AeroFit Study

April 3, 2024

1 AeroFit Case Study

1.1 Initial analysis of the data set

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

[2]:
```
df = pd.read_csv('aerofit_treadmill.csv')
```

[3]:
```
df.head(10)
```

[3]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47
5	KP281	20	Female	14	Partnered	3	3	32973	66
6	KP281	21	Female	14	Partnered	3	3	35247	75
7	KP281	21	Male	13	Single	3	3	32973	85
8	KP281	21	Male	15	Single	5	4	35247	141
9	KP281	21	Female	15	Partnered	2	3	37521	85

1.1.1 Data types of all the columns

[4]:
```
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
```

1.1.2 Shape of Data Frame

[5]:
```python
df.shape
```

[5]:
```
(180, 9)
```

1.1.3 Checking missing values if any

[6]:
```python
df.isnull().any()
```

[6]:
```
Product          False
Age              False
Gender           False
Education        False
MaritalStatus    False
Usage            False
Fitness          False
Income           False
Miles            False
dtype: bool
```

[7]:
```python
df.describe(include = "all")
```

[7]:
```
       Product         Age Gender   Education MaritalStatus       Usage  \
count      180  180.000000    180  180.000000           180  180.000000
unique       3         NaN      2         NaN             2         NaN
top      KP281         NaN   Male         NaN     Partnered         NaN
freq        80         NaN    104         NaN           107         NaN
mean       NaN   28.788889    NaN   15.572222           NaN    3.455556
std        NaN    6.943498    NaN    1.617055           NaN    1.084797
min        NaN   18.000000    NaN   12.000000           NaN    2.000000
25%        NaN   24.000000    NaN   14.000000           NaN    3.000000
50%        NaN   26.000000    NaN   16.000000           NaN    3.000000
75%        NaN   33.000000    NaN   16.000000           NaN    4.000000
max        NaN   50.000000    NaN   21.000000           NaN    7.000000

           Fitness         Income       Miles
count   180.000000     180.000000  180.000000
unique         NaN            NaN         NaN
top            NaN            NaN         NaN
freq           NaN            NaN         NaN
```

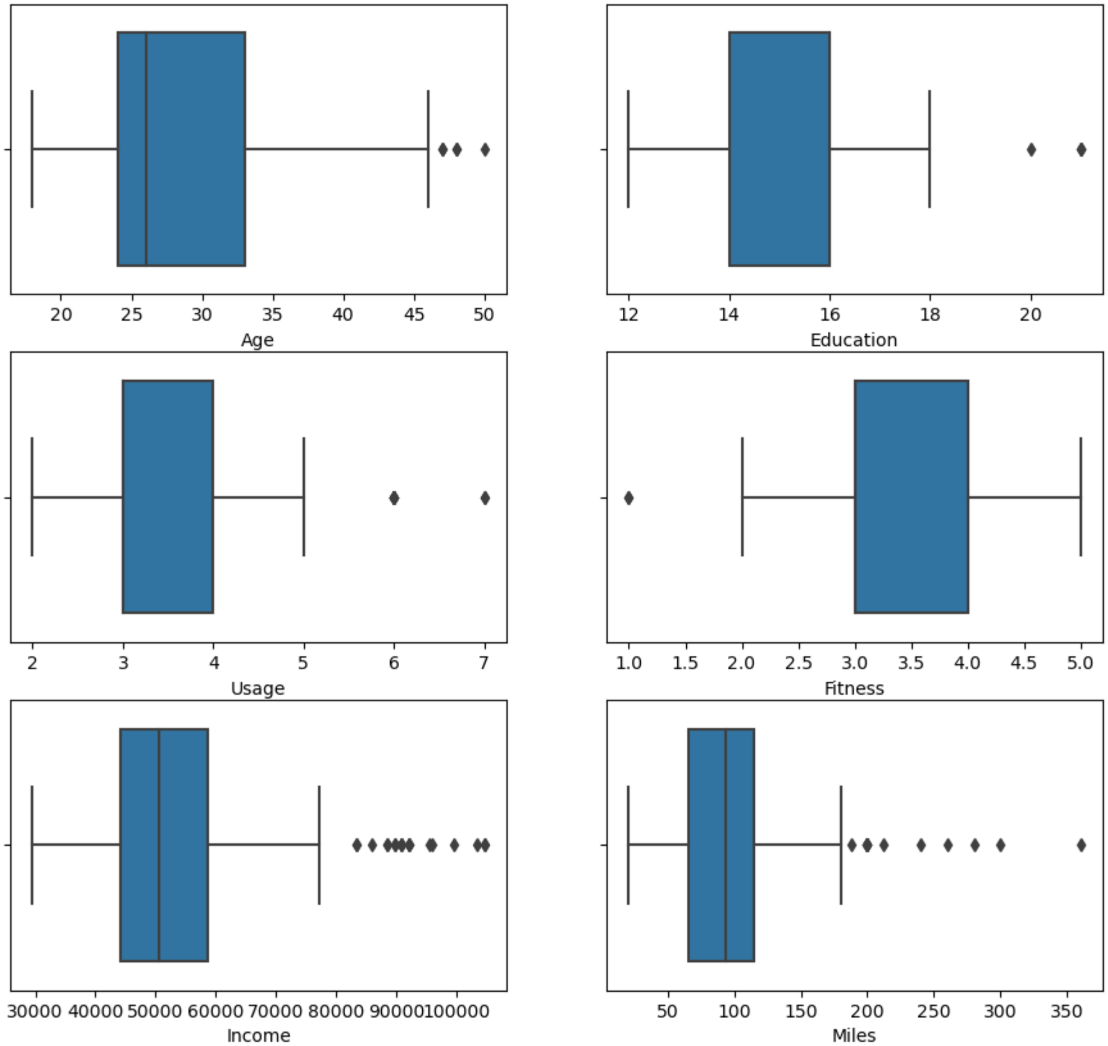
```
mean      3.311111   53719.577778  103.194444
std       0.958869   16506.684226   51.863605
min       1.000000   29562.000000   21.000000
25%       3.000000   44058.750000   66.000000
50%       3.000000   50596.500000   94.000000
75%       4.000000   58668.000000  114.750000
max       5.000000  104581.000000  360.000000
```

- There are no missing values in the dataset.
- There are 3 unique products in the dataset.
- KP281 is the most frequent product.
- Minimum & Maximum age of the customer is 18 & 50
- Mean is 28.79 and 75% of people have age less than or equal to 33.
- Out of 180 data points, 104 are of Male gender and rest are the female.
- Standard deviation for Income & Miles is very high. These variables might have outliers in it.

1.2 Detecting outliers

[8]:
```
fig,axis = plt.subplots(nrows=3, ncols=2, figsize=(11, 10))
sns.boxplot(data=df, x="Age", ax =axis[0,0])
sns.boxplot(data=df, x="Education", ax =axis[0,1])
sns.boxplot(data=df, x="Usage", ax=axis[1,0])
sns.boxplot(data=df, x="Fitness", ax=axis[1,1])
sns.boxplot(data=df, x="Income", ax=axis[2,0])
sns.boxplot(data=df, x="Miles", ax=axis[2,1])
```

[8]:
```
<Axes: xlabel='Miles'>
```



Age and education has few outliers

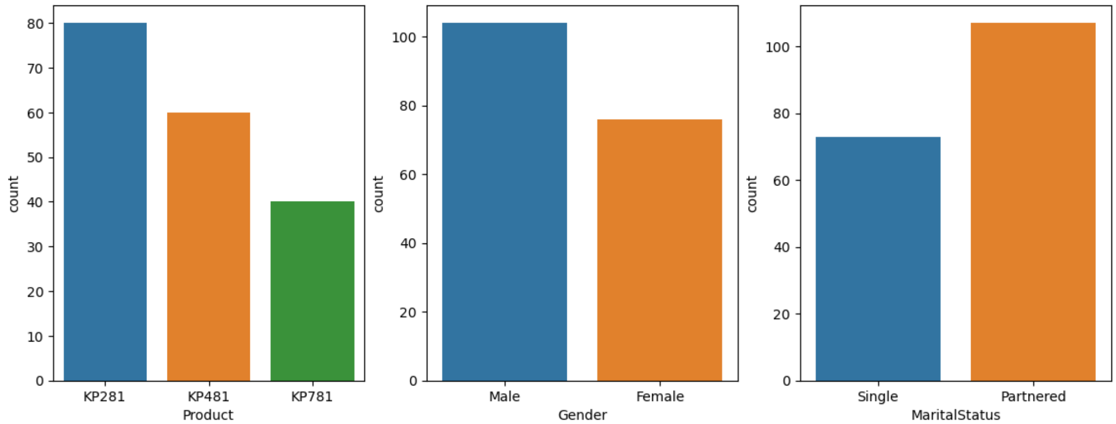
Income and miles have more outliers

1.3 Effects of features like Gender and Marital status on purchases

1.3.1 Relationship between the categorical variables and the output variables

[9]:
```
fig, axis = plt.subplots(1,3, figsize=(14,5))
sns.countplot(data=df, x='Product', ax=axis[0])
sns.countplot(data=df, x='Gender', ax=axis[1])
sns.countplot(data=df, x='MaritalStatus', ax=axis[2])
```

[9]: <Axes: xlabel='MaritalStatus', ylabel='count'>



KP281 is most frequently purchased product

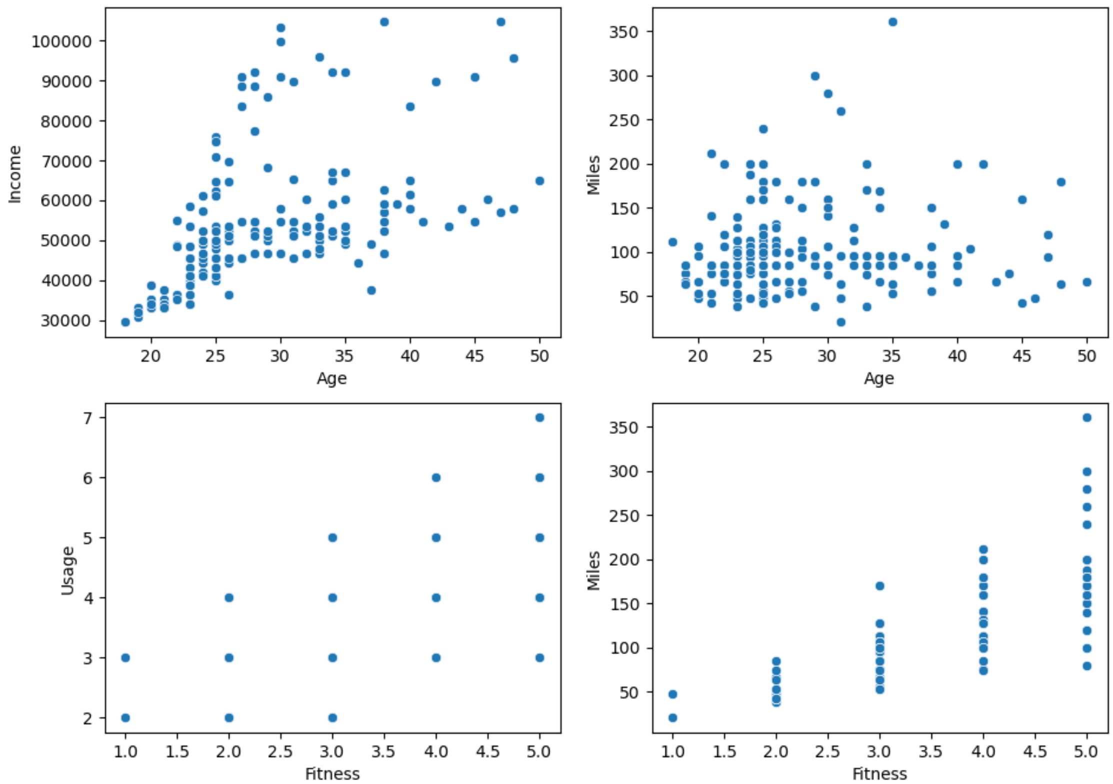
Men are more inclined to buy fitness equipment

Married couples are more likely to buy fitness equipment

1.3.2 Relationship between continuous variables and the output variables

[10]:
```
x_axis = df['Age']
fig, axis = plt.subplots(2,2, figsize=(11,8))
sns.scatterplot(data=df, x="Age", y="Income", ax=axis[0, 0] )
sns.scatterplot(data=df, x="Age", y="Miles", ax=axis[0, 1] )
sns.scatterplot(data=df, x="Fitness", y="Usage", ax=axis[1, 0] )
sns.scatterplot(data=df, x="Fitness", y="Miles", ax=axis[1, 1] )
```

[10]: <Axes: xlabel='Fitness', ylabel='Miles'>



1.4 Representing the Probability

1.4.1 Marginal probability of types of product sold

[11]:
```
product_sold = pd.crosstab(index=df['Product'], columns='percentage')
total_sold = product_sold['percentage'].sum()
marginal_probability = (product_sold / total_sold) * 100
marginal_probability
```

[11]:
```
col_0     percentage
Product
KP281      44.444444
KP481      33.333333
KP781      22.222222
```

- KP281 was bought by 44.44% of people
- KP481 was bought by 33.33% of people
- KP781 was bought by 22.22% of people

1.4.2 Probaility of purchase based on other criterias

[12]:
```
Gender = pd.crosstab(index=df['Gender'], columns='percentage')
total_sold = Gender['percentage'].sum()
percentage = (Gender / total_sold) * 100
percentage
```

[12]:
```
col_0    percentage
Gender
Female    42.222222
Male      57.777778
```

57.77% customers are Male and Feamle are other 42.22%

[13]:
```
Martial_status = pd.crosstab(index=df['MaritalStatus'], columns='percentage')
total_sold = Martial_status ['percentage'].sum()
percentage = (Martial_status  / total_sold) * 100
percentage
```

[13]:
```
col_0          percentage
MaritalStatus
Partnered       59.444444
Single          40.555556
```

59.44% of the customers are married and single people makeup the other 40.55%

1.4.3 Conditional probability

[14]:
```
product_sold = pd.crosstab(index=df['Product'], columns=df['Gender'],␣
 ↪margins=True)
prob = product_sold/180
prob['Female_Conditionl_P'] = prob['Female'] / prob.loc[prob.index == 'All',␣
 ↪'Female'].values[0]
prob['Male_Conditionl_P'] = prob['Male'] / prob.loc[prob.index == 'All',␣
 ↪'Male'].values[0]
prob
```

[14]:
```
Gender     Female      Male       All  Female_Conditionl_P  Male_Conditionl_P
Product
KP281    0.222222  0.222222  0.444444             0.526316           0.384615
KP481    0.161111  0.172222  0.333333             0.381579           0.298077
KP781    0.038889  0.183333  0.222222             0.092105           0.317308
All      0.422222  0.577778  1.000000             1.000000           1.000000
```

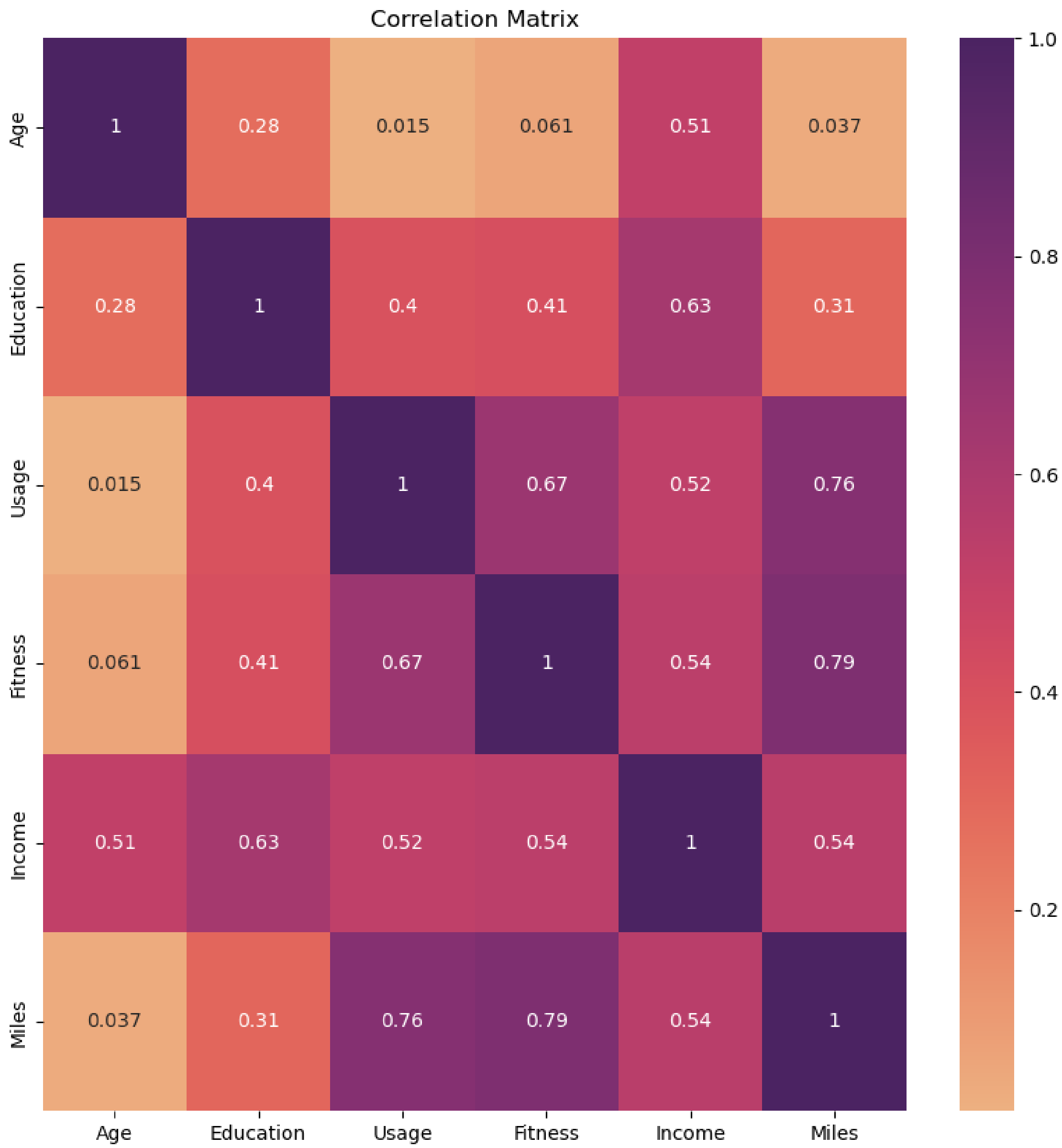
The probability of buying the following porgucts given they are Feamle are probability to buy: KP281 = 52.63% KP481 = 38.15% KP781 = 9.21%

The probability of buying the following porgucts given they are Male are probability to buy: KP281 = 38.46% KP481 = 29.80% KP781 = 31.71%

This shows Men are less biased towards the model of product and Women prefer the KP281 model.

1.5 Correlation among different factors

[15]:
```
correlation = df.corr(numeric_only=True)
plt.figure(figsize=(10, 10))
sns.heatmap(correlation, annot=True, cmap='flare')
plt.title('Correlation Matrix')
plt.show()
```

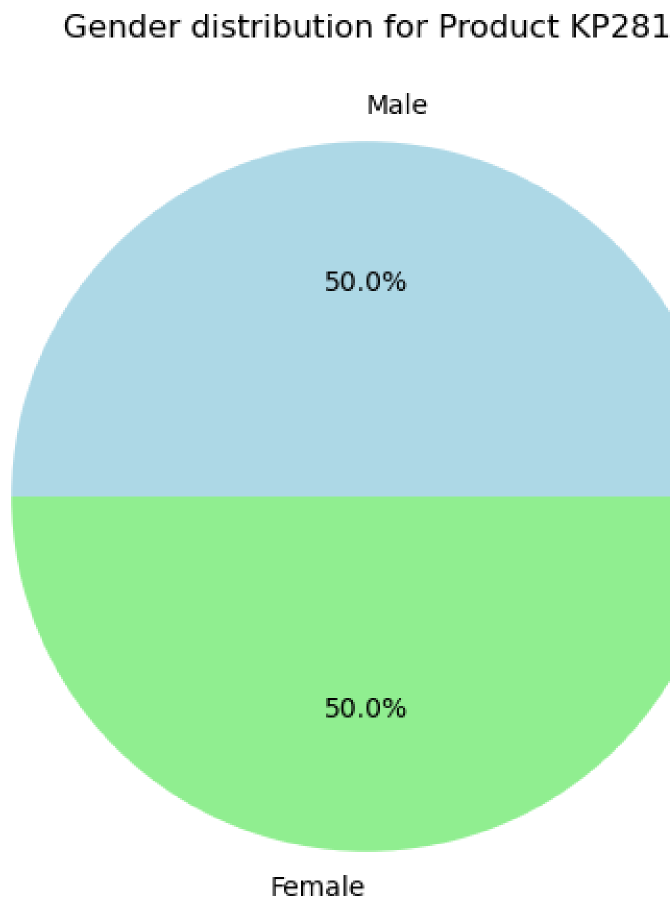


Some of the highly correlated data are: - Miles & Fitness - Miles & Usage - Usage & Fitness -

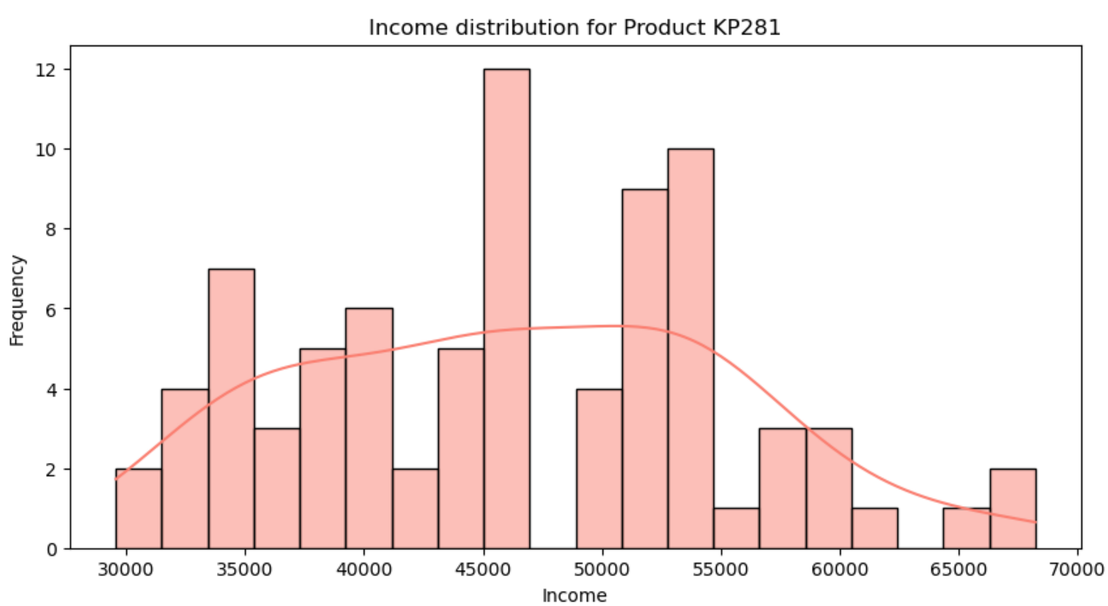
Income & Education

1.6 Customer Profiling and recomendation

[16]:
```
#Age Distribution
product_df = df[df['Product'] == 'KP281']
plt.figure(figsize=(10, 5))
sns.histplot(product_df['Age'], bins=20, kde=True, color='skyblue')
plt.title('Age Distribution for Product KP281')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
#Gender distribution
plt.figure(figsize=(6, 6))
product_df['Gender'].value_counts().plot(kind='pie', autopct='%1.1f%%',␣
↪colors=['lightblue', 'lightgreen'])
plt.title('Gender distribution for Product KP281')
plt.ylabel('')
plt.show()
#Income distribution
plt.figure(figsize=(10, 5))
sns.histplot(product_df['Income'], bins=20, kde=True, color='salmon')
plt.title('Income distribution for Product KP281')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.show()
```



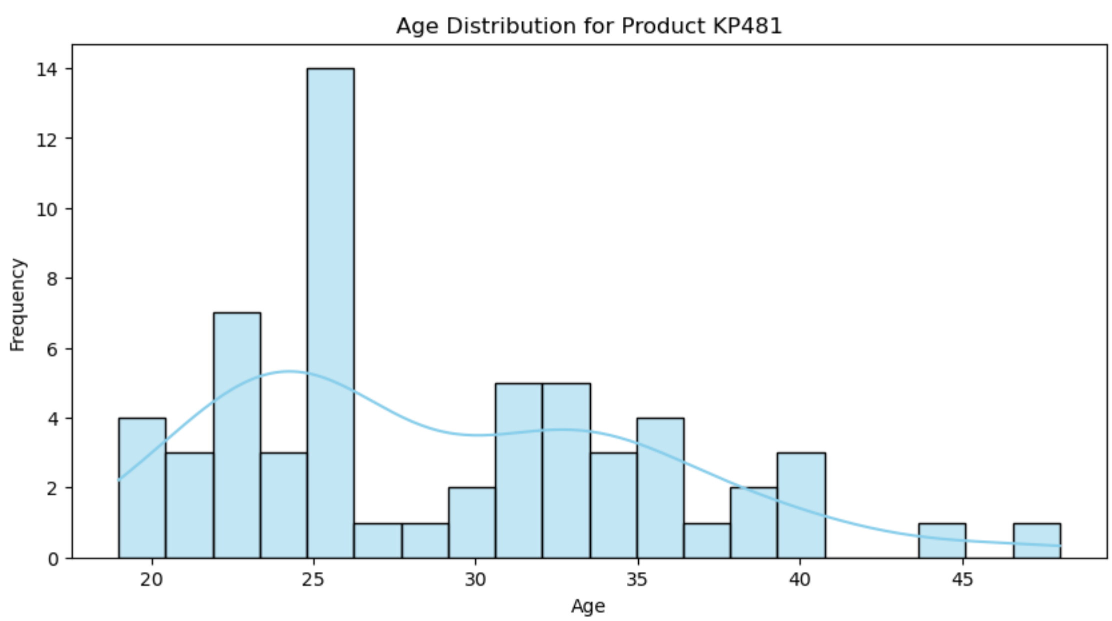
Gender distribution for Product KP281
Male
50.0%
50.0%
Female

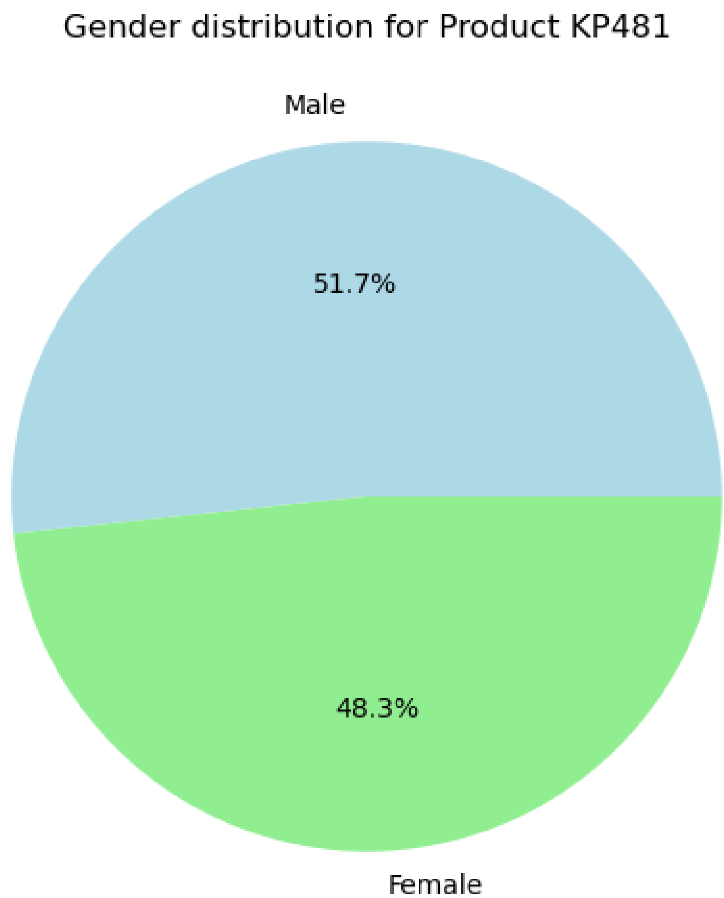


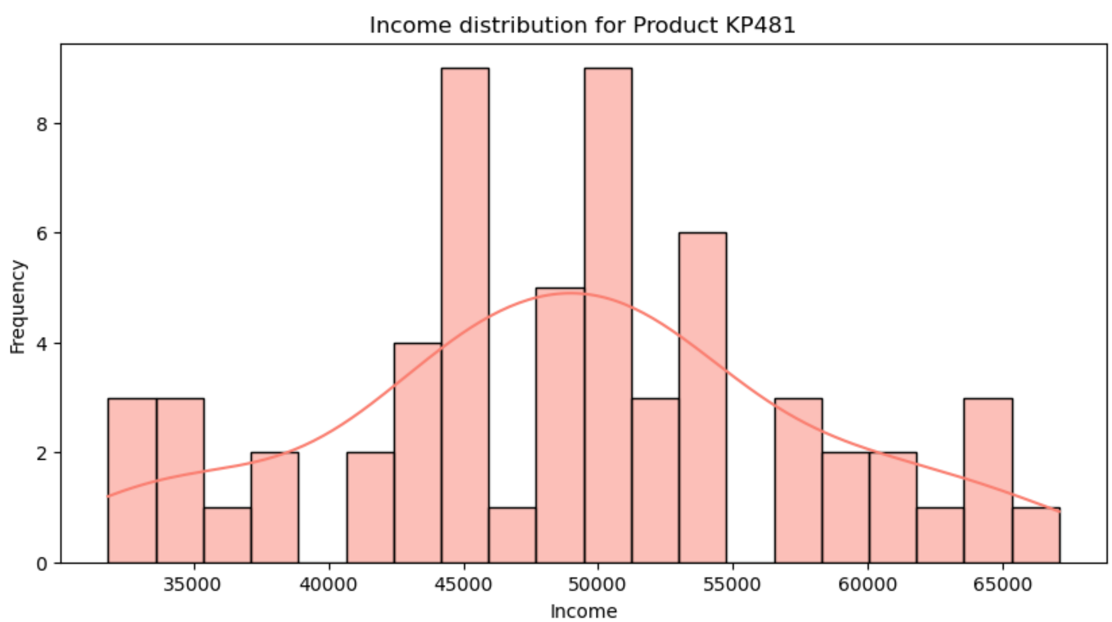
Income distribution for Product KP281
Frequency
Income
0
2
4
6
8
10
12
30000
35000
40000
45000
50000
55000
60000
65000
70000

[17]:

```
#Age Distribution
product_df = df[df['Product'] == 'KP481']
plt.figure(figsize=(10, 5))
sns.histplot(product_df['Age'], bins=20, kde=True, color='skyblue')
plt.title('Age Distribution for Product KP481')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
#Gender distribution
plt.figure(figsize=(6, 6))
product_df['Gender'].value_counts().plot(kind='pie', autopct='%1.1f%%',␣
↪colors=['lightblue', 'lightgreen'])
plt.title('Gender distribution for Product KP481')
plt.ylabel('')
plt.show()
#Income distribution
plt.figure(figsize=(10, 5))
sns.histplot(product_df['Income'], bins=20, kde=True, color='salmon')
plt.title('Income distribution for Product KP481')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.show()
```

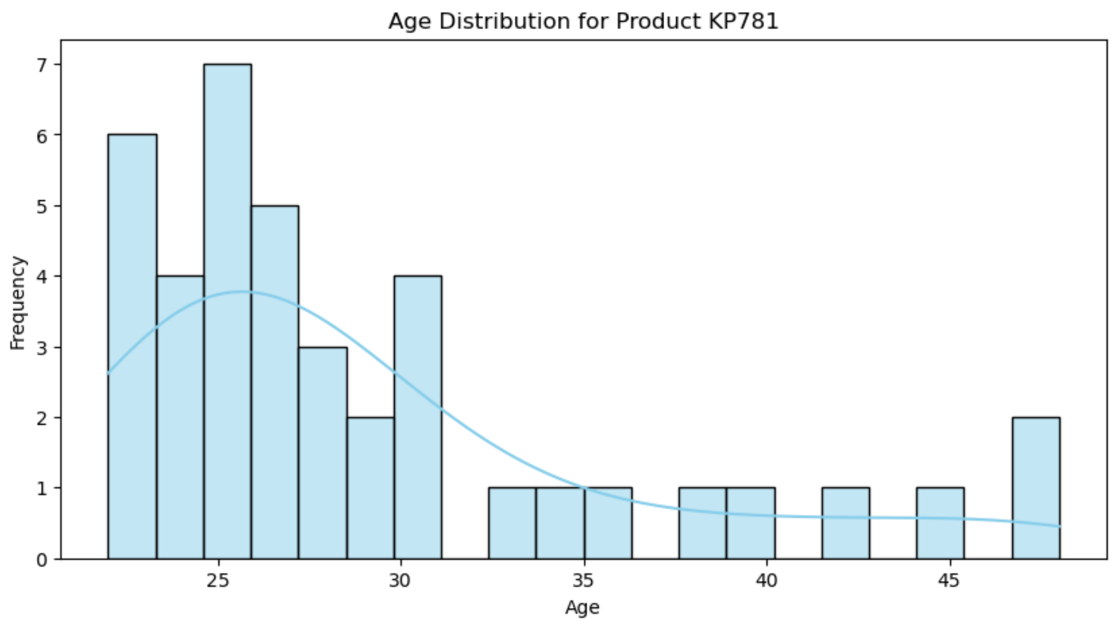


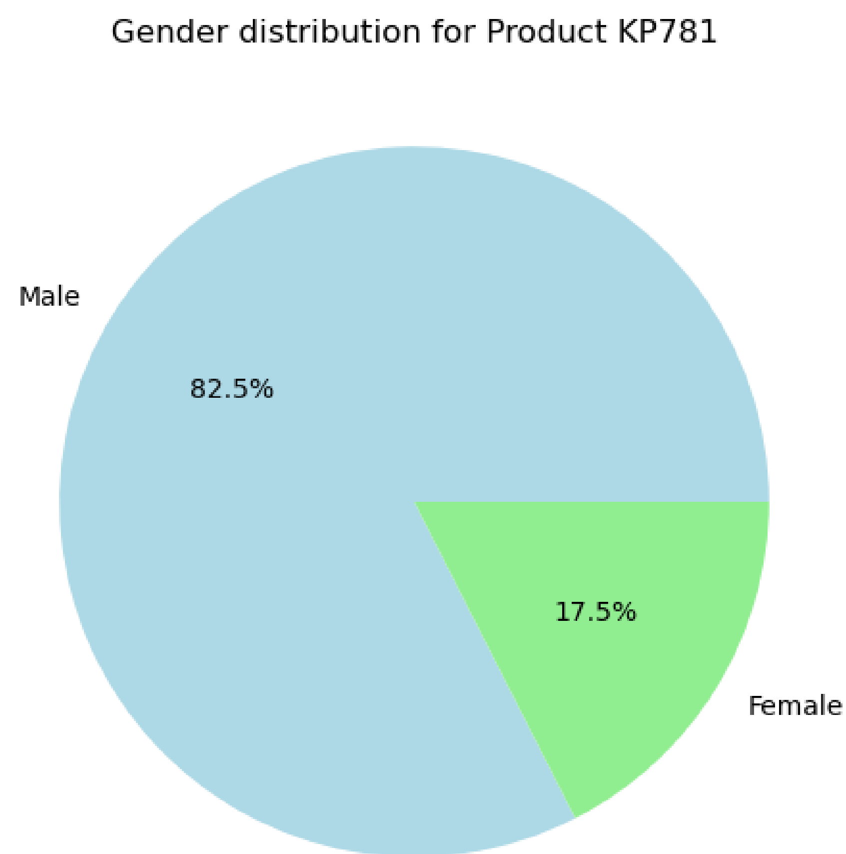

Gender distribution for Product KP481
Male
51.7%
48.3%
Female

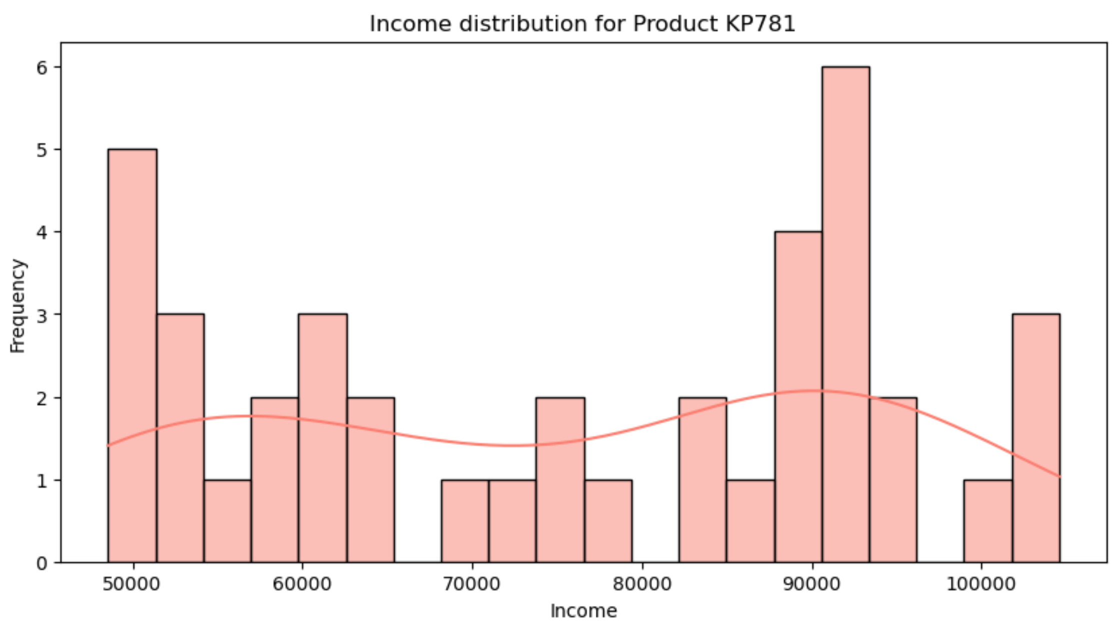

Income distribution for Product KP481
Frequency
0
2
4
6
8
35000
40000
45000
50000
55000
60000
65000
Income

[18]:

```
#Age Distribution
product_df = df[df['Product'] == 'KP781']
plt.figure(figsize=(10, 5))
sns.histplot(product_df['Age'], bins=20, kde=True, color='skyblue')
plt.title('Age Distribution for Product KP781')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
#Gender distribution
plt.figure(figsize=(6, 6))
product_df['Gender'].value_counts().plot(kind='pie', autopct='%1.1f%%',␣
↪colors=['lightblue', 'lightgreen'])
plt.title('Gender distribution for Product KP781')
plt.ylabel('')
plt.show()
#Income distribution
plt.figure(figsize=(10, 5))
sns.histplot(product_df['Income'], bins=20, kde=True, color='salmon')
plt.title('Income distribution for Product KP781')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.show()
```




Gender distribution for Product KP781
Male
82.5%
17.5%
Female


Income distribution for Product KP781
Frequency
0
1
2
3
4
5
6
50000
60000
70000
80000
90000
100000
Income

1.7 Recomendations

- Men are more likely to buy Fitness product so try marketing to women to expand your customer base.
- Married Coulples are more inclined to Fitness products so have target marketing to singles, this will help sales.
- KP281 is the most sold product across the board so focus on producing more KP281 to meet the demand.
- As the age increases the income also increases, hence older pople are more likely to buy more products.
- Women are more likly to buy the KP281 model and men dont particularly prefer a single model.
- Income is highly correlatd with all other criterias like Fitness, Usage and Education.