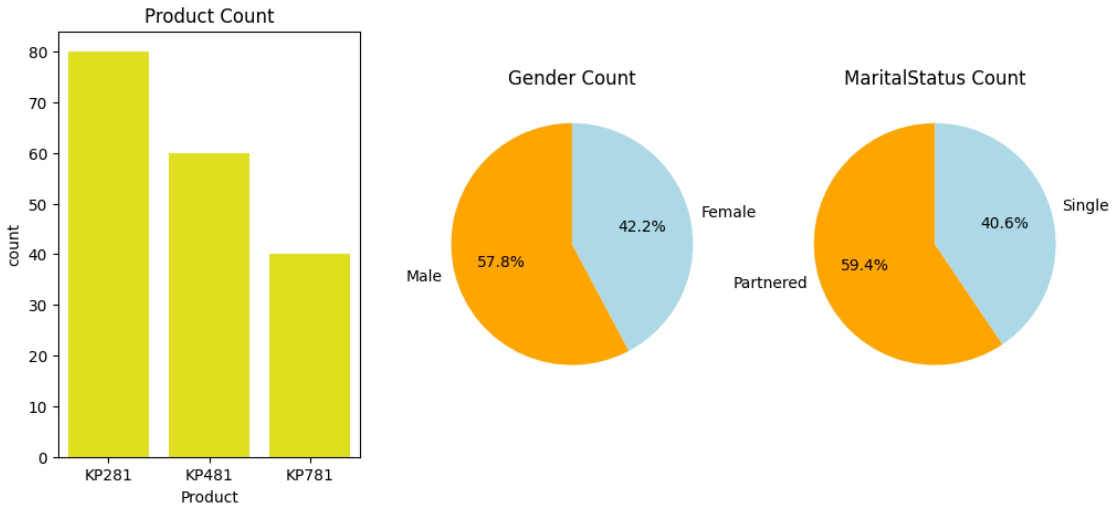
```
              Age   Education       Usage     Fitness         Income  \
count  180.000000  180.000000  180.000000  180.000000     180.000000
mean    28.788889   15.572222    3.455556    3.311111   53477.070000
```

```
std      6.943498   1.617055   1.084797   0.958869  15463.662523
min     18.000000  12.000000   2.000000   1.000000  34053.150000
25%     24.000000  14.000000   3.000000   3.000000  44058.750000
50%     26.000000  16.000000   3.000000   3.000000  50596.500000
75%     33.000000  16.000000   4.000000   4.000000  58668.000000
max     50.000000  21.000000   7.000000   5.000000  90948.250000

            Miles
count  180.000000
mean   101.088889
std     43.364286
min     47.000000
25%     66.000000
50%     94.000000
75%    114.750000
max    200.000000
```

1. Age - The age range of customers spans from 18 to 50 year, with an average age of 29 years.
2. Education - Customer education levels vary between 12 and 21 years, with an average education duration of 16 years.
3. Usage - Customers intend to utilize the product anywhere from 2 to 7 times per week, with an average usage frequency of 3 times per week.
4. Fitness - On average, customers have rated their fitness at 3 on a 5-point scale, reflecting a moderate level of fitness.
5. Income - The annual income of customers falls within the range of USD 34,000 to USD 90,000, with an average income of approximately USD 53,000.
6. Miles - Customers' weekly running goals range from 47 to 200 miles, with an average target of 101 miles per week

[109]:
```
Gender_counts=df_clipped['Gender'].value_counts()
MaritalStatus_counts=df_clipped['MaritalStatus'].value_counts()
Product_counts=df_clipped['Product'].value_counts()
fig, axes = plt.subplots(1,3, figsize=(12, 5))



axes[1].pie(Gender_counts, labels=Gender_counts.index, autopct='%1.1f%%',␣
↪startangle=90, colors=['orange', 'lightblue'])
axes[1].set_title('Gender Count')

sns.countplot(x='Product',data=df_clipped,order=Product_counts.
↪index,ax=axes[0],color='yellow')
axes[0].set_title('Product Count')
axes[2].pie(MaritalStatus_counts, labels=MaritalStatus_counts.index,␣
↪autopct='%1.1f%%', startangle=90, colors=['orange', 'lightblue'])
axes[2].set_title('MaritalStatus Count')
```

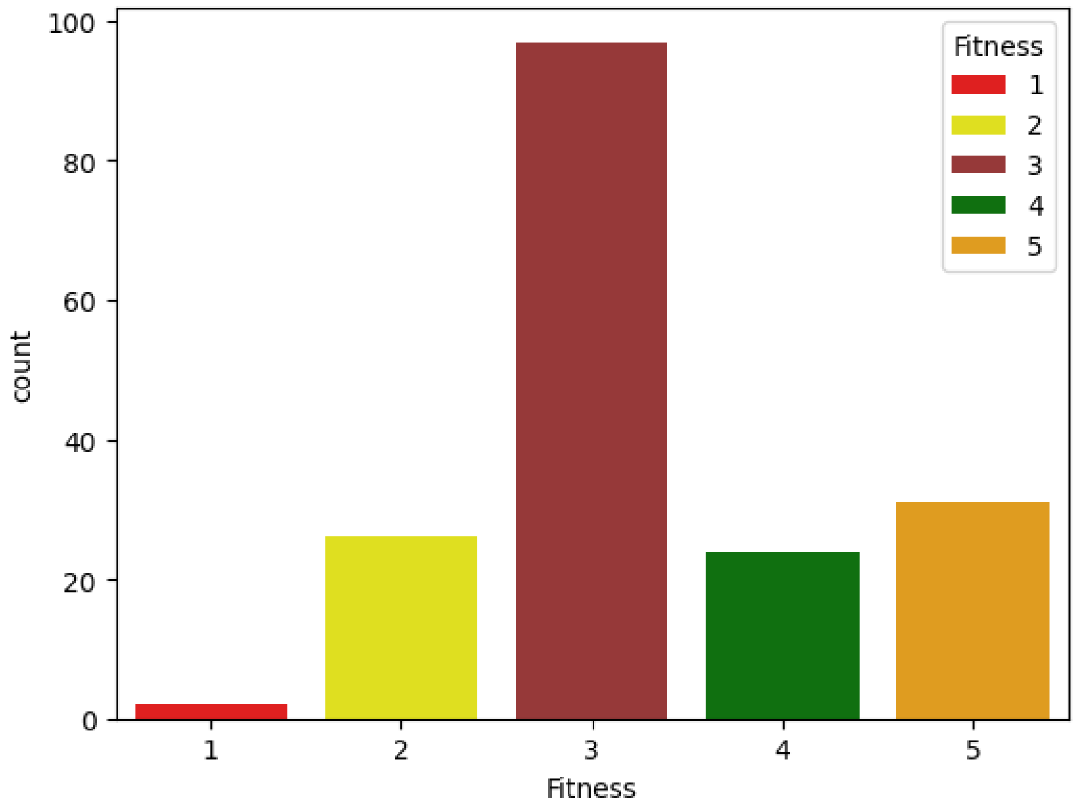
```
plt.show()
```



- KP281 is most purchased product

- Males have been found to have bought more products than females.
- Couples buy more products than singles

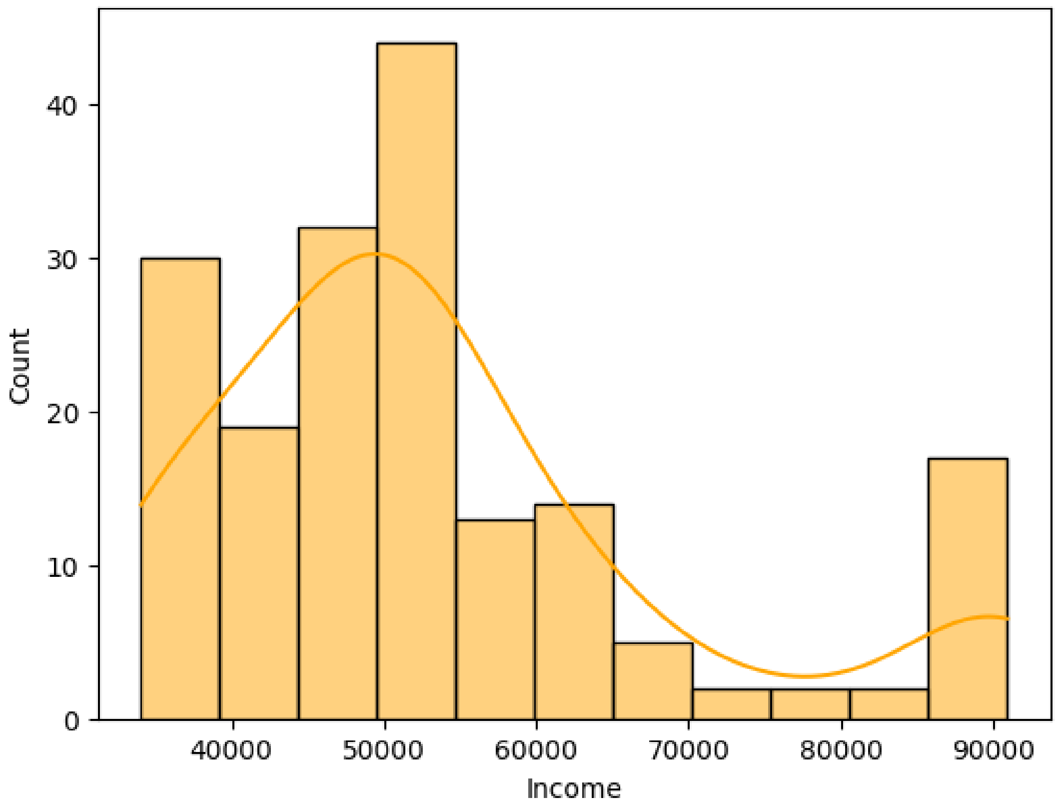
[110]:
```
# Fitness scale analysis - count plot
sns.
 ↪countplot(data=df_clipped,x='Fitness',hue='Fitness',palette=['red',"yellow","brown",␣
 ↪"green",'orange'])
plt.show()
```



More than 90 customers have rated their physical fitness scale as Average

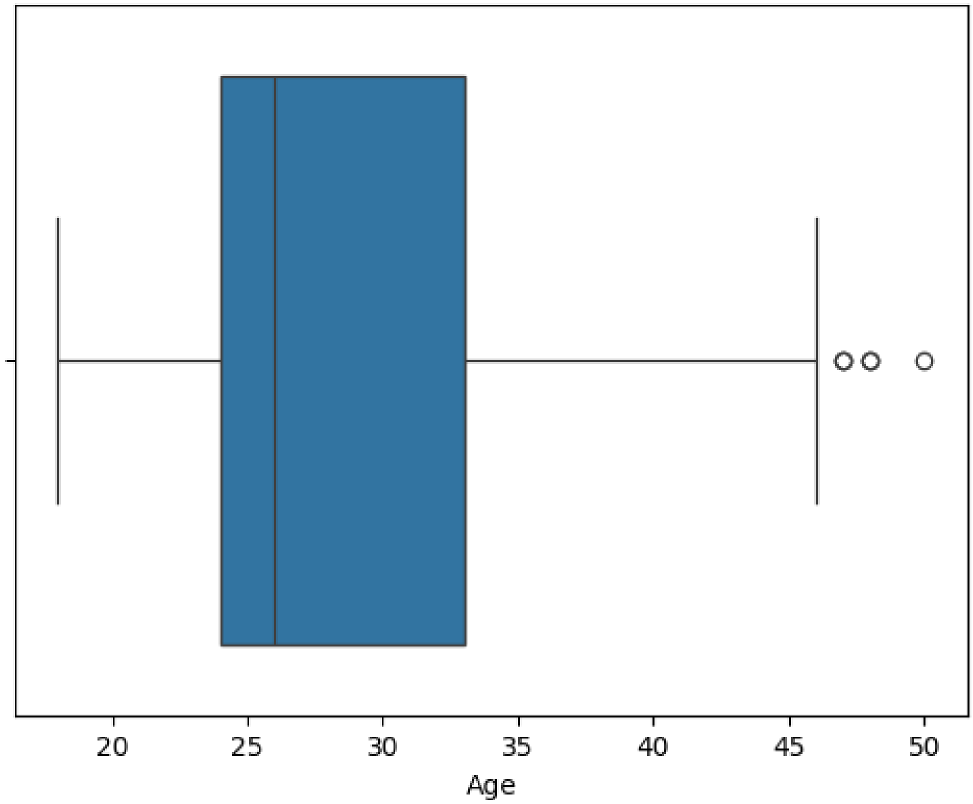
Second fitness scale is 5 which is the highest scale in Fitness column

[111]:
```
# Income Analysis - histplot
sns.histplot(df_clipped.Income,kde=True,color='orange')
plt.show()
```



Most of customers who have purchased the product have a average income between 40K to 60K

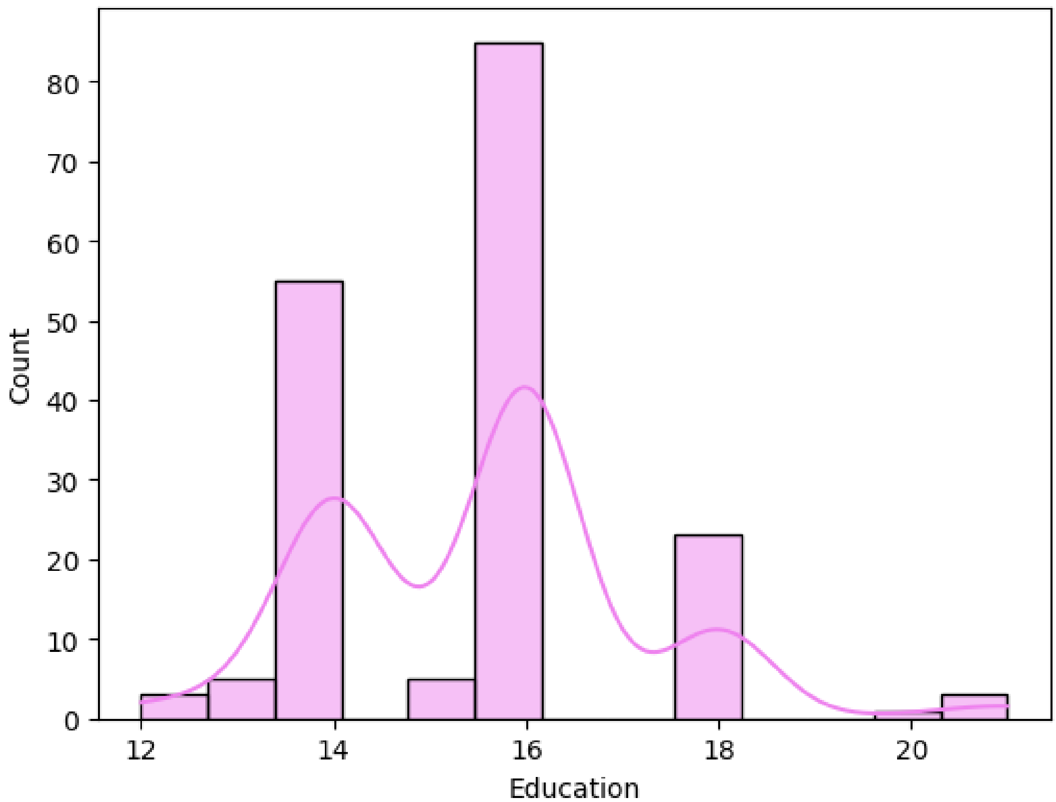
[112]:
```
# Age Analysis - Box plot
sns.boxplot(data=df_clipped,x='Age')
plt.show()
```



23 to 34 is the most common customer age group that has purchased the product

Above 45 years old customers are very few compared to the young age group given in the dataset

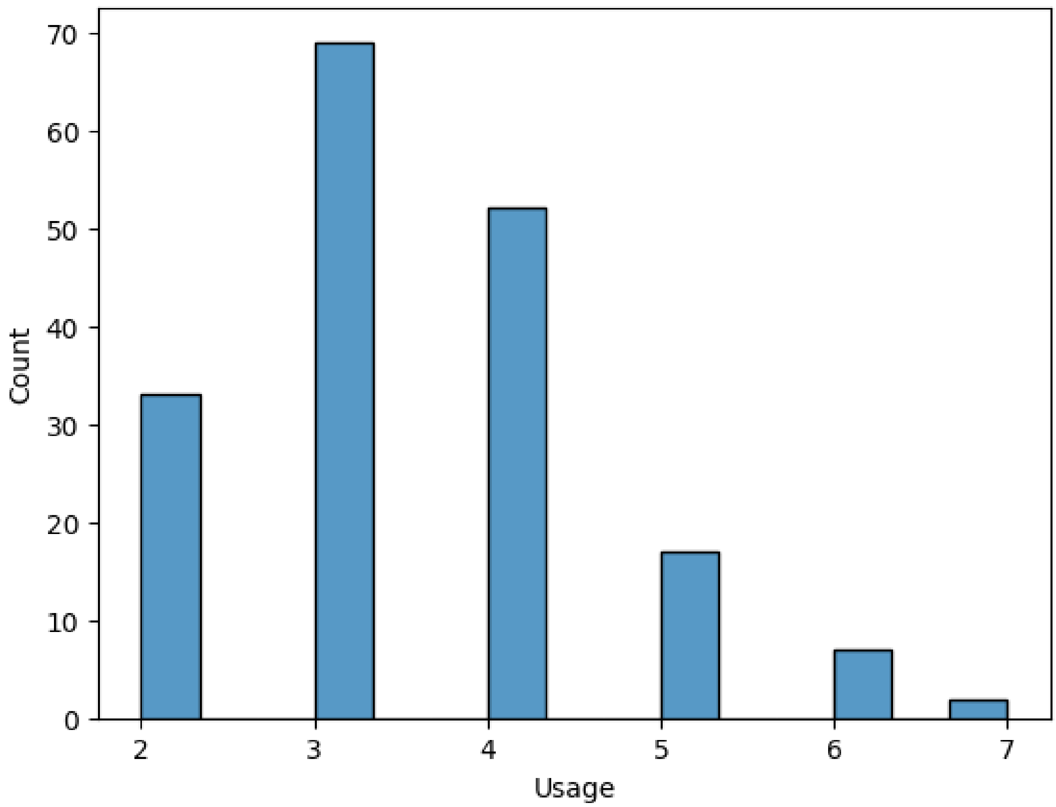
[113]:
```
# Education Analysis - histplot
sns.histplot(df_clipped.Education,kde=True,color='violet')
plt.show()
```



Customers who are educated for 16 years are high in purchasing the product

[114]:
```
# Usage Analysis - Histogram
sns.histplot(data=df_clipped,x='Usage')
```

[114]: <Axes: xlabel='Usage', ylabel='Count'>



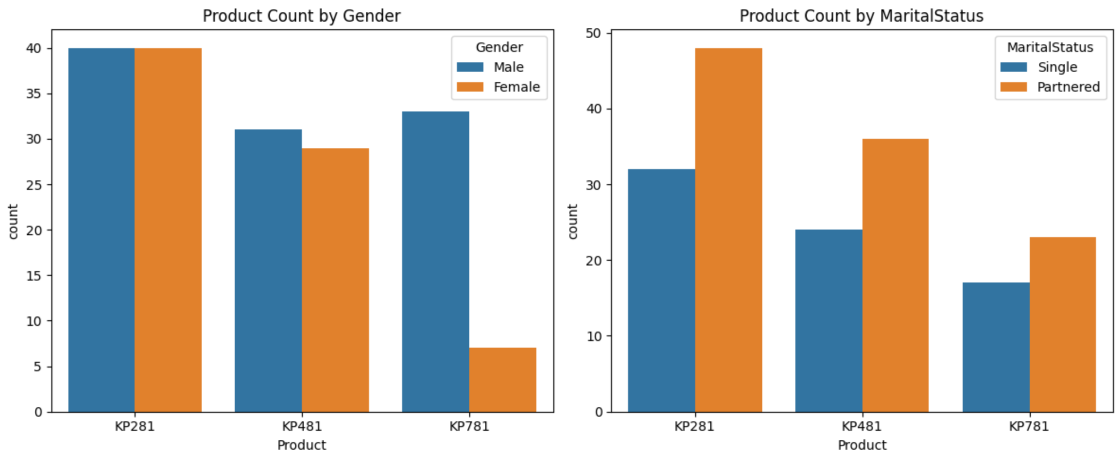
3 days per week is the most common usage among the customers

4 days and 2 days per week is the second and third highest usage among the customers

Very few customers use product 7 days per week

[115]:
```
import matplotlib.pyplot as plt
import seaborn as sns

#comparing the relationship with categorical variable with product

fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Count plot of Product by Gender
sns.countplot(x='Product', hue='Gender', data=df_clipped, ax=axes[0])
axes[0].set_title('Product Count by Gender')

# Count plot of Product by MaritalStatus
sns.countplot(x='Product', hue='MaritalStatus', data=df_clipped, ax=axes[1])
axes[1].set_title('Product Count by MaritalStatus')

plt.tight_layout()
```

```
plt.show()
```

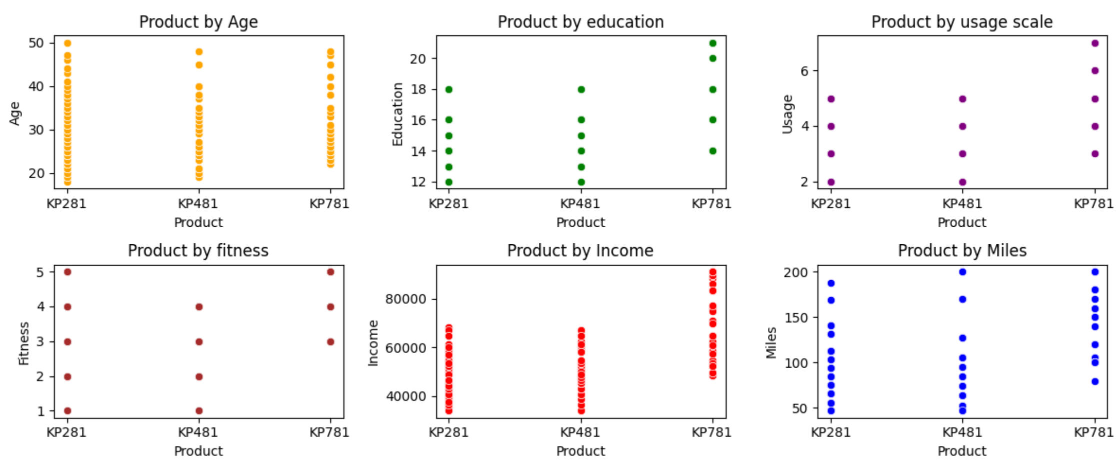


- The KP781 product shows the most significant difference in purchase patterns based on gender, with a clear preference among males
- Across all products, married individuals tend to purchase more than single individuals.

[116]:

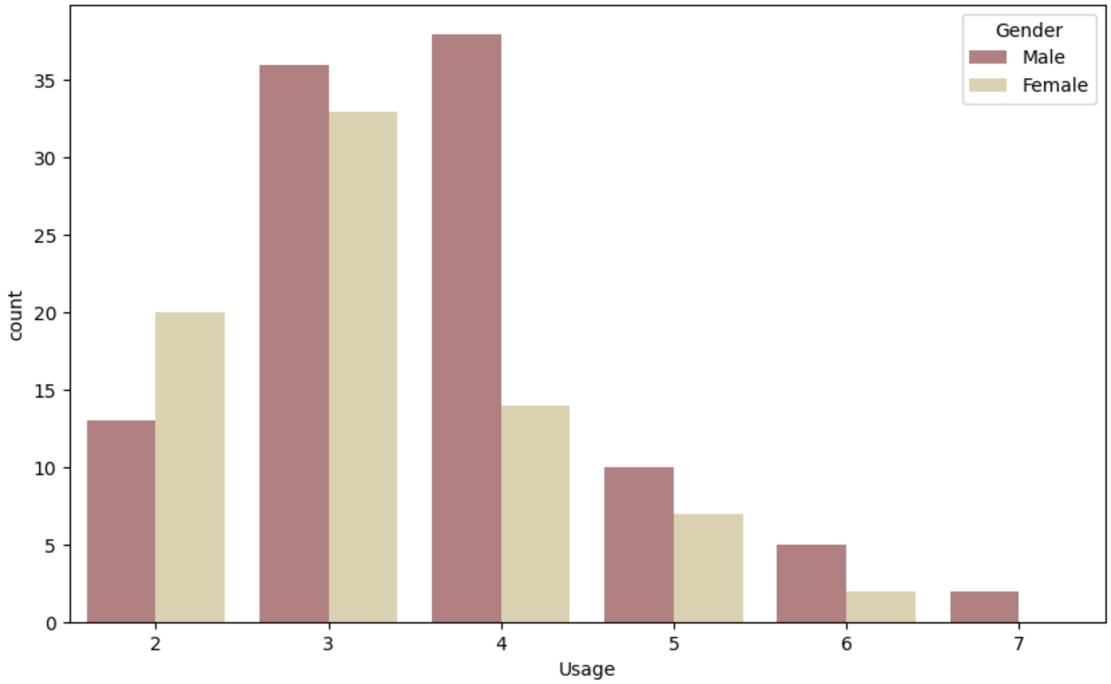
```
#comparing the relationship with numerical variable with product
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

fig, axes = plt.subplots(2, 3, figsize=(12, 5))

# Flatten the axes array for easy iteration
axes = axes.flatten()

# scatter plot of Product by Age
sns.scatterplot(x='Product', y='Age', data=df_clipped,␣
 ↪ax=axes[0],color='orange')
axes[0].set_title('Product by Age')

# scatter plot of Product by education
sns.scatterplot(x='Product', y='Education', data=df_clipped,␣
 ↪ax=axes[1],color='green')
axes[1].set_title('Product by education')

# scatter plot of Product by Usage scale
sns.scatterplot(x='Product', y='Usage', data=df_clipped,␣
 ↪ax=axes[2],color='purple')
axes[2].set_title('Product by usage scale')
```

```python
# scatter plot of Product by Fitness scale
sns.scatterplot(x='Product', y='Fitness', data=df_clipped, ax=axes[3],color='brown')
axes[3].set_title('Product by fitness')

# scatter plot of Product by Income
sns.scatterplot(x='Product', y='Income', data=df_clipped, ax=axes[4],color='red')
axes[4].set_title('Product by Income')

# scatter plot of Product by Miles
sns.scatterplot(x='Product', y='Miles', data=df_clipped, ax=axes[5],color='blue')
axes[5].set_title('Product by Miles')

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



- ** Age:** Appears to have some correlation with product, with KP781 buyers potentially skewing slightly older or having a wider age range.
- **Education:** Similarly, there might be a trend where buyers of higher-end products (like KP781) tend to have more years of education.
- ** Usage & Fitness**: These likely show a positive relationship with product. Buyers of more expensive treadmills are expected to have higher usage goals and consider themselves more fit.
- ** Income**: Income is a strong driver, with higher income individuals clearly favoring the KP781 model.
- ** Miles**: The target miles per week is also strongly correlated with product, with KP781 buyers targeting significantly more miles.

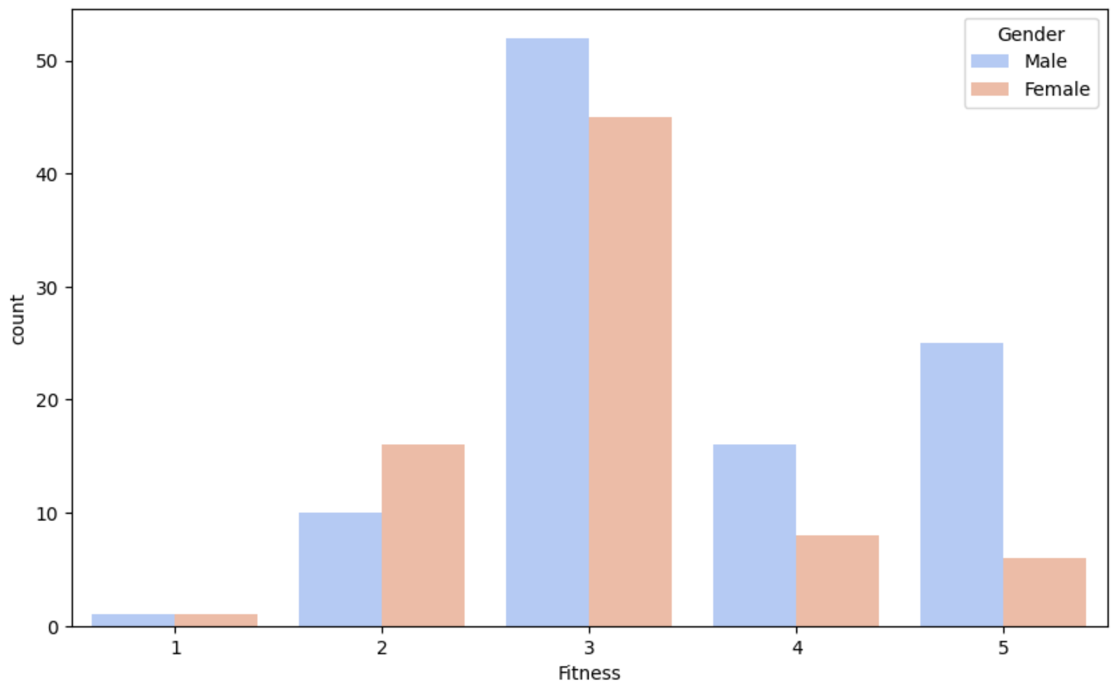
[117]:
```python
# Purchased product usage among Gender
plt.figure(figsize=(10,6))
sns.countplot(data=df_clipped,x='Usage',hue='Gender',palette='pink')
plt.show()
```



Male customers are using 4 days a week

0 Female customers uses 7 days a week

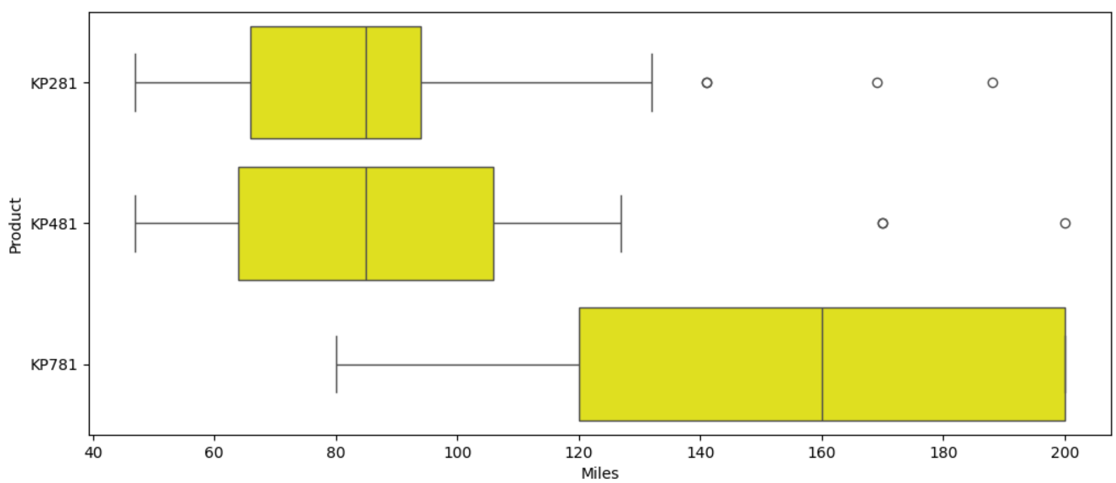
[118]:
```python
# Fitness rating among the customers categorised by Gender
plt.figure(figsize=(10,6))
sns.countplot(data=df_clipped,x='Fitness',hue='Gender',palette='coolwarm')
plt.show()
```



Average fitness scale for both Male and Female

Males are high in numbers for excellent 5 rating

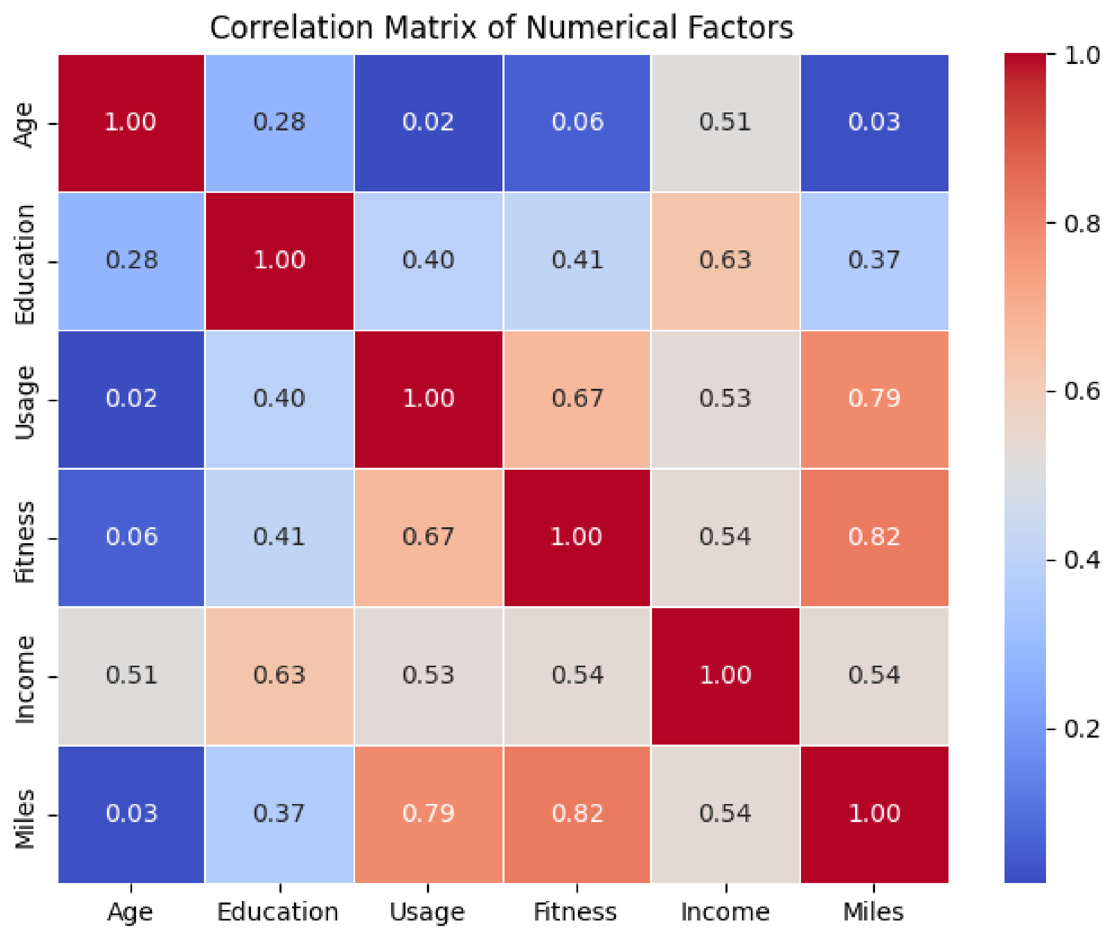
[119]:
```
# Miles with each product
plt.figure(figsize=(12,5))
sns.boxplot(x='Miles',y='Product',data=df_clipped,color='yellow')
plt.show()
```



KP781 product customers have covered more miles compared to other products

[120]:
```python
# Calculate the correlation matrix for numerical columns
numerical_cols = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
correlation_matrix = df_clipped[numerical_cols].corr()

# Print the correlation matrix
print("Correlation Matrix:")
print(correlation_matrix)

# Visualize the correlation matrix using a heatmap
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f",␣
↪linewidths=.5)
plt.title('Correlation Matrix of Numerical Factors')
plt.show()
```

```
Correlation Matrix:
                Age  Education     Usage   Fitness    Income     Miles
Age        1.000000   0.280496  0.015064  0.061105  0.512075  0.026168
Education  0.280496   1.000000  0.395155  0.410581  0.628908  0.367262
Usage      0.015064   0.395155  1.000000  0.668606  0.527707  0.786269
Fitness    0.061105   0.410581  0.668606  1.000000  0.535945  0.822393
Income     0.512075   0.628908  0.527707  0.535945  1.000000  0.537297
Miles      0.026168   0.367262  0.786269  0.822393  0.537297  1.000000
```



Strong Positive Correlations: 'Usage' and 'Fitness': Customers who plan to use the treadmill more often tend to rate themselves as more fit. This is a logical relationship.

'Usage' and 'Miles': Higher planned usage correlates with higher target mileage.

'Fitness' and 'Miles': Customers who rate themselves as more fit tend to have higher mileage goals. These three variables form a cluster related to fitness level and intended activity.

'Age' and 'Education': There is a moderate positive correlation, which is often observed in general populations (older individuals may have had more time to pursue education).

'Income' and 'Education': A moderate to strong positive correlation exists. Higher education levels are generally associated with higher earning potential.

'Income' and 'Miles': There's a notable positive correlation. Higher income likely enables customers to purchase higher-end treadmills suitable for covering more miles and potentially reflects a lifestyle that supports more intensive fitness activities.

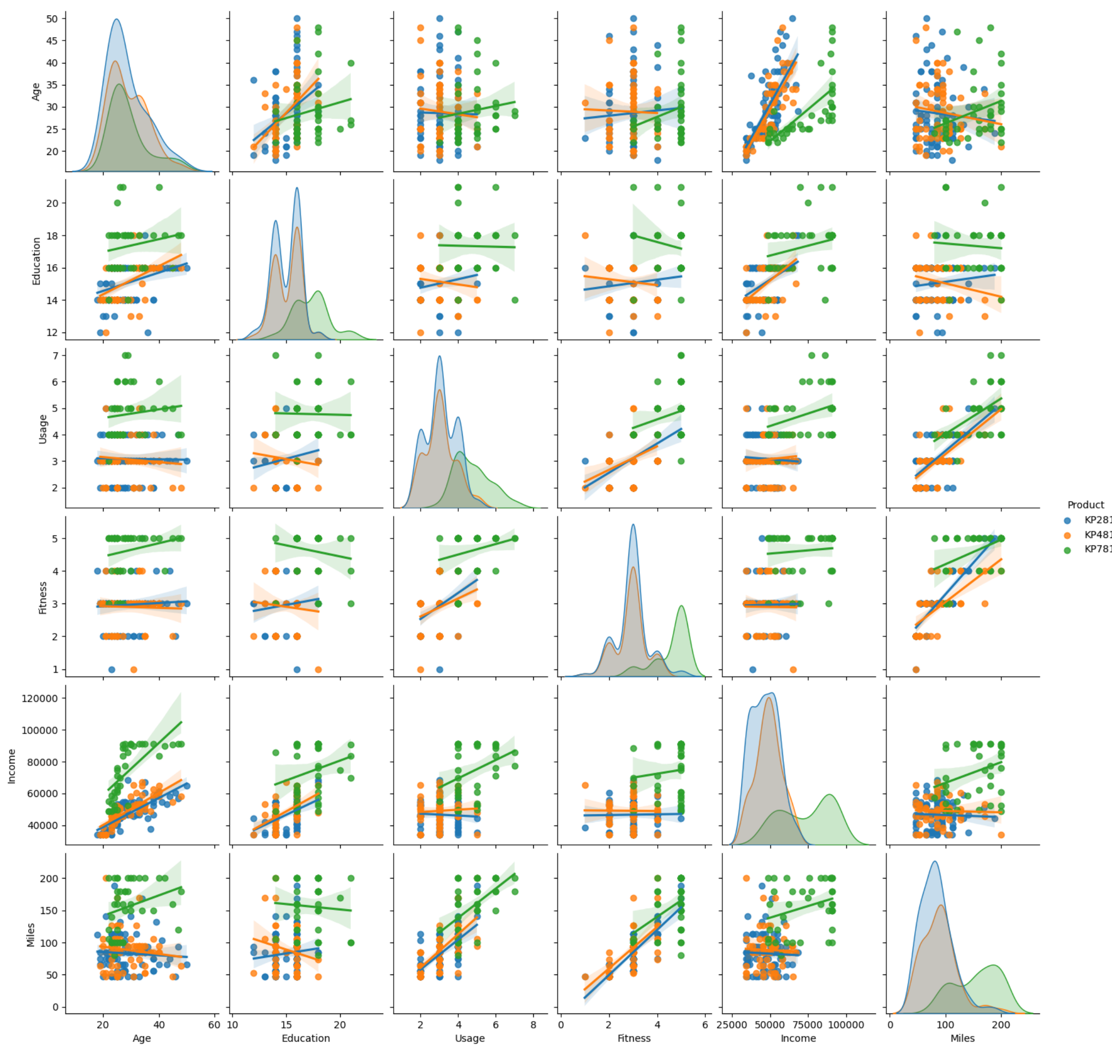
Moderate Correlations: 'Age' and 'Income': A moderate positive correlation, suggesting that income tends to increase with age, up to a certain point.

'Education' and 'Usage': A weak to moderate positive correlation. More educated individuals might have lifestyles that allow for more consistent treadmill usage.

'Education' and 'Fitness': A weak to moderate positive correlation. Similar to usage, education might indirectly influence fitness habits.

Weak or Near Zero Correlations: Relationships between 'Age' and the fitness/usage variables ('Usage', 'Fitness', 'Miles') appear weaker, suggesting age is not as strong a direct predictor of intended fitness activity compared to factors like income or education.

[121]:
```
# Product Analysis - Pair Plot
sns.pairplot(df_clipped,hue='Product',kind='reg')
plt.show()
```



KP281: This model appears to attract customers with lower income, education, and fitness levels, who typically target fewer miles and plan for less frequent usage.

KP481: This model generally falls between the KP281 and KP781, appealing to customers with moderate income, education, fitness, and usage/mileage goals.

KP781: This premium model is favored by customers with higher income, education, and fitness levels, who plan for more frequent usage and target significantly higher mileage.

[122]:
```
#Probability of product purchased with respect to gender
pd.
↪crosstab(index=df_clipped['Product'],columns=df_clipped['Gender'],margins=True,normalize=Tr
↪round(2)
```

[122]:
```
Gender   Female  Male   All
Product
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
```

The probability of a male purchasing a treadmill is 58%

The conditional probability of purchasing any model given the customer is male

- KP281 product - 22%
- KP481 Product - 17%
- KP781 Product - 18%

The probability of a female purchasing a treadmill is 42%

The conditional probability of purchasing any model given the customer is female

- KP281 product - 22%
- KP481 Product - 16%
- KP781 Product - 4%

[123]:
```
#converting to categorical values
df_category=df_clipped.copy()
df_category['Fitness_category'] = df.Fitness
df_category.head()
```

[123]:
```
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness    Income  \
0   KP281   18    Male         14        Single      3        4  34053.15
1   KP281   19    Male         15        Single      2        3  34053.15
2   KP281   19  Female         14     Partnered      4        3  34053.15
3   KP281   19    Male         12        Single      3        3  34053.15
4   KP281   20    Male         13     Partnered      4        2  35247.00

   Miles  Fitness_category
0    112                 4
1     75                 3
2     66                 3
3     85                 3
```

```
4     47                  2
```

[124]:
```
df_category['Fitness_category'] = df_category['Fitness_category'].replace({
    1: "Poor Shape",
    2: "Bad Shape",
    3: "Average Shape",
    4: "Good Shape",
    5: "Excellent Shape"
})
df_category.head()
```

[124]:
```
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness    Income  \
0   KP281   18    Male         14        Single      3        4  34053.15
1   KP281   19    Male         15        Single      2        3  34053.15
2   KP281   19  Female         14     Partnered      4        3  34053.15
3   KP281   19    Male         12        Single      3        3  34053.15
4   KP281   20    Male         13     Partnered      4        2  35247.00

   Miles Fitness_category
0    112       Good Shape
1     75    Average Shape
2     66    Average Shape
3     85    Average Shape
4     47        Bad Shape
```

[125]:
```
#creating bins for Age columns
# 0-21 -Teen
# 22-35 -Adult
# 36-45 -Middle Age
# 46-60 - Elder Age
df_category['age_group'] = df_category.Age
df_category.age_group = pd.cut(df_category.
 ↪age_group,bins=[0,21,35,45,60],labels=['Teen','Adult','Middle Aged','Elder'])
df_category.head()
```

[125]:
```
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness    Income  \
0   KP281   18    Male         14        Single      3        4  34053.15
1   KP281   19    Male         15        Single      2        3  34053.15
2   KP281   19  Female         14     Partnered      4        3  34053.15
3   KP281   19    Male         12        Single      3        3  34053.15
4   KP281   20    Male         13     Partnered      4        2  35247.00

   Miles Fitness_category age_group
0    112       Good Shape      Teen
1     75    Average Shape      Teen
2     66    Average Shape      Teen
3     85    Average Shape      Teen
```

```
4     47         Bad Shape       Teen
```

[126]:
```
#Probability of product purchased with respect to age group
pd.
 ↪crosstab(index=df_category['Product'],columns=df_category['age_group'],margins=True,normali
 ↪round(2)
```

[126]:

age_group	Teen	Adult	Middle Aged	Elder	All
Product					
KP281	0.06	0.31	0.06	0.02	0.44
KP481	0.04	0.25	0.04	0.01	0.33
KP781	0.00	0.19	0.02	0.01	0.22
All	0.09	0.75	0.12	0.03	1.00

The probability of a Adult age group of purchasing a treadmill is higher 75%

The conditional probability of purchasing any model given the customer age_group is Adult

KP281 product - 31%

KP481 Product - 25%

KP781 Product - 19%

The probability of a elder and teen age group purchasing a treadmill is very low 3% and 9%

The conditional probability of purchasing any model given the age group is Middle aged

KP281 product - 6%

KP481 Product - 4%

KP781 Product - 2%

[127]:
```
#Probability of product and fitness scale
pd.
 ↪crosstab(index=df_category['Product'],columns=df_category['Fitness_category'],margins=True,
 ↪round(2)
```

[127]:

Fitness_category	Average Shape	Bad Shape	Excellent Shape	Good Shape	\
Product					
KP281	0.30	0.08	0.01	0.05	
KP481	0.22	0.07	0.00	0.04	
KP781	0.02	0.00	0.16	0.04	
All	0.54	0.14	0.17	0.13	

Fitness_category	Poor Shape	All
Product		
KP281	0.01	0.44
KP481	0.01	0.33
KP781	0.00	0.22
All	0.01	1.00

The Probability of Excellent shape is higher in KP781 model 16%

The overall fitness probability is high in Average shape 54%

The conditional probability of using any model given the average_shape fitness category

KP281 product - 30%

KP481 Product - 22%

KP781 Product - 2%

[128]:
```
#Probability of buying the product with their usage
pd.
 ↪crosstab(index=df_clipped['Product'],columns=df_clipped['Usage'],margins=True,normalize=Tru
 ↪round(2)
```

[128]:

Usage Product	2	3	4	5	6	7	All
KP281	0.11	0.21	0.12	0.01	0.00	0.00	0.44
KP481	0.08	0.17	0.07	0.02	0.00	0.00	0.33
KP781	0.00	0.01	0.10	0.07	0.04	0.01	0.22
All	0.18	0.38	0.29	0.09	0.04	0.01	1.00

The 3 times a week usage customer has high probability the purchase the product 38%

The 7 times a week usage customer has lowest probability the purchase the product 1%

The probability of purchasing the product by a customer with usuage of 3 times per week 38%

The conditional probability of purchasing the treadmill model given that the customer has Usage 3 per week is

KP281 - 21%

KP481 - 17%

KP781 - 1%

[129]:
```
#Probability of buying the product with respect to the Gender
pd.
 ↪crosstab(index=df_clipped['Product'],columns=df_clipped['Gender'],margins=True,normalize=Tr
 ↪round(2)
```

[129]:

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

Male customers have higher probability in buying the product 58%

Probability of buying the each product given the total probability of female customer

KP281 - 22%

KP481 - 16%

KP781 - 4%

[130]:
```
#Probability of buying the product with respect to the MaritalStatus
pd.
 ↪crosstab(index=df_clipped['Product'],columns=df_clipped['MaritalStatus'],margins=True,norma
 ↪round(2)
```

[130]:

MaritalStatus Product	Partnered	Single	All
KP281	0.27	0.18	0.44
KP481	0.20	0.13	0.33
KP781	0.13	0.09	0.22
All	0.59	0.41	1.00

Partnered(couples) have the higher probability in purchasing the product 59%

Probability of purchasing each product given the customer is single

KP281 - 18%

KP481 - 13%

KP781 - 9%

Customer Profiling for each product

KP281

Income: Lower income range.

Education: Lower education levels.

Fitness: Lower fitness levels.

Usage: Target fewer miles and plan for less frequent usage.

Gender: Purchased by both males and females, with a slightly higher proportion of females compared to the other models.

Marital Status: Purchased by both single and partnered individuals.

Age: Purchased by a wider age range, including a notable proportion of younger adults.

KP481

Income: Moderate income range.

Education: Moderate education levels.

Fitness: Moderate fitness levels.

Usage: Generally falls between the KP281 and KP781 for usage frequency and target mileage.

Gender: Purchased by both males and females.

Marital Status: Purchased by both single and partnered individuals.

Age: Purchased by a range of ages, with a concentration in the adult age group.

KP781

Income: Higher income range.

Education: Higher education levels.

Fitness: Higher fitness levels, with a significant number of customers rating themselves as "Excellent Shape."

Usage: Plan for more frequent usage and target significantly higher mileage.

Gender: Favored by males.

Marital Status: Purchased by both single and partnered individuals, with a higher proportion of partnered individuals.

Age: Tends to attract slightly older adults and a wider age range compared to the KP281, but with a strong presence in the adult age group.

Recommendation

For the KP281, focus marketing towards a broader demographic, including individuals with lower income and education, who are looking for a basic treadmill for less frequent use. Highlight affordability and ease of use.

For the KP481, target the middle ground with messaging that emphasizes value and versatility for moderately active users.

For the KP781, direct marketing efforts towards higher-income, more educated males who prioritize high fitness levels and intensive workout routines. Highlight performance, advanced features, and durability.

KP781 product should be promotted using influencers and other international atheletes.

Given the target customer's age, education level, and income, it's important to offer the KP281 and KP481 Treadmill at an affordable price point.

For the KP281, consider placement in mass retail stores and online platforms that cater to a wider audience.

For the KP781, focus on specialized fitness stores, online channels targeting serious athletes