Aerofit - Business Case

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.

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

from scipy.stats import norm
from scipy.special import comb
from statsmodels.distributions.empirical_distribution import ECDF
from scipy.stats import norm
from scipy.stats import poisson
```

```
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749"
```

```
--2023-04-10 05:53:50--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 108.157.172.173, 108.157.172.10, 108.157.172.183, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|108.157.172.173|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7279 (7.1K) [text/plain]
Saving to: ‘aerofit_treadmill.csv?1639992749’

aerofit_treadmill.c 100%[===================>]   7.11K  --.-KB/s    in 0s

2023-04-10 05:53:50 (632 MB/s) - ‘aerofit_treadmill.csv?1639992749’ saved [7279/7279]
```

```
df = pd.read_csv("aerofit_treadmill.csv?1639992749")
df
```

	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
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

180 rows × 9 columns

```
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
```

```
df.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

```
df.describe(include="object")
```

	Product	Gender	MaritalStatus
count	180	180	180
unique	3	2	2
top	KP281	Male	Partnered
freq	80	104	107

Observations:

There are no missing values in the data.

There are 3 unique products in the dataset. KP281 is the most frequent product.

Out of 180 customers, 107 is for partnered and the rest is for singles.

Minimum & Maximum age of the person is 18 & 50, mean is 28.79 and 75% of persons have age less than or equal to 33.

Most of the people are having 16 years of education

Out of 180 Customers, 104's gender is Male and rest are the female.

Standard deviation for Income & Miles is very high.

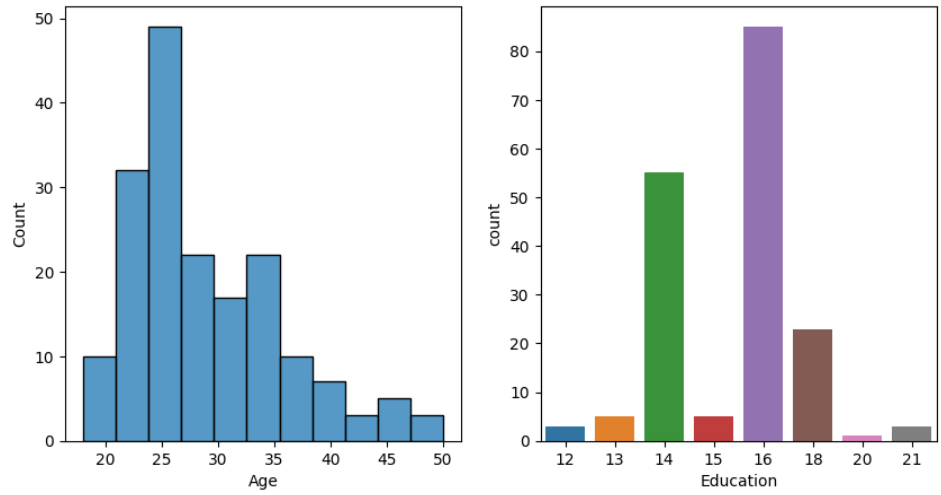
```
df["Product"].unique()
```

```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

Univariant Analysis

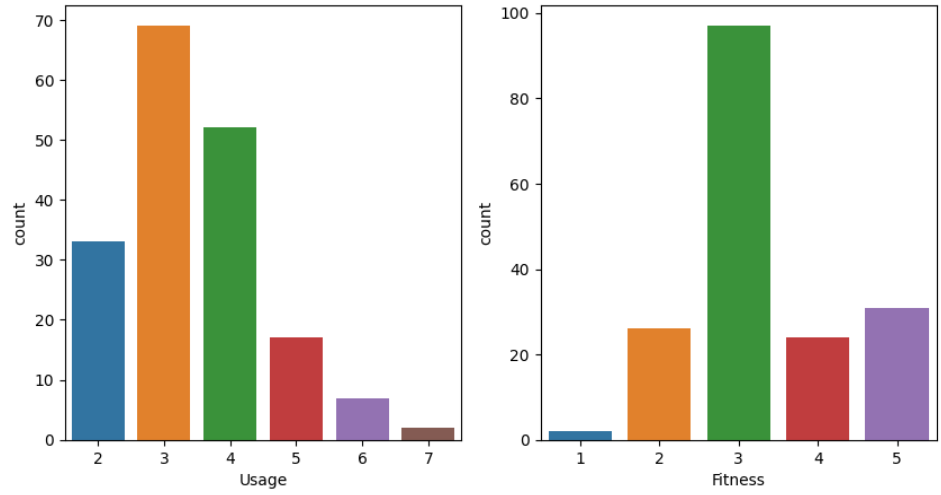
```
fig,ax = plt.subplots(1,2,figsize=(10,5))

sns.histplot(data = df , x= "Age", ax = ax[0])
sns.countplot(data = df , x= "Education", ax = ax[1])
```

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



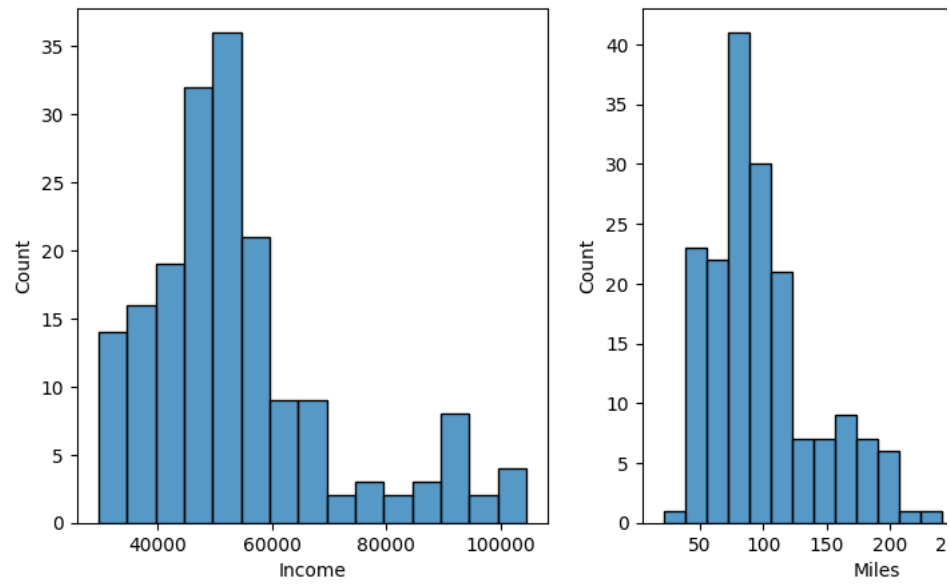

```
fig,ax = plt.subplots(1,2,figsize=(10,5))
sns.countplot(data = df , x= "Usage", ax = ax[0])
sns.countplot(data = df , x= "Fitness", ax = ax[1])
```

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



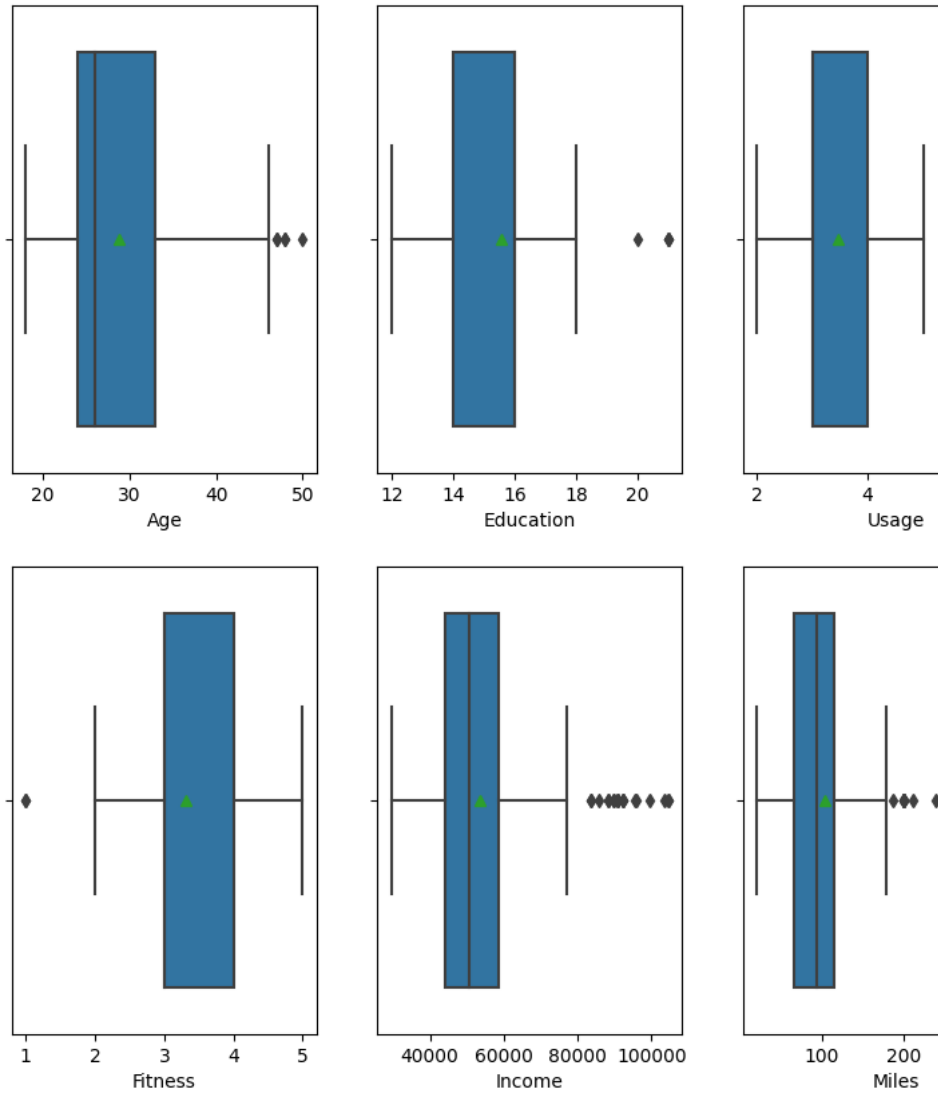

```
fig,ax = plt.subplots(1,2,figsize=(10,5))
sns.histplot(data = df , x= "Income", ax = ax[0])
sns.histplot(data = df , x= "Miles", ax = ax[1])
```

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



```
fig,ax = plt.subplots(2,3,figsize=(10,10))

sns.boxplot(data = df , x= "Age", ax = ax[0,0] , showmeans = True)
sns.boxplot(data = df , x= "Education", ax = ax[0,1] , showmeans = True)
sns.boxplot(data = df , x= "Usage", ax = ax[0,2] , showmeans = True)
sns.boxplot(data = df , x= "Fitness", ax = ax[1,0] , showmeans = True)
sns.boxplot(data = df , x= "Income", ax = ax[1,1] , showmeans = True)
sns.boxplot(data = df , x= "Miles", ax = ax[1,2] , showmeans = True)
```

```
<Axes: xlabel='Miles'>
```

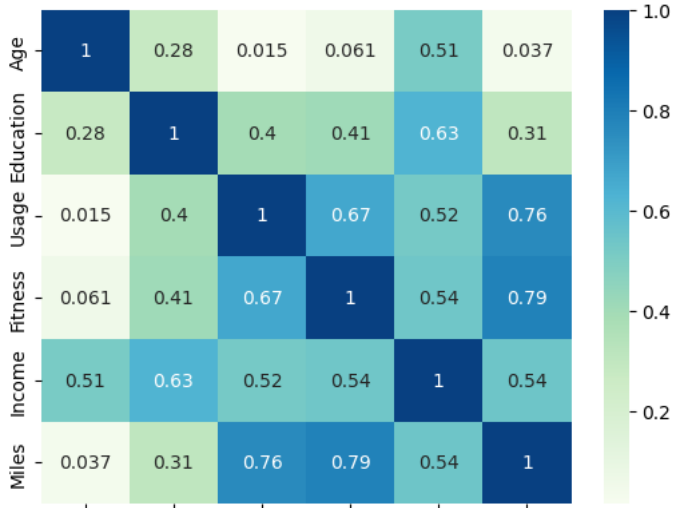


Observation:

Income and Miles has many outliers compared to others

```
sns.heatmap(df.corr(), cmap="GnBu", annot=True)
#df.corr()
```

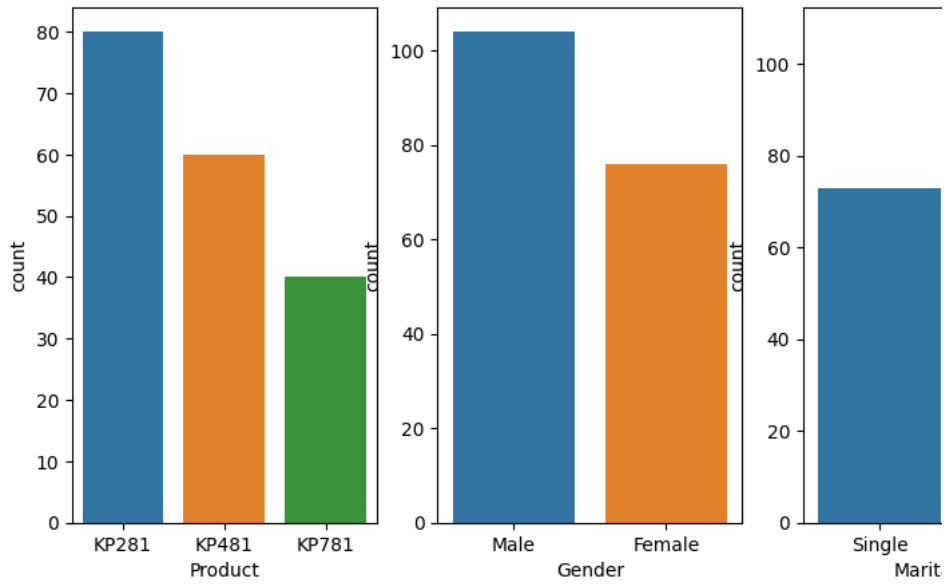
```
<Axes: >
```



Understanding of Data in Categorical type:

```
fig,ax = plt.subplots(1,3,figsize=(10,5))

sns.countplot(data = df , x= "Product", ax = ax[0])
sns.countplot(data = df , x= "Gender", ax = ax[1])
sns.countplot(data = df , x= "MaritalStatus", ax = ax[2])
```

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



Observations:

1) Among products the most frequent product is **KP281**

2) Most of them are males compared to females

3) Partnered persons are more frequent to buy products.

```
df1 = df[['Product', 'Gender', 'MaritalStatus']].melt()
```

```
df1.groupby(['variable', 'value'])[['value']].count()
```

variable	value	value
Gender	Female	76
	Male	104
MaritalStatus	Partnered	107
	Single	73
Product	KP281	80
	KP481	60
	KP781	40

```
df1.groupby(['variable', 'value'])[['value']].count() / len(df)
```

variable	value	value
Gender	Female	0.422222
	Male	0.577778
MaritalStatus	Partnered	0.594444
	Single	0.405556
Product	KP281	0.444444
	KP481	0.333333
	KP781	0.222222

Observations:

Gender:

1) 42.2 % of females have purchesed products from whole data

2) 57.8 % of males have purchased products from data.

Product

1) 44.4% of KP281 product has been sold to customers

2) 33.3% of KP281 product has been sold to customers

3) 22.2% of KP281 product has been sold to customers

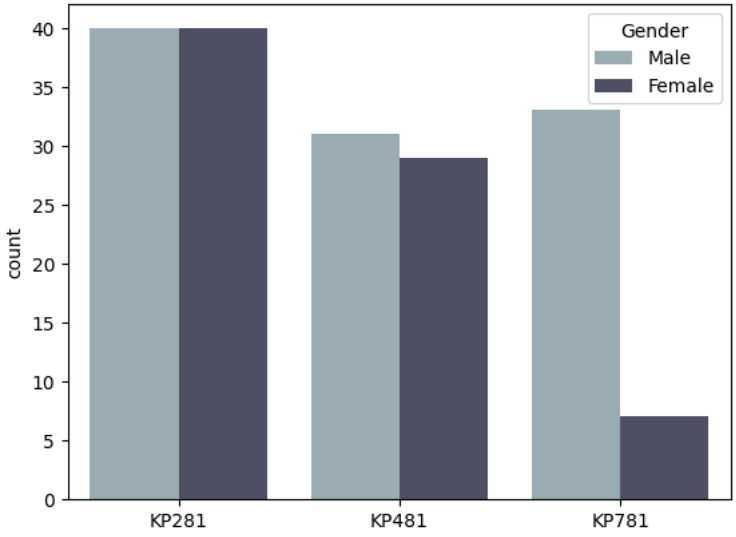
Marital Status

1) 59.9% of customers are Partnered Persons

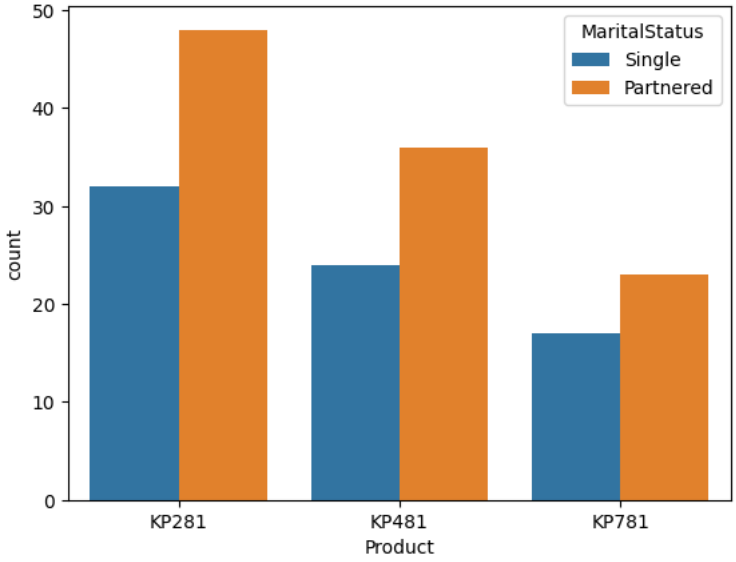
2) 40.6% of customers are singles

Bivariant Analysis

```
sns.countplot(data = df , x = "Product", hue="Gender", palette = "bone_r")
plt.show()
```




```
sns.countplot(data = df , x = "Product", hue="MaritalStatus")
plt.show()
```



Obervations:

Gender - Product

1) KP281 is the most purchased product among male and females

2) And of KP781 - Males are more tend to buy this then females

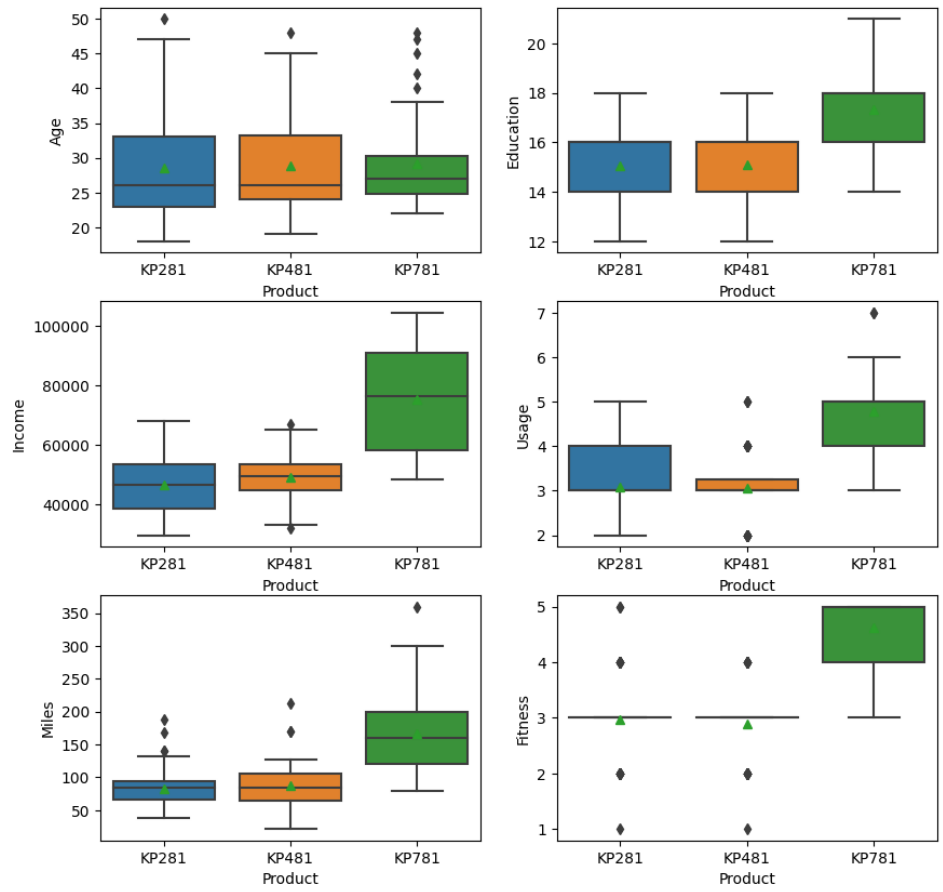
MaritalStatus - Product

1) KP281 is the most purchased product among singles and partnered

2) Customer who is Partnered, is more likely to purchase the product.

```
fig,ax = plt.subplots(3,2,figsize=(10,10))

sns.boxplot(data = df , x = "Product" , y = "Age", ax = ax[0,0],showmeans=True)
sns.boxplot(data = df , x = "Product" , y = "Education", ax = ax[0,1],showmeans=True)
sns.boxplot(data = df , x = "Product" , y = "Income", ax = ax[1,0],showmeans=True)
sns.boxplot(data = df , x = "Product" , y = "Usage", ax = ax[1,1],showmeans=True)
sns.boxplot(data = df , x = "Product" , y = "Miles", ax = ax[2,0],showmeans=True)
sns.boxplot(data = df , x = "Product" , y = "Fitness", ax = ax[2,1],showmeans=True)
```

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



Observations:

Product - Age:

1) More people are tend to purchase products KP281 & KP481 and has higher median values as well.

2) Customers whose age lies between 25-30, are more likely to buy KP781 product

Product - Education

1) Customers whose Education is higher than 16, have more chances to purchase the KP781 product.

2) While other customers less than 16 are more willing to buy KP281 or KP481.

Product - Usage

1) People working out for more then 4 times a week, are more likely to purchase the KP781 product.

While the other people are likely to purchasing KP281 or KP481.

Product - Fitness

1) The more the customer is fit (fitness >= 3), higher the chances of the customer to purchase the KP781 product.

Product - Income

1) Higher the Income of the customer (Income >= 60000), higher the chances of the customer to purchase the KP781 product.

Product - Miles

1) Customer expecting to walk more than 120 Miles per week, is more likely to buy KP781 product.

Computing Marginal & Conditional Probabilities

Marginal Probability

```
pd.crosstab(index = df["Product"], columns = df["Gender"], margins = True)
```

Gender	Female	Male	All
Product			
KP281	40	40	80
KP481	29	31	60
KP781	7	33	40

What is the probability of a male customer buying a KP781 treadmill?

In over all date probability of male buying KP781 is **0.18333**

```
male_kp281 = 40/80
male_kp281    ## Given KP281 probability of male buying kp281
```

```
0.5
```

```
df['Product'].value_counts(normalize=True)
```

```
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: Product, dtype: float64
```

Conditional Probability

Probability of each product given gender

```
## Using cross tab

female_kp781 = 7/76
female_kp781    ## Given female what is probability of kp281 buying by female
```

```
0.09210526315789473
```

```
## Given Male probability of each product

#df(KP781|male) = df(kp781 intersection male) / df(male)

df_781_and_male = df[(df["Product"] == "KP781") & (df["Gender"] == "Male")]
df_male = df[(df["Gender"] == "Male")]
df_481_and_male = df[(df["Product"] == "KP481") & (df["Gender"] == "Male")]
df_281_and_male = df[(df["Product"] == "KP281") & (df["Gender"] == "Male")]
print("Probility of KP781 given male = %.2f" %(len(df_781_and_male)/len(df_male)))
print("Probility of KP481 given male = %.2f" %(len(df_481_and_male)/len(df_male)))
print("Probility of KP281 given male = %.2f" %(len(df_281_and_male)/len(df_male)))
```

```
Probility of KP781 given male = 0.32
Probility of KP481 given male = 0.30
Probility of KP281 given male = 0.38
```

```
## Given Female probability of each product

#df(KP781|female) = df(kp781 intersection female) / df(female)

df_781_and_female = df[(df["Product"] == "KP781") & (df["Gender"] == "Female")]
df_female = df[(df["Gender"] == "Female")]
df_481_and_female = df[(df["Product"] == "KP481") & (df["Gender"] == "Female")]
df_281_and_female = df[(df["Product"] == "KP281") & (df["Gender"] == "Female")]
print("Probility of KP781 given female = %.2f" %(len(df_781_and_female)/len(df_female)))
print("Probility of KP481 given female = %.2f" %(len(df_481_and_female)/len(df_female)))
print("Probility of KP281 given female = %.2f" %(len(df_281_and_female)/len(df_female)))
```

```
Probility of KP781 given female = 0.09
Probility of KP481 given female = 0.38
Probility of KP281 given female = 0.53
```

Probability of each product given Marital Status

```
## Given Partnered probability of each product

#df(KP781|Partnered) = df(kp781 intersection Partnered) / df(Partnered)
```

```
df_781_and_Partnered = df[(df["Product"] == "KP781") & (df["MaritalStatus"] == "Partnered")]
df_Partnered = df[(df["MaritalStatus"] == "Partnered")]
df_481_and_Partnered = df[(df["Product"] == "KP481") & (df["MaritalStatus"] == "Partnered")]
df_281_and_Partnered = df[(df["Product"] == "KP281") & (df["MaritalStatus"] == "Partnered")]
print("Probility of KP781 given Partnered(P(KP781/Partnered)) = %.2f" % (len(df_781_and_Partnered)/len(df_Partnered)))
print("Probility of KP481 given Partnered(P(KP481/Partnered)) = %.2f" % (len(df_481_and_Partnered)/len(df_Partnered)))
print("Probility of KP281 given Partnered(P(KP281/Partnered)) = %.2f" % (len(df_281_and_Partnered)/len(df_Partnered)))
```

```
Probility of KP781 given Partnered(P(KP781/Partnered)) = 0.21
Probility of KP481 given Partnered(P(KP481/Partnered)) = 0.34
Probility of KP281 given Partnered(P(KP281/Partnered)) = 0.45
```

```
## Given Single probability of each product

#df(KP781|Single) = df(kp781 intersection Single) / df(Single)

df_781_and_Single = df[(df["Product"] == "KP781") & (df["MaritalStatus"] == "Single")]
df_Single = df[(df["MaritalStatus"] == "Single")]
df_481_and_Single = df[(df["Product"] == "KP481") & (df["MaritalStatus"] == "Single")]
df_281_and_Single = df[(df["Product"] == "KP281") & (df["MaritalStatus"] == "Single")]
print("Probility of KP781 given Single(P(KP781/Single)) = %.2f" %(len(df_781_and_Single)/len(df_Single)))
print("Probility of KP481 given Single(P(KP481/Single)) = %.2f" %(len(df_481_and_Single)/len(df_Single)))
print("Probility of KP281 given Single(P(KP281/Single)) = %.2f" %(len(df_281_and_Single)/len(df_Single)))
```

```
Probility of KP781 given Single(P(KP781/Single)) = 0.23
Probility of KP481 given Single(P(KP481/Single)) = 0.33
Probility of KP281 given Single(P(KP281/Single)) = 0.44
```

Recommendations:

1) Out of 180 Customers, 104's gender is Male and rest few members the female and too low this can should be worked upon.

2) Out of 180 customers, 107 is for partnered and the rest is for singles and singles buying products is too low and should be given coupons and tokens to attract them.

3) People working out for more then 4 times a week, are more likely to purchase the KP781 product.

While the other people are likely to purchasing KP281 or KP481, so working out for 3 times a week can suggest them to buy KP781 product inorder to higher up the sales.

4) Amongst 3 different products KP781 is the least bought and in which most of these are bought by males only so this needs to be focused on.

5) Among 3 products singles are least to buy these products and can make them buy by giving coupons and offers to attract them buy these products.

6) Probility of KP781 buying in female is very least which needs to be worked upon.