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

1.Importing the dataset and do usual data analysis steps like checking the structure & characteristics of the dataset

```
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
```

```
# Finfing the null values:
```

```
data.isna().sum()
```

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

```
# lenghth of the data:
```

```
len(data)
```

```
180
```

```
# checkpoints of data:

data.dtypes
```

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

```
#number of unique values in our data:

for i in data.columns:
  print(i,':',data[i].nunique())
```

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

```
# Checking the ocurrences of products:

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

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

```
# information abouth the dataset:

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 the shape of data:(rows,columns)

data.shape
```

```
(180, 9)
```

2.Detect Outliers (using boxplot, “describe” method by checking the difference between mean and median)

```
# Using the describe() method to find out the mean,median,mode and
other quanities od data;

data.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

data

    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]
```

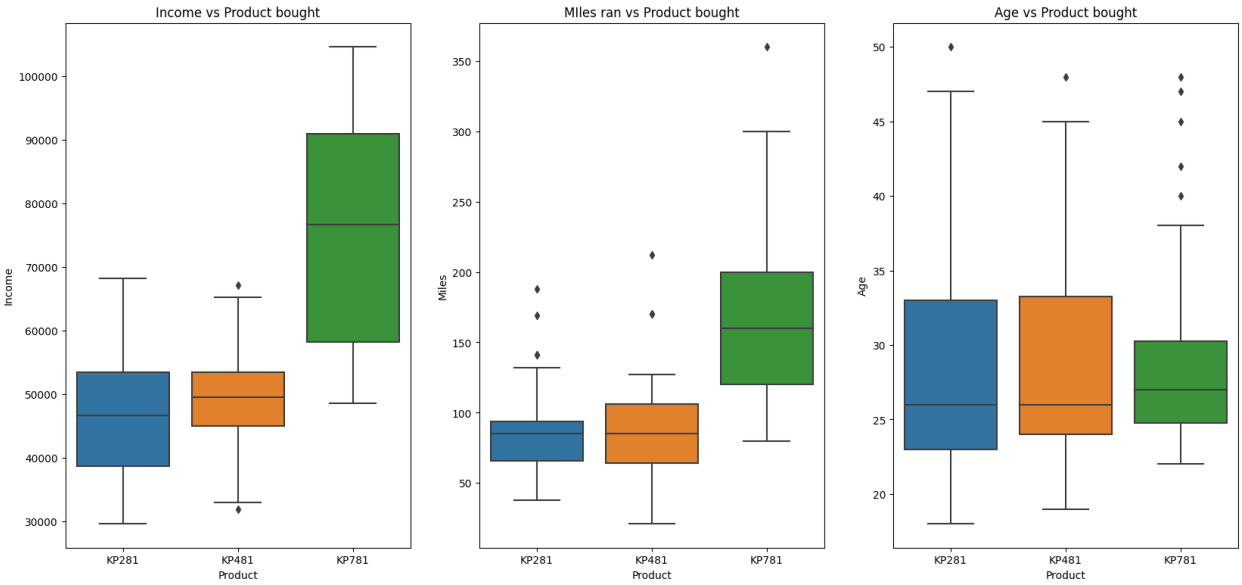
```
data['Product'].value_counts()
```

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

```
# Number of genders in the data:

data['Gender'].value_counts()
```

```
Male      104
Female     76
Name: Gender, dtype: int64
```

```
plt.figure(figsize = (20,9))
plt.subplot(1,3,1)
sns.boxplot(
    x = 'Product',
    y = 'Income',
    data = data
)
plt.title('Income vs Product bought')
plt.subplot(1,3,2)
sns.boxplot(
    x = 'Product',
    y = 'Miles',
    data = data
)
plt.title('MIles ran vs Product bought')
plt.subplot(1,3,3)
sns.boxplot(
    x = 'Product',
    y = 'Age',
    data = data
)
plt.title('Age vs Product bought')

plt.show()
```



1. We can see through the figures that the products brought are differntiated by the Age,income ans miles ran by the customers. 2.in the Income vs product bought **'KP781**' has more encouragment on the basis of buying from high income people and the 1 Quartile value for **'KP781'** ids far more higher then the other two thredmills.
2. The Median and mode value sare quite hight for** KP781** threadmill than the onther two.
3. In the'**Miles vs product bought**',People who runs more are intrested in the KP781 threadmills than any other one.
4. The median value for this plot is also higher for **KP781** threadmill.
5. In the **'Age vs product bought'** plot the young people of age group **22-32** invested more in the two threadmills -- **'KP281'** and **KP481**, Although the percentage of buying is more for these two threadmills the **median** value is higher for **KP781** fro ages from **25-30**.

3.Check if features like marital status, age have any effect on the product purchased (using countplot, histplots, boxplots etc)

```
data
```

```
    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]
```

```
#Grouing the data on basis of Marital staus and the product and
counting the products:


data_mrg = data.groupby(['MaritalStatus','Product']).agg(p_count =
('Product','count')).reset_index()

data_mrg
```

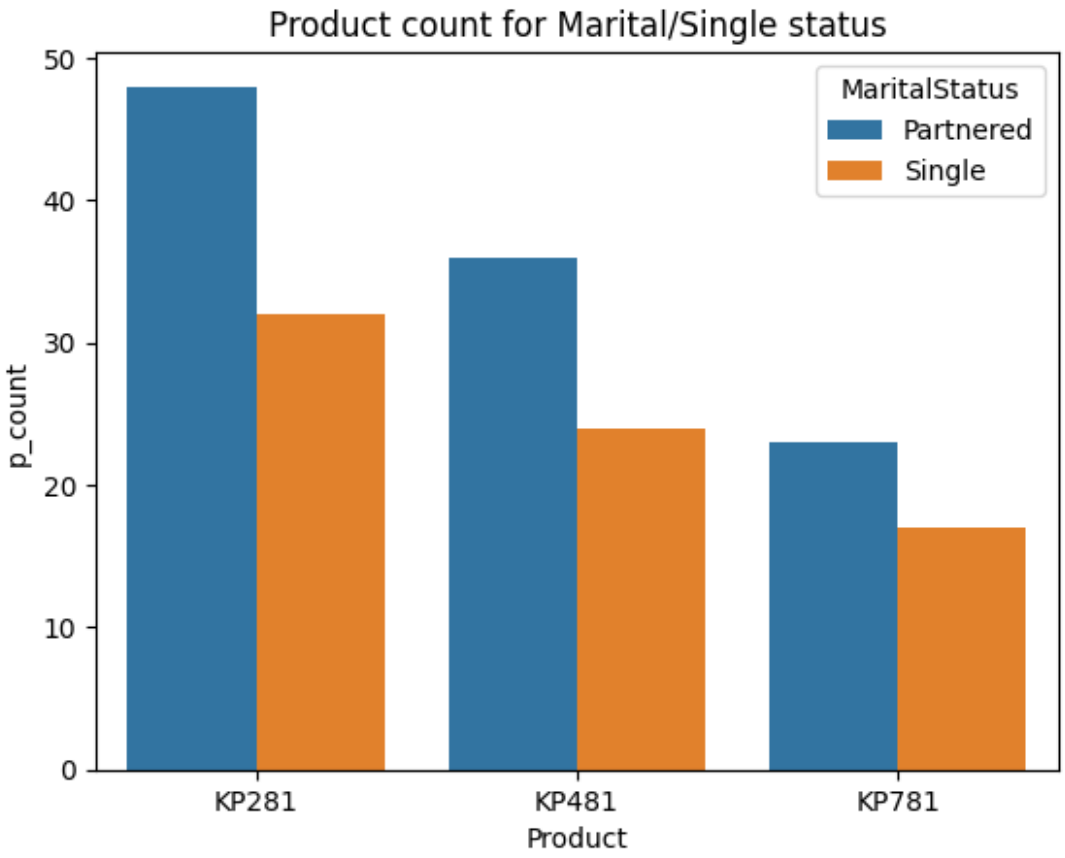
```
  MaritalStatus Product  p_count
0     Partnered   KP281       48
1     Partnered   KP481       36
2     Partnered   KP781       23
3        Single   KP281       32
4        Single   KP481       24
5        Single   KP781       17
```

```
#  grouping the data by age and counting the products:

data_age=data.groupby(['Age']).agg(p_count =
('Product','count')).sort_values(by = 'p_count',ascending =
False).reset_index()

# creating the bins for th eage groups:

bins1 = [-15,15,25,35,45,55]
labels1 = ['<15','15-25','25-35','35-45','45-55']
data_age['age_copy'] = pd.cut(data_age['Age'],bins = bins1,labels =
labels1)

data_age
```

	Age	p_count	age_copy
0	25	25	15-25
1	23	18	15-25
2	24	12	15-25
3	26	12	25-35
4	28	9	25-35
5	35	8	25-35
6	33	8	25-35
7	30	7	25-35
8	38	7	35-45
9	21	7	15-25
10	22	7	15-25
11	27	7	25-35
12	31	6	25-35
13	34	6	25-35
14	29	6	25-35
15	20	5	15-25
16	40	5	35-45
17	32	4	25-35
18	19	4	15-25
19	48	2	45-55
20	37	2	35-45
21	47	2	45-55
22	45	2	35-45
23	44	1	35-45
24	46	1	45-55
25	18	1	15-25
26	43	1	35-45
27	42	1	35-45
28	41	1	35-45
29	39	1	35-45
30	36	1	35-45
31	50	1	45-55

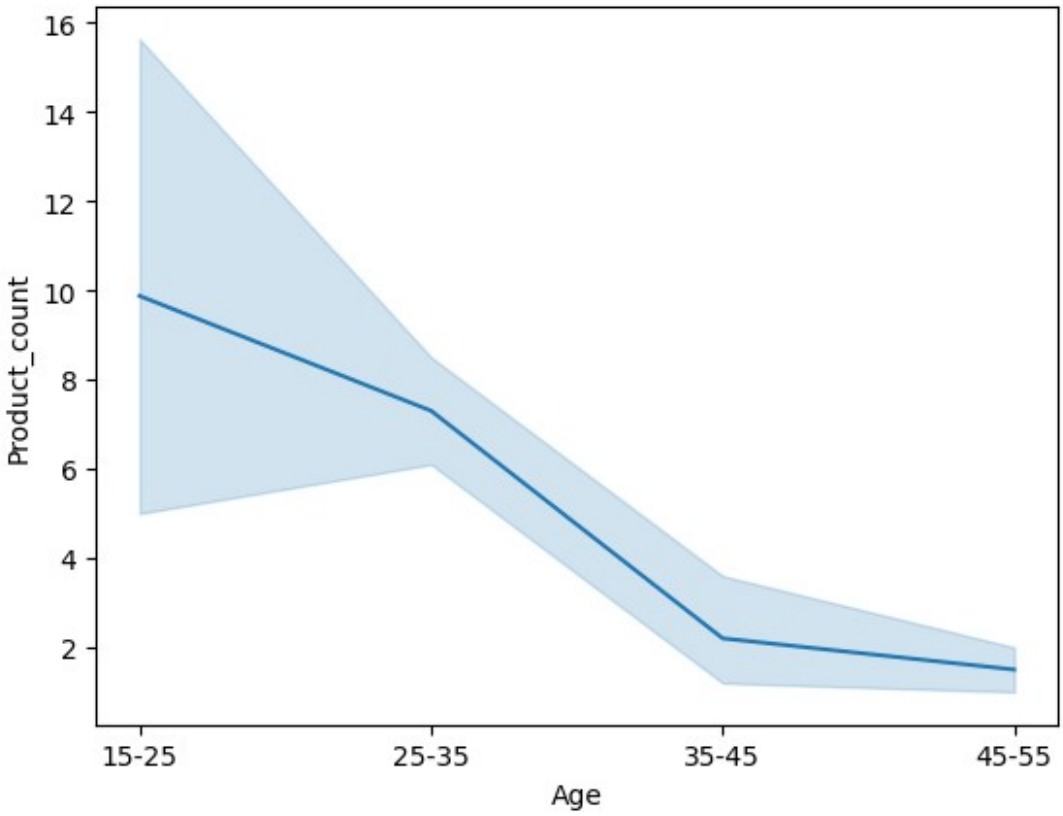
```
# plotting the bar plot for the above marital data

sns.barplot(
    x = 'Product',
    y = 'p_count',
    hue = 'MaritalStatus',
    data = data_mrg
)
plt.title('Product count for Marital/Single status')
```

```
Text(0.5, 1.0, 'Product count for Marital/Single status')
```




```
# line plot for age vs products count:

sns.lineplot(
    x = 'age_copy',
    y = 'p_count',
    data = data_age
)
plt.xlabel('Age'),
plt.ylabel('Product_count')
```

```
Text(0, 0.5, 'Product_count')
```



4.Representing the marginal probability like - what percent of customers have purchased KP281, KP481, or KP781 in a table (can use pandas.crosstab here)

```
# using the cross tab from pandas

pd.crosstab(
    index = data['Product'],
    columns = data['Gender'],
    margins = True,
    margins_name = "Total"


)
```

```
Gender   Female  Male  Total
Product
```

```
KP281        40    40     80
KP481        29    31     60
KP781         7    33     40
Total        76   104    180
```

```
# cross tab for the percentage of people bought the product:

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

data_tab.reset_index()
```

```
Gender Product    Female      Male
0        KP281  0.500000  0.500000
1        KP481  0.483333  0.516667
2        KP781  0.175000  0.825000
3          All  0.422222  0.577778
```

```
data_tab['Female'] = np.round(data_tab['Female']*100,2)

data_tab['Male'] = np.round(data_tab['Male']*100,2)

# percentage of people bought the product include in Gender:

data_tab
```

```
Gender   Female   Male
Product
KP281     50.00  50.00
KP481     48.33  51.67
KP781     17.50  82.50
All       42.22  57.78
```

5.Check correlation among different factors using heat maps or pair plots.

```
data
```

```
    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]
```

```
#plotting the pair plots for different qunatities in the data:

sns.pairplot(data = data,hue = "Product")
```

```
<seaborn.axisgrid.PairGrid at 0x780bf0390a60>
```

1. From the above pairplot the Quantities of the data has been plotted against each othe values. 2.Age vs Miles,education vs Miles,Usage vs Miles, fitness vs Miles,Income vs miles and many more has been plotted.
2. These plotting gives us the information about th eproduct purchased with the different qunatities measurments like income,Fitness,Education and Age.
3. The three products of threadmills KP281,Kp481 and KP781 has the signifcance customers towrads the metrics for each customer differnce.
4. People with young age has been more intreseted inthe KP281 and KP481 threadmill, whereas people with high income and more miles are fitness freaks who wanrs more advancement in their excecise ,hence they choosed KP781 threadmill.
5. People with more Usage and more Educations als preferred KP781.

6. People with age groups 25-30 and eduction level of low are msotly beginners hence they choosed either of KP281 or KP481.

6.With all the above steps you can answer questions like: What is the probability of a male customer buying a KP781 treadmill?

```
# Using the cross tba function to get the results fro male and
females:

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

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

```
# Converting th eresult into percentages:

round(pd.crosstab(
    index = data['Gender'],
    columns = data['Product'],
    margins = True,
    normalize = 'index'
)*100,2)
```

Product	KP281	KP481	KP781
Gender			
Female	52.63	38.16	9.21
Male	38.46	29.81	31.73
All	44.44	33.33	22.22

1.we can clearly see that the probability of male buying KP781 treadmill is 31.73%

1. Males has the high probability of Buying KP781 than female compared.

7.Customer Profiling - Categorization of users.

```
data
```

```
    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]
```

```
# Categorizing the customers on the basis of Income

data.groupby(['Income']).agg(prod_count = ('Product','count'))
```

```
        prod_count
Income
29562            1
30699            1
31836            2
32973            5
```

```
34110            5
...            ...
95508            1
95866            1
99601            1
103336           1
104581           2

[62 rows x 1 columns]
```

```
# max income:

data['Income'].max()
```

```
104581
```

```
# min income:

data['Income'].min()
```

```
29562
```

```
data1 = data

# creating the bins for thr income:

bins1 = [25000,40000,55000,60000,75000,90000,105000]
labels1 = ['25000-40000','40000-55000','55000-60000','60000-
75000','75000-90000','90000-105000']
data1['inc_grp'] = pd.cut(data1['Income'],bins = bins1,labels =
labels1)

#Ctegorizing the customers on th ebasis of miles they run:

data.groupby(['Miles']).agg(prod_count =
('Product','count')).sort_values(by = 'Miles',ascending = True)
```

```
       prod_count
Miles
21              1
38              3
42              4
47              9
53              7
56              6
64              6
66             10
74              3
75             10
80              1
85             27
94              8
```

```
95              12
100              7
103              3
106              9
112              1
113              8
120              3
127              5
132              2
140              1
141              2
150              4
160              5
169              1
170              3
180              6
188              1
200              6
212              1
240              1
260              1
280              1
300              1
360              1
```

```
# Creting the bins for the miles ran:

bins1 = [-50,50,100,150,200,250,300,350,400]
labels1 = ['<50','50-100','100-150','150-200','200-250','250-
300','300-350','350-400']
data1['miles_grp'] = pd.cut(data1['Miles'],bins = bins1,labels =
labels1)

data1
```

```
    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	inc_grp	miles_grp
0	112	25000-40000	100-150
1	75	25000-40000	50-100
2	66	25000-40000	50-100
3	85	25000-40000	50-100
4	47	25000-40000	<50
..	...	...	...
175	200	75000-90000	150-200
176	200	75000-90000	150-200
177	160	90000-105000	150-200
178	120	90000-105000	100-150
179	180	90000-105000	150-200

[180 rows x 11 columns]

```
data12 = data1.groupby(['Product','miles_grp']).agg(product_count =
('Product','count')).reset_index().sort_values(by =
'Product',ascending = True)
```

```
data12
```

	Product	miles_grp	product_count
0	KP281	<50	12
1	KP281	50-100	50
2	KP281	100-150	16
3	KP281	150-200	2
4	KP281	200-250	0
5	KP281	250-300	0
6	KP281	300-350	0
7	KP281	350-400	0
15	KP481	350-400	0
14	KP481	300-350	0
13	KP481	250-300	0
12	KP481	200-250	1
11	KP481	150-200	2
10	KP481	100-150	13
9	KP481	50-100	39
8	KP481	<50	5
16	KP781	<50	0
17	KP781	50-100	8

```
18   KP781   100-150              9
19   KP781   150-200             18
20   KP781   200-250              1
21   KP781   250-300              3
22   KP781   300-350              0
23   KP781   350-400              1
```

data1

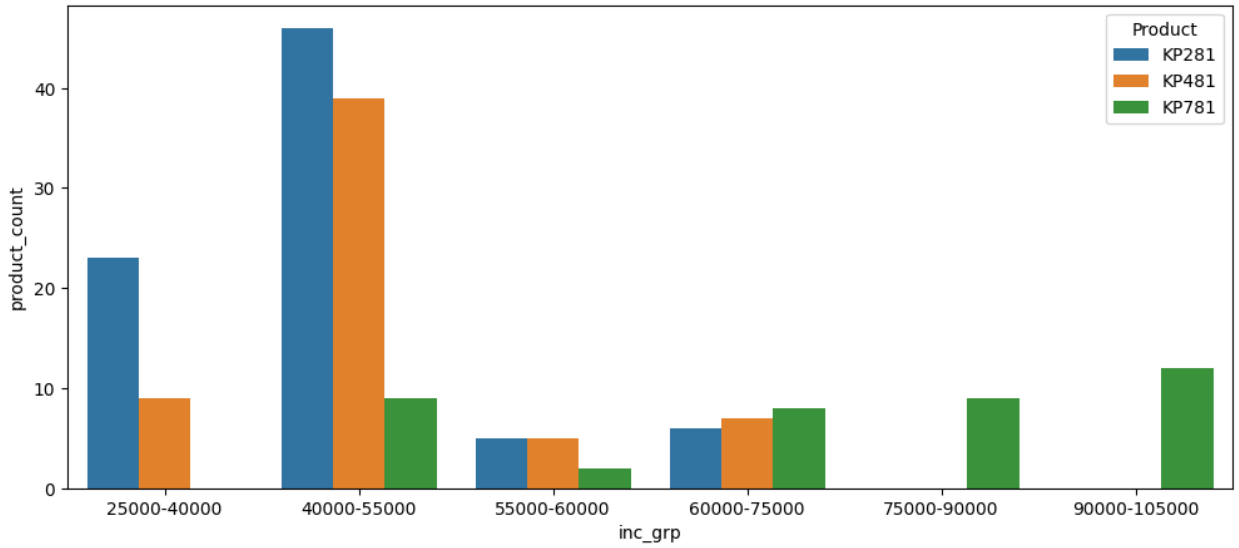
```
    Product  Age  Gender  Education MaritalStatus  Usage  Fitness
Income  \
0     KP281   18    Male         14        Single      3        4
29562
1     KP281   19    Male         15        Single      2        3
31836
2     KP281   19  Female         14     Partnered      4        3
30699
3     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       inc_grp miles_grp
0      112   25000-40000   100-150
1       75   25000-40000    50-100
2       66   25000-40000    50-100
3       85   25000-40000    50-100
4       47   25000-40000       <50
..     ...           ...       ...
175    200   75000-90000   150-200
176    200   75000-90000   150-200
177    160  90000-105000   150-200
178    120  90000-105000   100-150
179    180  90000-105000   150-200

[180 rows x 11 columns]
```

```
# grouping the data by product and income group:

data11 = data1.groupby(['Product','inc_grp']).agg(product_count =
('Product','count')).reset_index().sort_values(by =
'Product',ascending = True)

# Information about the customers income groups and the number of
products bought:

pd.crosstab(
    index = data1['inc_grp'],
    columns =data1['Product'],
    margins = True,
    margins_name = 'Total'
).reset_index()
```

Product	inc_grp	KP281	KP481	KP781	Total
0	25000-40000	23	9	0	32
1	40000-55000	46	39	9	94
2	55000-60000	5	5	2	12
3	60000-75000	6	7	8	21
4	75000-90000	0	0	9	9
5	90000-105000	0	0	12	12
6	Total	80	60	40	180

```
 # figure showing the income groups and the product bought:

plt.figure(figsize = (12,5))
sns.barplot(x = 'inc_grp',
            y = 'product_count',
            data = data11,
            hue = 'Product')
```

```
<Axes: xlabel='inc_grp', ylabel='product_count'>
```

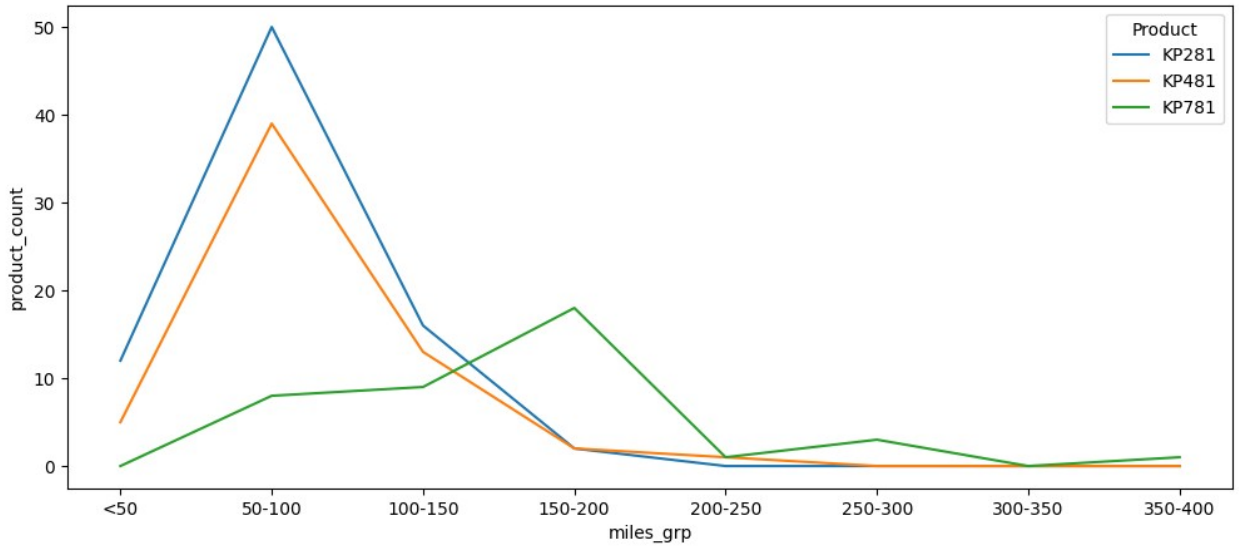



```
# information about the customers miles they ran and the product
bought:

pd.crosstab(
    index = data1['miles_grp'],
    columns =data1['Product'],
    margins = True,
    margins_name = 'Total'
).reset_index()
```

Product	miles_grp	KP281	KP481	KP781	Total
0	<50	12	5	0	17
1	50-100	50	39	8	97
2	100-150	16	13	9	38
3	150-200	2	2	18	22
4	200-250	0	1	1	2
5	250-300	0	0	3	3
6	350-400	0	0	1	1
7	Total	80	60	40	180

```
# line plot for the miles ran gropued and the number of products they
bought:


plt.figure(figsize=(12,5))
sns.lineplot(
    x = 'miles_grp',
    y = 'product_count',
    data = data12,
    hue = 'Product'
)
```

```
<Axes: xlabel='miles_grp', ylabel='product_count'>
```




```
data1
```

	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	inc_grp	miles_grp
0	112	25000-40000	100-150
1	75	25000-40000	50-100
2	66	25000-40000	50-100
3	85	25000-40000	50-100

```
4       47   25000-40000       <50
..     ...           ...       ...
175    200   75000-90000   150-200
176    200   75000-90000   150-200
177    160  90000-105000   150-200
178    120  90000-105000   100-150
179    180  90000-105000   150-200

[180 rows x 11 columns]
```

```
# GATble shwing the fitness levels for Male/Female

pd.crosstab(
    index = data1['Fitness'],
    columns = data1['Gender'],
    margins = True,
    margins_name = 'Total'

)
```

Gender	Female	Male	Total
Fitness			
1	1	1	2
2	16	10	26
3	45	52	97
4	8	16	24
5	6	25	31
Total	76	104	180

```
# percentage for the amle/female fitness levels:

round(pd.crosstab(
    index = data1['Fitness'],
    columns = data1['Gender'],
    normalize = 'index'

)*100,2)
```

Gender	Female	Male
Fitness		
1	50.00	50.00
2	61.54	38.46
3	46.39	53.61
4	33.33	66.67
5	19.35	80.65

RECOMMENDATIONS AND INSIGHTS:

1. The Aerofit Threadmill products KP281, KP481 and KP781 has good percentage of consumers from the given data.
2. More peoplebeteween the age groups 25-30 are intrested in the lower(KP281) and mid(KP481) versions of the threadmills
3. People with high income and high fitness are more attracted towards Advanced(KP781)model.
4. People with more age and newbies are good for the beginner friendlt KP281 and mid level KP481.
5. The Advance Options in the KP781 has to be explained to people who are in these conditions yet make goof effort for the fitness.
6. Buying the KP781 may be costly, so discounts according to the market and palce has to be introduced for the better sale of KP781.
7. More advertisements can be done for the Advanced machine.
8. Making customers comfaotable in all aspects in the KP781 model will increase the sales of it.
9. Getting the awrness of the people in different categories like usageyime,gender,age,health issues etc will also help in boosting of sales.
10. These are some recommendations from my side.