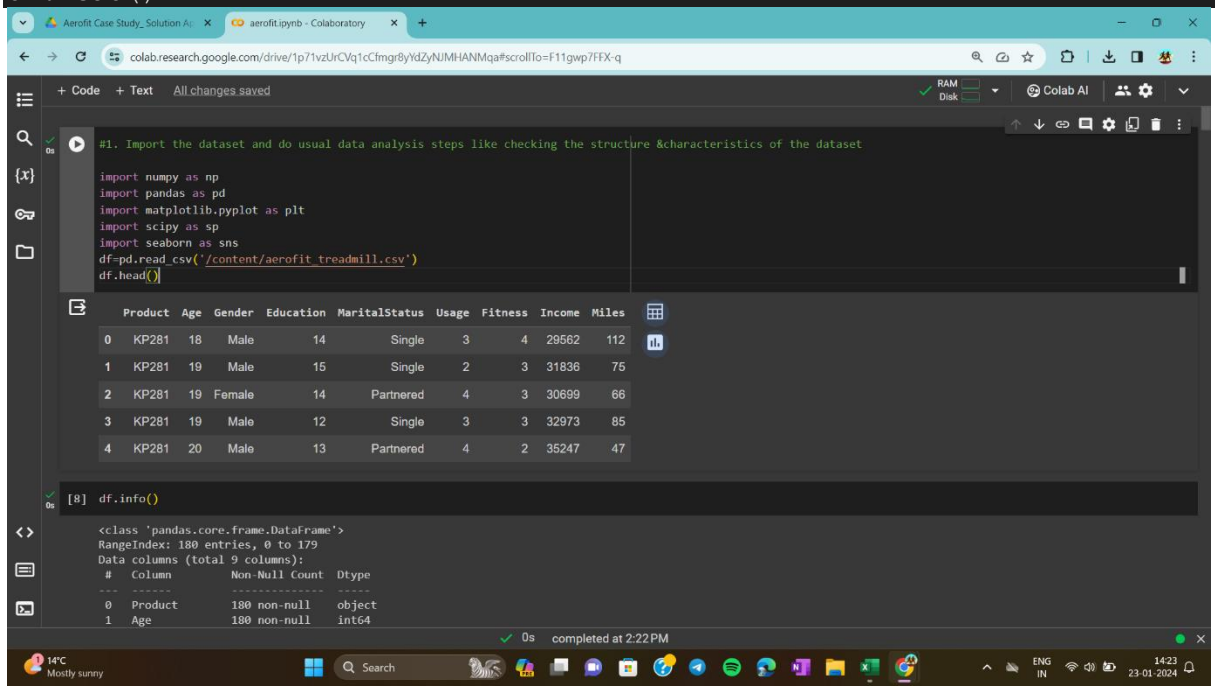
Aerofit Business Case Study

1. Import the dataset and do the usual data analysis steps like checking the structure & characteristics of the dataset.

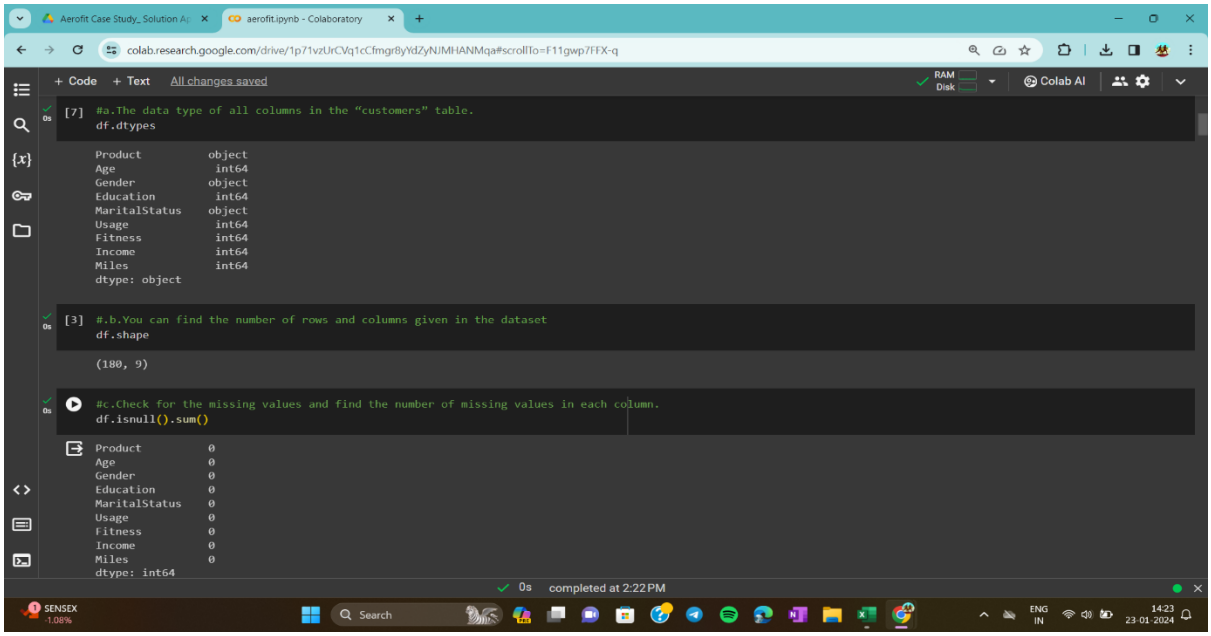
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scipy as sp
import seaborn as sns
df=pd.read_csv('/content/aerofit_treadmill.csv')
df.head()
```



	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

a. The data type of all columns in the “customers” table.
b. You can find the number of rows and columns given in the dataset
c. Check for the missing values and find the number of missing values in each column

```
df.dtypes
df.shape
df.isnull().sum()
```

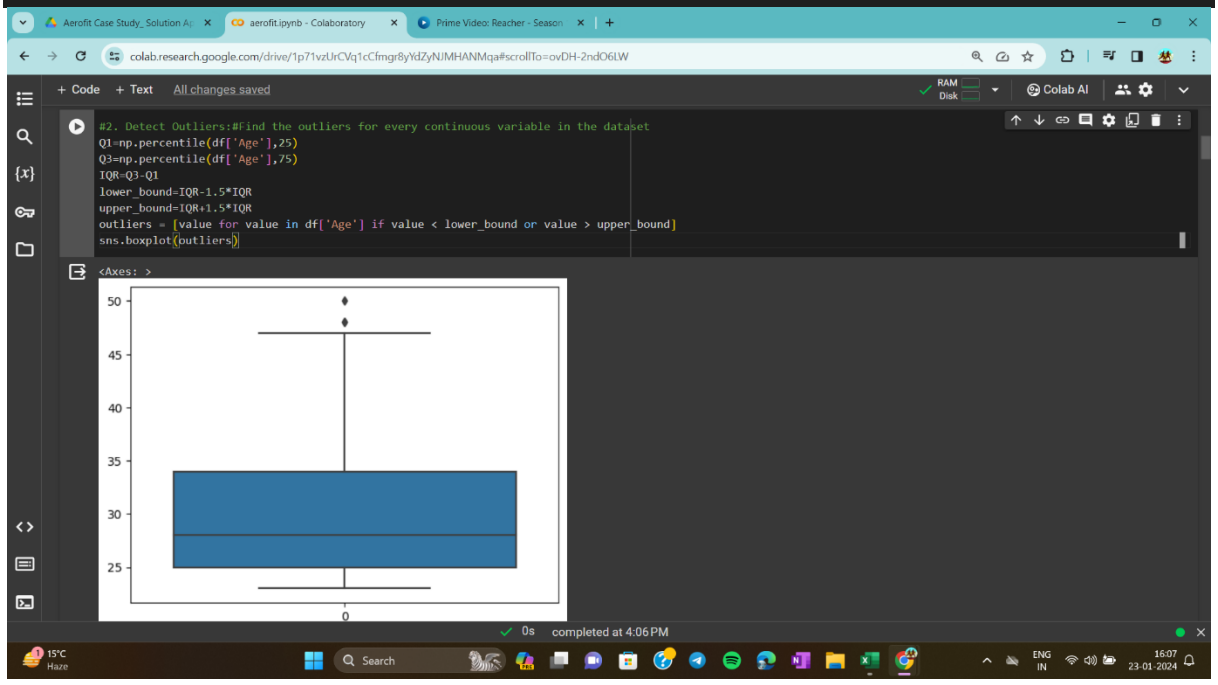


2. Detect Outliers:

a. Find the outliers for every continuous variable in the dataset.

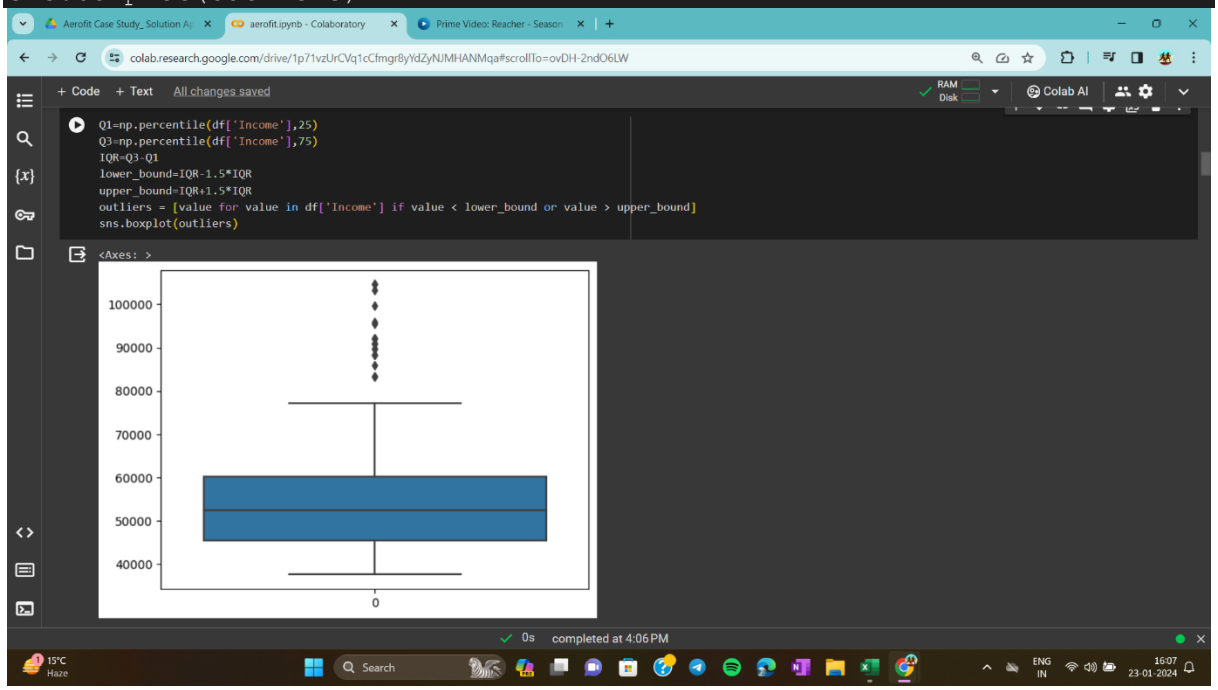
1.Outliers for "Age"

```
Q1=np.percentile(df['Age'],25)
Q3=np.percentile(df['Age'],75)
IQR=Q3-Q1
lower_bound=IQR-1.5*IQR
upper_bound=IQR+1.5*IQR
outliers = [value for value in df['Age'] if value < lower_bound or
value > upper_bound]
sns.boxplot(outliers)
```



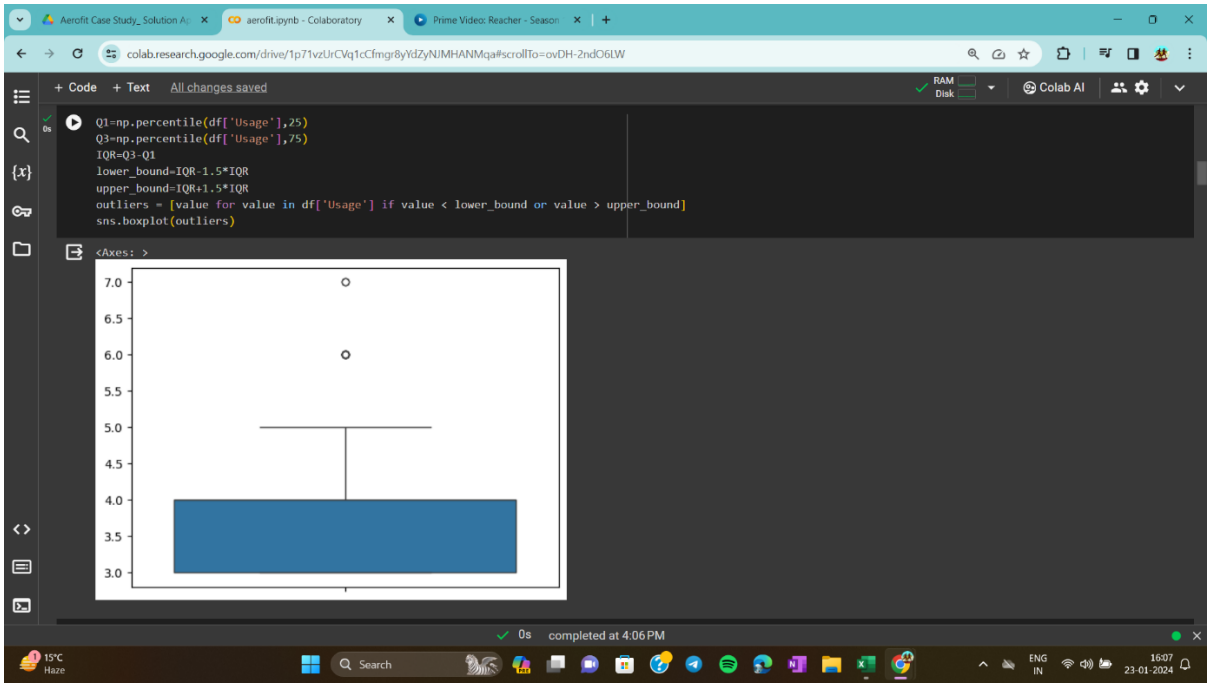
2. Outliers for “Income”

```
Q1=np.percentile(df['Income'],25)
Q3=np.percentile(df['Income'],75)
IQR=Q3-Q1
lower_bound=IQR-1.5*IQR
upper_bound=IQR+1.5*IQR
outliers = [value for value in df['Income'] if value < lower_bound or
value > upper_bound]
sns.boxplot(outliers)
```



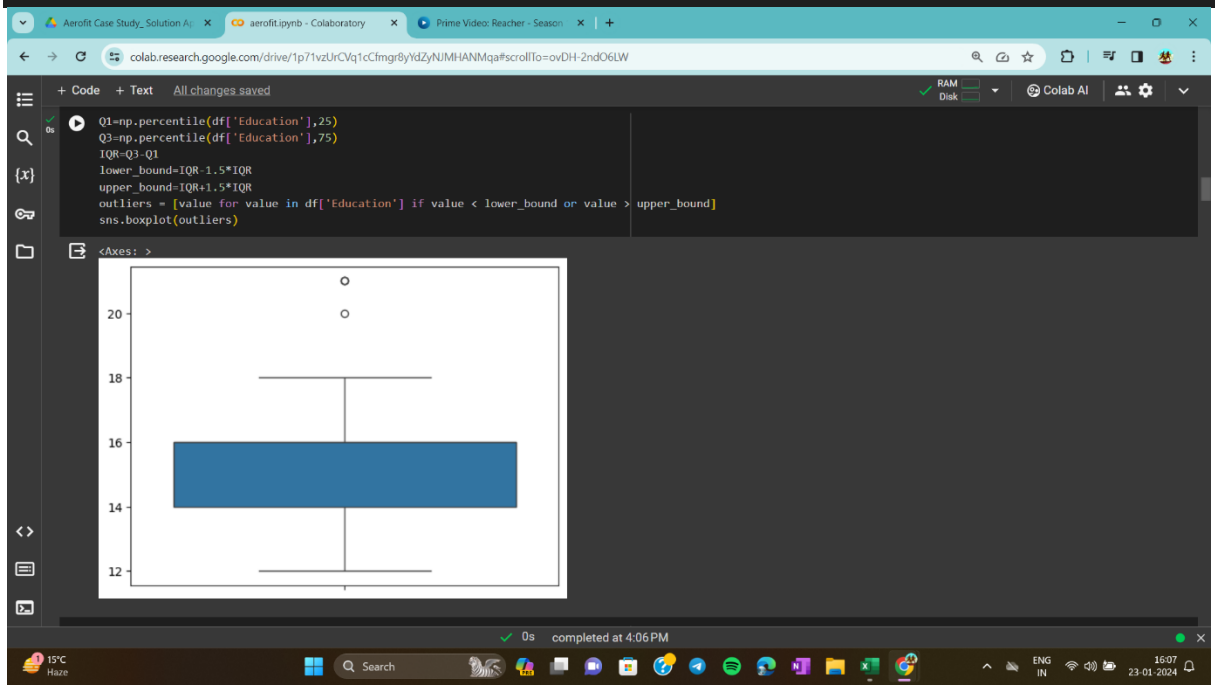
3. outliers for “Usage”

```
Q1=np.percentile(df['Usage'],25)
Q3=np.percentile(df['Usage'],75)
IQR=Q3-Q1
lower_bound=IQR-1.5*IQR
upper_bound=IQR+1.5*IQR
outliers = [value for value in df['Usage'] if value < lower_bound or
value > upper_bound]
sns.boxplot(outliers)
```



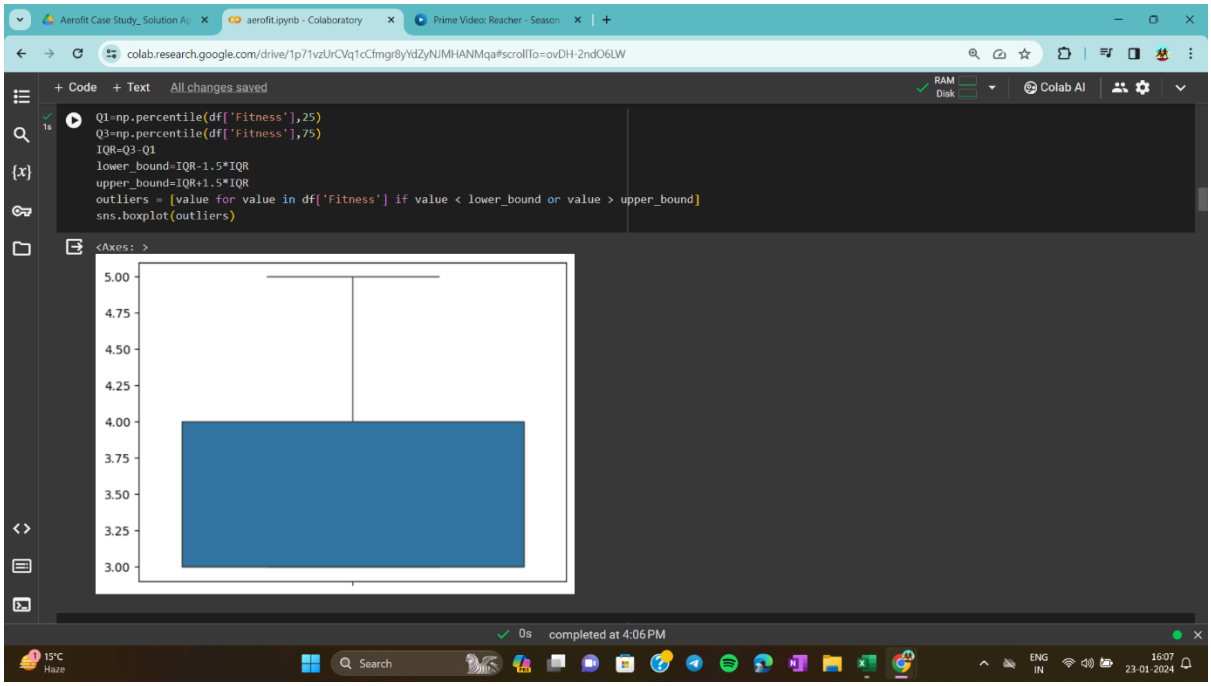
4. Outliers for “Education”

```
Q1=np.percentile(df['Education'],25)
Q3=np.percentile(df['Education'],75)
IQR=Q3-Q1
lower_bound=IQR-1.5*IQR
upper_bound=IQR+1.5*IQR
outliers = [value for value in df['Education'] if value < lower_bound
or value > upper_bound]
sns.boxplot(outliers)
```



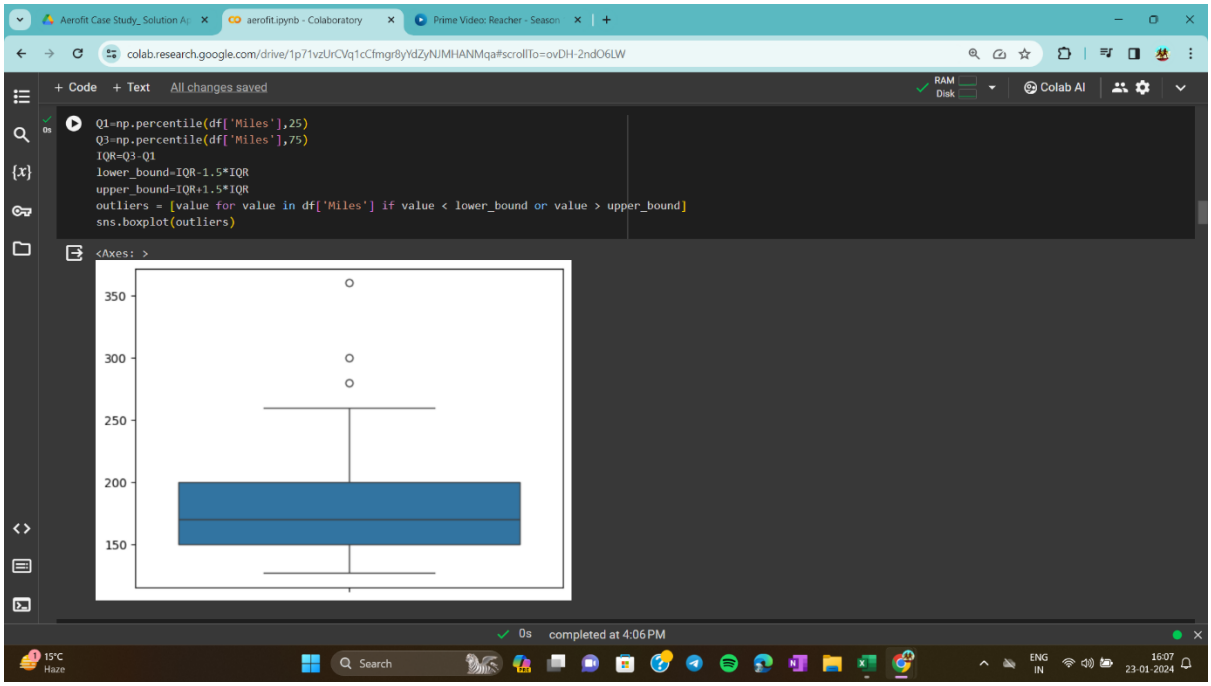
5. outliers for “Fitness”

```
Q1=np.percentile(df['Fitness'],25)
Q3=np.percentile(df['Fitness'],75)
IQR=Q3-Q1
lower_bound=IQR-1.5*IQR
upper_bound=IQR+1.5*IQR
outliers = [value for value in df['Fitness'] if value < lower_bound or
value > upper_bound]
sns.boxplot(outliers)
```



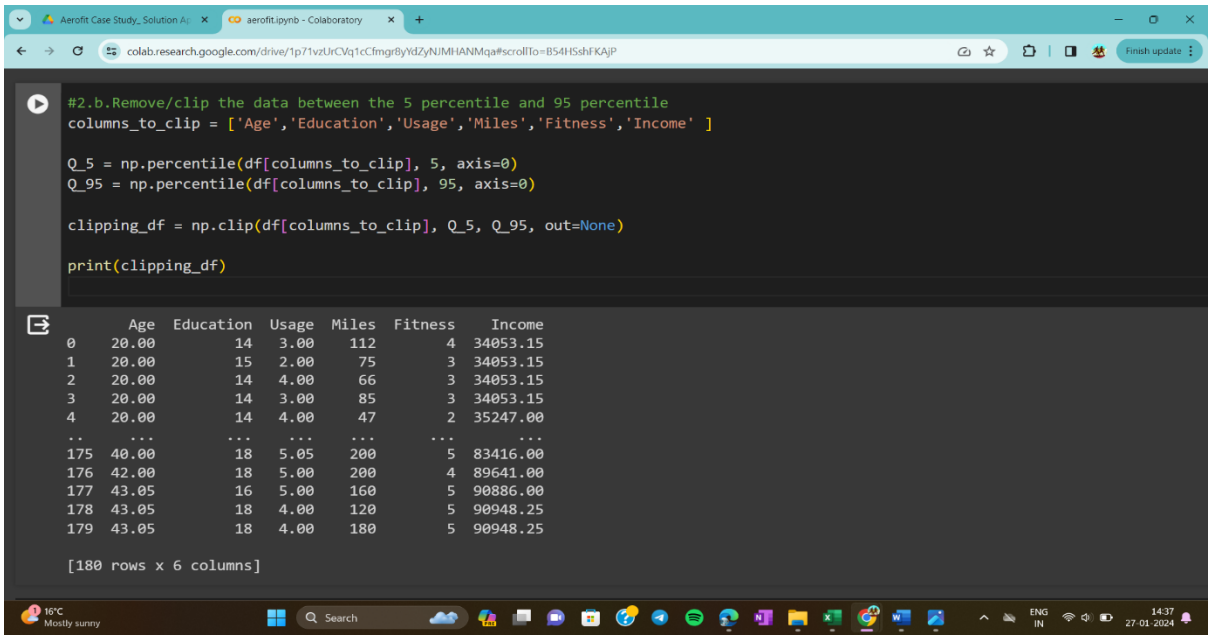
6. Outliers for “Miles”

```
Q1=np.percentile(df['Miles'],25)
Q3=np.percentile(df['Miles'],75)
IQR=Q3-Q1
lower_bound=IQR-1.5*IQR
upper_bound=IQR+1.5*IQR
outliers = [value for value in df['Miles'] if value < lower_bound or
value > upper_bound]
sns.boxplot(outliers)
```



2.b.Remove/clip the data between the 5 percentile and 95 percentile.

```
#2.b.Remove/clip the data between the 5 percentile and 95 percentile
columns_to_clip = ['Age','Education','Usage','Miles','Fitness','Income'
]

Q_5 = np.percentile(df[columns_to_clip], 5, axis=0)
Q_95 = np.percentile(df[columns_to_clip], 95, axis=0)

clipping_df = np.clip(df[columns_to_clip], Q_5, Q_95, out=None)

print(clipping_df)
```

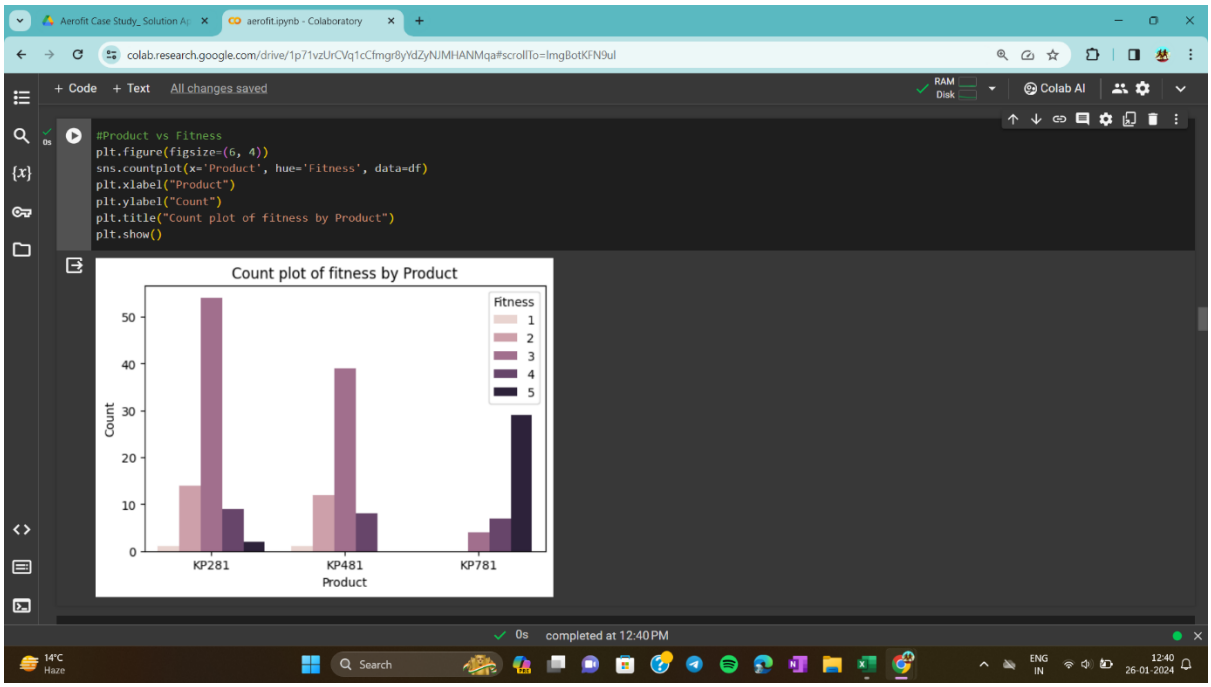


3. Check if features like marital status, Gender, and age have any effect on the product Purchased.

a. Find if there is any relationship between the categorical variables and the output variable in the data.(count plot to find the relationship between categorical variables and output variables.)

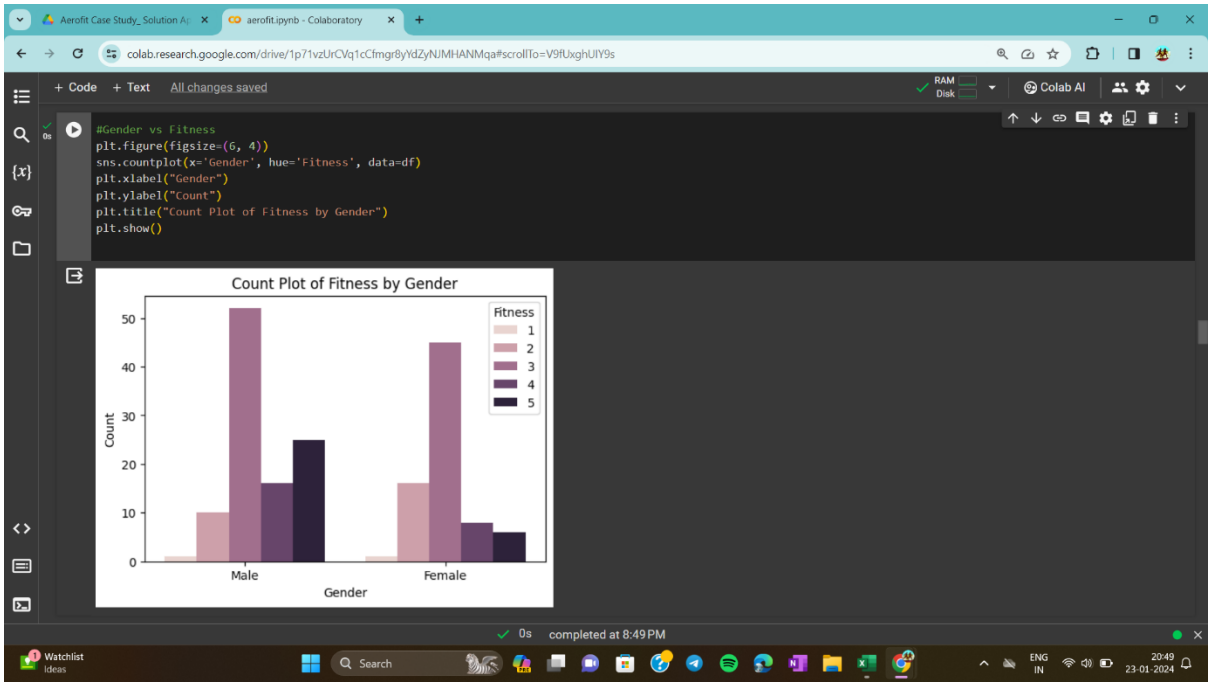
1.

```
#Product vs Fitness
plt.figure(figsize=(6, 4))
sns.countplot(x='Product', hue='Fitness', data=df)
plt.xlabel("Product")
plt.ylabel("Count")
plt.title("Count plot of fitness by Product")
plt.show()
```



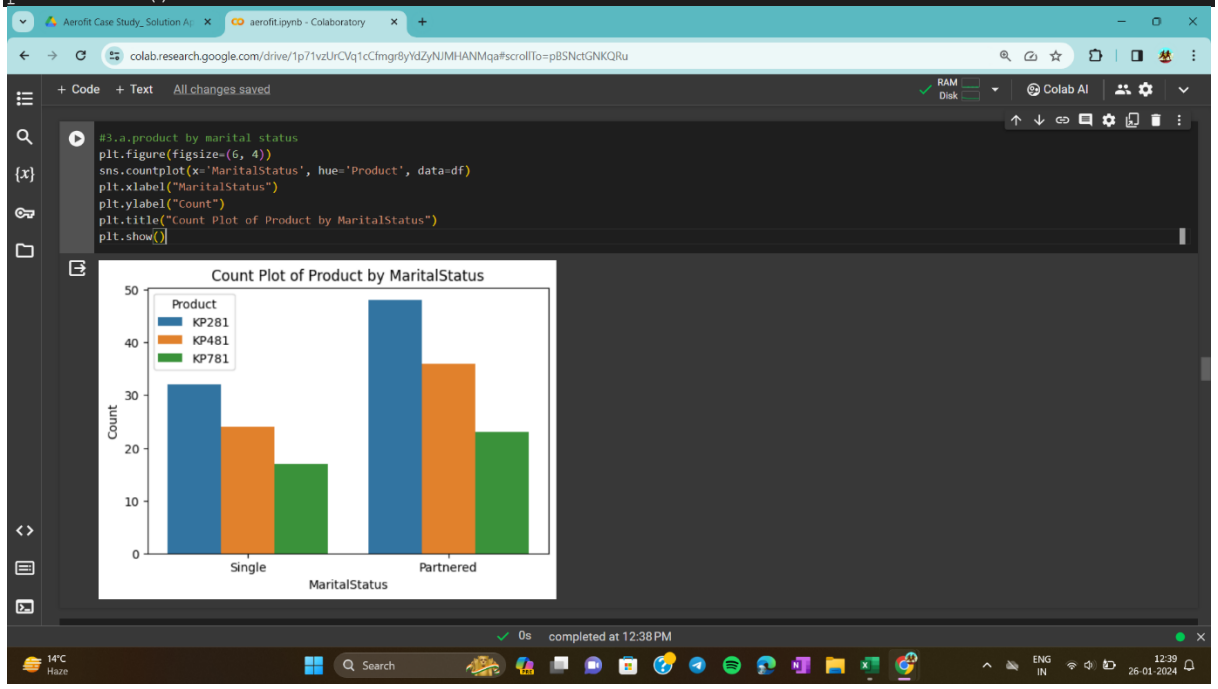
2.

```
#Gender vs Fitness
plt.figure(figsize=(6, 4))
sns.countplot(x='Gender', hue='Fitness', data=df)
plt.xlabel("Gender")
plt.ylabel("Count")
plt.title("Count Plot of Fitness by Gender")
plt.show()
```



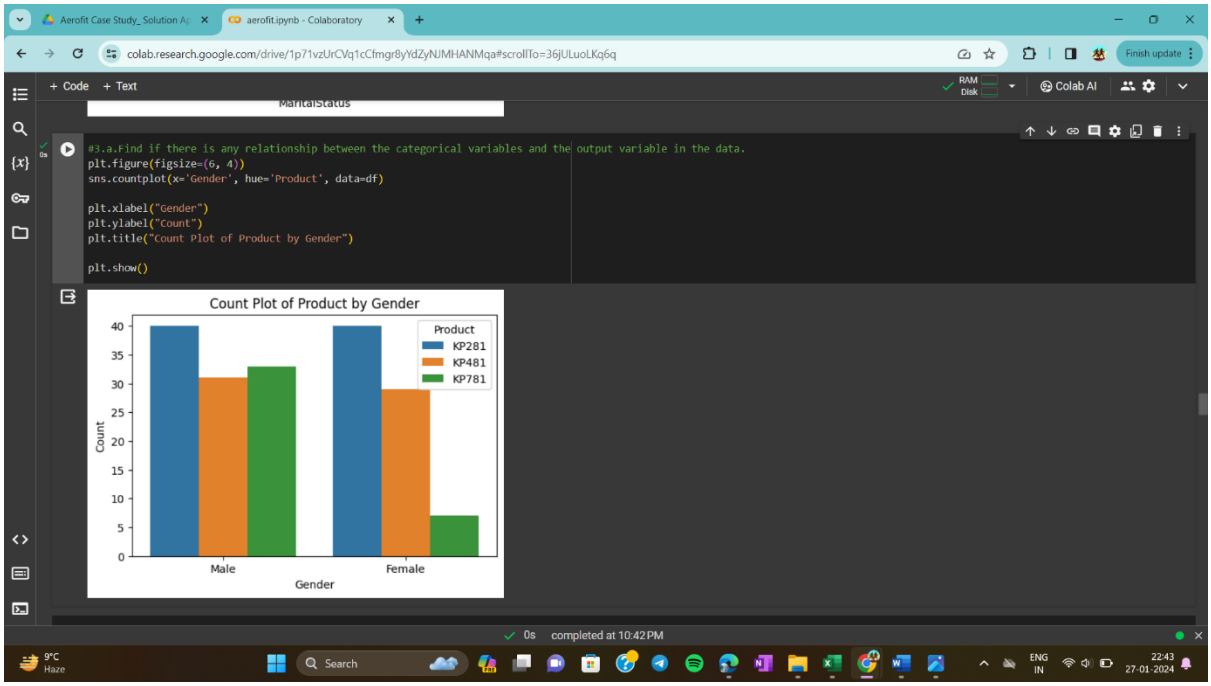
3.

```
#3.a.product by marital status
plt.figure(figsize=(6, 4))
sns.countplot(x='MaritalStatus', hue='Product', data=df)
plt.xlabel("MaritalStatus")
plt.ylabel("Count")
plt.title("Count Plot of Product by MaritalStatus")
plt.show()
```



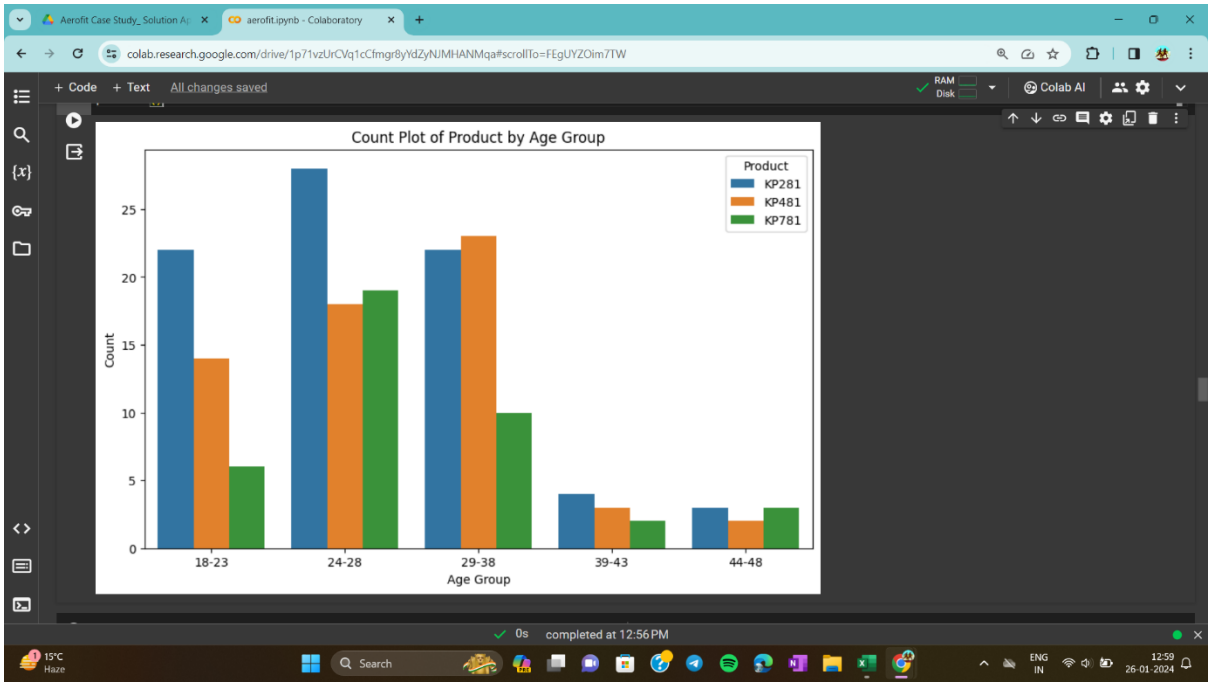
4. #3.a.Find if there is any relationship between the categorical variables and the output variable in the data.

```
plt.figure(figsize=(6, 4))
sns.countplot(x='Gender', hue='Product', data=df)

plt.xlabel("Gender")
plt.ylabel("Count")
plt.title("Count Plot of Product by Gender")

plt.show()
```



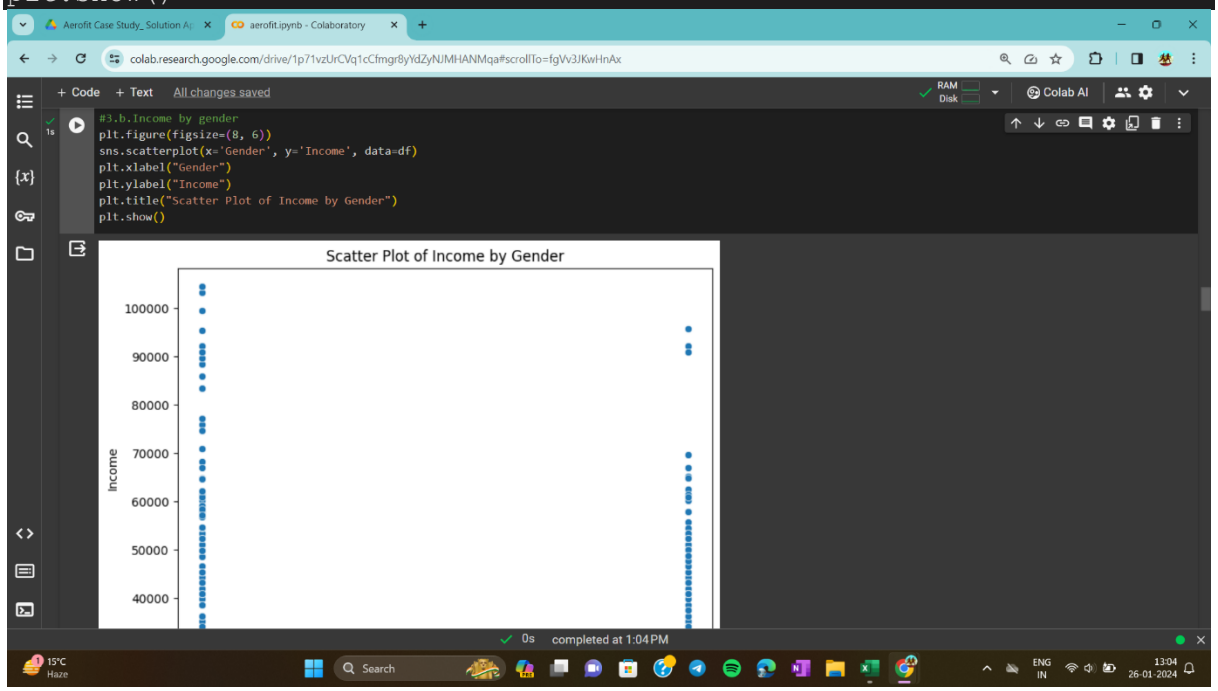
5.

```
#age vs product
bins = [18, 24, 29, 39, 44, 49]
labels = ['18-23','24-28','29-38','39-43','44-48']
df['Age Group'] = pd.cut(df['Age'], bins=bins, labels=labels,
right=False)

plt.figure(figsize=(10, 6))
sns.countplot(x='Age Group', hue='Product', data=df)

plt.xlabel("Age Group")
plt.ylabel("Count")
plt.title("Count Plot of Product by Age Group")

plt.show()
```

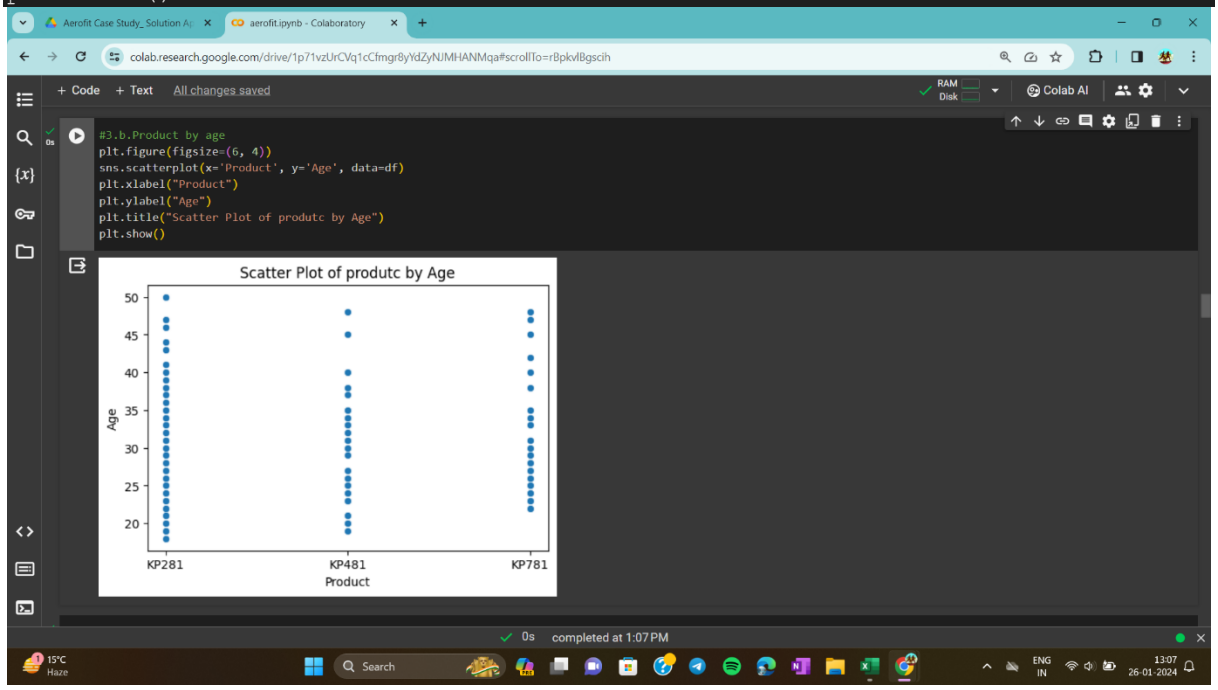


3. b. Find if there is any relationship between the continuous variables and the output variable in the data.

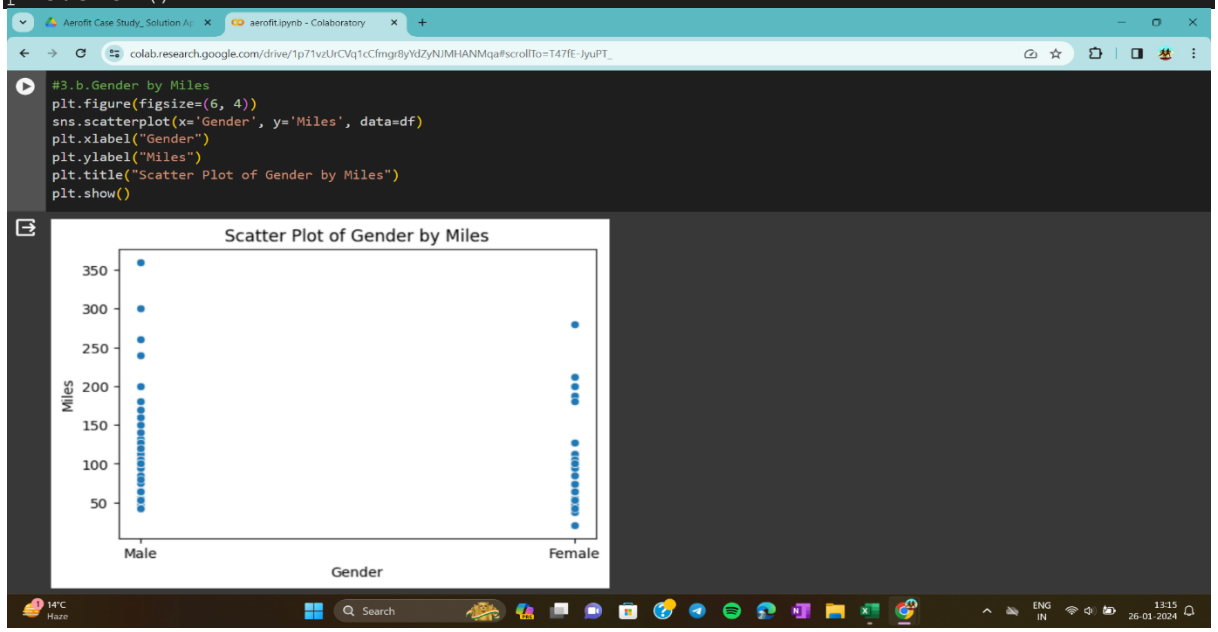
```
#3.b.Income by gender
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Gender', y='Income', data=df)
plt.xlabel("Gender")
plt.ylabel("Income")
plt.title("Scatter Plot of Income by Gender")
plt.show()
```



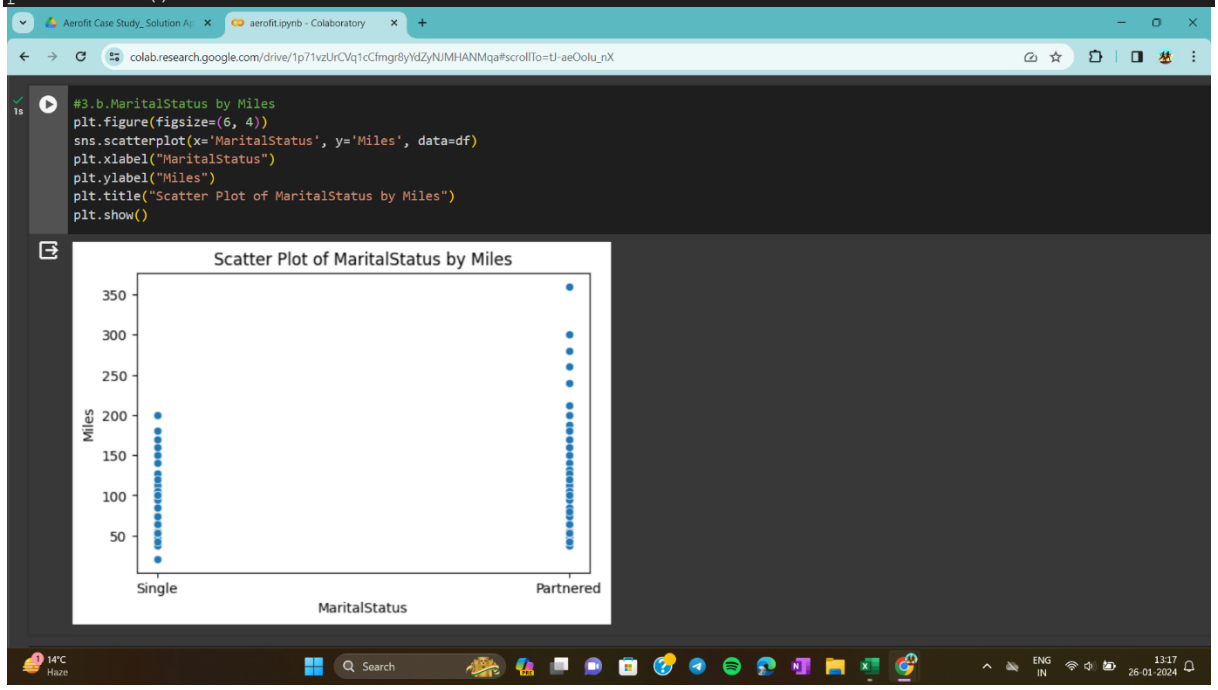
```
#3.b.Product by age
plt.figure(figsize=(6, 4))
sns.scatterplot(x='Product', y='Age', data=df)
plt.xlabel("Product")
plt.ylabel("Age")
plt.title("Scatter Plot of produtc by Age")
plt.show()
```



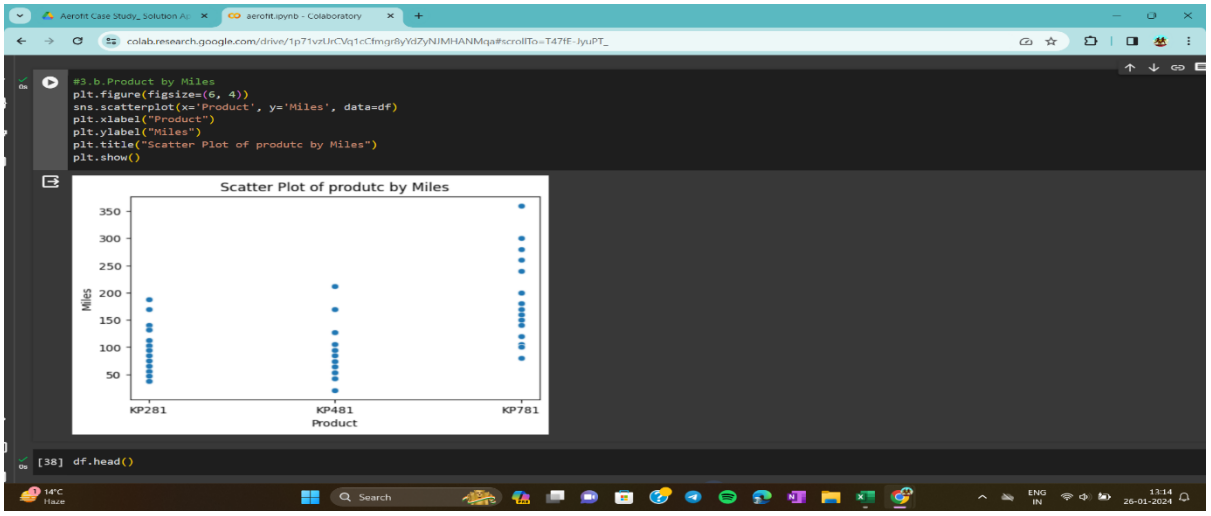

```
#3.b.Gender by Miles
plt.figure(figsize=(6, 4))
sns.scatterplot(x='Gender', y='Miles', data=df)
plt.xlabel("Gender")
plt.ylabel("Miles")
plt.title("Scatter Plot of Gender by Miles")
plt.show()
```



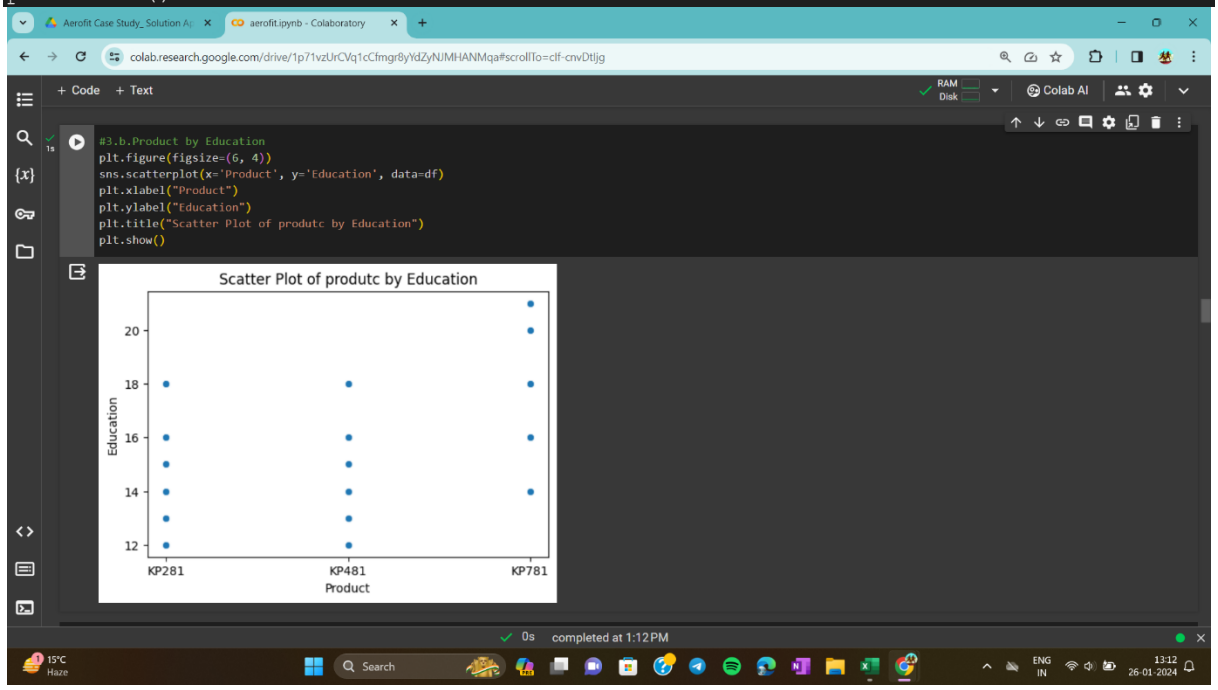
```
#3.b.MaritalStatus by Miles
plt.figure(figsize=(6, 4))
sns.scatterplot(x='MaritalStatus', y='Miles', data=df)
plt.xlabel("MaritalStatus")
plt.ylabel("Miles")
plt.title("Scatter Plot of MaritalStatus by Miles")
plt.show()
```




```
#3.b.Product by Miles
plt.figure(figsize=(6, 4))
sns.scatterplot(x='Product', y='Miles', data=df)
plt.xlabel("Product")
plt.ylabel("Miles")
plt.title("Scatter Plot of Product by Miles")
plt.show()
```



```
#3.b.Product by Education
plt.figure(figsize=(6, 4))
sns.scatterplot(x='Product', y='Education', data=df)
plt.xlabel("Product")
plt.ylabel("Education")
plt.title("Scatter Plot of produtc by Education")
plt.show()
```



4. Representing the Probability.

a. Find the marginal probability (what per cent of customers have purchased KP281, KP481, orKP781)

b. Find the probability that the customer buys a product based on each column.

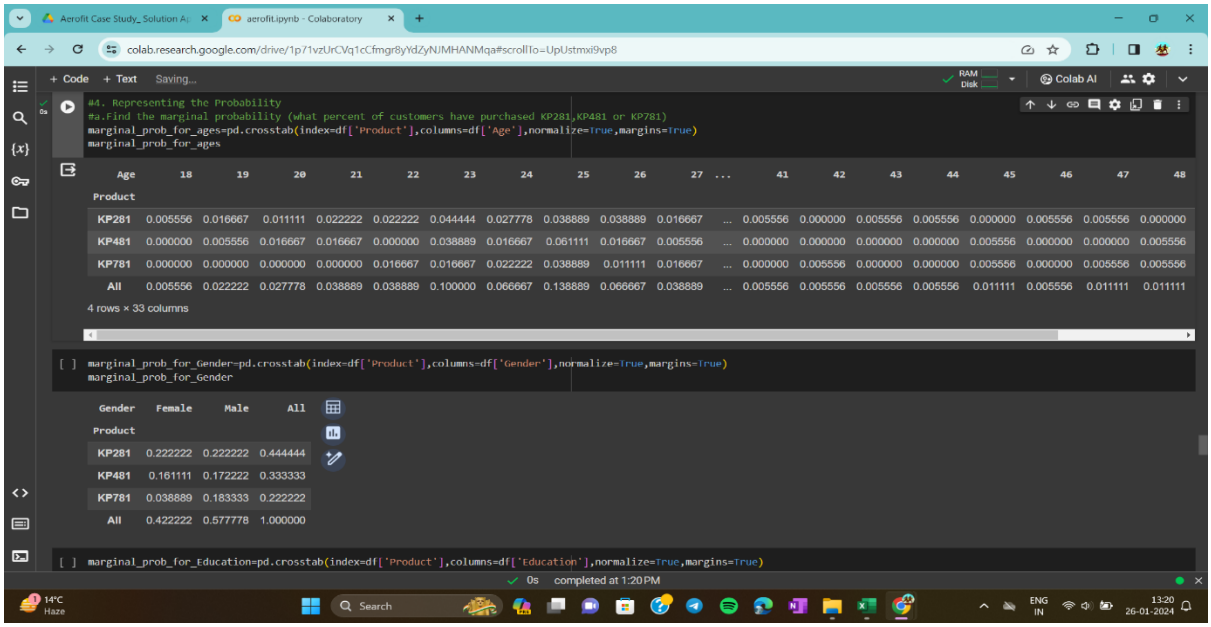
c. Find the conditional probability that an event occurs given that another event

has occurred. Find the conditional probability that an event occurs given that another event

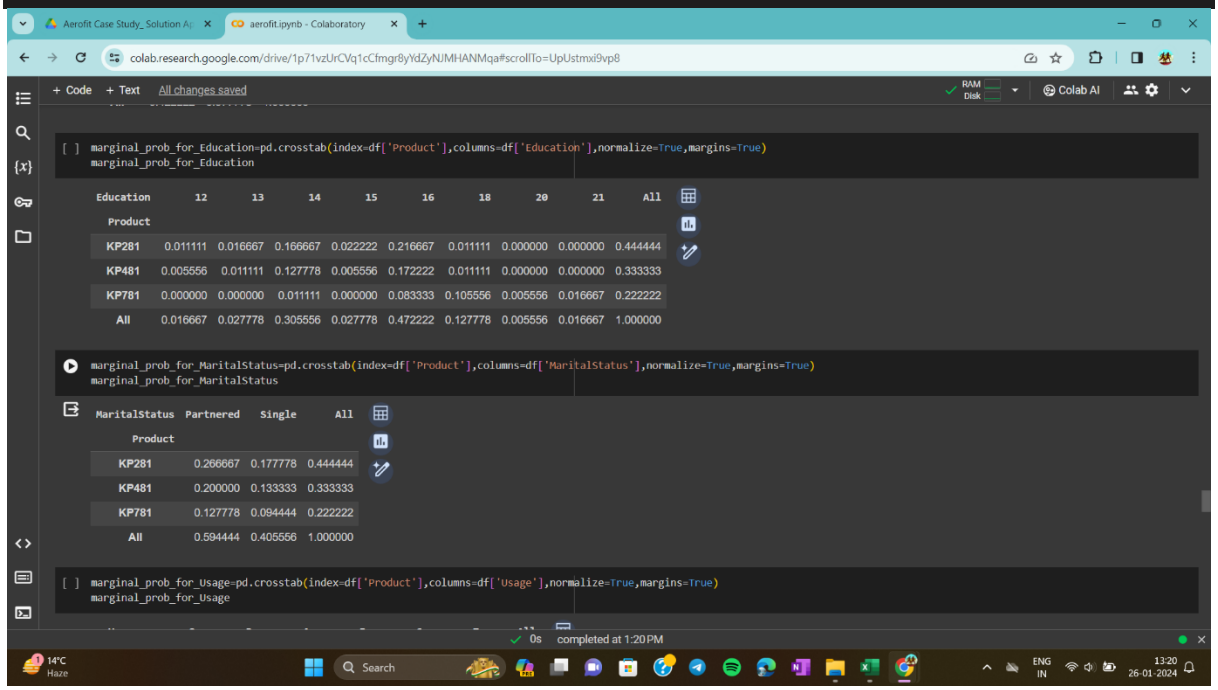
has occurred. (Example: given that a customer is female, what is the probability

she'll purchase a KP481)

```
#a.Find the marginal probability (what percent of customers have
purchased KP281,KP481 or KP781)
marginal_prob_for_ages=pd.crosstab(index=df['Product'],columns=df['Age'
],normalize=True,margins=True)
marginal_prob_for_ages



marginal_prob_for_Gender=pd.crosstab(index=df['Product'],columns=df['Ge
nder'],normalize=True,margins=True)
marginal_prob_for_Gender
```

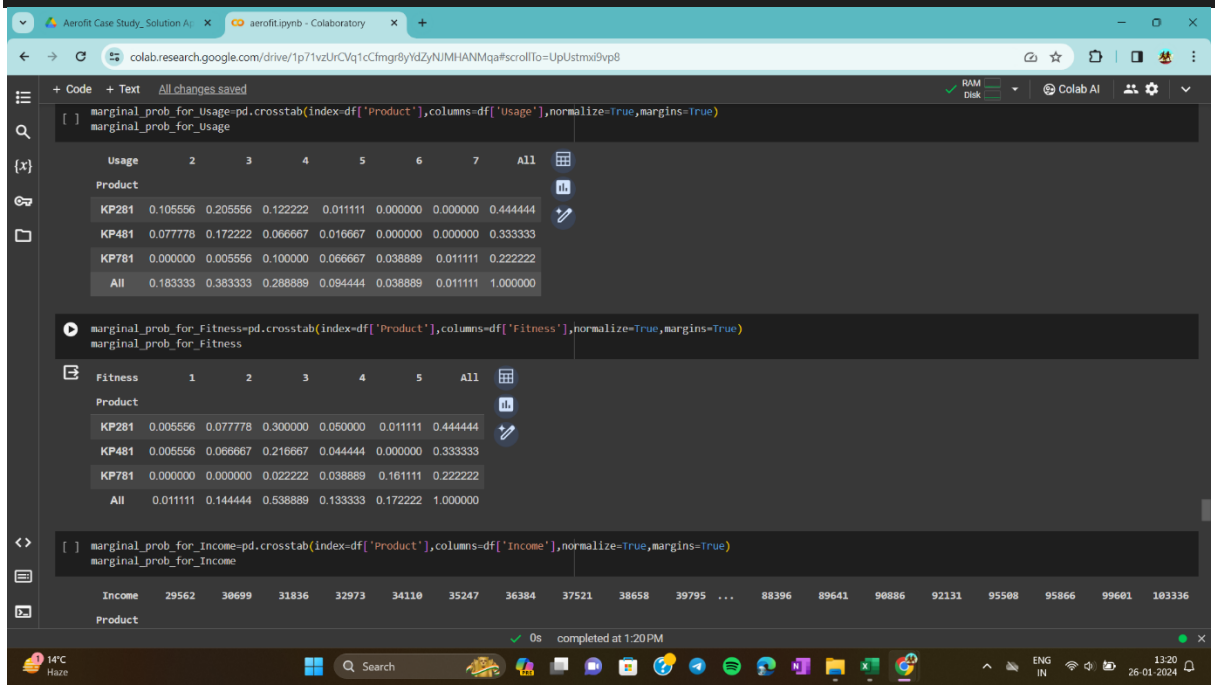



```
marginal_prob_for_Education=pd.crosstab(index=df['Product'],columns=df['Education'],normalize=True,margins=True)
marginal_prob_for_Education

marginal_prob_for_MaritalStatus=pd.crosstab(index=df['Product'],columns=df['MaritalStatus'],normalize=True,margins=True)
marginal_prob_for_MaritalStatus
```

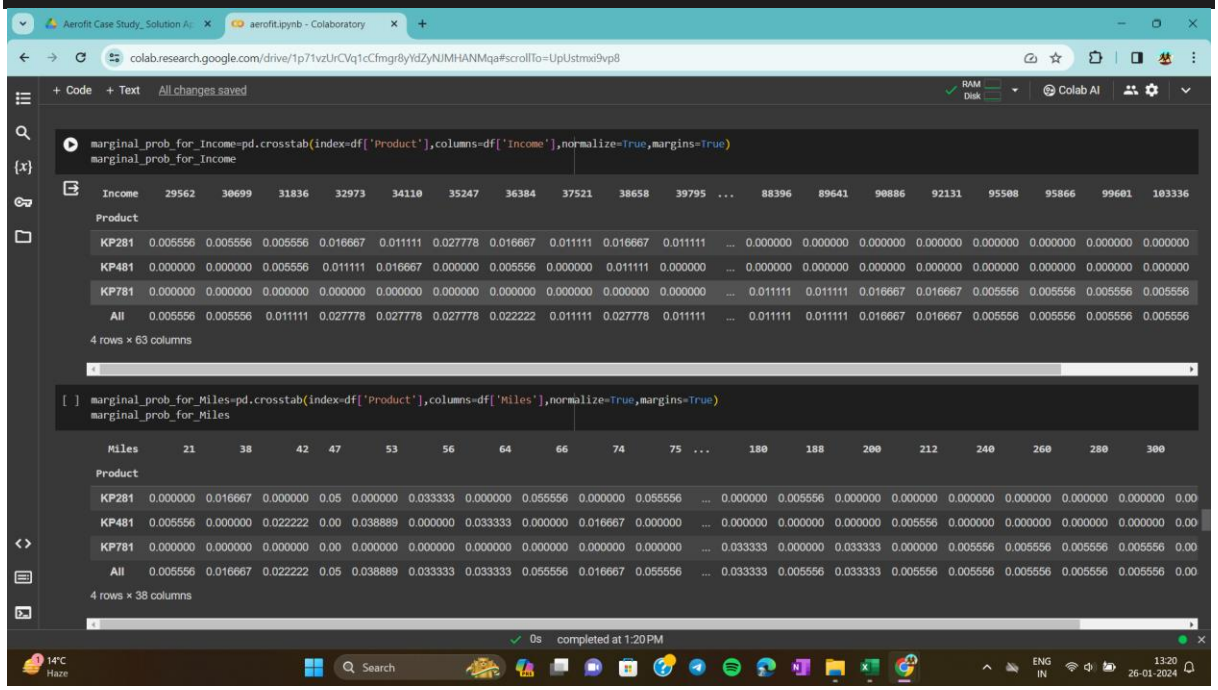



```
marginal_prob_for_Usage=pd.crosstab(index=df['Product'],columns=df['Usage'],normalize=True,margins=True)
marginal_prob_for_Usage
```

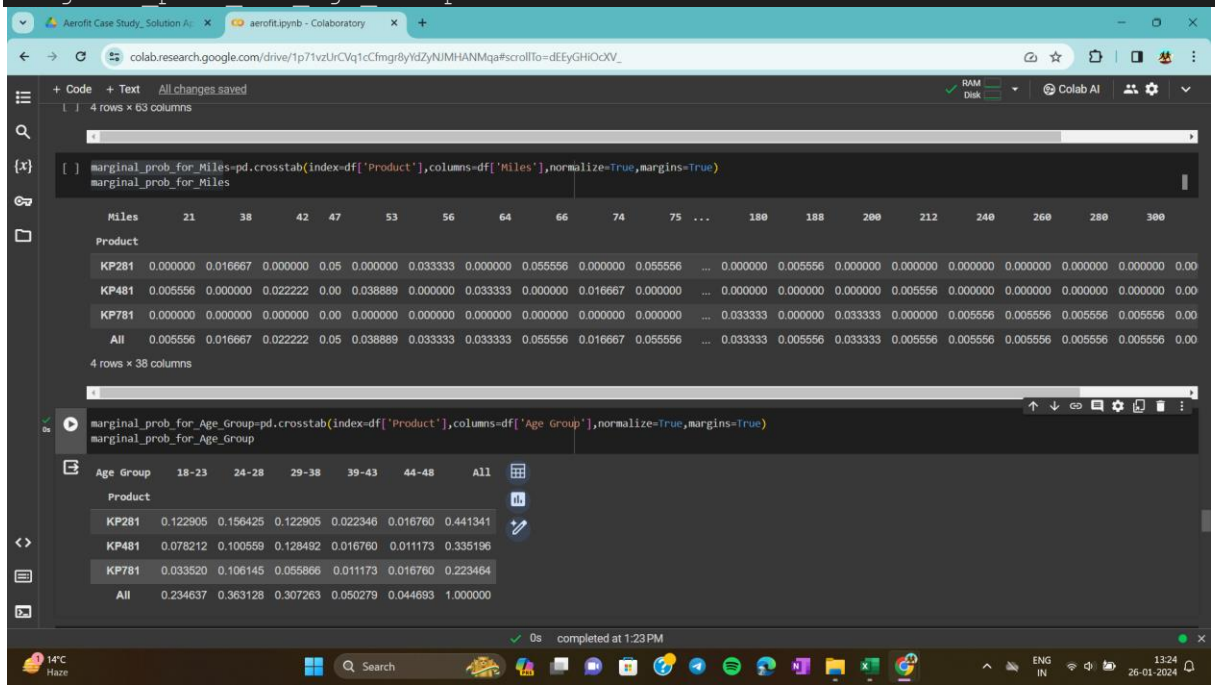
```
marginal_prob_for_Fitness=pd.crosstab(index=df['Product'],columns=df['Fitness'],normalize=True,margins=True)
marginal_prob_for_Fitness
```




```
marginal_prob_for_Income=pd.crosstab(index=df['Product'],columns=df['Income'],normalize=True,margins=True)
marginal_prob_for_Income

marginal_prob_for_Miles=pd.crosstab(index=df['Product'],columns=df['Miles'],normalize=True,margins=True)
marginal_prob_for_Miles
```



```
marginal_prob_for_Age_Group=pd.crosstab(index=df['Product'],columns=df[
'Age Group'],normalize=True,margins=True)
marginal_prob_for_Age_Group
```

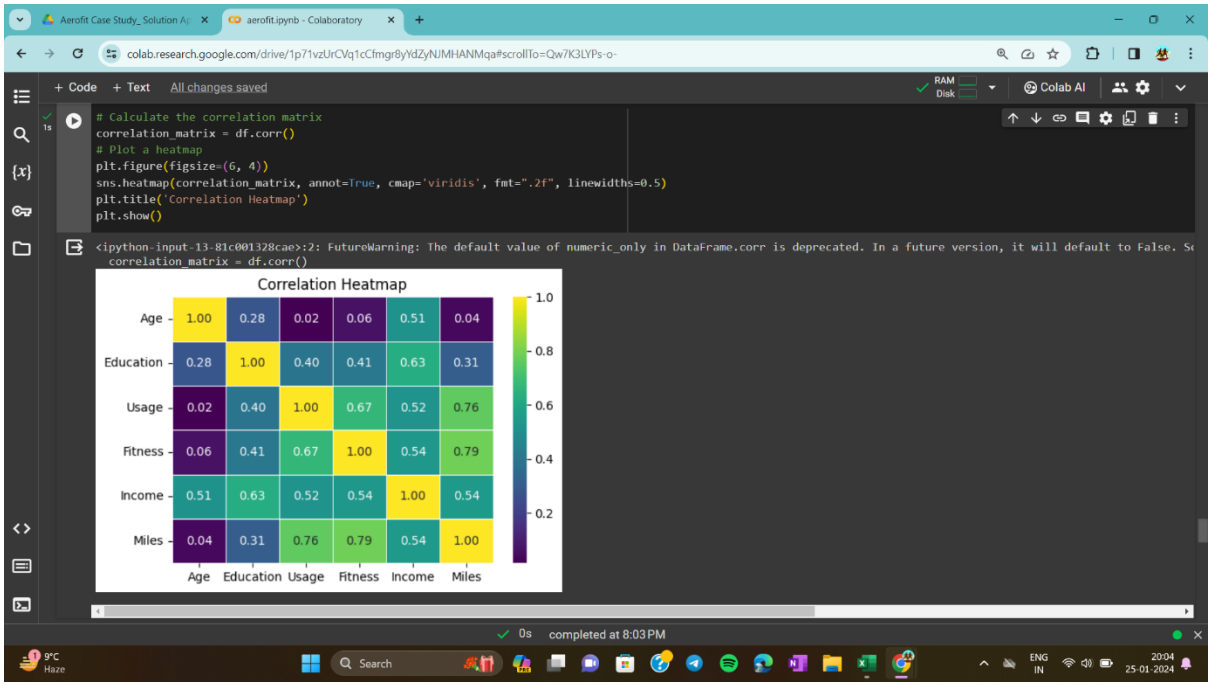


Age Group	18-23	24-28	29-38	39-43	44-48	All
Product						
KP281	0.122905	0.156425	0.122905	0.022346	0.016760	0.441341
KP481	0.078212	0.100559	0.128492	0.016760	0.011173	0.335196
KP781	0.033520	0.106145	0.055866	0.011173	0.016760	0.223464
All	0.234637	0.363128	0.307263	0.050279	0.044693	1.000000

5. Check the correlation among different factors.

Find the correlation between the given features in the table. (Use of heatmap and corr function to find the correlation between the variables)

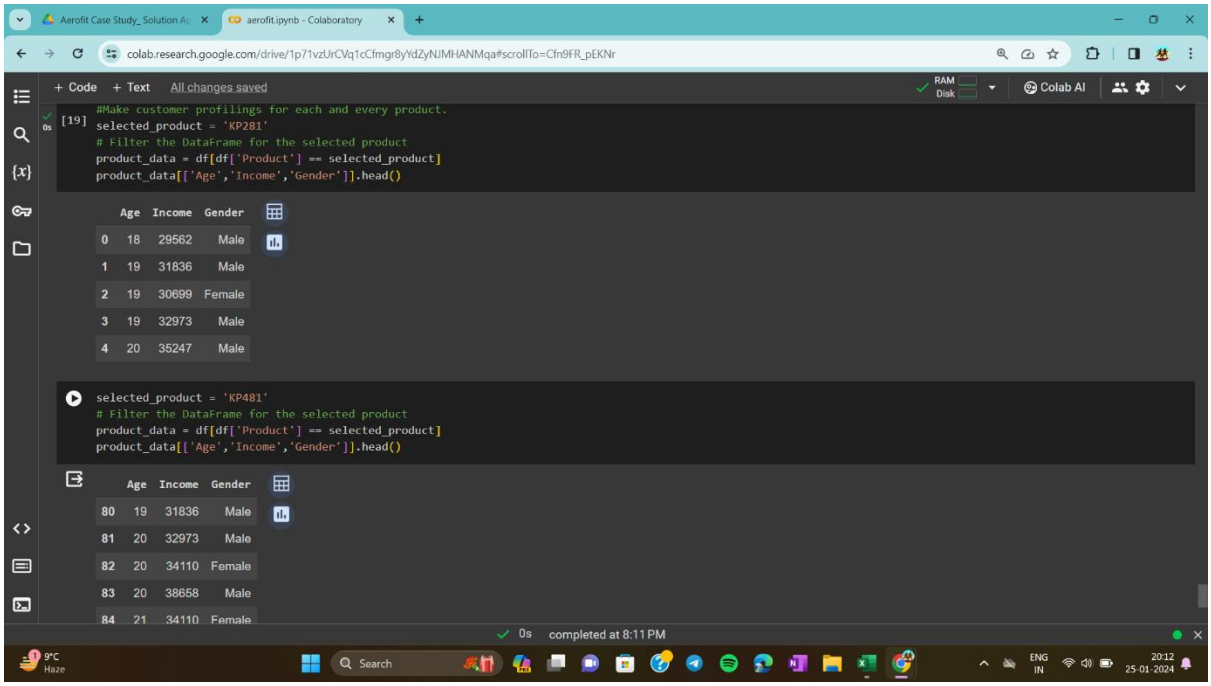
```
# Calculate the correlation matrix
correlation_matrix = df.corr()
# Plot a heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(correlation_matrix, annot=True, cmap='viridis', fmt=".2f",
linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



6. Customer profiling and recommendation

Make customer profiling for each and every product.

```
import pandas as pd

selected_product = 'KP281'

product_data = df[df['Product'] == selected_product]

product_data[['Age', 'Income', 'Gender']].head()


selected_product = 'KP481'

product_data = df[df['Product'] == selected_product]

product_data[['Age', 'Income', 'Gender']].head()
```




```
selected_product = 'KP781'
# Filter the DataFrame for the selected product
product_data = df[df['Product'] == selected_product]
product_data[['Age','Income','Gender']].head()
```