

```
In [8]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Problem statement:

This project is to identify the characteristics of the target audience for each type of treadmill offered by AeroFit, in order to provide better recommendations to new customers, Team at AeroFit wants to investigate whether there are differences across the products with respect to customer characteristics, such as age, gender, income level, education level, and marital status. By analyzing these characteristics, we can gain insights into the preferences and needs of the customers, and tailor the marketing and sales strategies accordingly.

Basic Metrics:

To analyze the basic metrics for the given dataframe and to address the business problem of identifying the characteristics of the target audience for each type of treadmill offered by AeroFit, we can perform the following steps: ¶

1. Identify the overall size of the data frame by using the shape attribute of pandas.
2. Analyze the data types and missing values in the data frame using info() method.
3. Detect Outliers.
4. Identify the basic statistical measures like mean, median, standard deviation, minimum, maximum values for the numerical columns using describe() method.
5. Analyze the distribution of the numerical columns using histograms and boxplots.
6. Check if features like marital status, age have any effect on the product purchased.
7. Customer Profiling - Categorization of users.
8. Probability- marginal, conditional probability.
9. Some recommendations and actionable insights, based on the inferences.

```
In [10]: df=pd.read_csv('/Users/praneetcb/Downloads/aerofit_treadmill.csv')
```

Non-graphical visualization

```
In [14]: df.head()
```

```
Out[14]:
```

	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 [12]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Product         180 non-null   object
1   Age             180 non-null   int64
2   Gender          180 non-null   object
3   Education        180 non-null   int64
4   MaritalStatus   180 non-null   object
5   Usage           180 non-null   int64
6   Fitness         180 non-null   int64
7   Income          180 non-null   int64
8   Miles           180 non-null   int64
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

```
In [13]: df.value_counts()
```

```
Out[13]:
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	In
	come	Miles						
	KP281	18	Male	14	Single	3	4	29
	562	112	1					
	KP481	30	Female	13	Single	4	3	46
	617	106	1					
		31	Female	16	Partnered	2	3	51
	165	64	1					
				18	Single	2	1	65
	220	21	1					
			Male	16	Partnered	3	3	52
	302	95	1					
	..							
	KP281	34	Female	16	Single	2	2	52
	302	66	1					
			Male	16	Single	4	5	51
	165	169	1					
		35	Female	16	Partnered	3	3	60
	261	94	1					
				18	Single	3	3	67
	083	85	1					
	KP781	48	Male	18	Partnered	4	5	95
	508	180	1					

Name: count, Length: 180, dtype: int64

```
In [16]: df.nunique()
```

```
Out[16]:
```

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

dtype: int64

```
In [17]: df.describe()
```

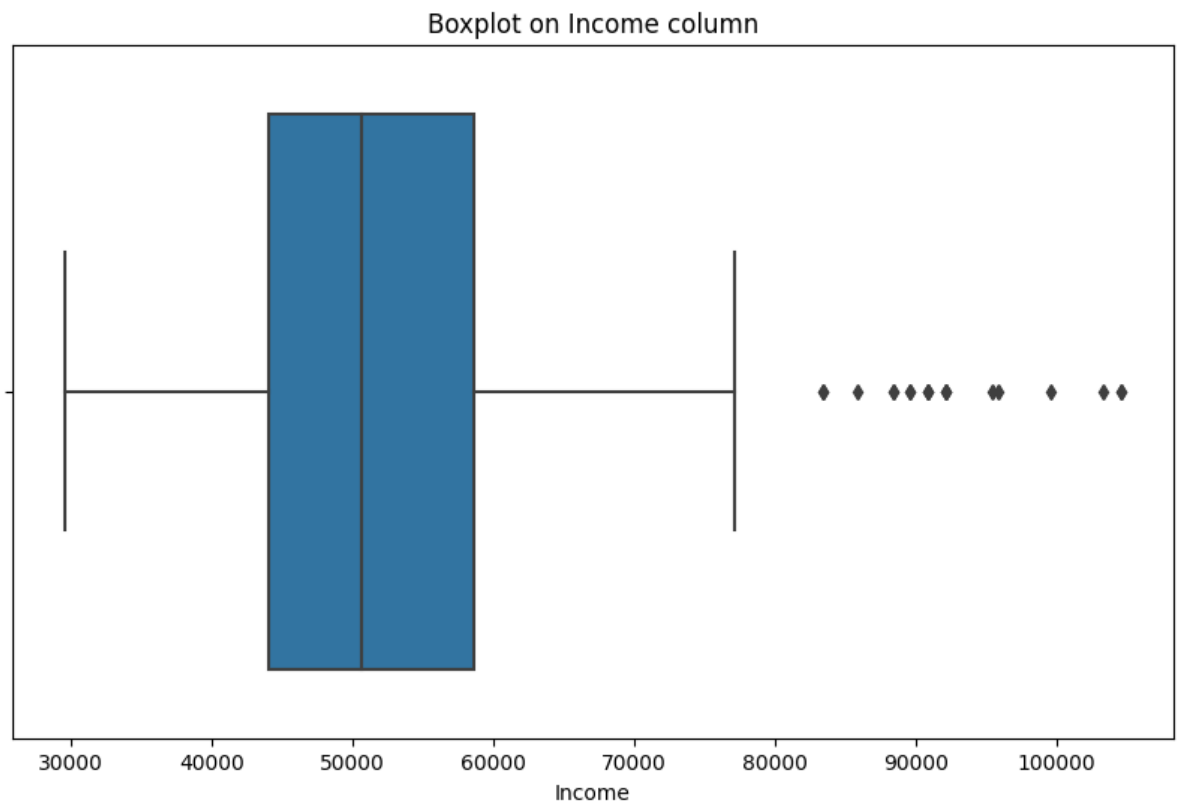
```
Out[17]:
```

	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

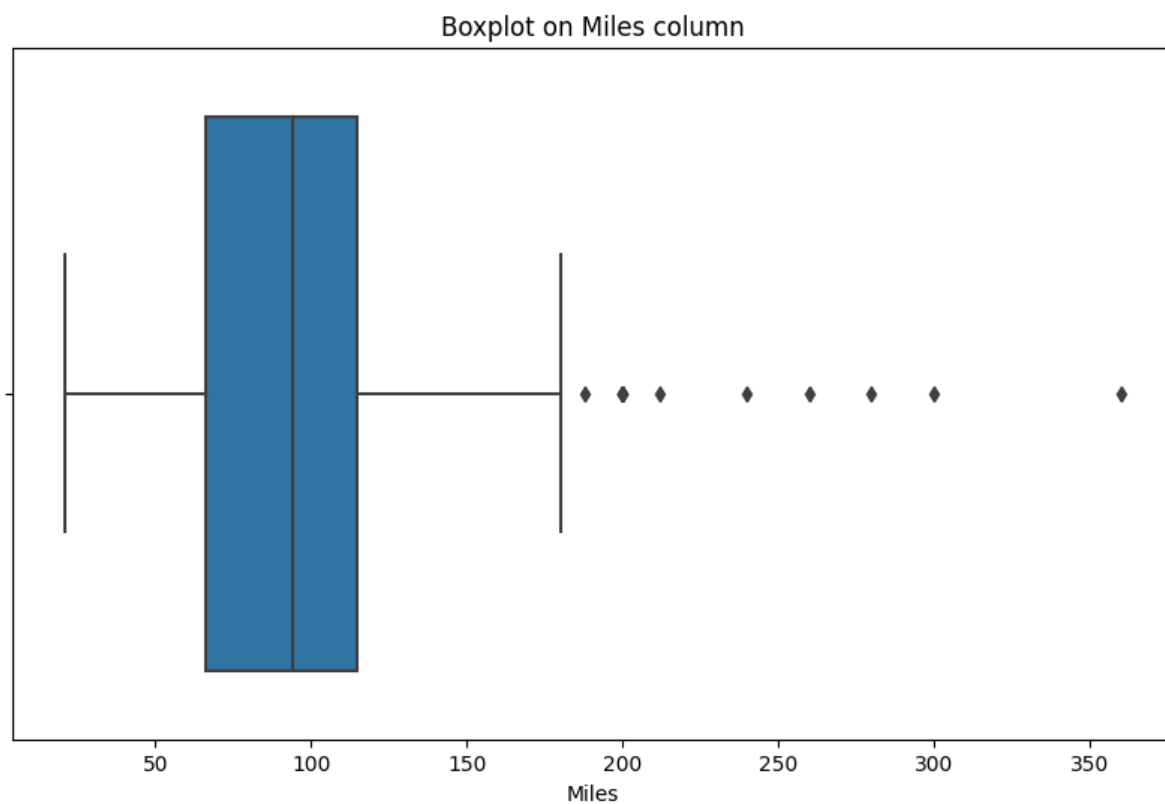
Graphical visualization

```
In [47]: # Boxplots to visualize outliers in the Income column
```

```
fig, ax = plt.subplots(figsize=(10,6))  
sns.boxplot(data=df, x='Income',ax=ax)  
ax.set_title("Boxplot on Income column")  
plt.show()
```

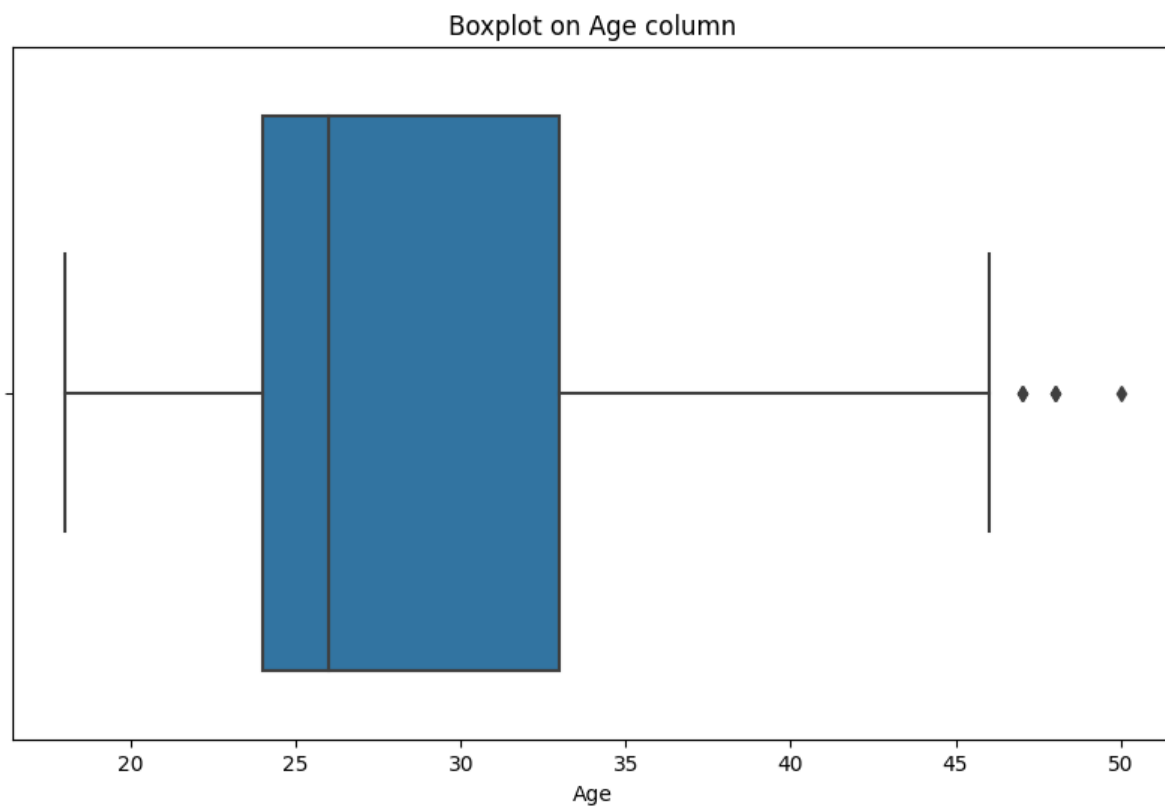


```
In [111]: # Boxplots to visualize outliers in the Miles column,  
## Miles – The average number of miles the customer expects to walk  
  
fig, ax = plt.subplots(figsize=(10,6))  
sns.boxplot(data=df, x='Miles',ax=ax)  
ax.set_title("Boxplot on Miles column")  
plt.show()
```



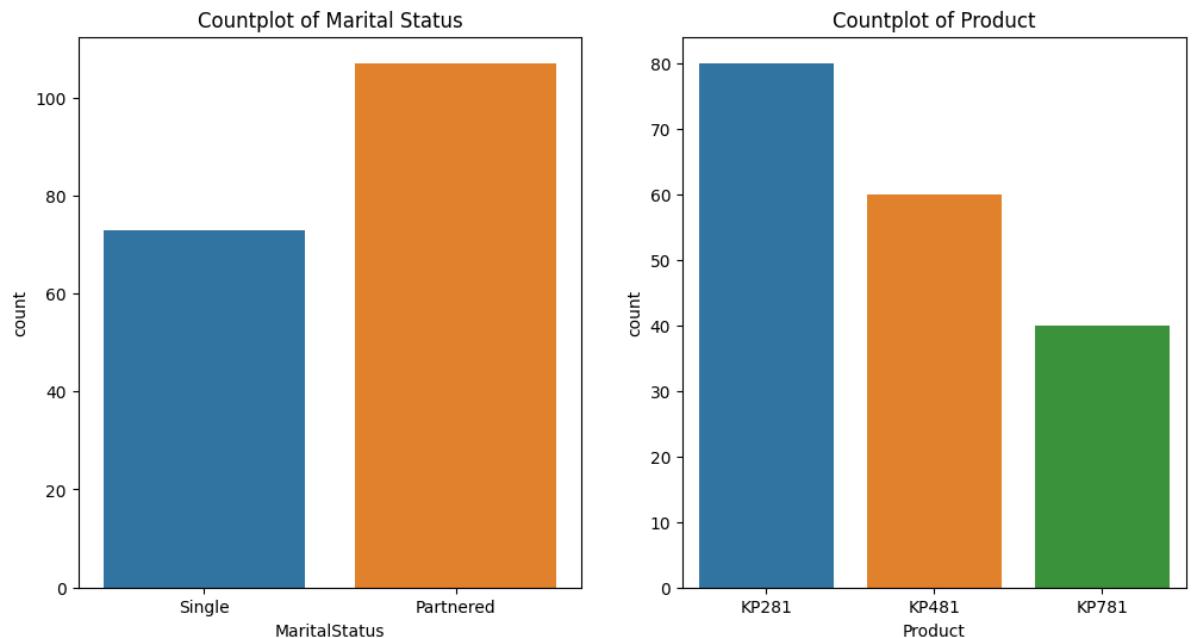
```
In [113]: # Boxplots to visualize outliers in the Age column
```

```
fig, ax = plt.subplots(figsize=(10,6))  
sns.boxplot(data=df, x='Age',ax=ax)  
ax.set_title("Boxplot on Age column")  
plt.show()
```



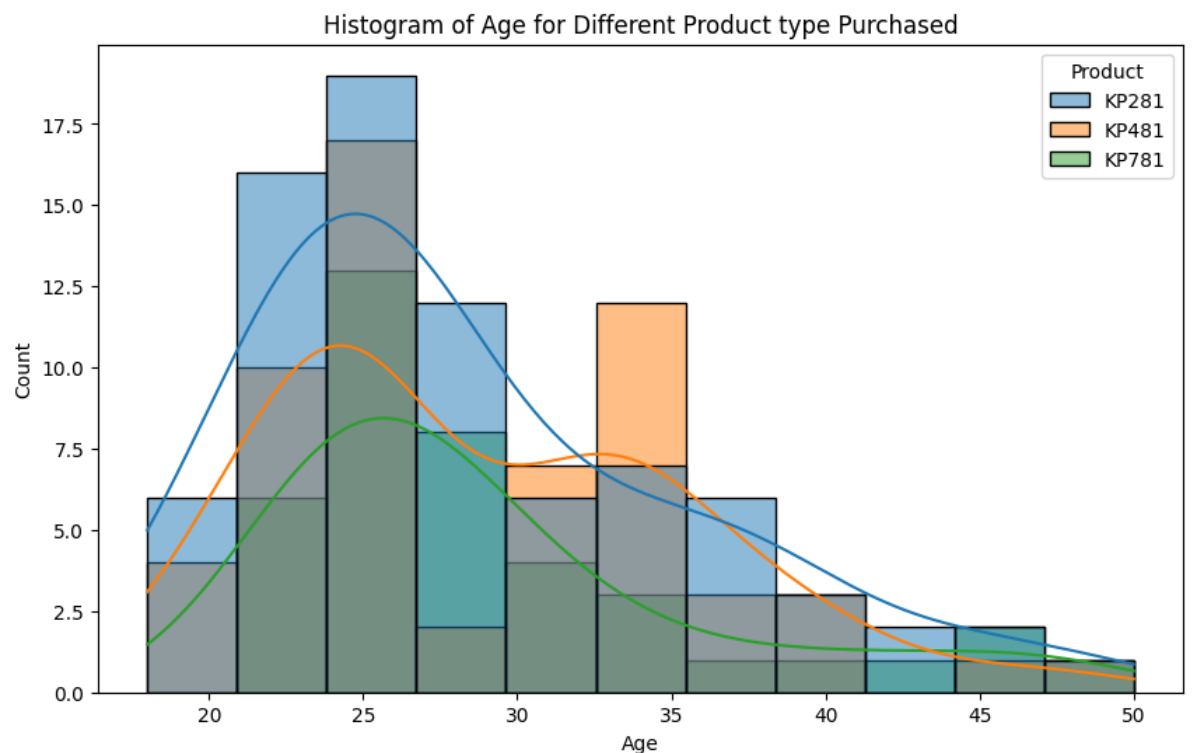
In [44]: *# Countplots of marital status and product purchased*

```
fig, ax = plt.subplots(1, 2, figsize=(12,6))
sns.countplot(x='MaritalStatus', data=df, ax=ax[0])
sns.countplot(x='Product', data=df, ax=ax[1])
ax[0].set_title("Countplot of Marital Status")
ax[1].set_title("Countplot of Product")
plt.show()
```



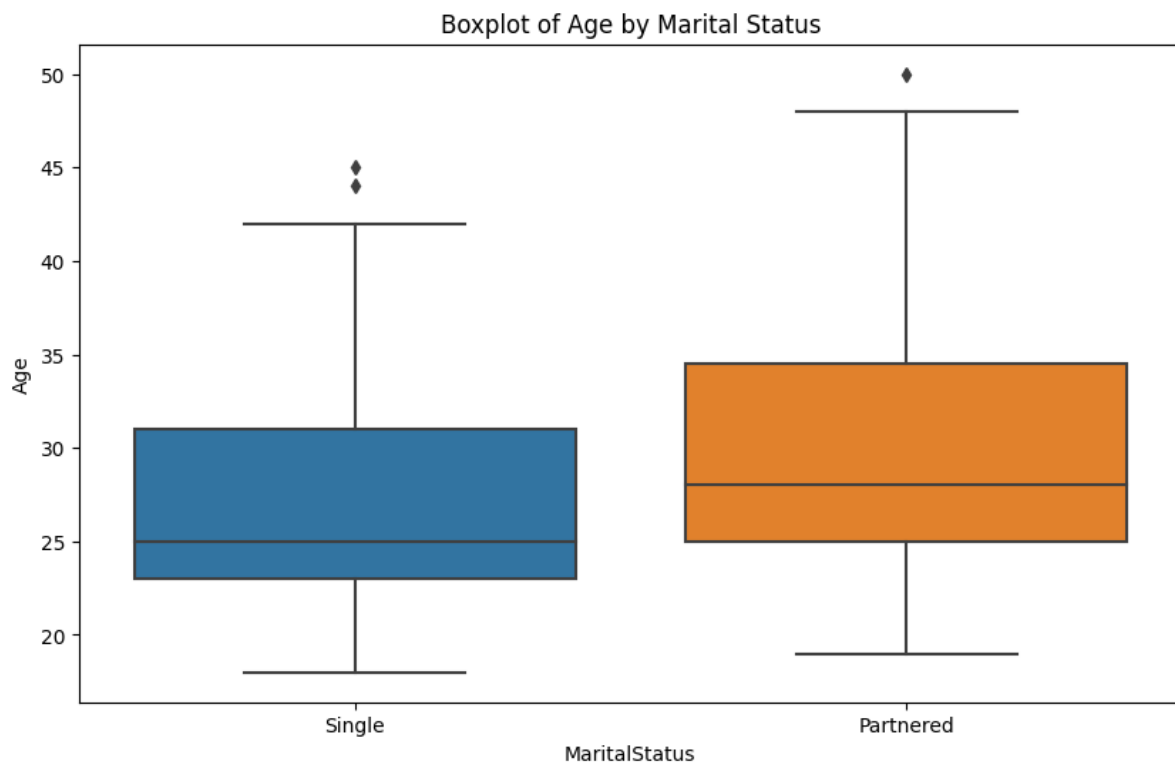
In [27]: *# Plot histograms of age for each product purchased category*

```
fig, ax = plt.subplots(figsize=(10,6))
sns.histplot(x='Age', data=df, hue='Product', ax=ax, kde=True)
ax.set_title("Histogram of Age for Different Product type Purchased")
plt.show()
```



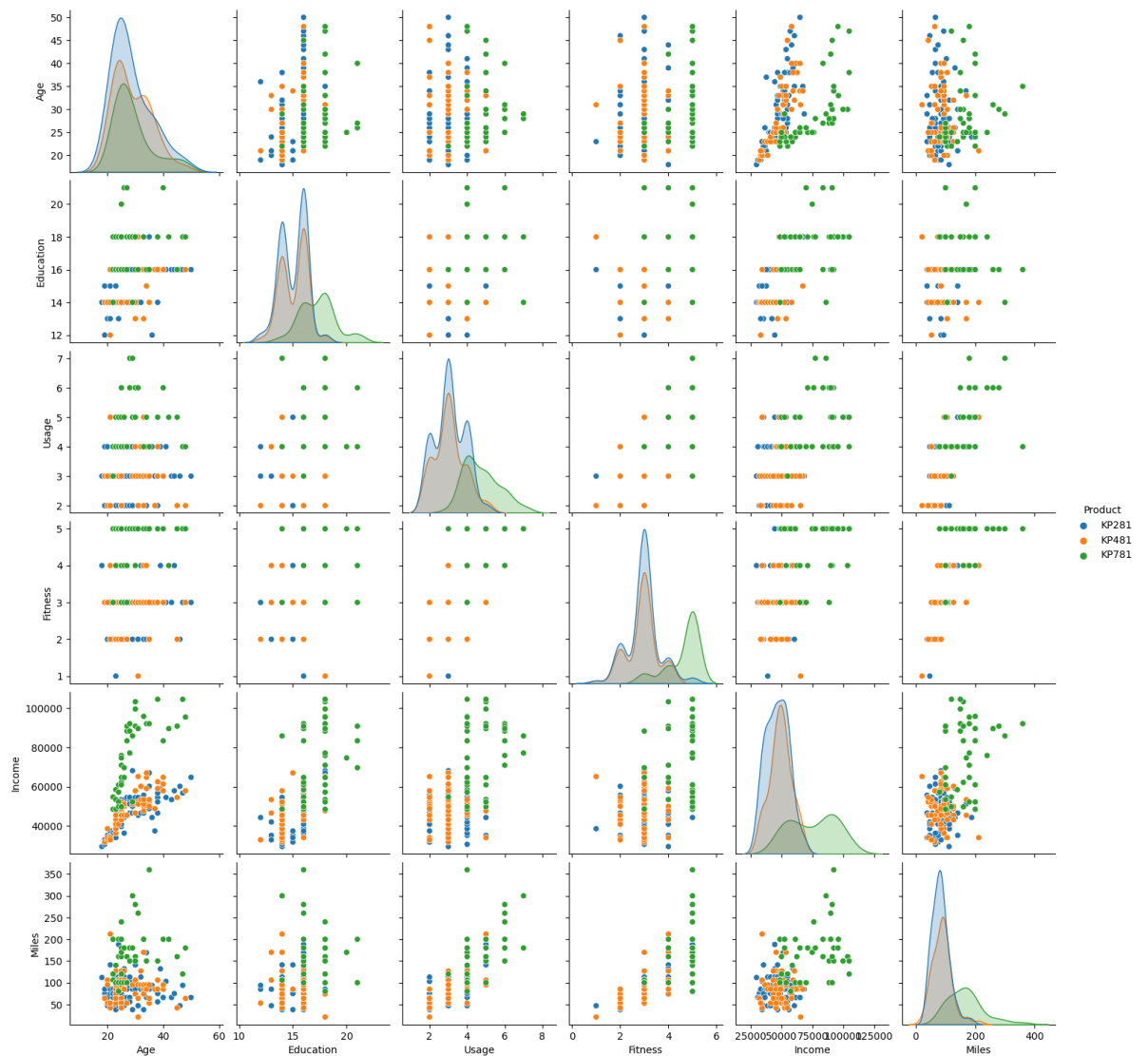
```
In [29]: # Plot boxplots of age for each marital status category
```

```
fig, ax = plt.subplots(figsize=(10,6))  
sns.boxplot(x='MaritalStatus', y='Age', data=df, ax=ax)  
ax.set_title("Boxplot of Age by Marital Status")  
plt.show()
```



Pairplot of the Data


```
In [83]: # Plot a pairplot of the data
sns.pairplot(df, hue='Product')
plt.show()
```



```
In [ ]:
```

Marginal probability on the product

In [125]: *# Marginal probability of each product*

```
marginal_prob = pd.crosstab(index=df['Product'], columns='count', n
round(marginal_prob,2)
```

Out[125]:

	col_0	count	All
Product			
KP281		44.44	44.44
KP481		33.33	33.33
KP781		22.22	22.22
All		100.00	100.00

Conditional probability of purchasing a product given marital status

In [107]: *# Conditional probability of purchasing a product given marital sta*

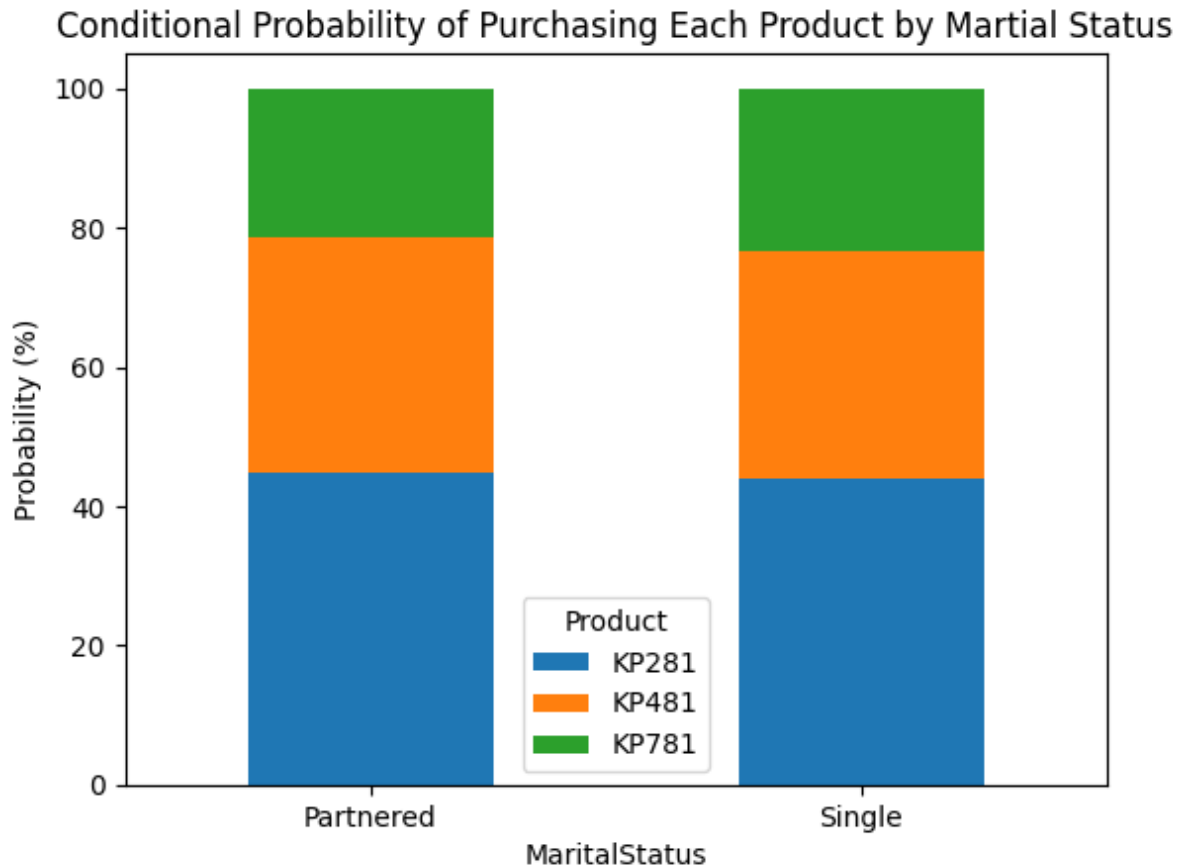
```
conditional_prob_m = pd.crosstab(index=df['MaritalStatus'], columns
round(conditional_prob_m,2)
```

Out[107]:

	Product	KP281	KP481	KP781
MaritalStatus				
Partnered		44.86	33.64	21.50
Single		43.84	32.88	23.29

In [108]: *#Stacked Bar plot to show Probability % with Martial status*

```
conditional_prob_m.plot(kind='bar', stacked=True)
plt.title('Conditional Probability of Purchasing Each Product by Ma
plt.ylabel('Probability (%)')
plt.xticks(rotation=0)
plt.legend(title='Product')
plt.show()
```



Conditional probability of purchasing a product given Gender

In [102]: *# Conditional probability of purchasing a product given Gender*

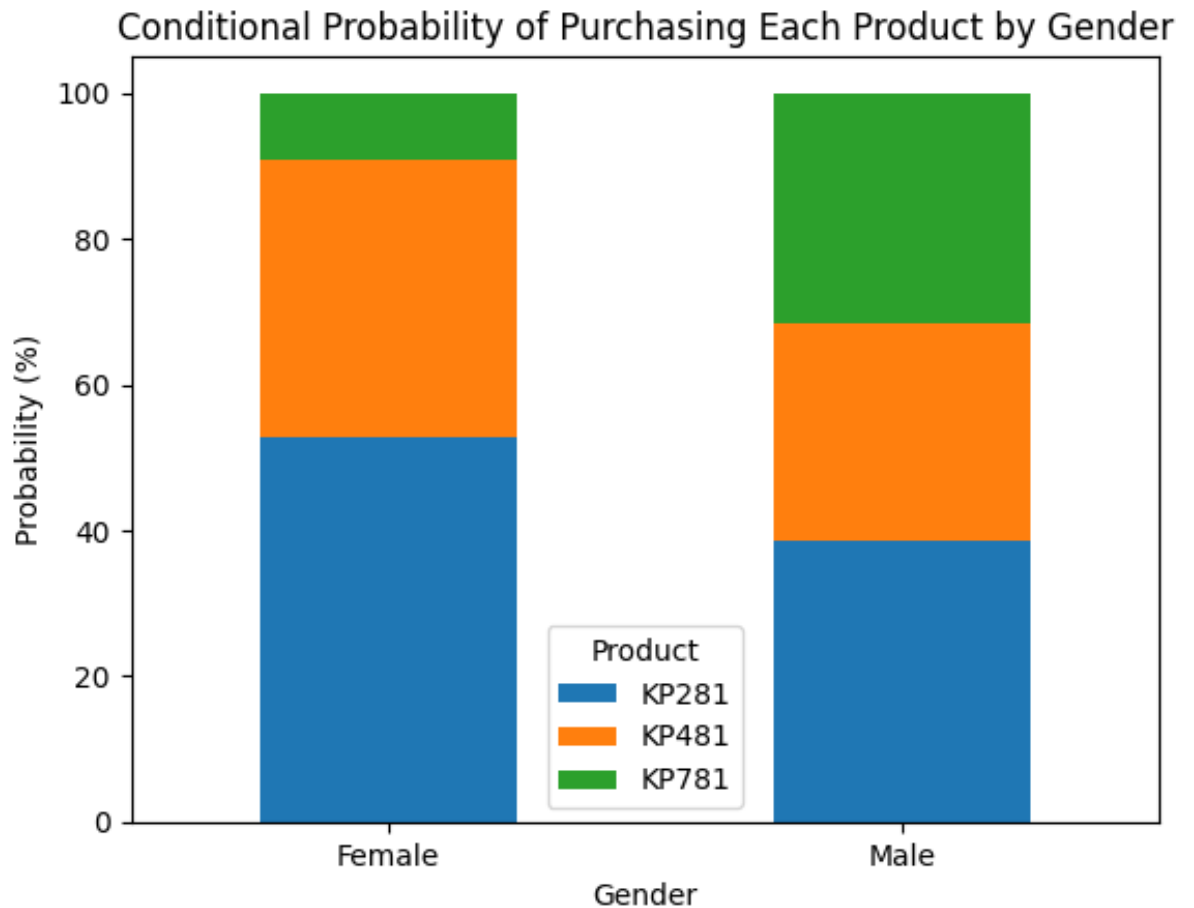
```
conditional_prob = pd.crosstab(index=df['Gender'], columns=df['Prod
round(conditional_prob,2)
```

Out[102]:

Product	KP281	KP481	KP781
Gender			
Female	52.63	38.16	9.21
Male	38.46	29.81	31.73

In [103]: *#Stacked Bar plot to show Probability % with Gender*

```
conditional_prob.plot(kind='bar', stacked=True)
plt.title('Conditional Probability of Purchasing Each Product by Ge
plt.xlabel('Gender')
plt.ylabel('Probability (%)')
plt.xticks(rotation=0)
plt.legend(title='Product')
plt.show()
```



Conditional probability of purchasing a product given the customer's age group

```
In [75]: # Create age groups
df['age_group'] = pd.cut(df['Age'], bins=[18, 25, 35, 50], labels=[

# Calculate the conditional probability of purchasing a product giv
conditional_prob_age = pd.crosstab(index=df['age_group'], columns=d

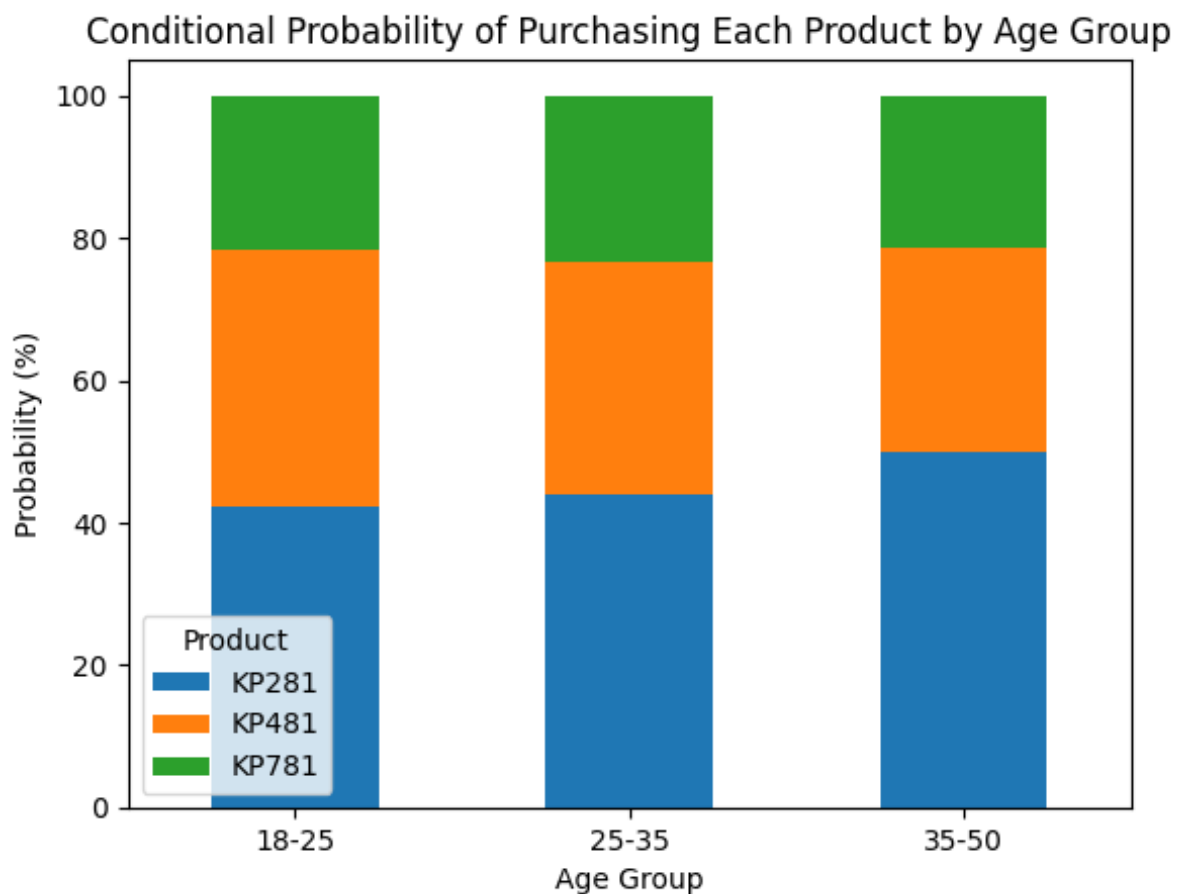
round((conditional_prob_age),2)
```

Out [75]:

Product	KP281	KP481	KP781
age_group			
18-25	42.31	35.90	21.79
25-35	43.84	32.88	23.29
35-50	50.00	28.57	21.43

```
In [95]: #Stacked Bar plot to show Probablity % with Age group

conditional_prob_age.plot(kind='bar', stacked=True)
plt.title('Conditional Probability of Purchasing Each Product by Ag
plt.xlabel('Age Group')
plt.ylabel('Probability (%)')
plt.xticks(rotation=0)
plt.legend(title='Product')
plt.show()
```



Conditional probability of purchasing a product given the customer's income group

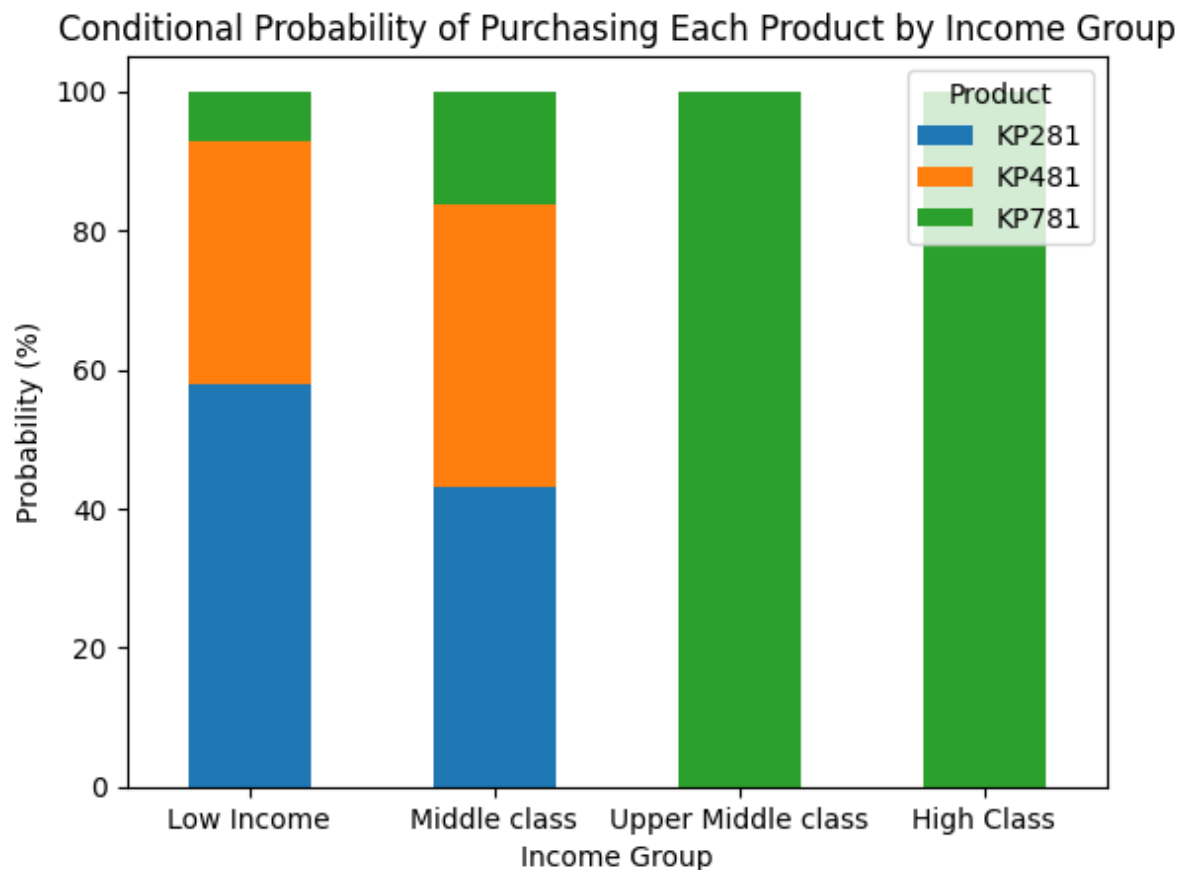
```
In [80]: #Create Income Group
df['Income_group'] = pd.cut(df['Income'], bins=[35000, 50000, 70000,
#Calculate the conditional probability of purchasing a product given
conditional_prob_income = pd.crosstab(index=df['Income_group'], col
round((conditional_prob_income),2)
```

Out [80]:

	Product		
	KP281	KP481	KP781
Income_group			
Low Income	57.97	34.78	7.25
Middle class	43.24	40.54	16.22
Upper Middle class	0.00	0.00	100.00
High Class	0.00	0.00	100.00

In [96]: *#Stacked Bar plot to show Probablity % with Income group*

```
conditional_prob_income.plot(kind='bar', stacked=True)  
plt.title('Conditional Probability of Purchasing Each Product by In  
plt.xlabel('Income Group')  
plt.ylabel('Probability (%)')  
plt.xticks(rotation=0)  
plt.legend(title='Product')  
plt.show()
```

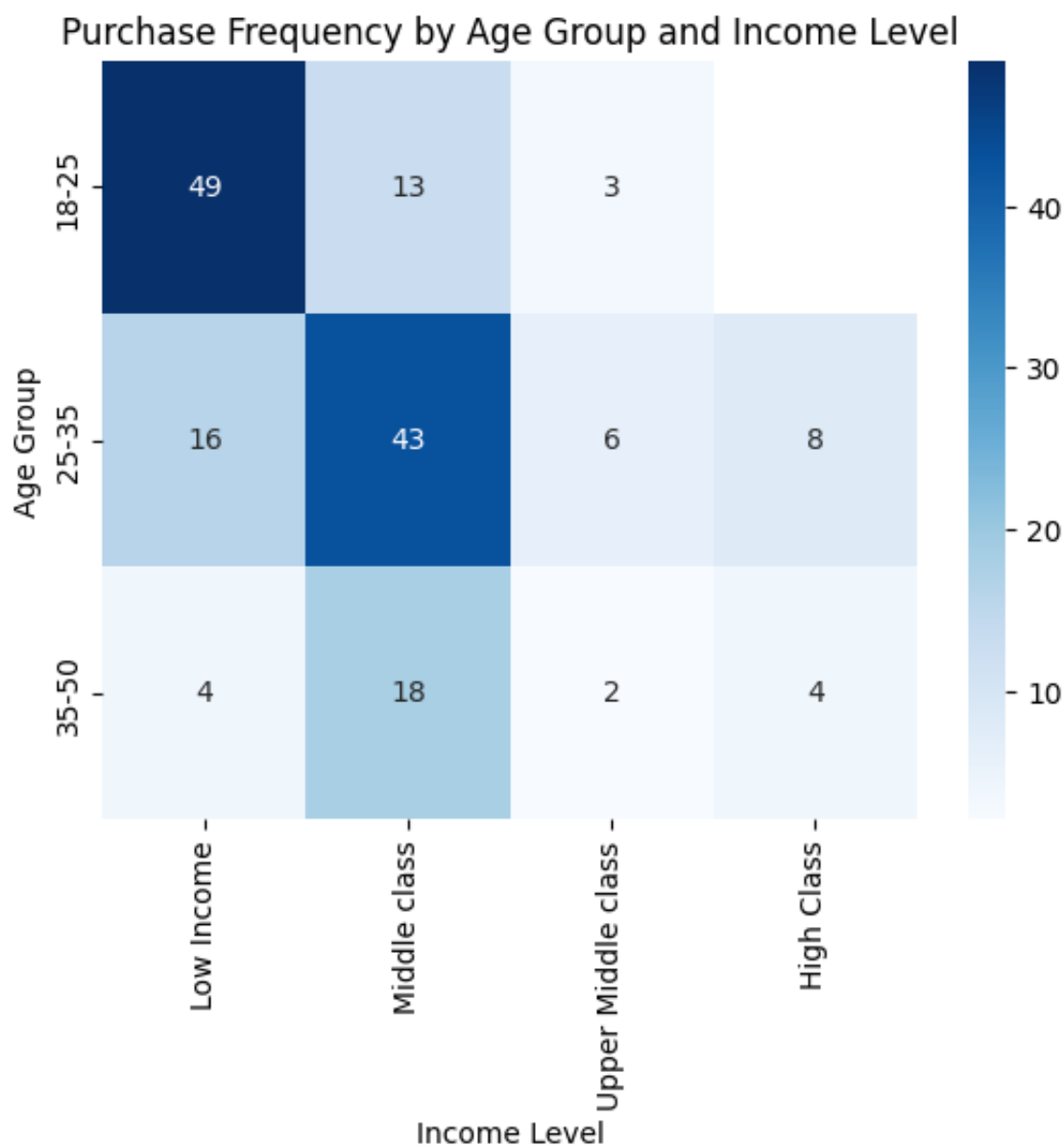


```
In [117]: # A heat map can be created to visualize the frequency of product
pivot_table1 = pd.pivot_table(df, values='Product', index='age_group', columns='income_level')

# Create the heat map using seaborn
sns.heatmap(pivot_table1, cmap='Blues', annot=True, fmt='g')

# Add title and axis labels
plt.title('Purchase Frequency by Age Group and Income Level')
plt.xlabel('Income Level')
plt.ylabel('Age Group')

# Show the plot
plt.show()
```



Based on the analysis of the AeroFit Treadmill dataset, here are some strong recommendations and actionable insights that can be derived:

Increase marketing efforts for KP281: The product KP281 has the highest purchase frequency among all products, so it would be beneficial to increase marketing efforts for this product to target the customer segments that are more likely to purchase it.

Target younger age groups: The highest frequency of purchases is observed in the youngest age group (18 -35 years). This indicates that AeroFit should target younger age groups in their marketing efforts to increase the overall sales of their products.

Increase product offerings for higher income levels: The purchase frequency is highest for the highest income level (Upper middle class and High class), which suggests that AeroFit should focus on offering more premium products or services that cater to customers in this income bracket. This could involve the development of higher-end treadmill models or add-on services, such as personal training or virtual coaching.

We see that Married couples have the highest purchase frequency, which suggests that AeroFit should create marketing campaigns that target married couples.

Offer financing options: The average income of the customers is relatively high, but the range of incomes is wide, which suggests that offering financing options may increase sales.

Target customers who value fitness: The data shows that customers who rate their fitness level as "Very Fit" are more likely to purchase a treadmill. Therefore, AeroFit can create targeted marketing campaigns that appeal to fitness enthusiasts to attract this customer segment.

Provide customer support: Providing excellent customer support is crucial for building long-term relationships with customers.

In []:

In []:

In []:

