Name: Raghu
Business case: Aerofit_Descriptive Statistics & Probability
Date: 12-06-2023

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.

1. Perform descriptive analytics **to create a customer profile** for each AeroFit treadmill product by developing appropriate tables and charts.
2. 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.

Name: Raghu
Business case: Aerofit
Date: 12-06-2023

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

```
[2] Aerofit_data = pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749")
```

```
[3] Aerofit_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

```
# Shape of data frame
Aerofit_data.shape
```

```
(180, 9)
```

```
# Structure & Characterstics of the data set
Aerofit_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
```

```
# Checking count of null values in each column
Aerofit_data.isna().sum()
```

```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

```
# Describing Numerical data

Aerofit_data.describe()
```

	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

Name: Raghu
Business case: Aerofit_Descriptive Statistics & Probability
Date: 12-06-2023

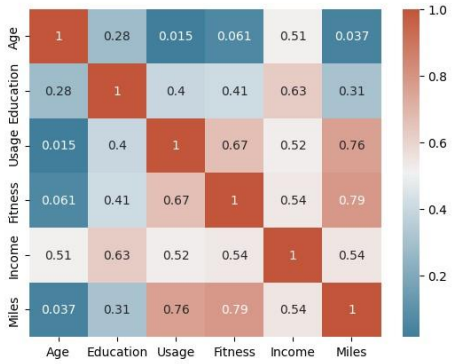
```
# Count of unique products

Aerofit_data['Product'].nunique()
```

```
3
```

```
# Correlation among different factors
corr = Aerofit_data.corr()
cmap = sns.diverging_palette(230, 20, as_cmap=True)
sns.heatmap(corr, annot=True, cmap=cmap)
```

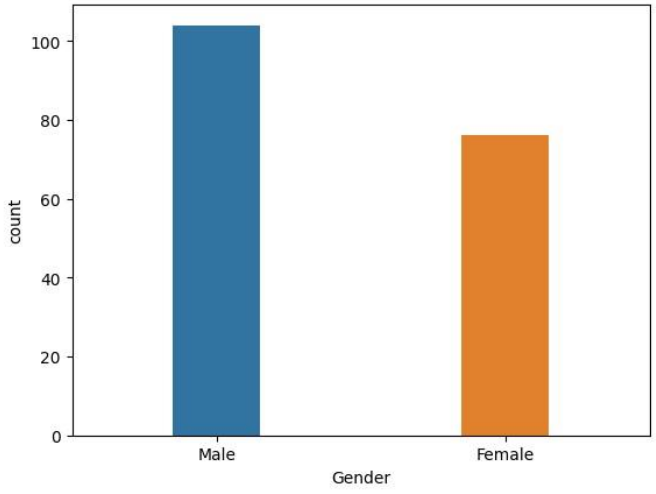
```
<ipython-input-68-1719fdd18f6e>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid
  corr = Aerofit_data.corr()
<Axes: >
```



Fitness, Miles & Usage have high correlation with one another and education & income are also correlated.

```
# Count of each gender purchasing product
sns.countplot(Aerofit_data, x = 'Gender', width = 0.3)
```

```
<Axes: xlabel='Gender', ylabel='count'>
```

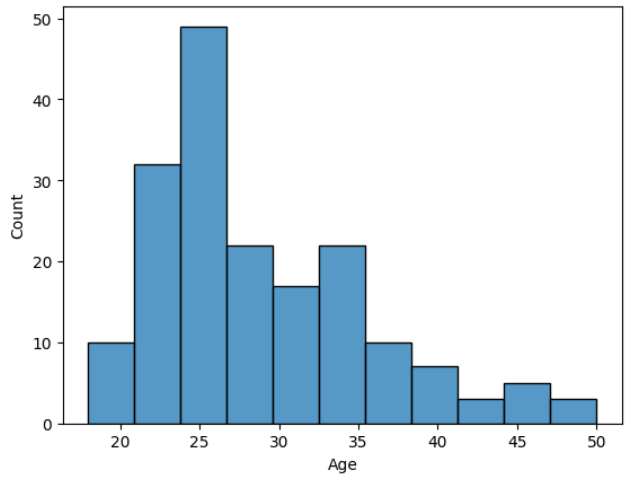


**From the above chart, out of 180, there are 100 males & 80 females purchased the product.Compared to Female, male have slightly higher tendency of purchasing trademill products **

Name: Raghu
Business case: Aerofit
Date: 12-06-2023

```
[10] sns.histplot(Aerofit_data, x = 'Age')
```

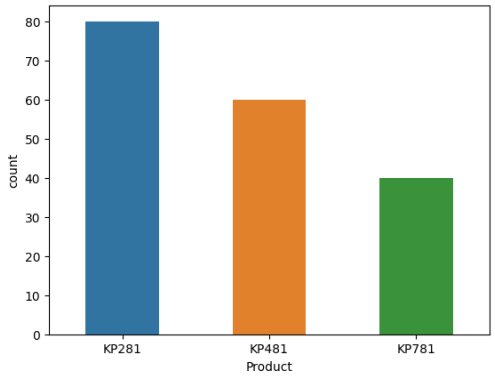
```
<Axes: xlabel='Age', ylabel='Count'>
```



*People aged between 22 and 27 years old are the age group most likely to purchase trademills *

```
# Count of customers for each product
sns.countplot(Aerofit_data, x = 'Product', width = 0.5)
```

```
<Axes: xlabel='Product', ylabel='count'>
```

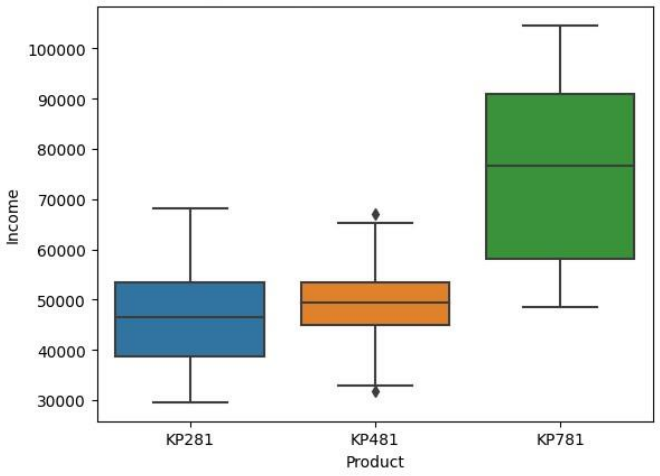


Number of people want to buy KP281 is greater than that of KP481 & KP781. It is reccomended to always have product KP281 product in stock

Name: Raghu
Business case: Aerofit_Descriptive Statistics & Probability
Date: 12-06-2023

```
sns.boxplot(Aerofit_data, x = "Product", y = "Income")
```

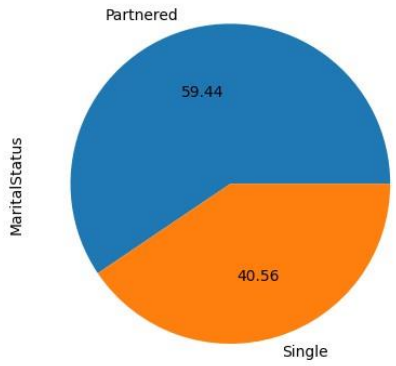
```
<Axes: xlabel='Product', ylabel='Income'>
```



*Since cost of KP781 is higher, the average income of people most likely to purchase KP781 is much higher than that of people most likely to purchase other two products.From the above, it is concluded that customers having income around 80000$ or more most likely to purchase KP781 *

```
Aerofit_data["MaritalStatus"].value_counts().plot(kind = "pie", autopct = "%.2f")
```

```
<Axes: ylabel='MaritalStatus'>
```



Name: Raghu
Business case: Aerofit
Date: 12-06-2023

```
# Constructinh two way contingency tables
Prod_Gender_cont = pd.DataFrame(Aerofit_data.groupby("Product")['Gender'].value_counts())
Prod_Gender_cont
```

		Gender
Product	Gender	
KP281	Female	40
	Male	40
KP481	Male	31
	Female	29
KP781	Male	33
	Female	7

```
Prod_Gender_cont.columns = ['Count']
Prod_Gender_cont.columns.name = None
Prod_Gender_cont.reset_index(inplace = True)
Prod_Gender_cont
```

	Product	Gender	Count
0	KP281	Female	40
1	KP281	Male	40
2	KP481	Male	31
3	KP481	Female	29
4	KP781	Male	33
5	KP781	Female	7

```
[46] # two-way Product - Count of each gender contingency tables
Prod_Gender_cont_table = pd.pivot(Prod_Gender_cont, index = ["Product"], columns = "Gender", values = "Count" )
Prod_Gender_cont_table.reset_index(inplace= True)
Prod_Gender_cont_table
```

Gender	Product	Female	Male
0	KP281	40	40
1	KP481	29	31
2	KP781	7	33

```
[48] # Probability of each gender purchasing each product
probOfGender_per_each_product_table = Prod_Gender_cont_table.copy()
probOfGender_per_each_product_table['Prob_Of_Female'] = (Prod_Gender_cont_table["Female"]/(Prod_Gender_cont_table["Female"]+Prod_Gender_cont_table["Male"])).round(2)
probOfGender_per_each_product_table['Prob_Of_Male'] = (Prod_Gender_cont_table["Male"]/(Prod_Gender_cont_table["Female"]+Prod_Gender_cont_table["Male"])).round(2)
probOfGender_per_each_product_table
```

Gender	Product	Female	Male	Prob_Of_Female	Prob_Of_Male
0	KP281	40	40	0.50	0.50
1	KP481	29	31	0.48	0.52
2	KP781	7	33	0.18	0.82

From the above, probability of Male and probability of Female purchasing product KP281/KP481 are almost same. But probability of Male purchasing KP781(0.82) is much higher than that of female(0.18) which means 82 % of the poeople purcasing KP781 are male

Name: Raghu
Business case: Aerofit
Date: 12-06-2023

```
[83] # Probability of fitness rating > =3 for each product

     Product_fitness_rating = pd.DataFrame(Aerofit_data[Aerofit_data["Fitness"]>=3]["Product"].value_counts())
     Product_fitness_rating.columns.name = None
     Product_fitness_rating.reset_index(inplace = True)
     Product_fitness_rating.columns = ["Product","countOfcustomers_fitness_level>=3"]
     Product_fitness_rating
```

	Product	countOfcustomers_fitness_level>=3
0	KP281	65
1	KP481	47
2	KP781	40

```
[84] Product_count = pd.DataFrame(Aerofit_data['Product'].value_counts())
     Product_count.columns.name = None
     Product_count.reset_index(inplace = True)
     Product_count.columns = ["Product", "Total_customers"]
     Product_count
```

	Product	Total_customers
0	KP281	80
1	KP481	60
2	KP781	40

```
[ ] Product_fitness_rating = pd.merge(Product_fitness_rating)
```

```
Product_fitness_rating_prob_tab = pd.merge(Product_count,Product_fitness_rating, on = "Product" )
Product_fitness_rating_prob_tab["Prob(Fitnessrating>=3)"] = (Product_fitness_rating_prob_tab["countOfcustomers_fitness_level>=3"]/Product_fitness_rating_prob_tab["Total_customers"]).round(2)
Product_fitness_rating_prob_tab
```

	Product	Total_customers	countOfcustomers_fitness_level>=3	Prob(Fitnessrating>=3)
0	KP281	80	65	0.81
1	KP481	60	47	0.78
2	KP781	40	40	1.00

```
# Customer profile for each product

Aerofit_data.groupby("Product")[["Age", "Education","Income"]].agg({'Age':['min','max','mean'],'Education':['min','max','mean'],'Income':['min','max','mean']}).reset_index()
```

	Product	Age			Education			Income		
		min	max	mean	min	max	mean	min	max	mean
0	KP281	18	50	28.55	12	18	15.037500	29562	68220	46418.025
1	KP481	19	48	28.90	12	18	15.116667	31836	67083	48973.650
2	KP781	22	48	29.10	14	21	17.325000	48556	104581	75441.575

From the above table, Age & education do not have any impact on purchasing different products, but income has. People having higher income have tendency to buy KP781

```
# Probability of fitness rating >= 4 for each product

Product_fitness_rating = pd.DataFrame(Aerofit_data[Aerofit_data["Fitness"]>=4]["Product"].value_counts())
Product_fitness_rating.columns.name = None
Product_fitness_rating.reset_index(inplace = True)
Product_fitness_rating.columns = ["Product","countOfcustomers_fitness_level>=4"]
Product_fitness_rating
```

	Product	countOfcustomers_fitness_level>=4
0	KP781	36
1	KP281	11
2	KP481	8

```
[97] Product_count = pd.DataFrame(Aerofit_data['Product'].value_counts())
     Product_count.columns.name = None
     Product_count.reset_index(inplace = True)
     Product_count.columns = ["Product", "Total_customers"]
     Product_count
```

	Product	Total_customers
0	KP281	80
1	KP481	60
2	KP781	40

```
Product_fitness_rating_prob_tab = pd.merge(Product_count,Product_fitness_rating, on = "Product" )
Product_fitness_rating_prob_tab["Prob(Fitness_rating>=4)"] = (Product_fitness_rating_prob_tab["countOfcustomers_fitness_level>=4"]/Product_fitness_rating_prob_tab["Total_customers"]).round(2)
Product_fitness_rating_prob_tab
```

	Product	Total_customers	countOfcustomers_fitness_level>=4	Prob(Fitness_rating>=4)
0	KP281	80	11	0.14
1	KP481	60	8	0.13
2	KP781	40	36	0.90

*From the above table, prob of fitness_rating >=4 for KP781 is 90% but for other two products, it is less than 15 %. This may be due to design of treadmill to optimize performance and maintenance for customers while providing a comfortable and effective workout for customers exercisers.But at the same time, fitness rating also have correlation with factors "usage" & "miles". *