Business_Case_Aerofit

May 11, 2024

1 Business Case: Aerofit - Descriptive Statistics & Probability

#Aerofit

Aerofit is a leading brand in the field of fitness equipment. Aerofit provides a product range including machines such as treadmills, exercise bikes, gym equipment, and fitness accessories to cater to the needs of all categories of people.

Business Problem

The market research team at AeroFit wants to identify the characteristics of the target audience for each type of treadmill offered by the company, to provide a better recommendation of the treadmills to the new customers.

Importing the required Libraries

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[ ]:
```
!pip install pandas_profiling
```

[ ]:
```
from ydata_profiling import ProfileReport
```

Downloading the Aerofit Dataset

[ ]:
```
!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, 31.9MB/s]
```

[ ]:
```
df=pd.read_csv('/content/aerofit_treadmill.csv?1639992749')
print('Data Set read successfully')
```

```
Data Set read successfully
```

[ ]:
```
df
```

[ ]:
```
     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]
```

2 Analysing basic metrics of the Aerofit Dataset

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

[ ]: (180, 9)

[ ]:
```
df.size
```

[ ]: 1620

[ ]:
```
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
```

First 5 rows of the Dataset

```
[ ]: df.tail()
```

```
[ ]:     Product  Age Gender  Education MaritalStatus  Usage  Fitness  Income  \
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
175    200
176    200
177    160
178    120
179    180
```

Last five rows of Dataset

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

```
[ ]: 0
```

Observed that there is no duplicate values in the dataset

3 Data types of all the attributes

```
[ ]: df.dtypes
```

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

```
[ ]: 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
```

The above information shows that there is No Null values in the Dataset.

4 Statistical Information

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

[ ]:
```
              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   53719.577778
std      6.943498    1.617055    1.084797    0.958869   16506.684226
min     18.000000   12.000000    2.000000    1.000000   29562.000000
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  104581.000000

            Miles
count  180.000000
mean   103.194444
std     51.863605
min     21.000000
25%     66.000000
50%     94.000000
75%    114.750000
max    360.000000
```

[ ]:
```
df.describe(include=object)
```

[ ]:
```
      Product Gender MaritalStatus
count     180    180           180
```

```
unique       3      2            2
top      KP281   Male    Partnered
freq        80    104          107
```

[ ]:
```
ProfileReport(df)
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

[ ]:

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

[ ]:
```
Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  Miles
KP281    18   Male    14         Single         3      4        29562   112
1
KP481    30   Female  13         Single         4      3        46617   106
1
         31   Female  16         Partnered      2      3        51165   64
1
                      18         Single         2      1        65220   21
1
              Male    16         Partnered      3      3        52302   95
1
..
KP281    34   Female  16         Single         2      2        52302   66
1
              Male    16         Single         4      5        51165   169
1
         35   Female  16         Partnered      3      3        60261   94
1
                      18         Single         3      3        67083   85
1
KP781    48   Male    18         Partnered      4      5        95508   180
1
Name: count, Length: 180, dtype: int64
```

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

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

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

[ ]:
```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

[ ]:
```python
df['Income'].value_counts()
```

[ ]:
```
Income
45480    14
52302     9
46617     8
54576     8
53439     8
         ..
65220     1
55713     1
68220     1
30699     1
95508     1
Name: count, Length: 62, dtype: int64
```

[ ]:
```python
df.head(2)
```

[ ]:

	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

[ ]:
```python
def income_level(x):
  if x>=60000:
    return "High"
  elif x>=30000 and x<60000:
    return "Medium"
  else:
    return "Low"
```

[ ]:
```python
df["Income_Range"]=df['Income'].apply(income_level)
```

[ ]:
```python
df.head()
```

[ ]:
```
  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

   Miles  Income_Range
0    112           Low
1     75        Medium
2     66        Medium
3     85        Medium
4     47        Medium
```

[ ]:
```
bins=[18,25,35,50]

# creating labels for the bins
labels=['18-25','26-35','36-50']

#creating new column in df

df['Age_Range']=pd.cut(df['Age'],bins=bins,labels=labels,include_lowest=True)

df
```

[ ]:
```
    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 Income_Range Age_Range
0     112          Low     18-25
1      75       Medium     18-25
2      66       Medium     18-25
3      85       Medium     18-25
4      47       Medium     18-25
..    …          …       …
175   200         High     36-50
176   200         High     36-50
177   160         High     36-50
178   120         High     36-50
```

```
179    180          High     36-50

[180 rows x 11 columns]
```

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

[ ]:
```
              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   53719.577778
std      6.943498    1.617055    1.084797    0.958869   16506.684226
min     18.000000   12.000000    2.000000    1.000000   29562.000000
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  104581.000000

            Miles
count  180.000000
mean   103.194444
std     51.863605
min     21.000000
25%     66.000000
50%     94.000000
75%    114.750000
max    360.000000
```

[ ]:
```
df.describe(include=object)
```

[ ]:
```
       Product Gender MaritalStatus Income_Range
count      180    180           180          180
unique       3      2             2            3
top      KP281   Male     Partnered       Medium
freq        80    104           107          137
```

Statistical Summary

Age Distribution: The average age of the participants is approximately 28.79 years, with a minimum age of 18 years and a maximum age of 50 years. The majority of participants (25th to 75th percentile) fall within the age range of 24 to 33 years.

Education: Participants have a mean education level of approximately 15.57 years, with a minimum of 12 years and a maximum of 21 years. The interquartile range (25th to 75th percentile) for education level ranges from 14 to 16 years.

Usage and Fitness: On average, participants use fitness equipment 3.46 times per week and rate their fitness level at 3.31 on a scale of 1 to 5. The distribution of usage and fitness levels appears to be relatively consistent, with standard deviations of approximately 1.08 and 0.96, respectively.

Income: The average income of participants is approximately 53,719 USD per yaer, with a minimum income of 29,562 USD per year and a maximum income of 104,581 USD per year. Income levels vary widely among participants, with a standard deviation of approximately $16,506.

Miles: Participants cover an average distance of approximately 103.19 miles per week using fitness equipment, with a minimum of 21 miles and a maximum of 360 miles. The interquartile range for weekly miles ranges from 66 to 114.75 miles.

Product: There are three unique products. The product with code 'KP281' is the most frequently occurring product, appearing 80 times in the data.

Gender: There are two unique genders: Male and Female. Male is the most common gender, occurring 104 times in the data.

Marital Status: There are two unique marital statuses: Partnered and Single. Partnered is the most common marital status, occurring 107 times in the data.

Income Range: There are three unique income ranges: Low, Medium, and High. The Medium income range is the most common, appearing 137 times in the data.

5 Descriptive analytics to create a customer profile for each AeroFit treadmill product.

[ ]:
```
df['Product'].value_counts()
```

[ ]:
```
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
```

[ ]:
```
Product_KP281=df[df['Product']=="KP281"]
Product_KP281
```

[ ]:
```
Product_KP281.shape
```

[ ]:
```
(80, 11)
```

[ ]:
```
Product_KP481=df[df['Product']=="KP481"]
Product_KP481
```

[ ]:
```
Product_KP481.shape
```

[ ]:
```
(60, 11)
```

[ ]:
```
Product_KP781=df[df['Product']=="KP781"]
Product_KP781
```

[ ]:
```
Product_KP781.shape
```

[ ]: (40, 11)

```
[ ]: Product_KP281.head(2)
```

```
[ ]:   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

       Income_Range Age_Range
     0          Low     18-25
     1       Medium     18-25
```

6 Product_KP281

```
[ ]: Product_KP281.describe()
```

	Age	Education	Usage	Fitness	Income	Miles
count	80.000000	80.000000	80.000000	80.00000	80.00000	80.000000
mean	28.550000	15.037500	3.087500	2.96250	46418.02500	82.787500
std	7.221452	1.216383	0.782624	0.66454	9075.78319	28.874102
min	18.000000	12.000000	2.000000	1.00000	29562.00000	38.000000
25%	23.000000	14.000000	3.000000	3.00000	38658.00000	66.000000
50%	26.000000	16.000000	3.000000	3.00000	46617.00000	85.000000
75%	33.000000	16.000000	4.000000	3.00000	53439.00000	94.000000
max	50.000000	18.000000	5.000000	5.00000	68220.00000	188.000000

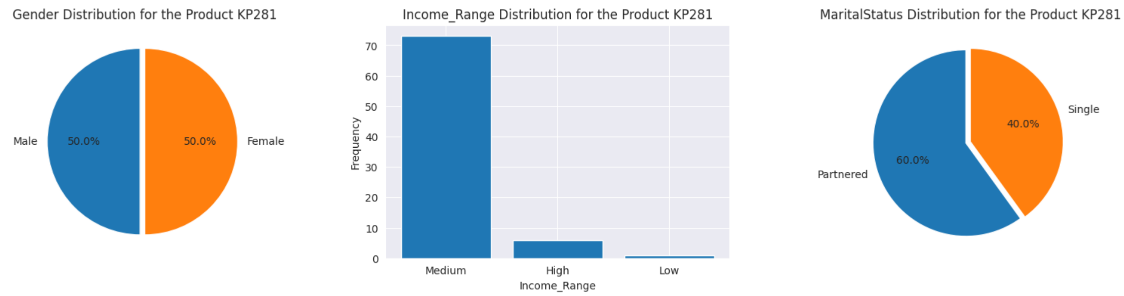
```
[ ]: Product_KP281.describe(include=object)
```

	Product	Gender	MaritalStatus	Income_Range
count	80	80	80	80
unique	1	2	2	3
top	KP281	Male	Partnered	Medium
freq	80	40	48	73

```
[ ]: P1_gender=Product_KP281['Gender'].value_counts()
     P1_Income_Range=Product_KP281['Income_Range'].value_counts()
     P1_MaritalStatus=Product_KP281['MaritalStatus'].value_counts()
```

```
[ ]: plt.figure(figsize=(20,4))

     # Gender Distribution
     plt.subplot(1,3,1)
     plt.pie(P1_gender,labels=P1_gender.index,autopct="%1.1f%%",startangle=90,␣
      ↪explode=[0.05,0])
     plt.title("Gender Distribution for the Product KP281")
```

```
#Income_Range Distribution
plt.subplot(1,3,2)
plt.bar(P1_Income_Range.index,P1_Income_Range.values)
plt.title("Income_Range Distribution for the Product KP281")
plt.xlabel("Income_Range")
plt.ylabel("Frequency")

#MaritalStatus Distribution
plt.subplot(1,3,3)
plt.pie(P1_MaritalStatus,labels=P1_MaritalStatus.index,autopct="%1.
 ↪1f%%",startangle=90,explode=[0.05,0])
plt.title("MaritalStatus Distribution for the Product KP281")

plt.show()
```



7 Insights:

- The purchase of Product KP281 shows gender parity, with an equal number of males and females making the purchase.
- Product KP281 exhibits a notable preference among customers with incomes falling within the medium range.
- Product KP281 demonstrates significant popularity among married individuals, who represent approximately 60% of its consumer base.

8 Product_KP481

[ ]:
```
Product_KP481.describe()
```

[ ]:

	Age	Education	Usage	Fitness	Income	Miles
count	60.000000	60.000000	60.000000	60.00000	60.000000	60.000000
mean	28.900000	15.116667	3.066667	2.90000	48973.650000	87.933333
std	6.645248	1.222552	0.799717	0.62977	8653.989388	33.263135
min	19.000000	12.000000	2.000000	1.00000	31836.000000	21.000000
25%	24.000000	14.000000	3.000000	3.00000	44911.500000	64.000000

```
50%     26.000000  16.000000   3.000000   3.00000  49459.500000   85.000000
75%     33.250000  16.000000   3.250000   3.00000  53439.000000  106.000000
max     48.000000  18.000000   5.000000   4.00000  67083.000000  212.000000
```

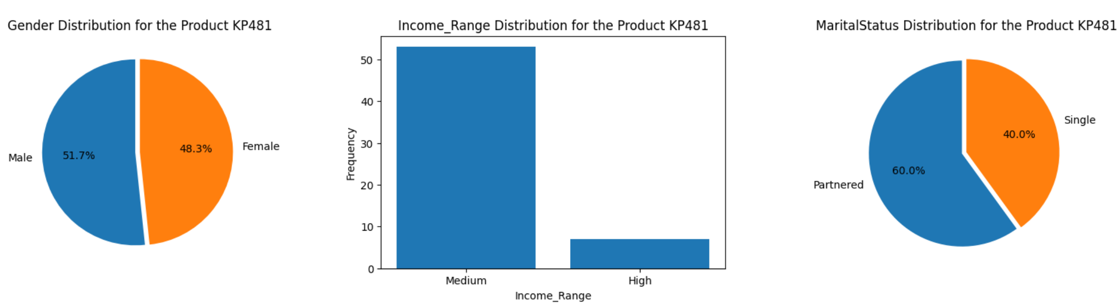
[ ]:
```
Product_KP481.describe(include=object)
```

[ ]:

	Product	Gender	MaritalStatus	Income_Range
count	60	60	60	60
unique	1	2	2	2
top	KP481	Male	Partnered	Medium
freq	60	31	36	53

[ ]:
```
P2_gender=Product_KP481['Gender'].value_counts()
P2_Income_Range=Product_KP481['Income_Range'].value_counts()
P2_MaritalStatus=Product_KP481['MaritalStatus'].value_counts()
```

[ ]:
```
plt.figure(figsize=(20,4))

# Gender Distribution
plt.subplot(1,3,1)
plt.pie(P2_gender,labels=P2_gender.index,autopct="%1.1f%%",startangle=90,␣
 ↪explode=[0.05,0])
plt.title("Gender Distribution for the Product KP481")

#Income_Range Distribution
plt.subplot(1,3,2)
plt.bar(P2_Income_Range.index,P2_Income_Range.values)
plt.title("Income_Range Distribution for the Product KP481")
plt.xlabel("Income_Range")
plt.ylabel("Frequency")

#MaritalStatus Distribution
plt.subplot(1,3,3)
plt.pie(P2_MaritalStatus,labels=P2_MaritalStatus.index,autopct="%1.
 ↪1f%%",startangle=90,explode=[0.05,0])
plt.title("MaritalStatus Distribution for the Product KP481")

plt.show()
```



- The purchase pattern for Product KP481 indicates a slight majority of male customers at 51.7%, while females constitute 48.3% of the buyers.
- Product KP481 enjoys significant popularity among customers with incomes categorized as medium, suggesting a strong appeal within this income range.
- Approximately 60% of the purchasers of Product KP481 are married individuals, signifying a notable preference among this demographic segment.

9 Product_KP781

[ ]:
```
Product_KP781.describe()
```

[ ]:

	Age	Education	Usage	Fitness	Income	Miles
count	40.000000	40.000000	40.000000	40.000000	40.00000	40.000000
mean	29.100000	17.325000	4.775000	4.625000	75441.57500	166.900000
std	6.971738	1.639066	0.946993	0.667467	18505.83672	60.066544
min	22.000000	14.000000	3.000000	3.000000	48556.00000	80.000000
25%	24.750000	16.000000	4.000000	4.000000	58204.75000	120.000000
50%	27.000000	18.000000	5.000000	5.000000	76568.50000	160.000000
75%	30.250000	18.000000	5.000000	5.000000	90886.00000	200.000000
max	48.000000	21.000000	7.000000	5.000000	104581.00000	360.000000

[ ]:
```
Product_KP781.describe(include=object)
```

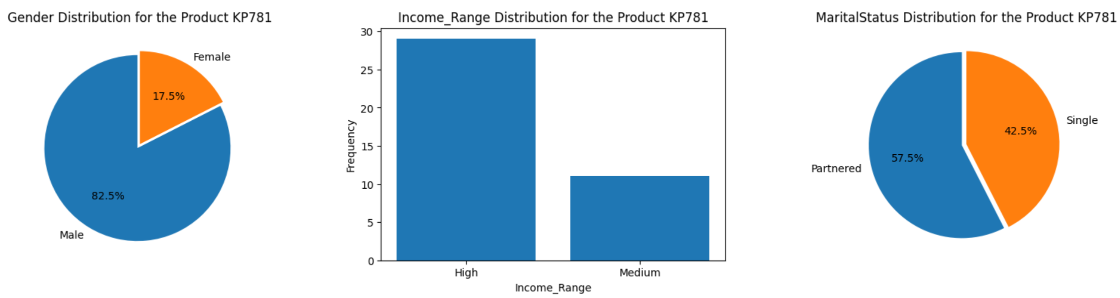
[ ]:

	Product	Gender	MaritalStatus	Income_Range
count	40	40	40	40
unique	1	2	2	2
top	KP781	Male	Partnered	High
freq	40	33	23	29

[ ]:
```
P3_gender=Product_KP781['Gender'].value_counts()
P3_Income_Range=Product_KP781['Income_Range'].value_counts()
P3_MaritalStatus=Product_KP781['MaritalStatus'].value_counts()
```

[ ]:
```
plt.figure(figsize=(20,4))

# Gender Distribution
plt.subplot(1,3,1)
plt.pie(P3_gender,labels=P3_gender.index,autopct="%1.1f%%",startangle=90,␣
↪explode=[0.05,0])
plt.title("Gender Distribution for the Product KP781")

#Income_Range Distribution
plt.subplot(1,3,2)
```

```
plt.bar(P3_Income_Range.index,P3_Income_Range.values)
plt.title("Income_Range Distribution for the Product KP781")
plt.xlabel("Income_Range")
plt.ylabel("Frequency")

#MaritalStatus Distribution
plt.subplot(1,3,3)
plt.pie(P3_MaritalStatus,labels=P3_MaritalStatus.index,autopct="%1.
↪1f%%",startangle=90,explode=[0.05,0])
plt.title("MaritalStatus Distribution for the Product KP781")

plt.show()
```

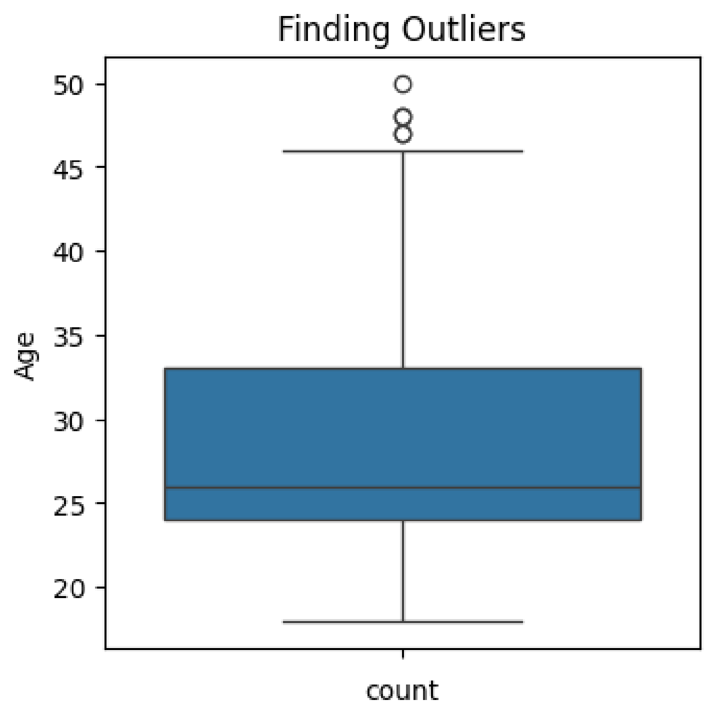


- The distribution of customers purchasing Product KP781 is notably skewed towards males, comprising 82.5% of buyers, while females account for 17.5% of purchases.
- Product KP781 exhibits a strong preference among customers with high incomes, indicating that a significant portion of its buyers belong to this income bracket.
- A substantial proportion, approximately 60%, of purchasers of Product KP781 are married individuals, highlighting the product's appeal within this demographic segment.

10 Detecting Outliers

[ ]:
```
plt.figure(figsize=(4,4))
sns.boxplot(df['Age'])
plt.title('Finding Outliers')
plt.xlabel("count")
plt.ylabel("Age")
```

[ ]: Text(0, 0.5, 'Age')



[ ]:
```
df['Age'].describe()
```

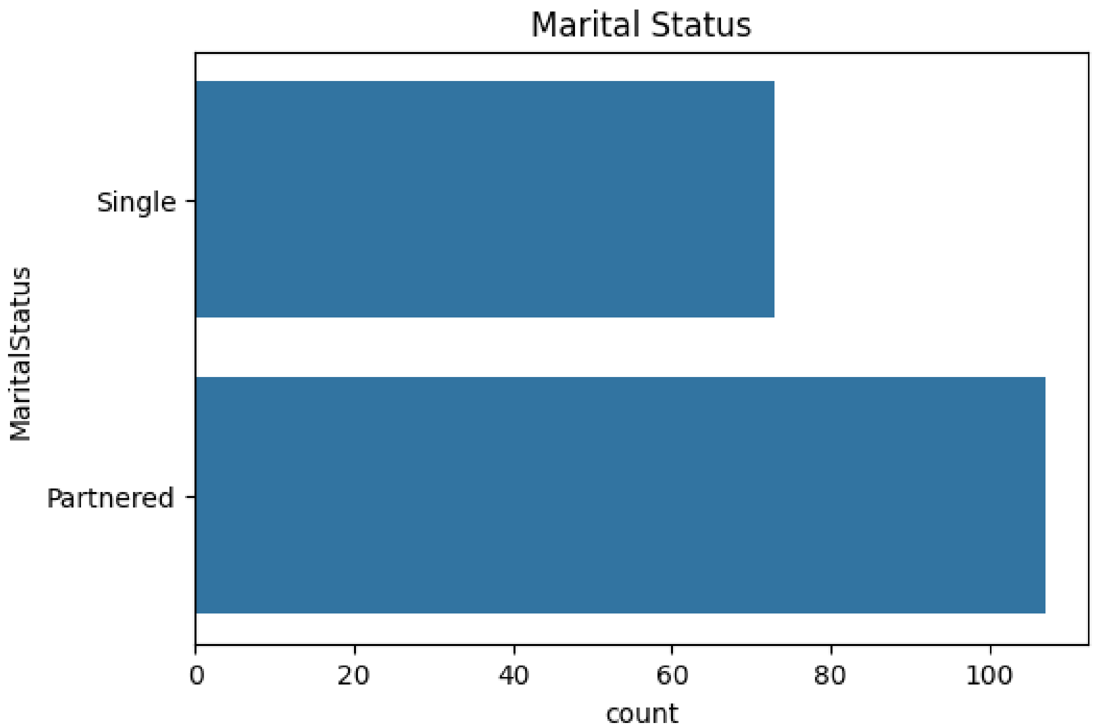
[ ]:
```
count    180.000000
mean      28.788889
std        6.943498
min       18.000000
25%       24.000000
50%       26.000000
75%       33.000000
max       50.000000
Name: Age, dtype: float64
```

[ ]:
```
df['Age'].mean()-df['Age'].median()
```

[ ]:
```
2.788888888888888
```

[ ]:
```
plt.figure(figsize=(6,4))
sns.countplot(df['MaritalStatus'])
plt.title("Marital Status")
```

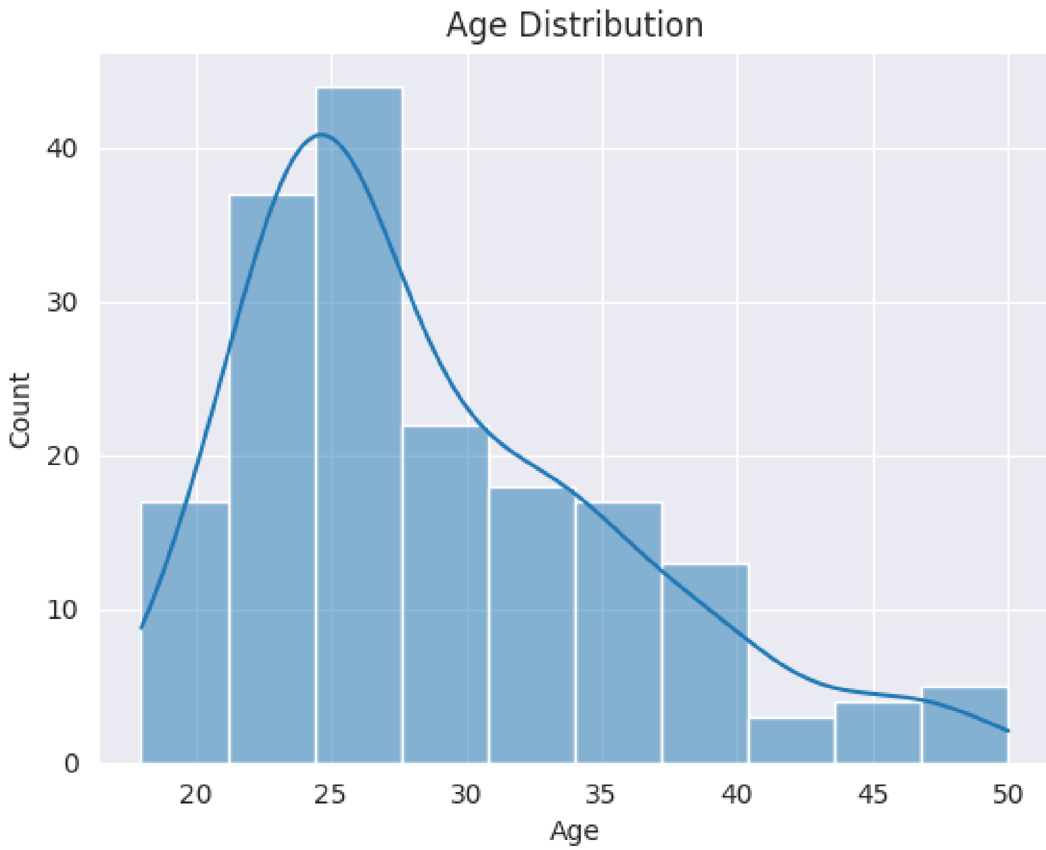
[ ]:
```
Text(0.5, 1.0, 'Marital Status')
```



Observed that Married customers show a higher purchasing propensity compared to single customers.

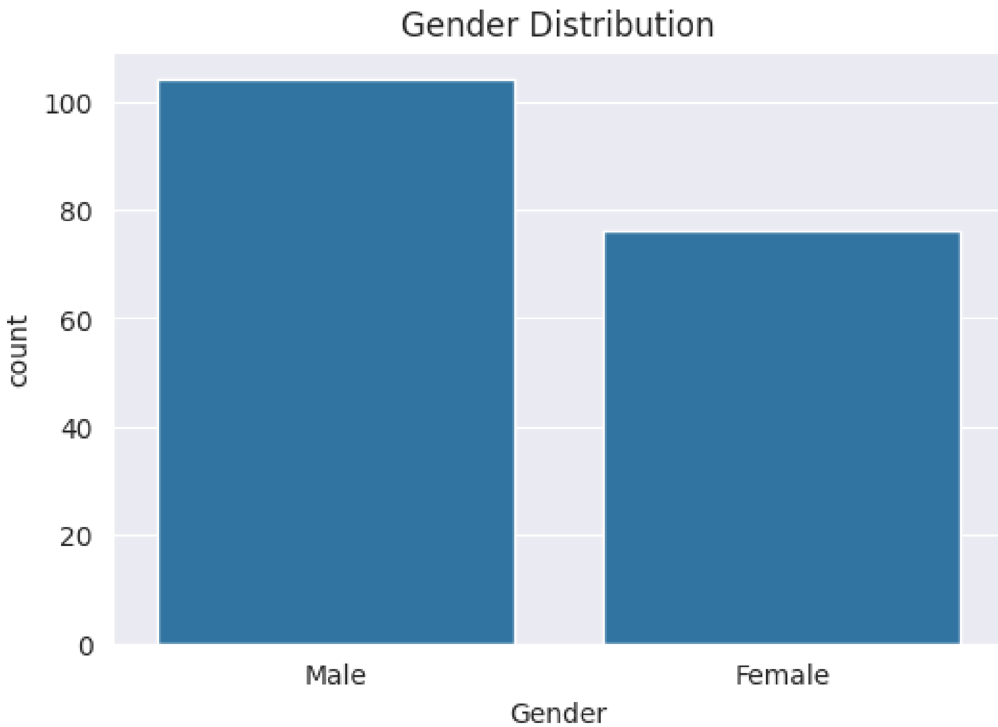
[ ]:
```
sns.set_style("darkgrid")
sns.histplot(df['Age'],bins=10,kde=True)
plt.title("Age Distribution")
```

[ ]: Text(0.5, 1.0, 'Age Distribution')



[ ]:
```
plt.figure(figsize=(6,4))
sns.countplot(x=df['Gender'])
plt.title("Gender Distribution")
```

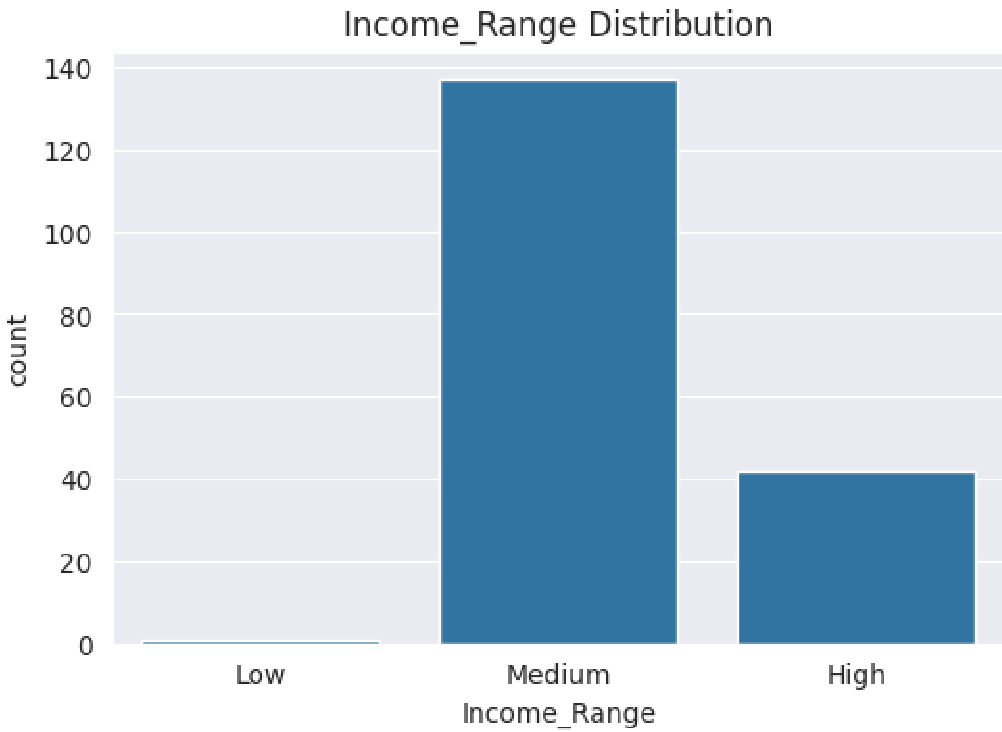
[ ]: Text(0.5, 1.0, 'Gender Distribution')



Observed that Male customers show a higher purchasing propensity compared to Female customers

[ ]:
```
plt.figure(figsize=(6,4))
sns.countplot(x=df['Income_Range'])
plt.title("Income_Range Distribution")
```

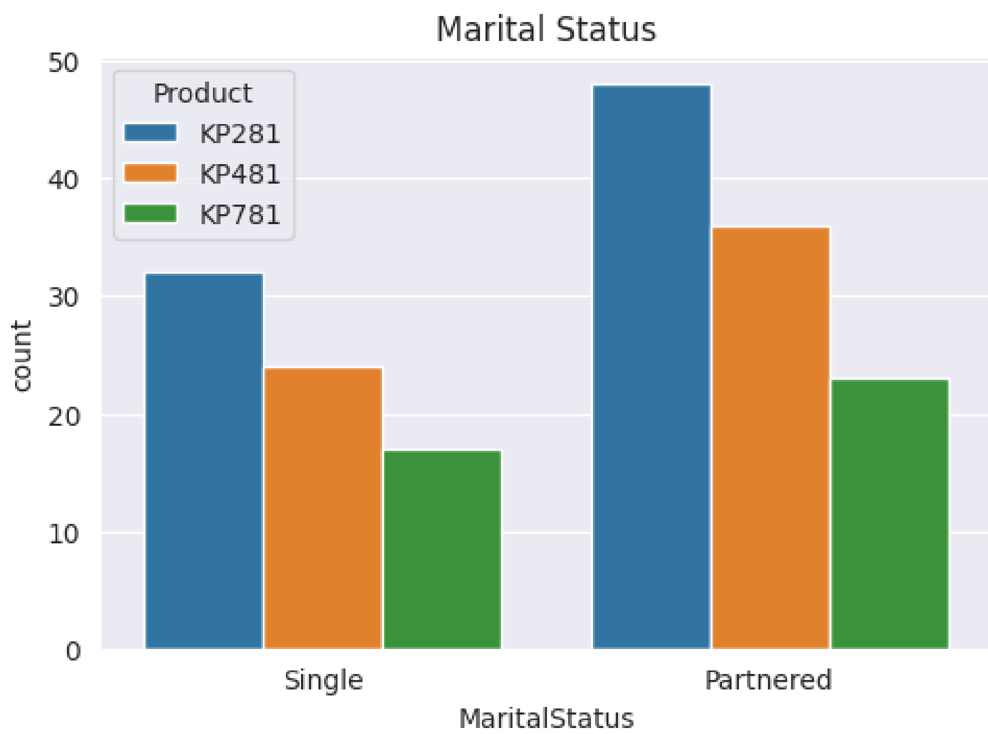
[ ]:
```
Text(0.5, 1.0, 'Income_Range Distribution')
```



Customers who have income range between 30000 and 60000 are highly purchasing these products

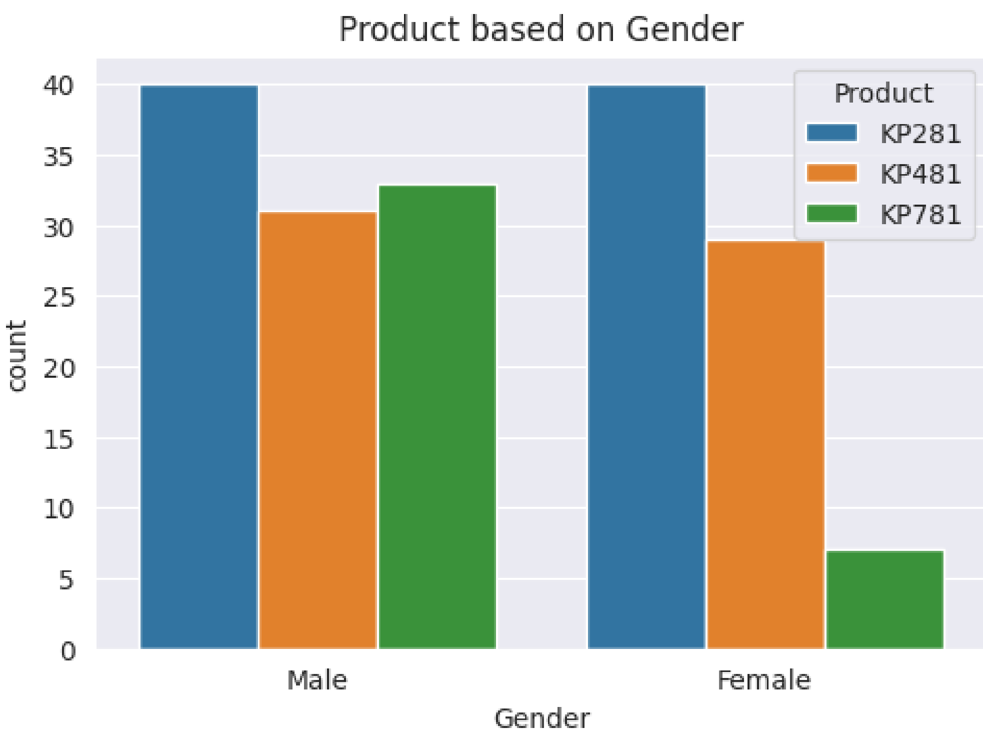
[ ]:
```
plt.figure(figsize=(6,4))
sns.countplot(x=df['MaritalStatus'],hue=df['Product'])
plt.title("Marital Status")
```

[ ]:
```
Text(0.5, 1.0, 'Marital Status')
```



[ ]:
```
plt.figure(figsize=(6,4))
sns.countplot(x=df['Gender'],hue=df['Product'])
plt.title("Product based on Gender")
```

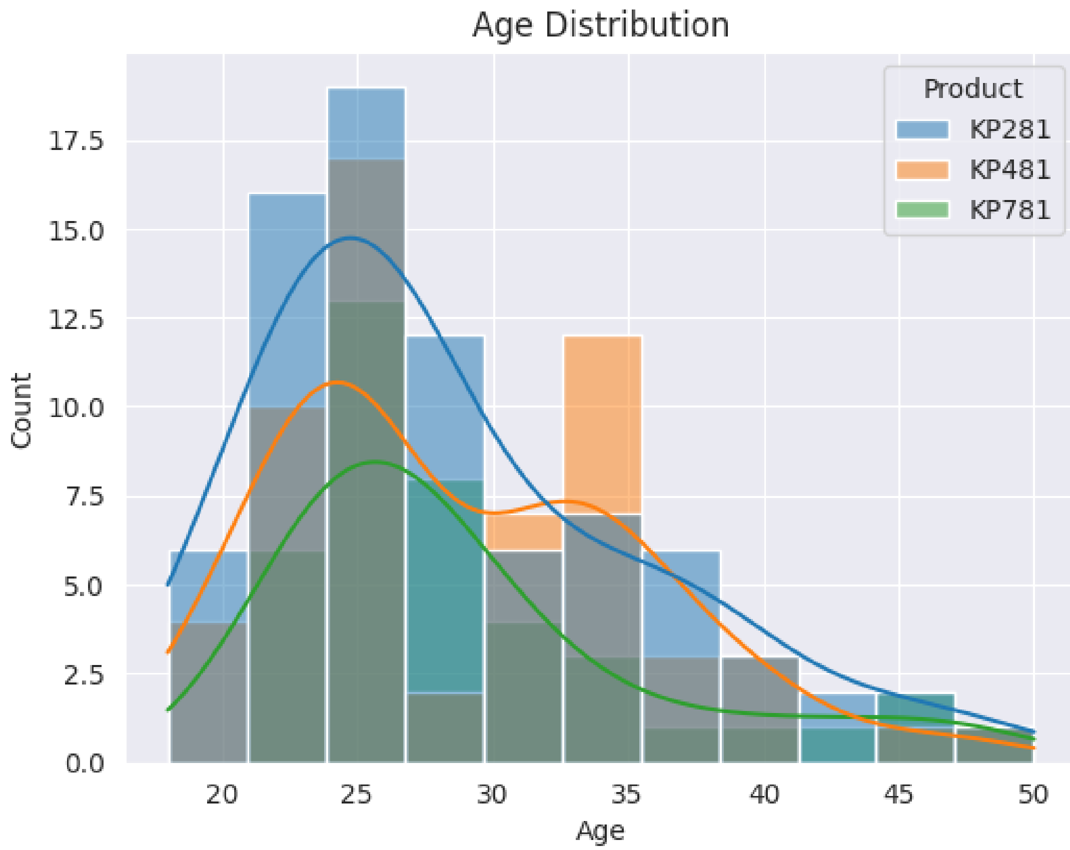
[ ]: Text(0.5, 1.0, 'Product based on Gender')



KP781 treadmill is having advanced feature which is highly puchased by male compared to female

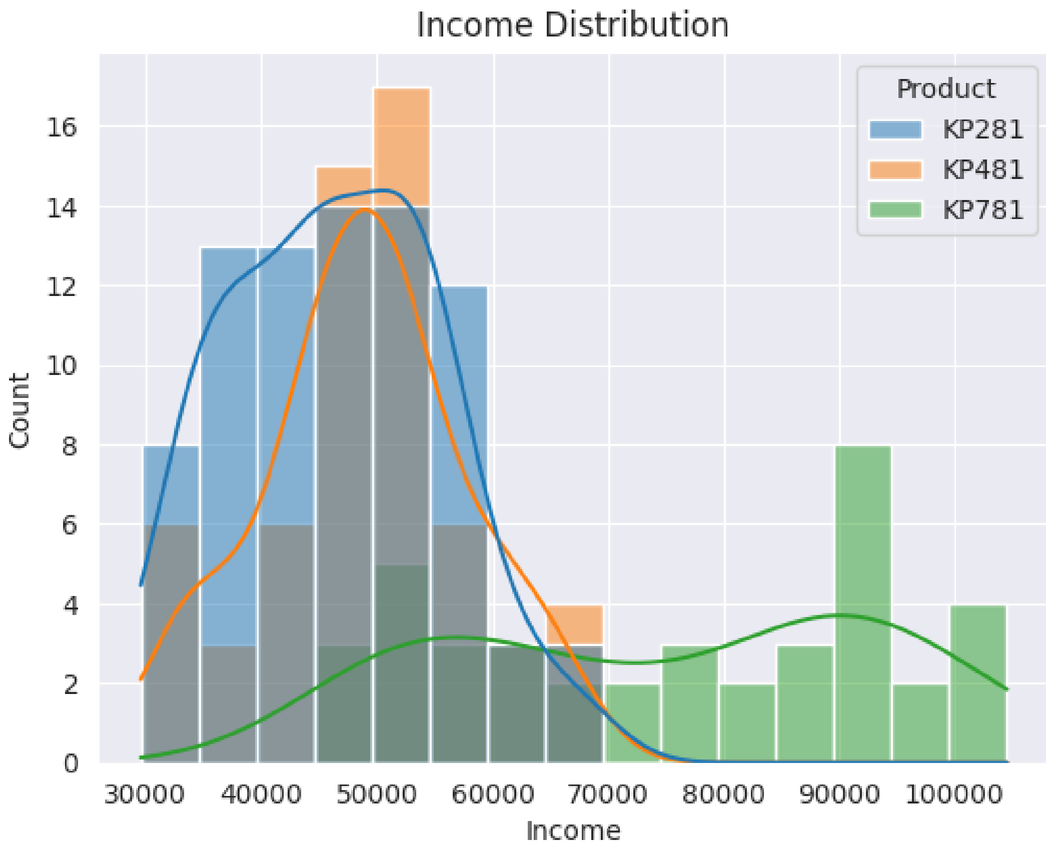
[ ]:
```
sns.set_style("darkgrid")
sns.histplot(data=df,x='Age',hue='Product',kde=True)
plt.title("Age Distribution")
```

[ ]: Text(0.5, 1.0, 'Age Distribution')




```
[ ]: sns.set_style("darkgrid")
     sns.histplot(data=df,x='Income',hue='Product',kde=True)
     plt.title("Income Distribution")
```

```
[ ]: Text(0.5, 1.0, 'Income Distribution')
```

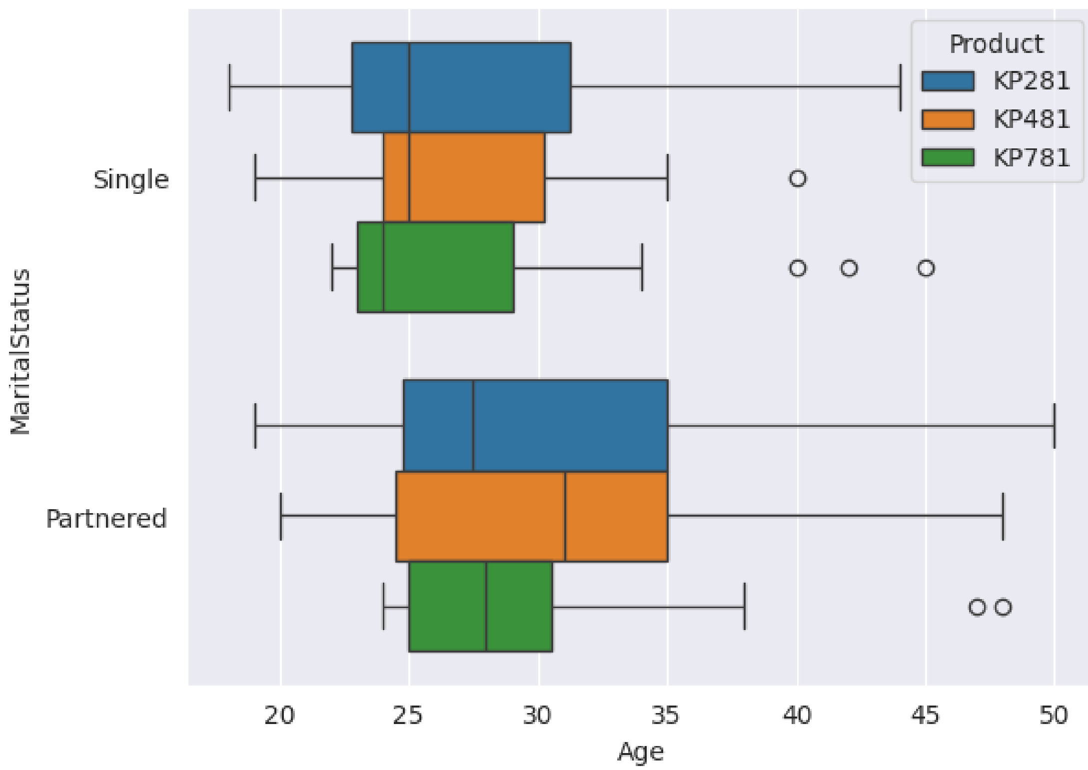


[ ]:
```
sns.set_style("darkgrid")
sns.histplot(data=df,x='Income_Range',hue='Product',kde=True)
plt.title("Income_Range Distribution")
```

[ ]: Text(0.5, 1.0, 'Income_Range Distribution')

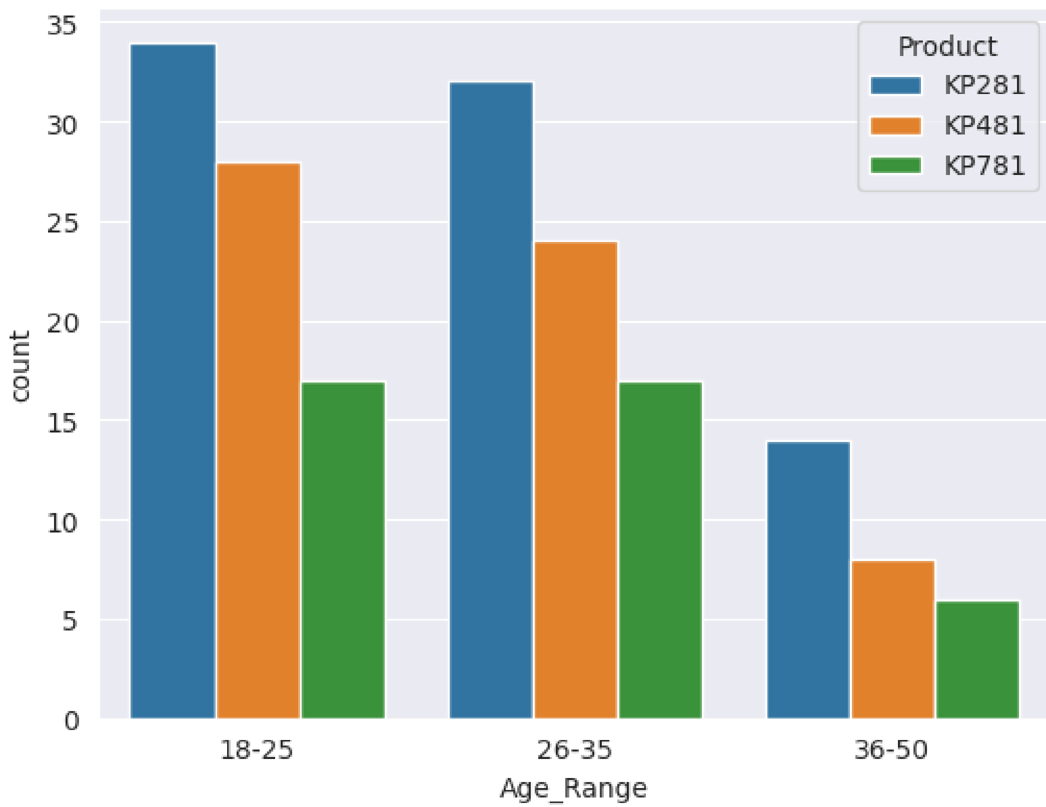
[ ]:
```
sns.boxplot(data=df,x='Age',y='MaritalStatus',hue='Product')
```

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




```
[ ]: sns.countplot(data=df,x='Age_Range',hue='Product')
```

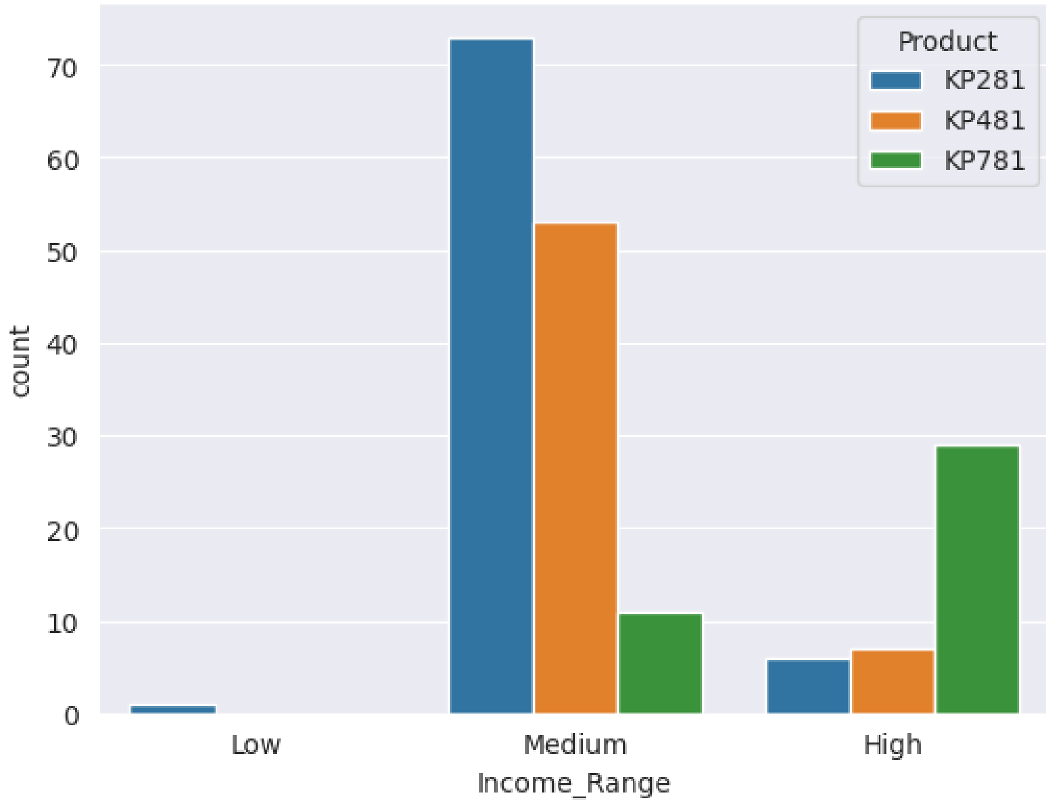
```
[ ]: <Axes: xlabel='Age_Range', ylabel='count'>
```



Customers in the age range of 18 to 25 exhibit a strong propensity for purchasing these products.

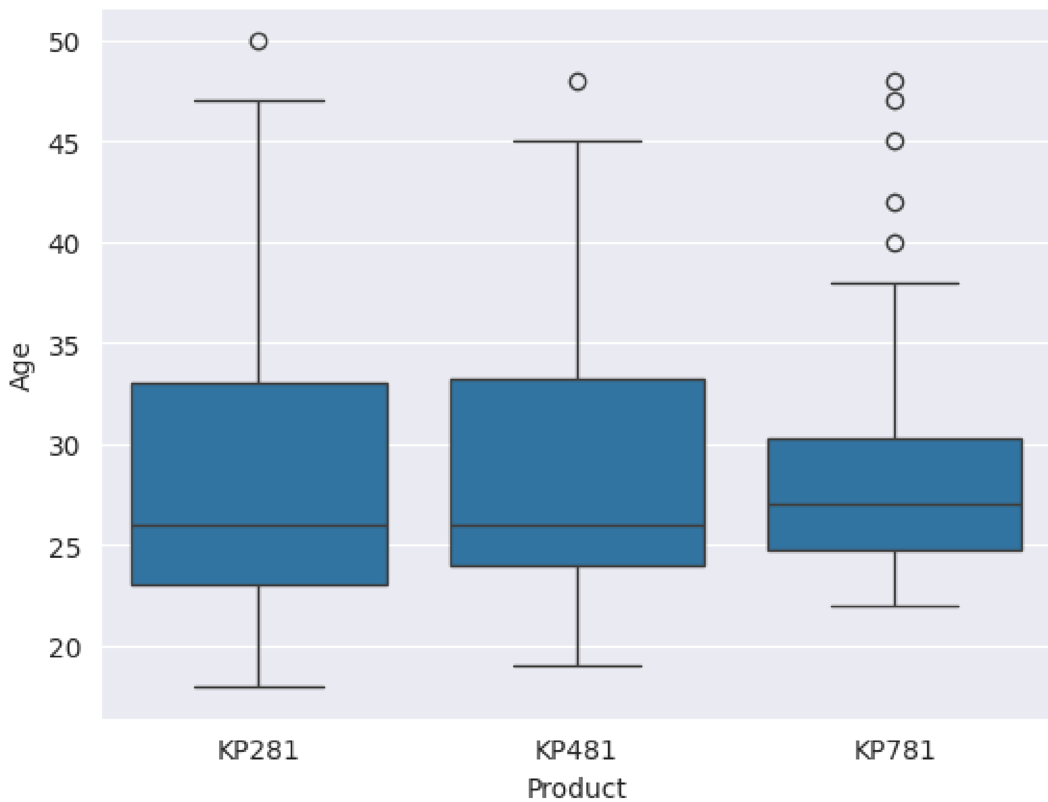
[ ]:
```
sns.countplot(data=df,x='Income_Range',hue='Product')
```

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



[ ]:
```
sns.boxplot(data=df,x='Product',y='Age')
```

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




```
[ ]: sns.pairplot(df)
```

```
[ ]: <seaborn.axisgrid.PairGrid at 0x790419445810>
```

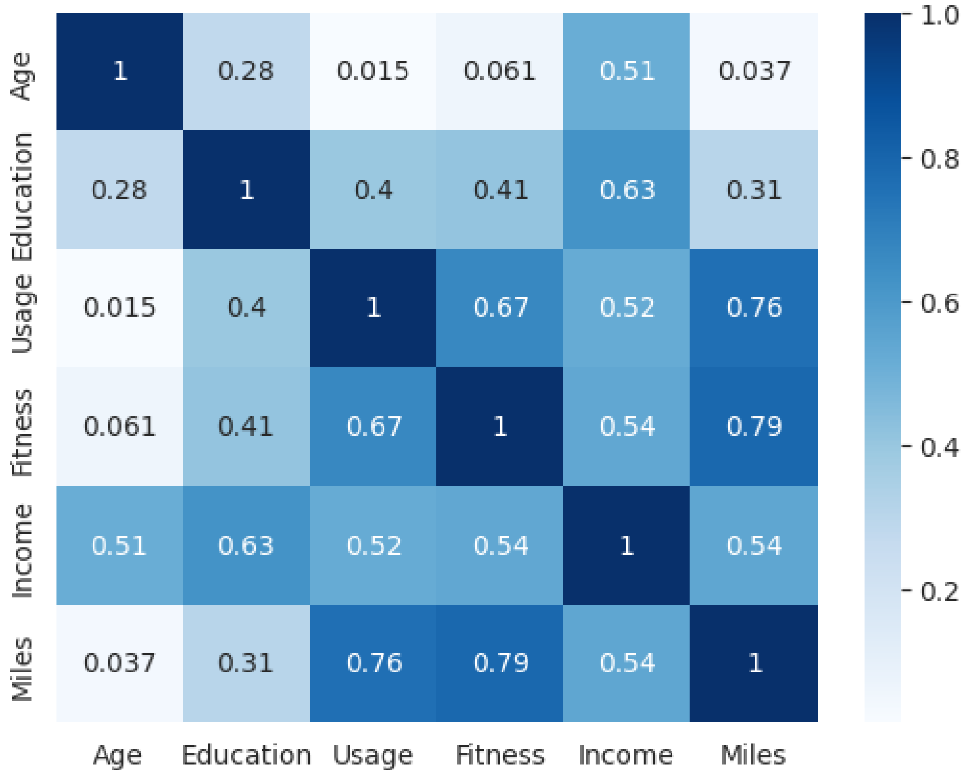
[ ]:
```python
c_df=df.select_dtypes(include=['number'])
c_df.corr()
```

[ ]:

	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

[ ]:
```python
sns.heatmap(c_df.corr(),cmap='Blues',annot=True)
```

[ ]: <Axes: >



- Age is highly overall correlated with Income
- Education is highly overall correlated with Income
- Fitness is highly overall correlated with Miles
- Income is highly overall correlated with Age

11 conditional and marginal probabilities

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

[ ]:
```
Product           KP281     KP481     KP781
MaritalStatus
Partnered      0.266667  0.200000  0.127778
Single         0.177778  0.133333  0.094444
```

[ ]:
```
pd.crosstab(index=df['Age_Range'],columns=df['Product'],normalize=True)
```

[ ]:
```
Product       KP281     KP481     KP781
Age_Range
18-25      0.188889  0.155556  0.094444
```

```
26-35      0.177778  0.133333  0.094444
36-50      0.077778  0.044444  0.033333
```

[ ]:
```python
pd.crosstab(index=df['Income_Range'],columns=df['Product'],normalize=True)
```

[ ]:
```
Product         KP281     KP481     KP781
Income_Range
High         0.033333  0.038889  0.161111
Low          0.005556  0.000000  0.000000
Medium       0.405556  0.294444  0.061111
```

[ ]:
```python
pd.crosstab(index=df['Gender'],columns=df['Product'],normalize=True)
```

[ ]:
```
Product     KP281     KP481     KP781
Gender
Female   0.222222  0.161111  0.038889
Male     0.222222  0.172222  0.183333
```

[ ]:
```python
pd.crosstab(index=df['Gender'],columns=df['Product'])
```

[ ]:
```
Product  KP281  KP481  KP781
Gender
Female      40     29      7
Male        40     31     33
```

[ ]:
```python
# What is the probability of a male customer buying a KP781 treadmill?

33/40
```

[ ]:
```
0.825
```

12 Insights

The Age attribute ranges from 18 to 50, which indicates that the data represents a sample of individuals aged between 18 and 50 years. The Education attribute ranges from 12 to 18, which indicates that the individuals in the dataset have completed education ranging from high school to college. The Income attribute ranges from 29562 to 104581, indicating a wide range of income levels.

The Miles attribute ranges from 38 to 188, which indicates that individuals in the dataset are using the fitness equipment for varying distances. The Usage attribute ranges from 2 to 5, which indicates how frequently individuals use the fitness equipment. The Fitness attribute ranges from 1 to 5, indicating the level of fitness of individuals in the dataset.

Overall, the range of attributes in this dataset is quite diverse, which may provide valuable insights into the behavior and characteristics of customers who purchase fitness equipment.

13 Recommendations based on the analysis of the customer data for Aerofit:

- Focus marketing efforts on promoting the KP781 model to customers in the high and medium income brackets. The data indicates that customers with higher incomes are more inclined to purchase higher-end products like the KP781. By tailoring marketing campaigns to these income ranges, Aerofit can capitalize on the purchasing power of these segments and potentially boost sales of the KP781 model.
- Direct marketing efforts towards married individuals, as they show a higher likelihood of purchasing Aerofit products compared to unmarried individuals. This could involve creating targeted advertisements or campaigns that highlight the benefits of using Aerofit products for couples, such as promoting fitness as a shared activity. Additionally, considering gender preferences in marketing strategies can further enhance the effectiveness of these campaigns.
- Develop gender-specific marketing campaigns for the KP281 and KP781 models based on the observed preferences. Females show a greater inclination towards the KP281 model, while males prefer the KP781. By tailoring marketing messages and highlighting features that resonate with each gender, Aerofit can effectively engage with its target audience and drive sales for both models. These recommendations aim to leverage the insights gained from the analysis of customer data to optimize marketing strategies and enhance sales performance for Aerofit's treadmill products.