About 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. The team decides to investigate whether there are differences across the product with respect to customer characteristics.

Perform descriptive analytics to create a customer profile for each AeroFit treadmill product by developing appropriate tables and charts. For each AeroFit treadmill product, construct two-way contingency tables and compute all conditional and marginal probabilities along with their insights/impact on the business.

Dataset

The company collected the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months. The dataset has the following features:

- Product Purchased: KP281, KP481, or KP781
- Age: In years
- Gender: Male/Female
- Education: In years
- MaritalStatus: Single or partnered
- Usage: The average number of times the customer plans to use the treadmill each week.
- Income: Annual income (in $)
- Fitness: Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 is the excellent shape.
- Miles: The average number of miles the customer expects to walk/run each week

Product Portfolio:

- The KP281 is an entry-level treadmill that sells for $1,500. * The KP481 is for mid-level runners that sell for 1,750.
- The KP781 treadmill is having advanced features that sell for $2,500.

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

```
!gdown 1fZMfXRvPISsZjrZod73RWsRVlG38ZnvO
```

```
Downloading...
From: https://drive.google.com/uc?id=1fZMfXRvPISsZjrZod73RWsRVlG38ZnvO
To: /content/aerofit_treadmill.csv
100% 7.28k/7.28k [00:00<00:00, 27.8MB/s]
```

```
data = pd.read_csv("aerofit_treadmill.csv")
```

```
data.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

Next steps: Generate code with `data` | View recommended plots | New interactive sheet

```
data.shape
```

```
(180, 9)
```

```
data.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
```

```
(data.isnull()).sum()
```

	0
Product	0
Age	0
Gender	0
Education	0
MaritalStatus	0
Usage	0
Fitness	0
Income	0
Miles	0

dtype: int64

```
data.columns
```

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

```
data["Age"].describe()
```

	Age
count	180.000000
mean	28.788889
std	6.943498
min	18.000000
25%	24.000000
50%	26.000000
75%	33.000000
max	50.000000

dtype: float64

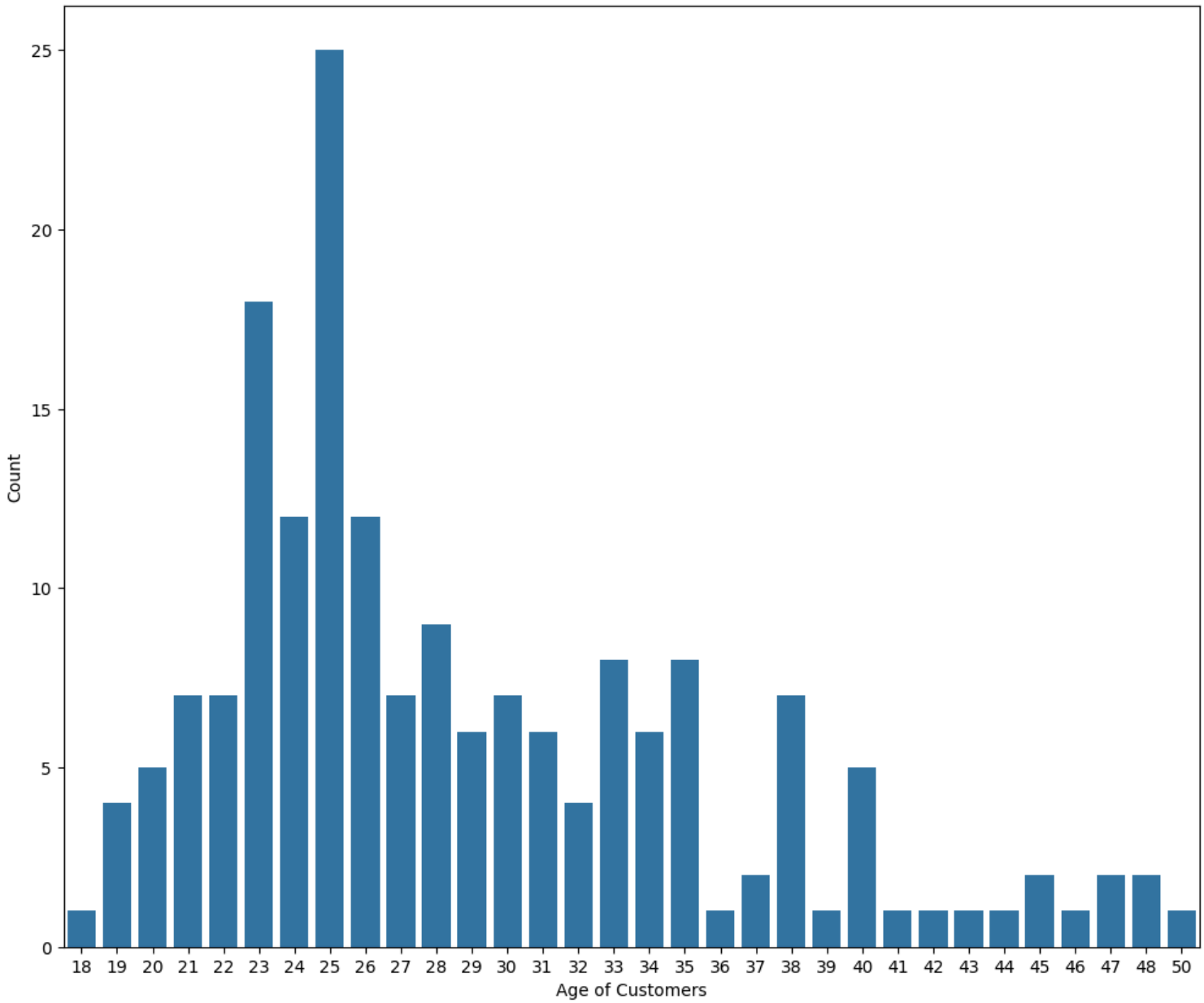
```
data["Age"].value_counts()[:10]
x = data["Age"].value_counts().index
y = data["Age"].value_counts().values
```

```
data["Age"].value_counts()[:10]
```

Age	count
25	25
23	18
24	12
26	12
28	9
35	8
33	8
30	7
38	7
21	7

dtype: int64

```
plt.figure(figsize=(12, 10))
sns.countplot(data=data, x = data["Age"])
plt.xlabel("Age of Customers")
plt.ylabel("Count")
plt.show()
```




```
np.mean(data["Income"])
```

```
53719.57777777778
```

```
np.median(data["Income"])
```

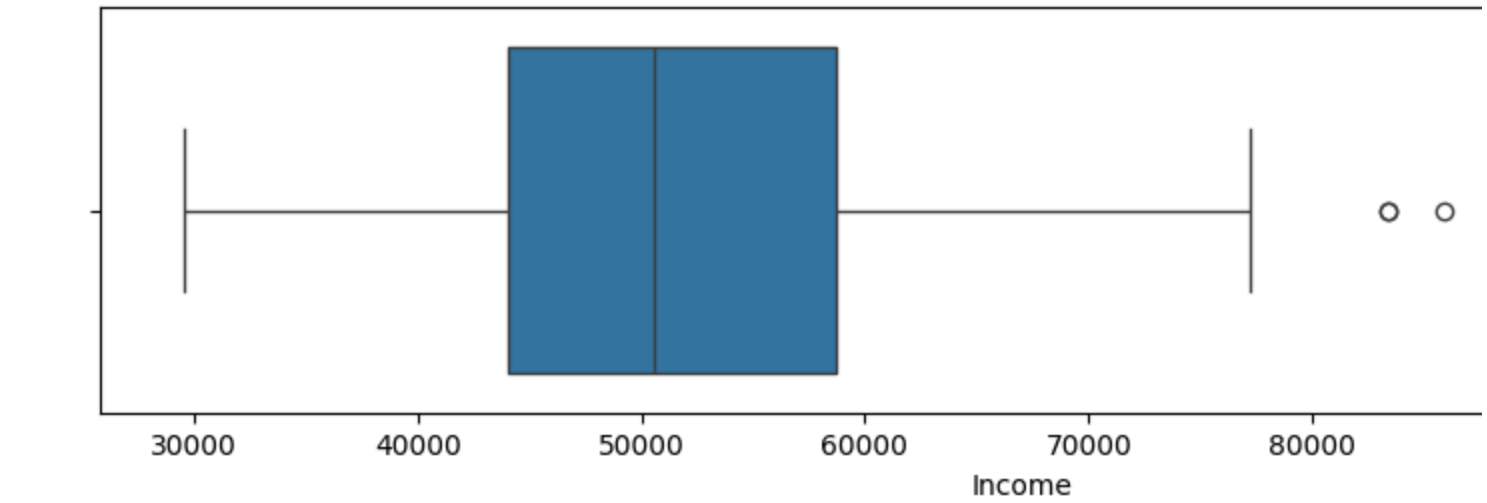
```
50596.5
```

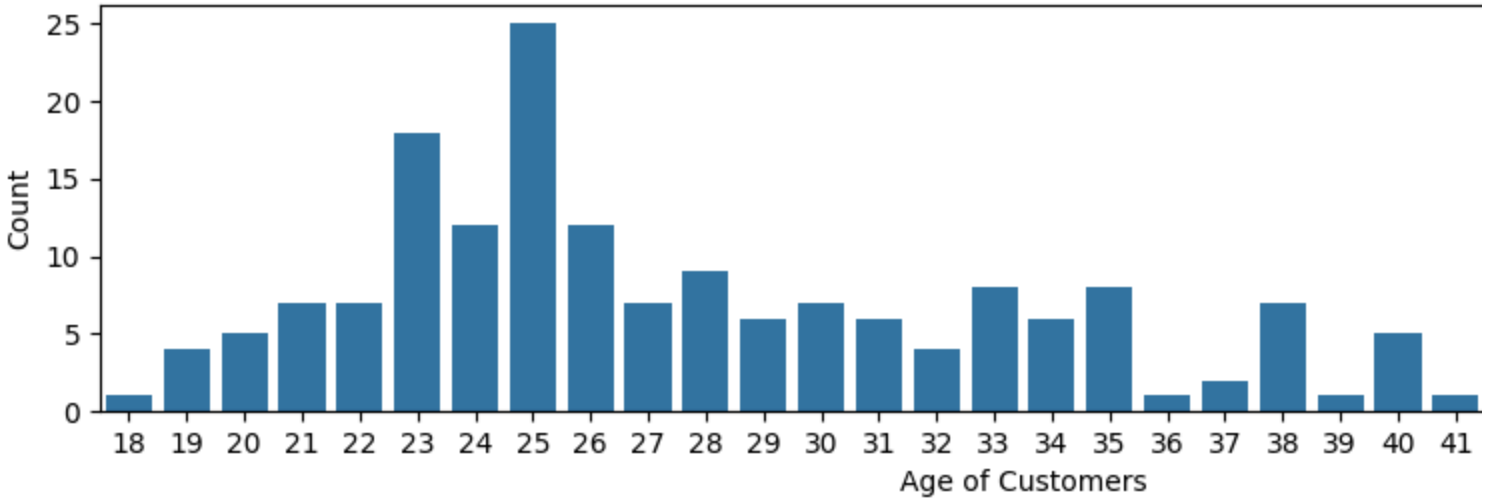
```
data["Income"].describe()
```

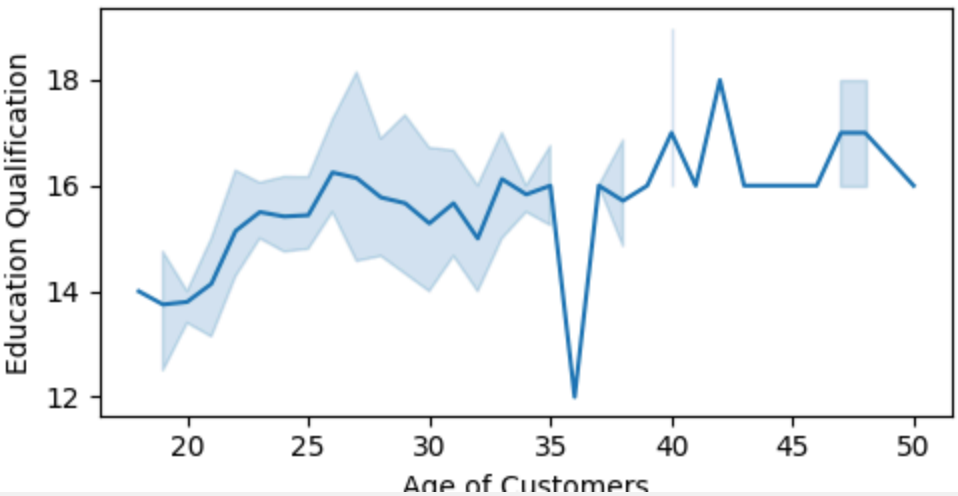
	Income
count	180.000000
mean	53719.577778
std	16506.684226
min	29562.000000
25%	44058.750000
50%	50596.500000
75%	58668.000000
max	104581.000000

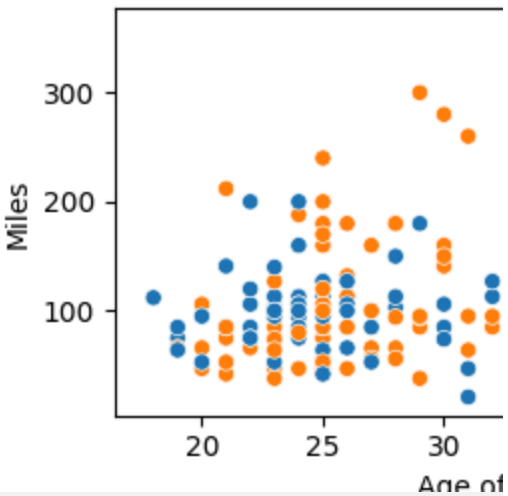
dtype: float64

```
plt.figure(figsize=(10,8))

plt.subplot(3,2, (1, 2))
sns.boxplot(data=data, x=data["Income"], fill=True, orient="V")

plt.subplot(3,2, (3,4))
sns.countplot(data=data, x = data["Age"])
plt.xlabel("Age of Customers")
plt.ylabel("Count")

plt.subplot(3,2, 5)
sns.lineplot(data=data, x=data["Age"] ,y = data["Education"], estimator='mean', legend='auto')
plt.xlabel("Age of Customers")
plt.ylabel("Education Qualification")


plt.subplot(3,2, 6)
sns.scatterplot(data=data, y=data["Miles"], x=data["Age"], hue=data["MaritalStatus"])
plt.xlabel("Age of Customers")
plt.ylabel("Miles")



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










```
data["Miles"].describe()
```

	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

dtype: float64

```
data["Education"].describe()
```

	Education
count	180.000000
mean	15.572222
std	1.617055
min	12.000000
25%	14.000000
50%	16.000000
75%	16.000000
max	21.000000

dtype: float64

```
data["Usage"].describe()
```

	Usage
count	180.000000
mean	3.455556
std	1.084797
min	2.000000
25%	3.000000
50%	3.000000
75%	4.000000
max	7.000000

dtype: float64

```
data["Fitness"].describe()
```

	Fitness
count	180.000000
mean	3.311111
std	0.958869
min	1.000000
25%	3.000000
50%	3.000000
75%	4.000000
max	5.000000

dtype: float64

```
data["Income"].describe()
```

	Income
count	180.000000
mean	53719.577778
std	16506.684226
min	29562.000000
25%	44058.750000
50%	50596.500000
75%	58668.000000
max	104581.000000

dtype: float64

Non-Graphical Analysis: Value counts and unique attributes

```
data["Product"].value_counts()
```

Product	count
KP281	80
KP481	60
KP781	40

dtype: int64

```
data["Gender"].value_counts()
labels = data["Gender"].value_counts().index
values = data["Gender"].value_counts().values
```

```
data["Gender"].value_counts(normalize=True)
```

Gender	proportion
Male	0.577778
Female	0.422222

dtype: float64

```
data["MaritalStatus"].value_counts()
```

MaritalStatus	count
Partnered	107
Single	73

dtype: int64

```
plt.figure(figsize=(12,8))
plt.subplot(2, 2, 1)
plt.pie(values, labels=labels,  autopct="%1.1f%%", startangle=90)
plt.title("Number of Males and Females")


plt.subplot(2,2,2)
sns.countplot(data=data, x=data["Product"], width=0.5)
plt.xlabel("Products")
plt.title("Total number of Products")

plt.subplot(2, 2, 3)
sns.countplot(data=data, x=data["Product"], hue="Gender")

# plt.subplot(2, 2, 4)
# sns.lineplot(data=data, y=data["Product"], x=data["Gender"])



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