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

```
! gdown '1Z2ON0JqcQw25HwmVhq8clpHMnVfM654V'
```

```
Downloading...
From: https://drive.google.com/uc?id=1Z2ON0JqcQw25HwmVhq8clpHMnVfM654V
To: /content/aerofit_treadmill.txt
100% 7.28k/7.28k [00:00<00:00, 20.4MB/s]
```

```
data = pd.read_csv('/content/aerofit_treadmill.txt')
```

```
data
```

	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

Next steps: Generate code with data View recommended plots

Insights :- There are total 180 rows in the data with 9 columns.

```
data.shape
```

```
(180, 9)
```

```
data.dtypes
```

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

Insights :- The mean and median for all the columns in the dataset is slightly different from each other, so there are some outliers in the data.

```
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

Insights :- There are null values in the data, the data is already sanitized.

```
data.isnull().any()
```

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

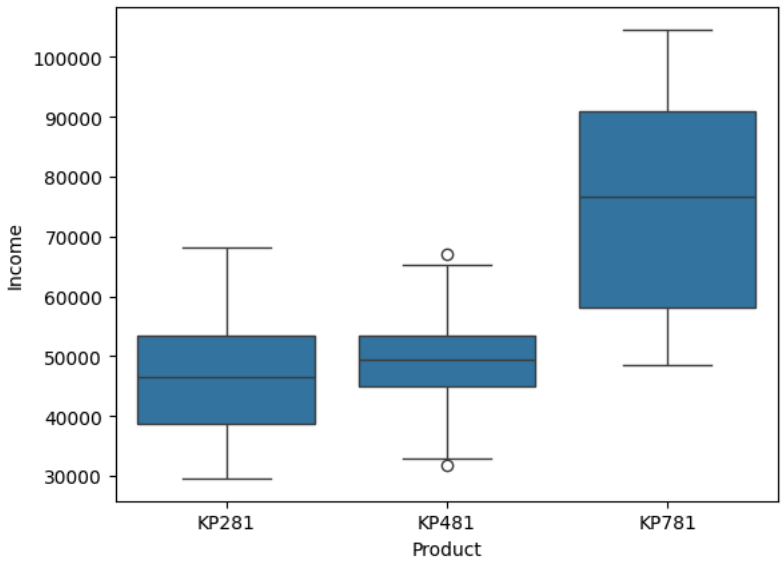
Insights :-

1. The median of the Income for the people buying the KP781 product is around 80K.
2. The median of the Income of people buying KP281 and KP481 are nearly similar i.e. around 50K.
3. We also see there are outliers for the product KP481 on either side of the wick.

Recommendations :- We need to advertise the product in such a way that the people who are buying the product KP481, should feel the tech of the product KP781 relevent to them.

```
sns.boxplot(x='Product',y ='Income',data = data)
```

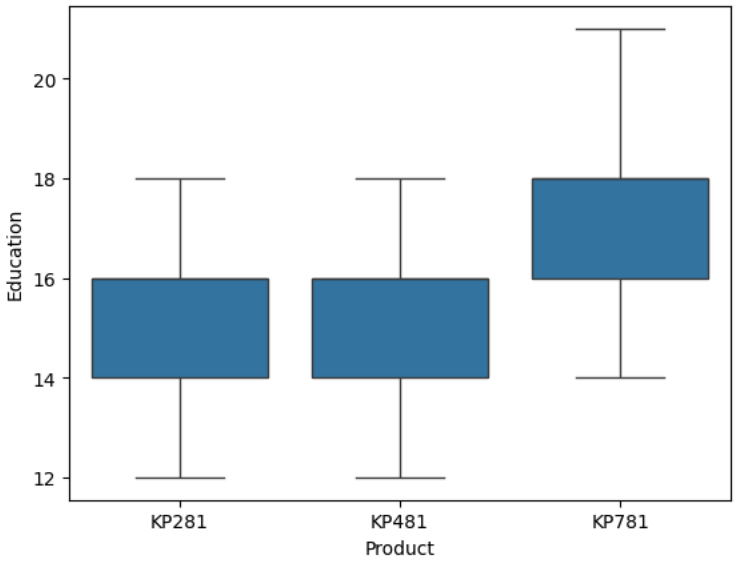
```
<Axes: xlabel='Product', ylabel='Income'>
```



Insights :- The people with higher education i.e. 18 tends to buy more of KP781 product, whereas people with low education i.e. 14 or 16 tends to buy KP281 and KP481

```
sns.boxplot(x='Product',y ='Education',data = data)
```

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




```
data.groupby('Product')['Education'].describe()
```

Product	count	mean	std	min	25%	50%	75%	max
KP281	80.0	15.037500	1.216383	12.0	14.0	16.0	16.0	18.0
KP481	60.0	15.116667	1.222552	12.0	14.0	16.0	16.0	18.0
KP781	40.0	17.325000	1.639066	14.0	16.0	18.0	18.0	21.0

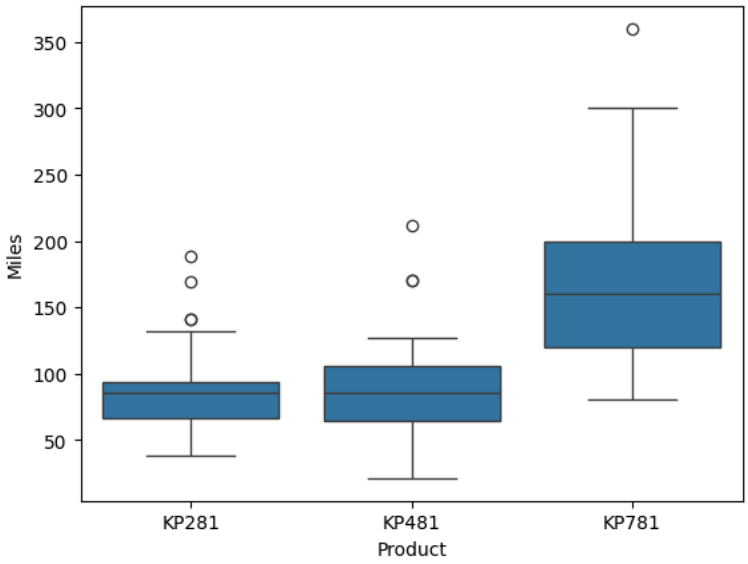
Insights :-

1. The people who have bought KP781 tends to use the product more.
2. The people who have bought KP281,KP481 tends to use the product less.
3. There are outliers in this data.

Recommendation :- This insight implies that the people who are more serious regarding the health uses the product KP781. We should use this fact to woo the customers.

```
sns.boxplot(x='Product',y ='Miles',data = data)
```

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




```
data.groupby('Product')['Miles'].describe()
```

Product	count	mean	std	min	25%	50%	75%	max
KP281	80.0	82.787500	28.874102	38.0	66.0	85.0	94.0	188.0
KP481	60.0	87.933333	33.263135	21.0	64.0	85.0	106.0	212.0
KP781	40.0	166.900000	60.066544	80.0	120.0	160.0	200.0	360.0

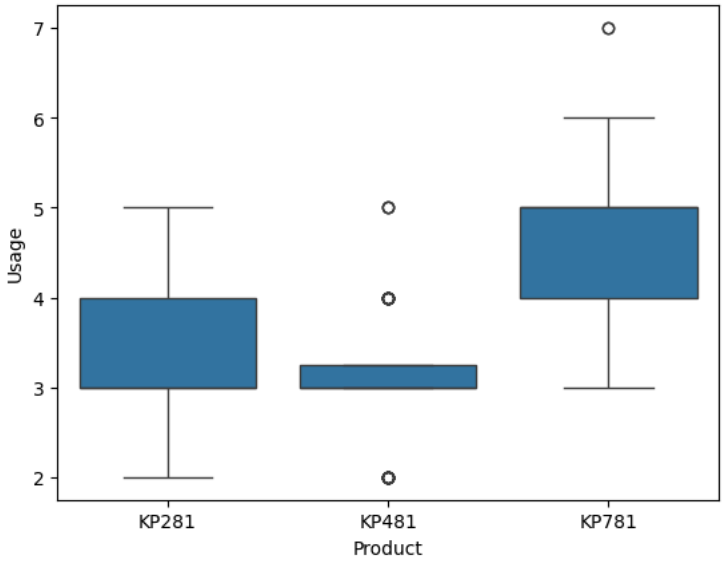
Insights :-

1. The people who have bought KP781 tends to use the product more in a week.
2. The people who have bought KP281,KP481 tends to use the product less in a week.
3. There are outliers in this data.

Recommendation :- This insight implies that the people who are more serious regarding the health uses the product KP781. We should use this fact to woo the customers, they can buy this upper end model of the product.

```
sns.boxplot(x='Product',y ='Usage',data = data)
```

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




```
data.groupby('Product')['Usage'].describe()
```

Product	count	mean	std	min	25%	50%	75%	max
KP281	80.0	3.087500	0.782624	2.0	3.0	3.0	4.00	5.0
KP481	60.0	3.066667	0.799717	2.0	3.0	3.0	3.25	5.0
KP781	40.0	4.775000	0.946993	3.0	4.0	5.0	5.00	7.0

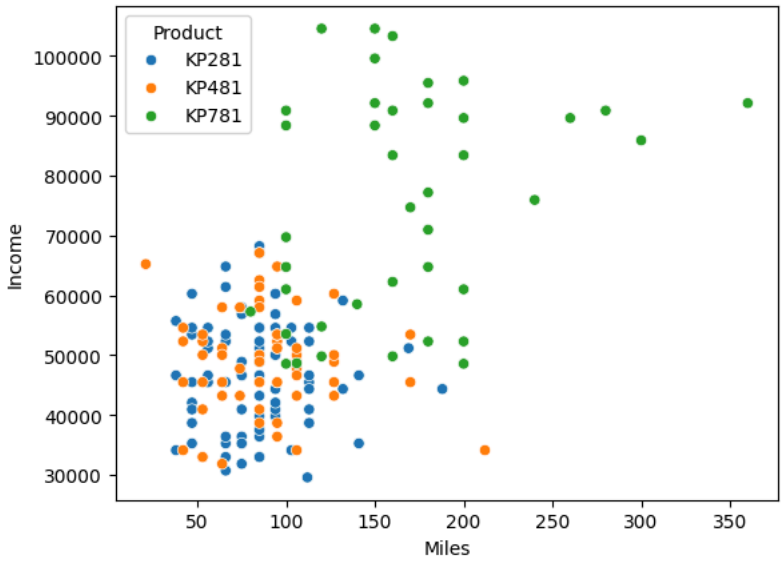
Insight :- After checking for product across Eduction, Miles, Usage, we found that the people who have more education, more miles, more usage per week tends to buy KP781.

Insight :-

1. Here we can observe that the people with higher income and higher miles are interested in KP781 product.
2. People with lower Income and lower miles are interested in KP281 and KP481 products.

```
sns.scatterplot(x = 'Miles',y='Income',data = data,hue='Product')
```

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

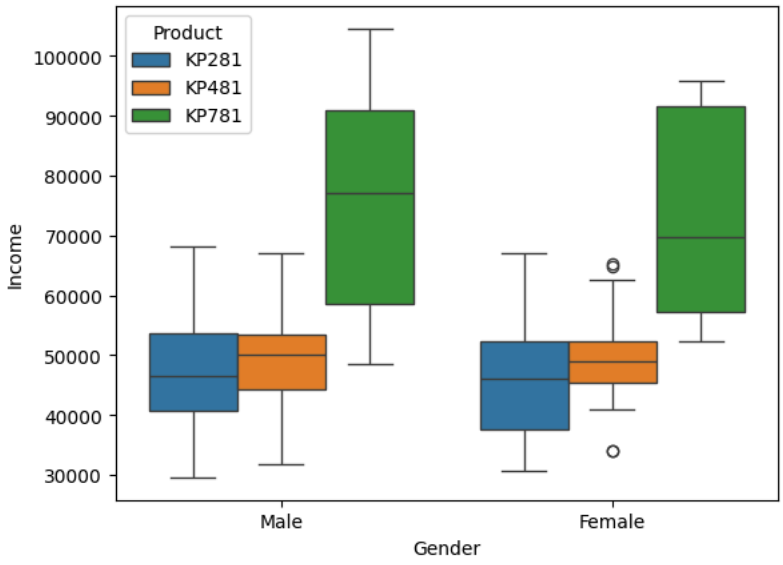


Insight :-

1. We can see that irrespective of the genders, the people with higher income tends to buy higher variant of the product.
2. We see some outliers for females for the mid variant KP481.
3. The IQR of the higher variant KP781 in female is litte bigger than that of male.

```
sns.boxplot(x='Gender',y ='Income',data = data,hue ='Product')
```

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



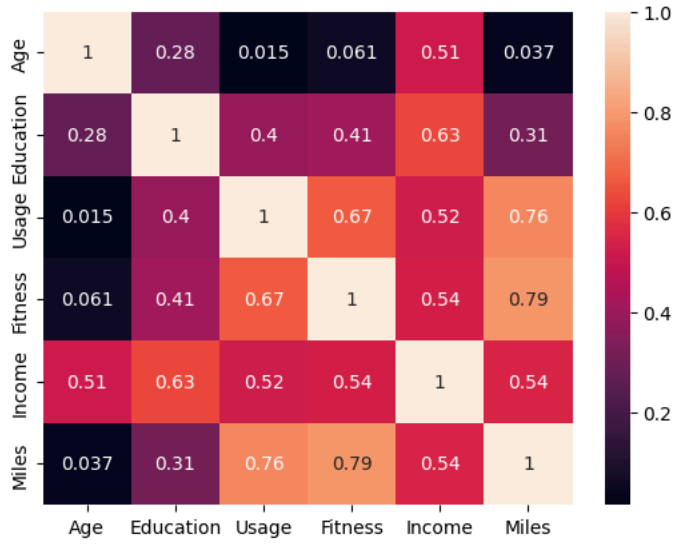
Insights :-

Here we can see many correlations :

1. People with more age tends to walk/run less average number of miles.
2. People with more usage and fitness tends to walk/run more average number of miles.
3. People who have more education, they have higher income.
4. There is direct relation of usage with fitness, people who tends to use the product more is more fit.

```
sns.heatmap(data.drop(['Product','MaritalStatus','Gender'],axis = 1).corr(), annot=True)
```

```
<Axes: >
```



Probabilty

```
# CATEGORICAL SERIES
pd.crosstab(index=data['Gender'], columns=data['Product'],margins=True)
```

Product	KP281	KP481	KP781	All
Gender				
Female	40	29	7	76
Male	40	31	33	104
All	80	60	40	180

Q. #If 500 males walk into my store, what should be my min inventory for KP481

```
#Ans :- From above data there are total of 104 males and 31 have bought KP481
ans = 31/104 * 500
print(f'We will need total of {ans} number of KP481.')
```

```
We will need total of 149.03846153846155 number of KP481.
```

Q. There are 5000 pieces of KP781, How many of them will be bought by females?

```
# Ans :- From above data there are total of 76 females and 7 have bought KP781
ans = 7/40 * 5000
print(f'We will need total of {ans} number of KP781.')
```

```
We will need total of 875.0 number of KP781.
```

This is percent wise contirbution of each Product over both the genders.

```
pd.crosstab(index=data['Gender'], columns=data['Product'],margins=True,normalize = True) * 100
```

Product	KP281	KP481	KP781	All
Gender				
Female	22.222222	16.111111	3.888889	42.222222
Male	22.222222	17.222222	18.333333	57.777778
All	44.444444	33.333333	22.222222	100.000000

Q. How much should be the inventory of KP481 if :-

a. There are around 45000 potential customers online to buy.

b. How many males & females will buy KP481

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

Product	KP281	KP481	KP781	All
Gender				
Female	40	29	7	76
Male	40	31	33	104
All	80	60	40	180

```
#a. ans :- there are total 60 KP481 sold from total of 180
ans = 60/180 * 45000
print(f'We will need total of {ans} number of KP481.')
```

```
We will need total of 15000.0 number of KP481.
```

```
#b. total males and females buying KP481
male = round(76/180 * 45000 * 29/76)
female = round(104/180 * 45000 * 31/104)
print(f'We would be expecting a total of {male} Males and {female} Females buying the product KP481.')
```

```
We would be expecting a total of 7250 Males and 7750 Females buying the product KP481.
```