Aerofit case study

- **Dataset:** Aerofit Dataset
- **Name:** S.Tejeswara Rao
- **Email:** tejavishnu2000@gmail.com

Problem statement:

- 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.

Importing required python libraries

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

Read aerofit dataset

```
aerofit_data=pd.read_csv('/content/sample_data/aerofit_treadmill.csv')
```

```
aerofit_data.head(7)
```

	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
5	KP281	20	Female	14	Partnered	3	3	32973	66
6	KP281	21	Female	14	Partnered	3	3	35247	75

Next steps: Generate code with `aerofit_data` View recommended plots

Dataset size in terms of rows/records and columns/properties.

```
aerofit_data.shape
```

```
(180, 9)
```

- No.of rows/records = 180
- Nof columns/properties = 9

Datatypes, non-null values and how much memory it consumes.

```
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
```

we coluld observe that the all columns contains the proper data types.

we could see Numerical and categorical columns below.

```
numeric_columns=aerofit_data.select_dtypes(include='number').columns
categorical_columns=aerofit_data.select_dtypes(include='object').columns
print(f"There are {len(numeric_columns)} numeric_columns = {list(numeric_columns)}")
print(f"There are {len(categorical_columns)} categorical_columns = ",list(categorical_columns))
```

```
There are 6 numeric_columns = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
There are 3 categorical_columns =  ['Product', 'Gender', 'MaritalStatus']
```

Let's verifying our dataset contains any duplicate values or not.

```
aerofit_data.duplicated().sum()
```

```
0
```

There are no duplicate values in our dataset.

Data Cleaning and Handling Missing values:

```
aerofit_data.isnull().sum()
```

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

```
Miles            0
dtype: int64
```

Our aerofit daset doesn't contain even single null value. so dataset is clean and no null values and proper, no need to do any processing for data cleaning / handling null values since no null values.

Value counts VS Nunique for every column in our dataset.

```
for i in aerofit_data.columns:
  print(f'{i} : {aerofit_data[i].nunique()}')
```

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

```
for i in aerofit_data.columns:
  print(f'{i} : {aerofit_data[i].value_counts()}')
  print("-"*50)
```

```
132     2
141     2
280     1
260     1
300     1
240     1
112     1
212     1
80      1
140     1
21      1
169     1
188     1
360     1
Name: count, dtype: int64
--------------------------------------------------
```

We can see statastical description of all the numerical columns of our dataset.

```
for i in numeric_columns:
  print(f'{i} : {aerofit_data[i].describe()}')
  print("-"*50)
```

```
std        6.943498
min       18.000000
25%       24.000000
50%       26.000000
75%       33.000000
max       50.000000
Name: Age, dtype: float64
--------------------------------------------------
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
Name: Education, dtype: float64
--------------------------------------------------
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
Name: Usage, dtype: float64
--------------------------------------------------
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
Name: Fitness, dtype: float64
--------------------------------------------------
Income : count       180.000000
mean      53719.577778
std       16506.684226
min       29562.000000
25%       44058.750000
50%       50596.500000
75%       58668.000000
```

```
--------------------------------------------------
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
Name: Miles, dtype: float64
--------------------------------------------------
```

```
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

Outliers Detection:

```
for i in numeric_columns:
  lower_wisk=aerofit_data[i].quantile(0.25)
  upper_wisk=aerofit_data[i].quantile(0.75)
  IQR=upper_wisk-lower_wisk
  print(f'{i} : outliers which are at high peack are: {(aerofit_data[i]>(upper_wisk+1.5*IQR)).sum()}')
  print(f'{i} : outliers which are at low peack are: {(aerofit_data[i]<(lower_wisk-1.5*IQR)).sum()}')
  print('_'*50)
```

```
Age : outliers which are at high peack are: 5
Age : outliers which are at low peack are: 0
__________________________________________________
Education : outliers which are at high peack are: 4
Education : outliers which are at low peack are: 0
__________________________________________________
Usage : outliers which are at high peack are: 9
Usage : outliers which are at low peack are: 0
__________________________________________________
Fitness : outliers which are at high peack are: 0
Fitness : outliers which are at low peack are: 2
__________________________________________________
Income : outliers which are at high peack are: 19
Income : outliers which are at low peack are: 0
__________________________________________________
Miles : outliers which are at high peack are: 13
Miles : outliers which are at low peack are: 0
__________________________________________________
```

Age : There are 5 members which are too aged.

Education : There are 4 members which are profitionals and well educated.

Usage : There are 9 members which are mostly uses the treadmill.

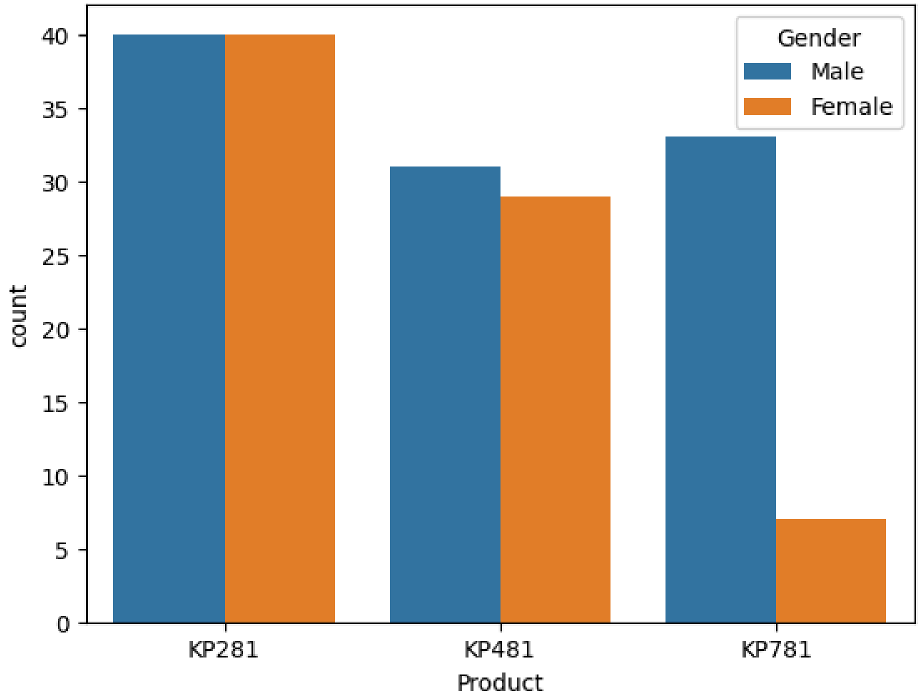
Fitness : There are 2 members which are 2 members which are less fintness.

Income : There are 19 members which we can get the more income.

Miles : There are 13 members which are running/walking more more distance.

Now we can see the different product wise outliers:

```
sns.countplot(x=aerofit_data['Product'],hue=aerofit_data['Gender'])
plt.show()
```




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

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

Products KP281 and KP481 are having almost similar number of males and females.

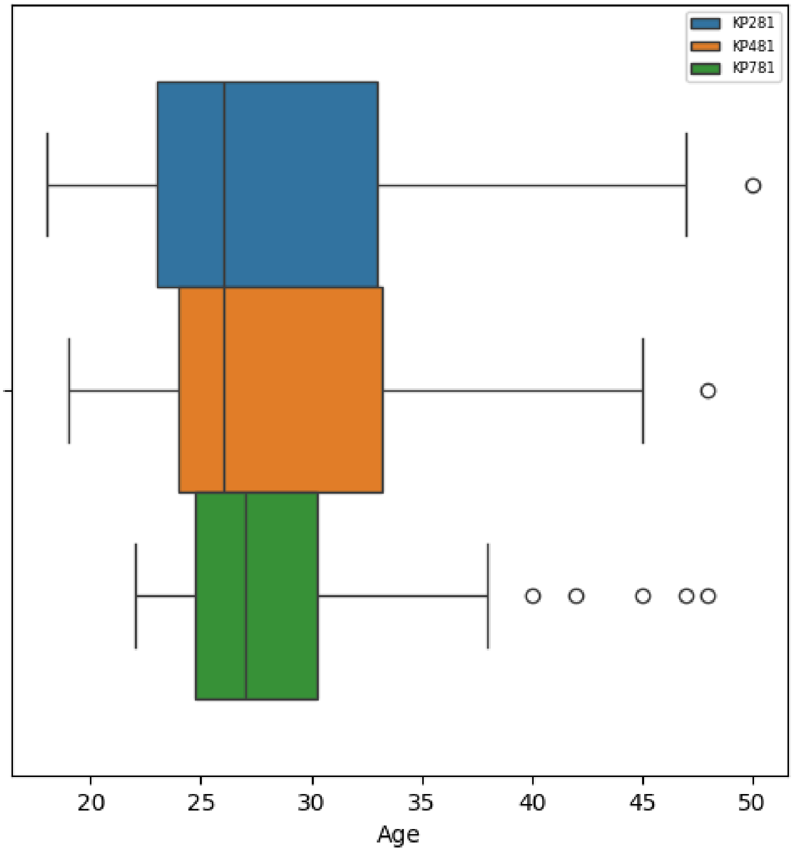
Total KP281 customers = 80

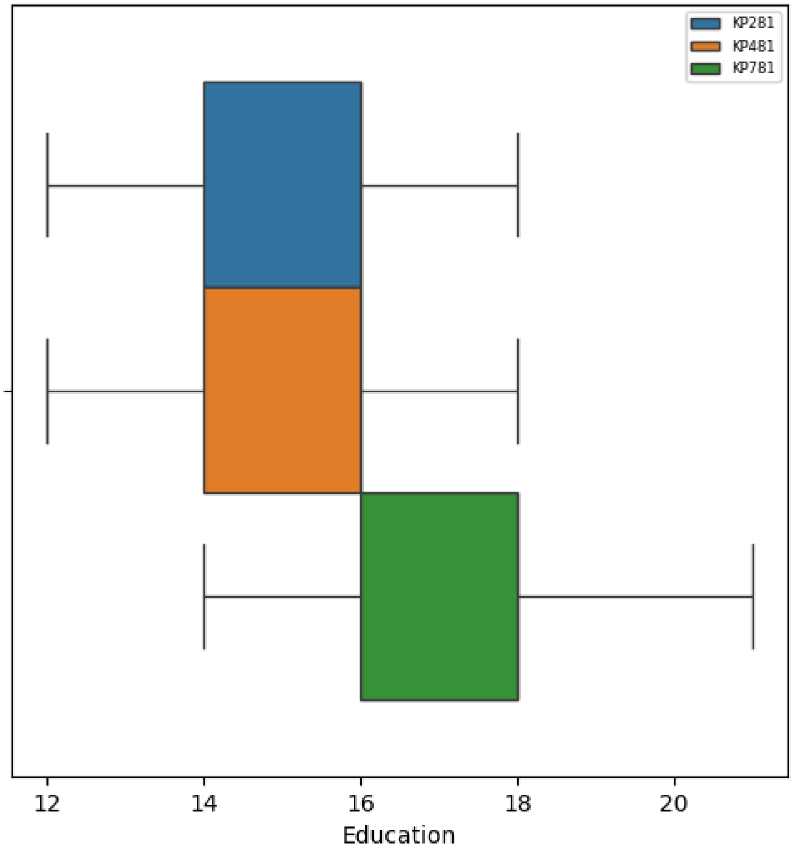
Total KP481 customers = 60

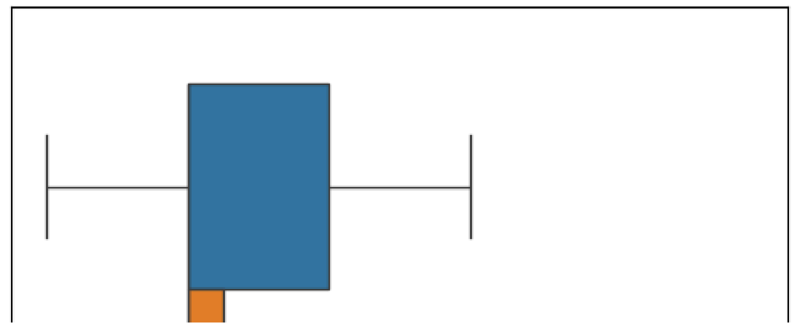
But product KP781 are almost males there is no female customers are there.

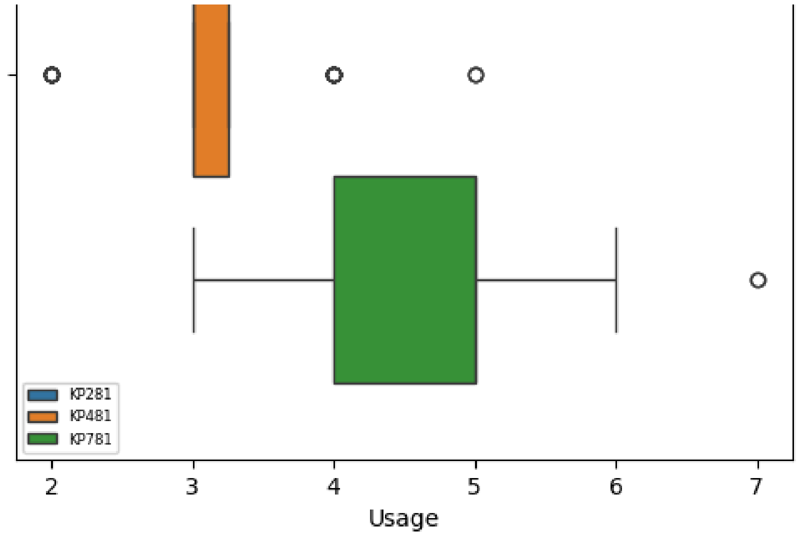
Total KP781 customers = 40

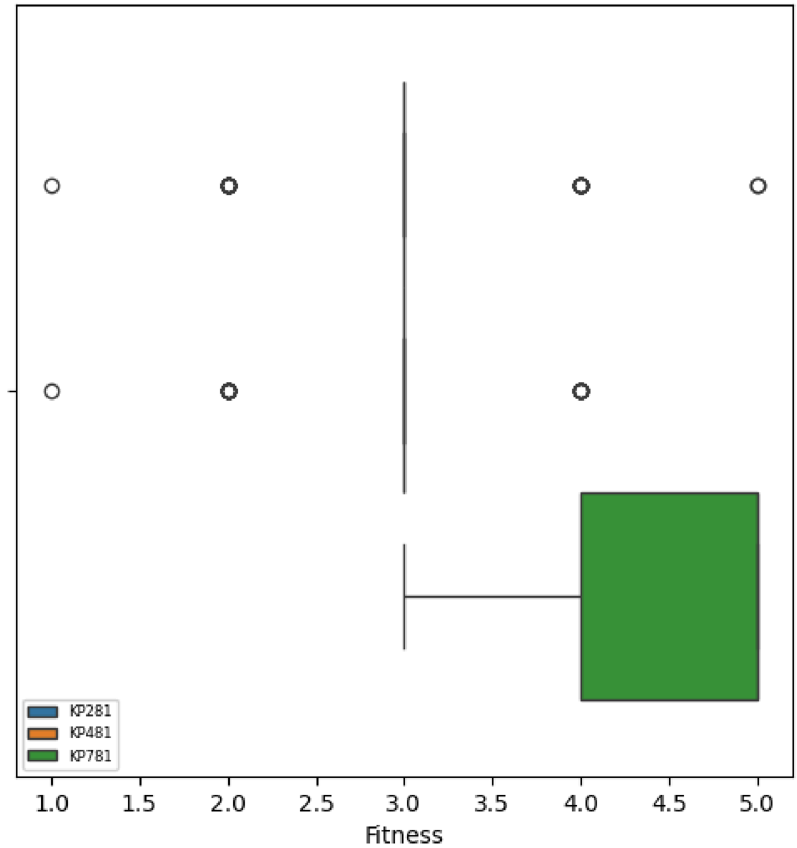
```
p=0
for i in numeric_columns:
  plt.figure(figsize=(6,6))
  sns.boxplot(data=aerofit_data,x=i,hue='Product')
  if p>=2:
    plt.legend(loc='lower left',prop={'size': 6})
  else:
    plt.legend(loc='upper right',prop={'size': 6})
  p+=1
  plt.show()
```

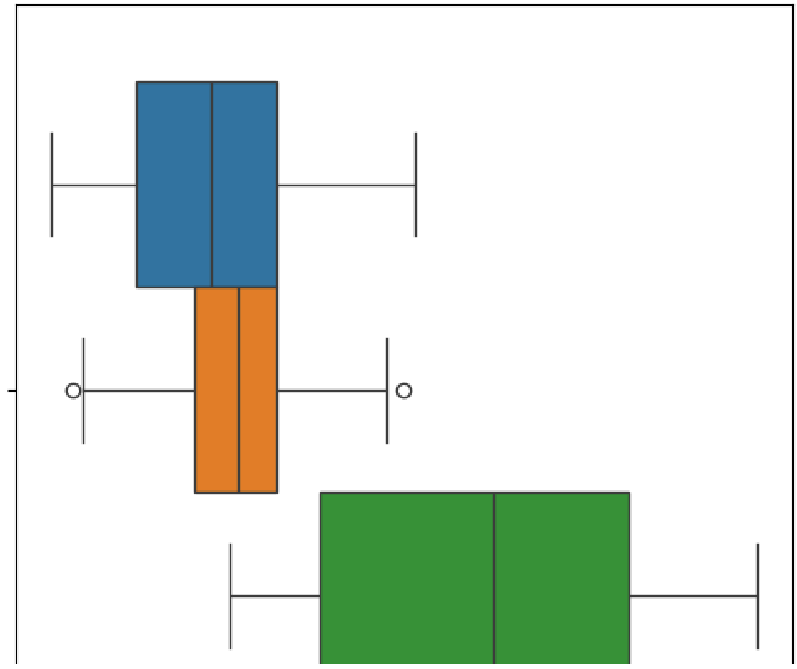

KP281
KP481
KP781
20
25
30
35
40
45
50
Age

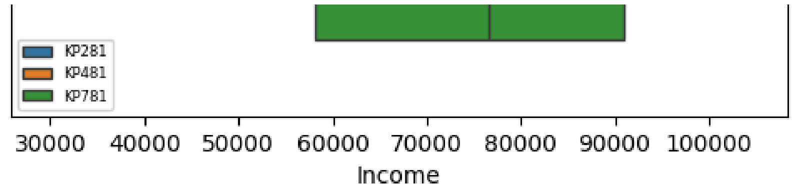

KP281
KP481
KP781
12
14
16
18
20
Education

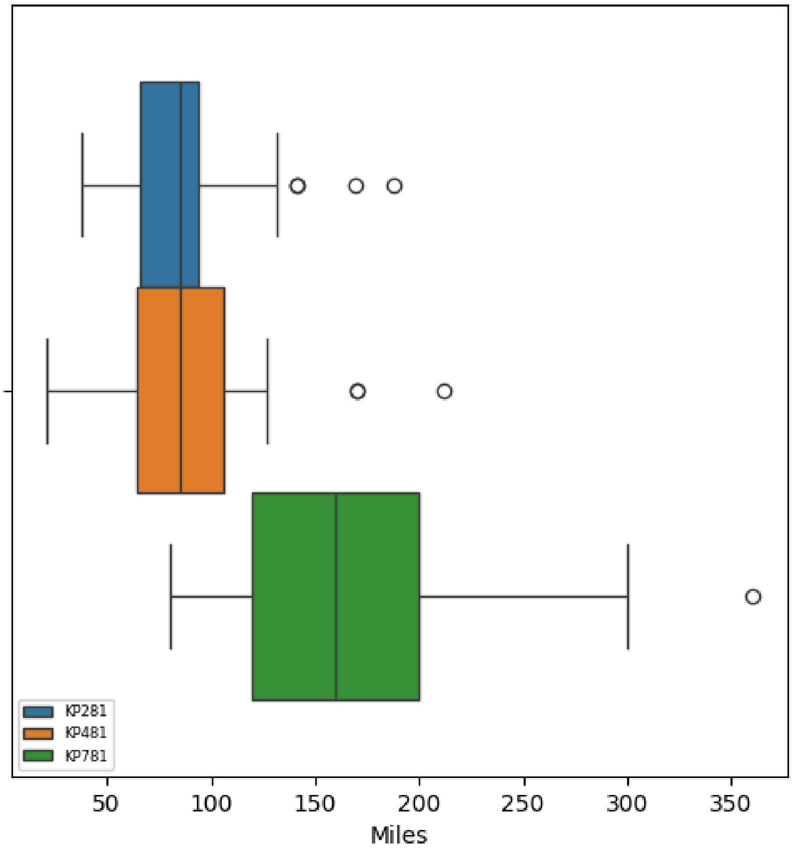



KP281
KP481
KP781
2
3
4
5
6
7
Usage


KP281
KP481
KP781
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Fitness







Outlier Treatment:

```
outlier_columns=['Age','Education','Usage','Fitness','Income','Miles']
for i in outlier_columns:
  aerofit_data[i]=np.where(aerofit_data[i]<aerofit_data[i].quantile(0.05),aerofit_data[i].quantile(0.05),ae
  aerofit_data[i]=np.where(aerofit_data[i]>aerofit_data[i].quantile(0.95),aerofit_data[i].quantile(0.95),ae
print(aerofit_data.head())
```

```
  Product   Age  Gender  Education MaritalStatus  Usage  Fitness    Income  \
0   KP281  20.0    Male       14.0        Single    3.0      4.0  34053.15
1   KP281  20.0    Male       15.0        Single    2.0      3.0  34053.15
2   KP281  20.0  Female       14.0     Partnered    4.0      3.0  34053.15
3   KP281  20.0    Male       14.0        Single    3.0      3.0  34053.15
4   KP281  20.0    Male       14.0     Partnered    4.0      2.0  35247.00

   Miles
0  112.0
1   75.0
2   66.0
3   85.0
4   47.0
```

Age & maritual status

We could see how age and maritual status effect and they bought product.

```
aerofit_data['Age'].value_counts().reset_index().sort_values(by='Age').reset_index(drop=True)
```

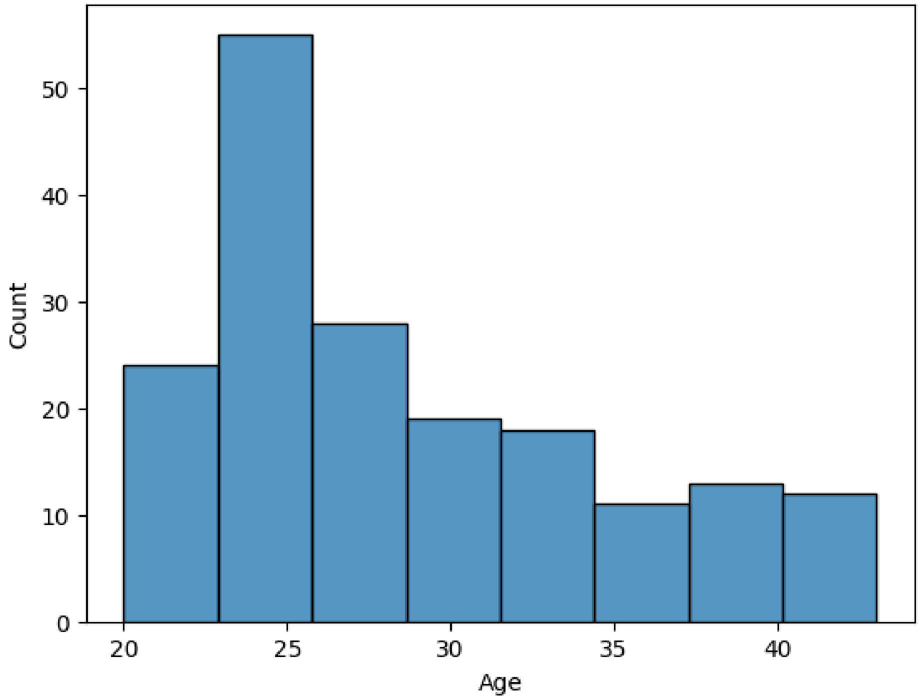
	Age	count
0	20.00	10
1	21.00	7
2	22.00	7
3	23.00	18
4	24.00	12
5	25.00	25
6	26.00	12
7	27.00	7
8	28.00	9
9	29.00	6
10	30.00	7
11	31.00	6
12	32.00	4
13	33.00	8
14	34.00	6
15	35.00	8
16	36.00	1
17	37.00	2
18	38.00	7
19	39.00	1
20	40.00	5
21	41.00	1
22	42.00	1
23	43.00	1
24	43.05	9

Unique number of ages for all the customers.

```
print(f"Customers with staring age: {(aerofit_data['Age']).min()}")
print(f"Customers with ending age: {aerofit_data['Age'].max()}")
print(f"Total unique ages: {aerofit_data['Age'].nunique()}")
```

```
Customers with staring age: 20.0
Customers with ending age: 43.04999999999998
Total unique ages: 25
```

```
sns.histplot(data=aerofit_data,x='Age',bins=8)
plt.show()
```



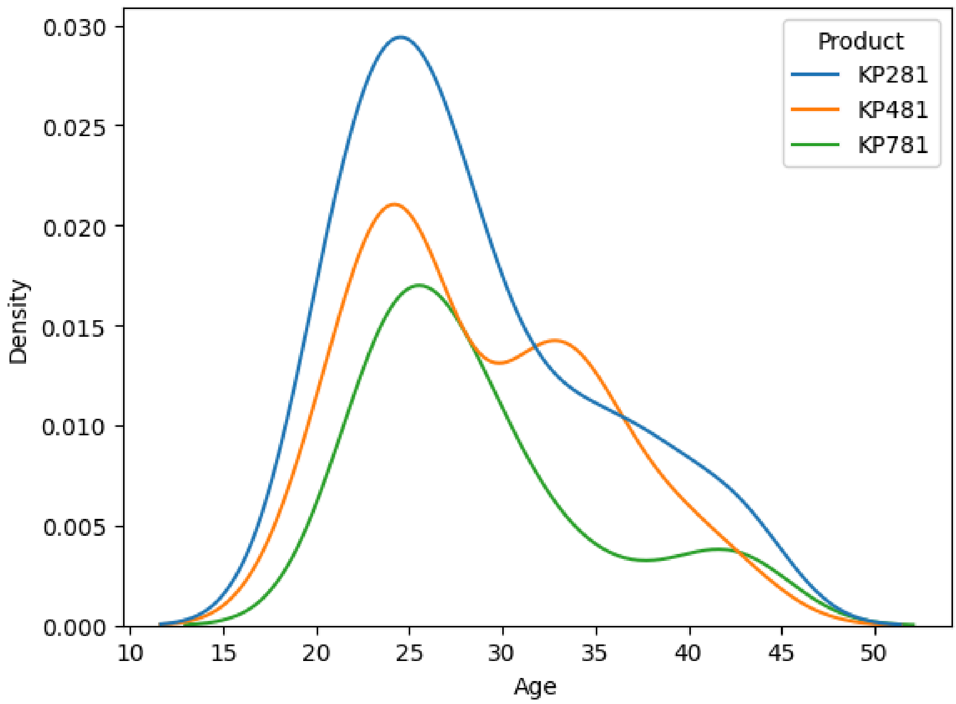
Insights:

We can clearly see that the more peple between the age 23 to 26 are bought more number of products.

After 24 years slowly decreses the number of product buying people.

Product wise different age group customers

```
sns.kdeplot(data=aerofit_data,x='Age',hue='Product')
plt.show()
```



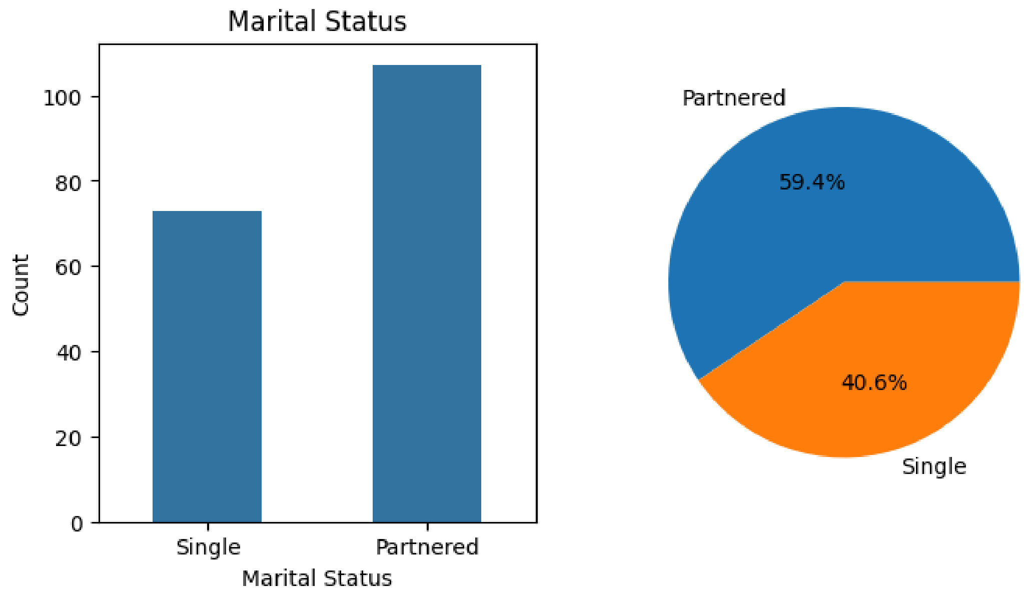
Insights:

KP781 product has more no.of customers with high propability. Most of this customers are 20-35 age group people.

```
# Maritual status

aerofit_data['MaritalStatus'].value_counts()
```

```
MaritalStatus
Partnered    107
Single        73
Name: count, dtype: int64
```

```
plt.figure(figsize=(8,4))
plt.subplot(1,2,1)
sns.countplot(data=aerofit_data,x='MaritalStatus',width=0.5)
plt.title('Marital Status')
plt.xlabel('Marital Status')
plt.ylabel('Count')

plt.subplot(1,2,2)
plt.pie(aerofit_data['MaritalStatus'].value_counts(),labels=aerofit_data['MaritalStatus'].value_counts().ir

plt.show()
```



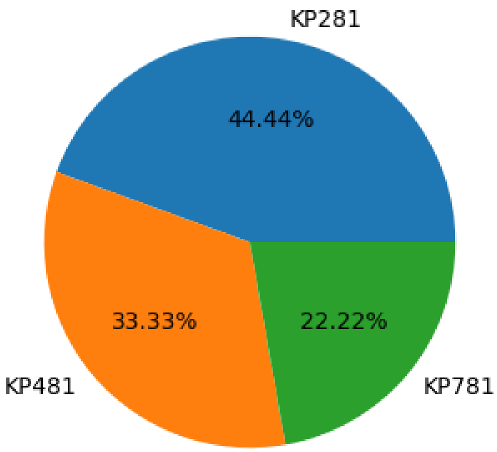
Insights:

No.of married customers =107 and 59.4% customers are married.

No.of un-married customers = 73 and 40.6% customers are single.

what percent of customers have purchased KP281, KP481, or KP781 in a table

```
plt.figure(figsize=(8,4))
plt.pie(aerofit_data['Product'].value_counts(),labels=aerofit_data['Product'].value_counts().index,autopct=
plt.show()
```



Out of all the customers

- KP281 : 44.44%
- KP481 : 33.34%
- KP781 : 22.22%

```
cross_tab=pd.crosstab(index=aerofit_data['Product'],columns=aerofit_data['Gender'],normalize=True,margins=T
cross_tab.reset_index(inplace=True)
cross_tab
```

Gender	Product	Female	Male	All
0	KP281	0.222222	0.222222	0.444444
1	KP481	0.161111	0.172222	0.333333
2	KP781	0.038889	0.183333	0.222222
3	All	0.422222	0.577778	1.000000

Next steps: Generate code with `cross_tab` View recommended plots

Insights:

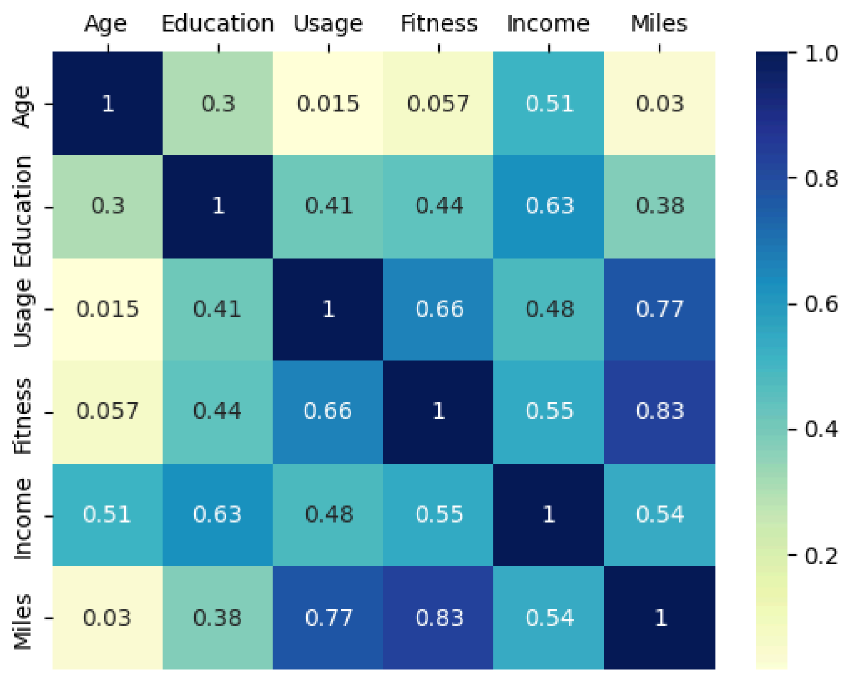
From the cross table we can easily say that below propabilities.

- Probability of prodact KP281 is P(Product_KP281) = 0.4444 ==> 44.44% of the user uses this product.
- Probability of prodact KP481 is P(Product_KP481) = 0.333 ==> 33.33% of the user uses this product.
- Probability of prodact KP781 is P(Product_KP781) = 0.222 ==> 22.22% of the user uses this product.
- Probability of female users P(Female) = 0.4222 ==> 42.22% users are female candidates.
- Probability of female users P(Female) = 0.5777 ==> 57.77% users are male candidates.
- Probability of male users for product KP281 is P(Product_KP281 & male) = 0.2222 ==> 22.22% male users uses the product KP281.
- Probability of female users for product KP281 is P(Product_KP281 & female) = 0.2222 ==> 22.22% female users uses the product KP281.
- Probability of male users for product KP481 is P(Product_KP481 & male) = 0.1722 ==> 17.22% male users uses the product KP481.
- Probability of female users for product KP481 is P(Product_KP481 & female) = 0.1611 ==> 16.11% female users uses the product KP481.

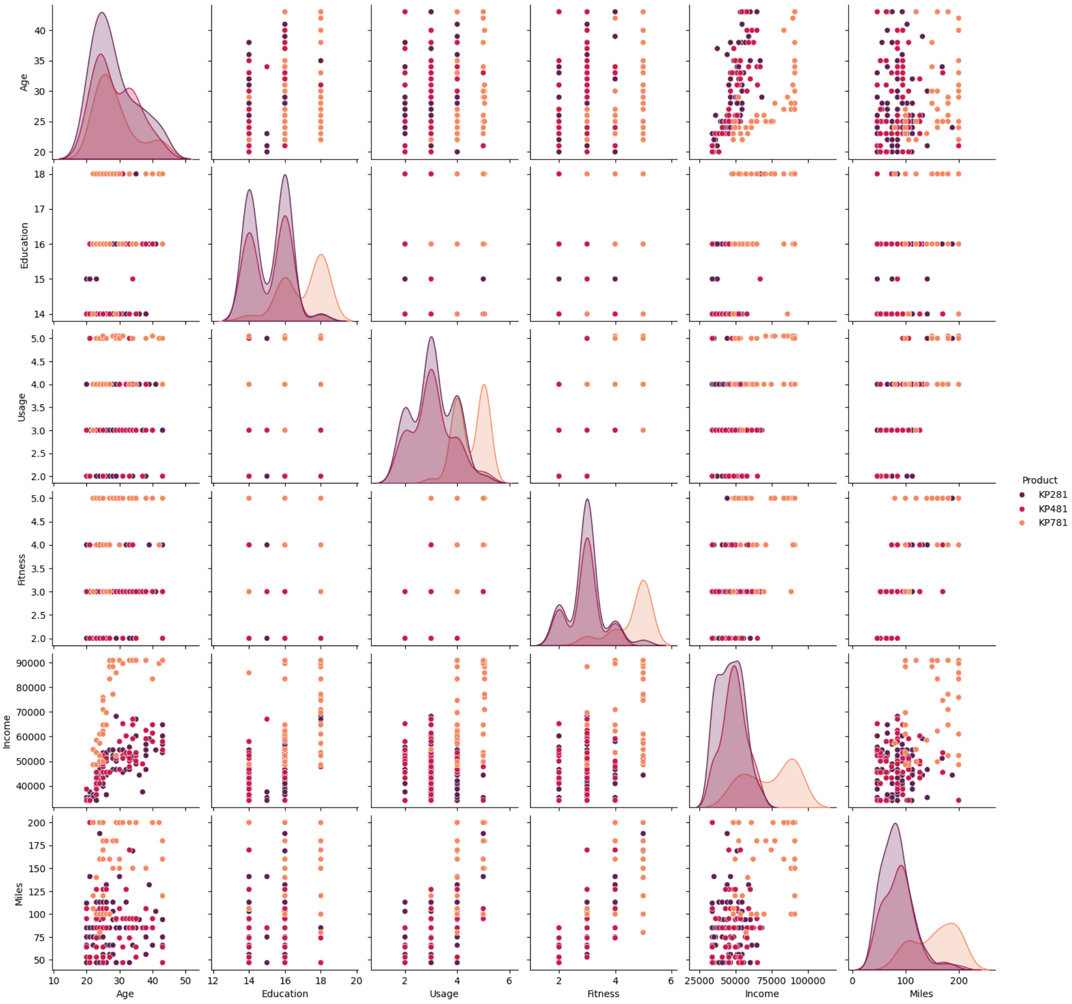
- Probability of male users for product KP781 is P(Product_KP781 & male) = 0.1833 ==> 18.33% male users uses the product KP781.
- Probability of female users for product KP781 is P(Product_KP781 & female) = 0.0388 ==> 3.88% female users uses the product KP781.

correlation among different factors using heat maps or pair plots.

```
ax=sns.heatmap(aerofit_data[numeric_columns].corr(),annot=True,cmap='YlGnBu')
ax.set(xlabel="", ylabel="")
ax.xaxis.tick_top()
plt.show()
```




```
sns.pairplot(aerofit_data,hue='Product',palette='rocket')
plt.show()
```



Insights:

Age and Income: Age and income are positively corelated which means older age people with high income and lower age with lower income.

Education and Income: Strong positive correlation between these 2.

Education and Fitness: Well educated people utilizes/uses the trendmill equipment and they plan to become more fit. These are in good positive corelation.

Fitness and Usage: People who are fit alwas uses the trendmill quipment and they plan accordinggly.

Income and usage: The people who got more income are always uses the trendmill properly -Positive corelation.

Income and fitness: The pople who are more fit, got more income. Psitively corelated.

Distance to walk/run and fitness: who are ruuning/walking more to be more fit. Here we can see strong positive correlation between them means if walks more distance to become more fit.

Distance travel by walk and usage: who walks more, they uses trendmill properly.

Income and distance travel by walk: These are positively corelated, more income people walked more avg distance per week.

Final summary: People who are well educayted and experienced uses the trendmill properly and the are more fit and walked more distance on an average per week.

Customer Profiling and Recomendations:

```
#Converting age to descrite column with low,medium and above_medium and high
aerofit_data['Age_group']=pd.cut(aerofit_data['Age'],bins=[18,26,34,42,50],labels=['low','medium','above_me

#Converting No.of years they educated to descrite column with level_1,level_2 and level_3
aerofit_data['Education_group']=pd.cut(aerofit_data['Education'],bins=[12,15,18,21],labels=['level_1','leve

#Converting finess to descrite column with level_1,level_2 and level_3
aerofit_data['Fitness_group']=pd.cut(aerofit_data['Fitness'],bins=[1,2,3,5],labels=['fitness_poor','fitness
```

```
aerofit_data['Age_group'].value_counts()
```

```
Age_group
low             91
medium          53
above_medium    26
high            10
Name: count, dtype: int64
```

```
aerofit_data['Education_group'].value_counts()
```

```
Education_group
level_2    112
level_1     68
level_3      0
Name: count, dtype: int64
```

```
aerofit_data['Fitness_group'].value_counts()
```

```
Fitness_group
fitness_average    97
fitness_good       55
fitness_poor       28
Name: count, dtype: int64
```

Note:

Age_group

low = between 18 and 26 years

medium = between 26 and 33 years

above_medium = between 34 and 42 years

high = between 42 and 50 years

Education_group:

level_1 = 12-15 education duration in years

level_2 = 15-18 education duration in years

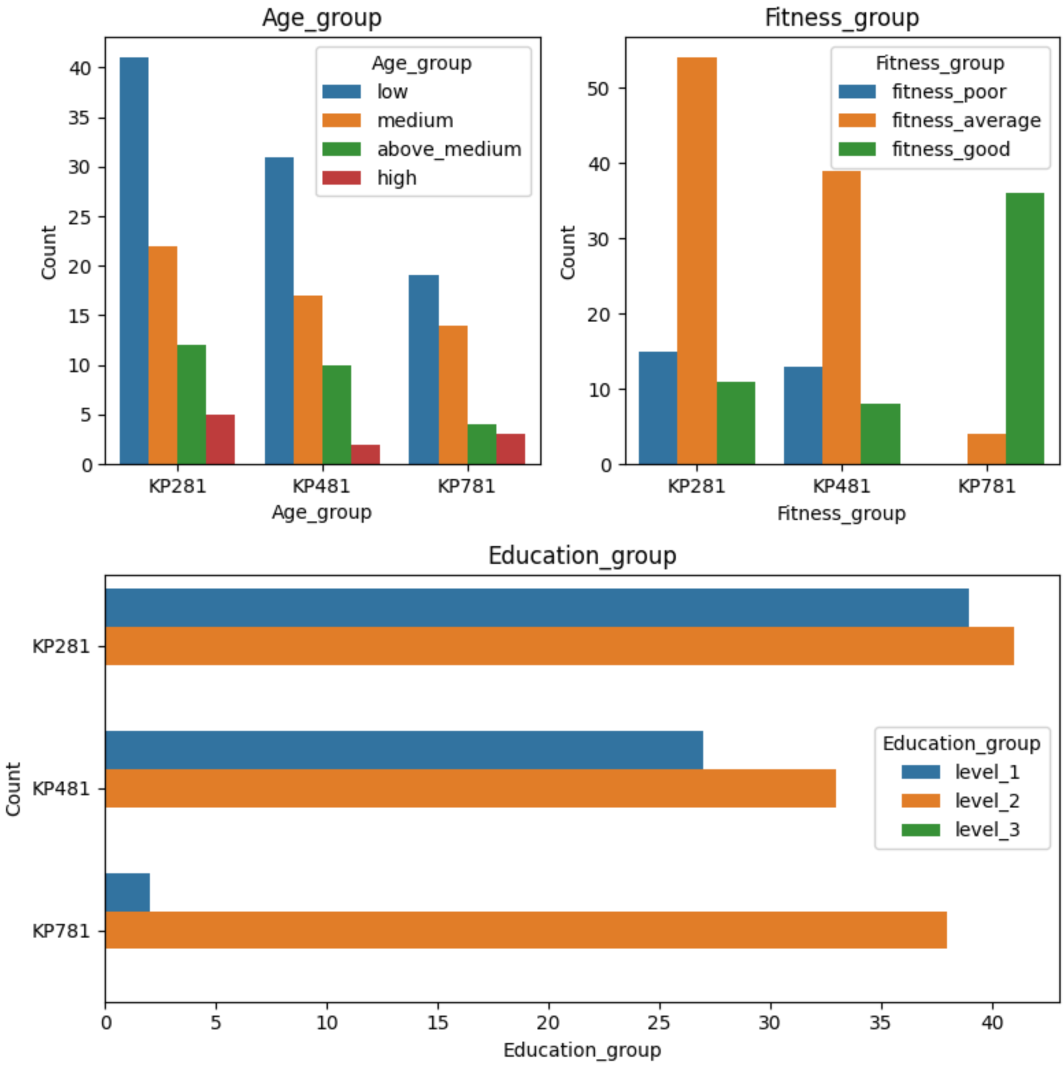
level_3 = 18-21 education duration in years

Fitness_group:

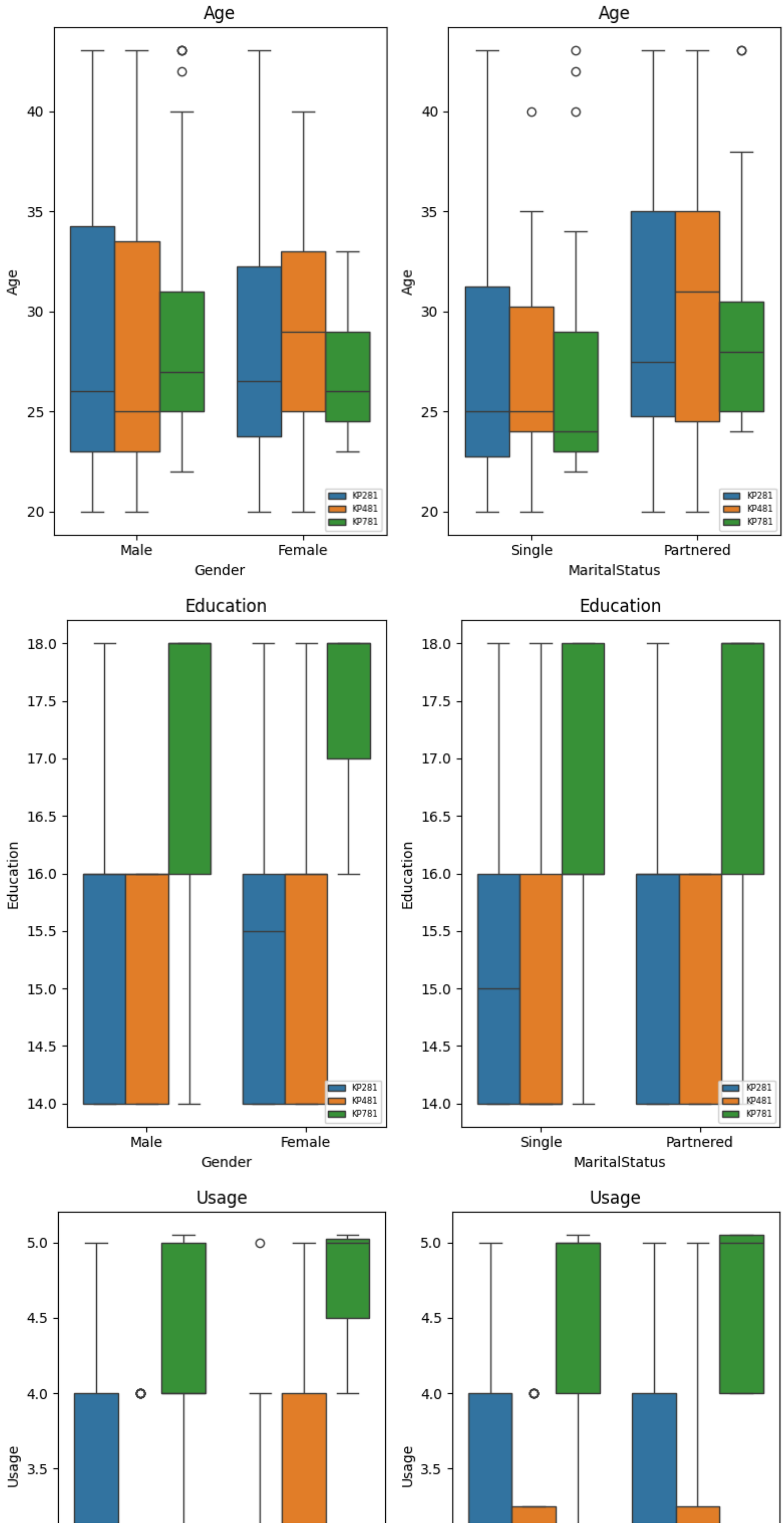
fitness_poor = 1-2 score

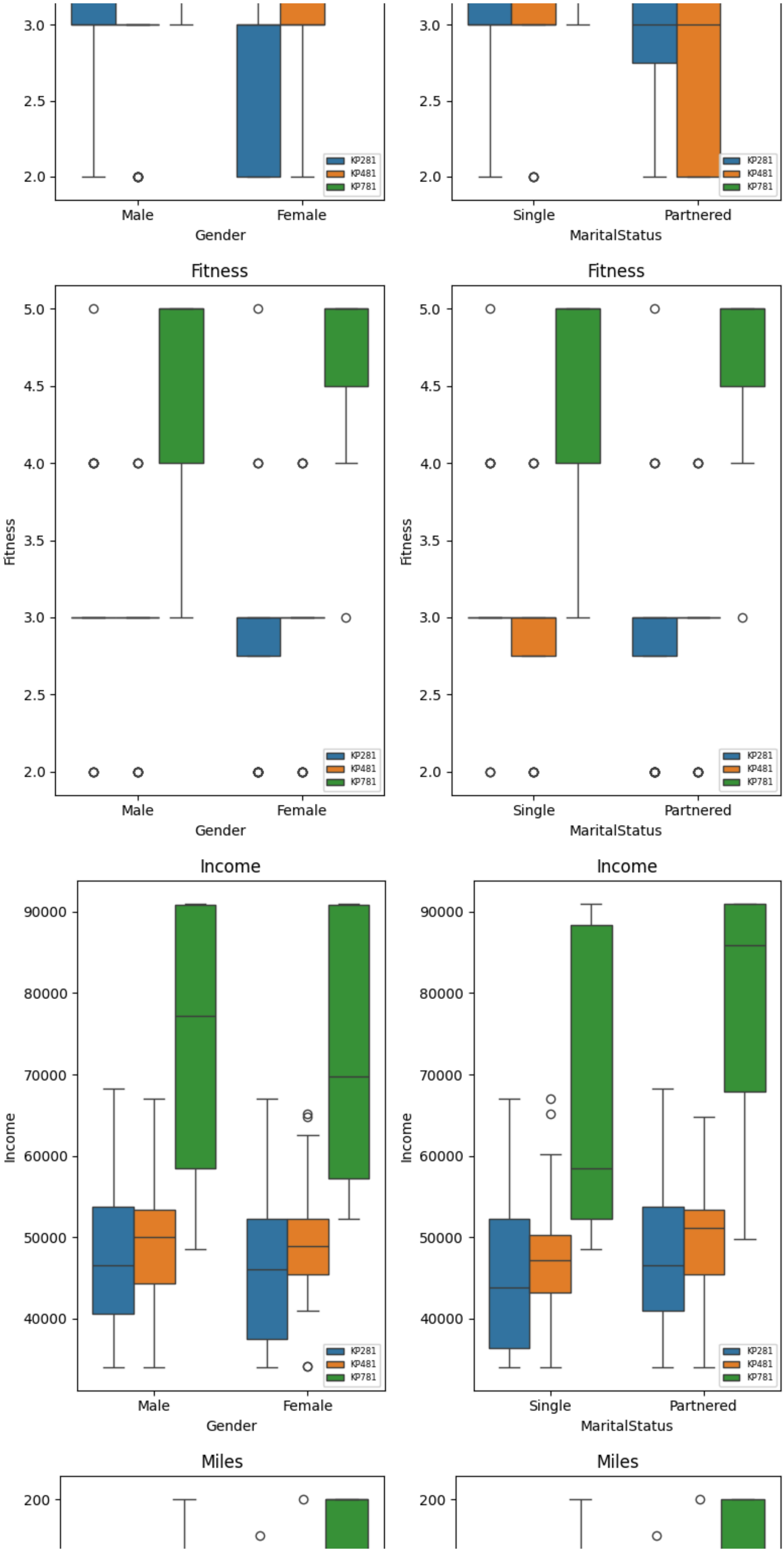
fitness_average = 2-3 fitness_good = 3-5

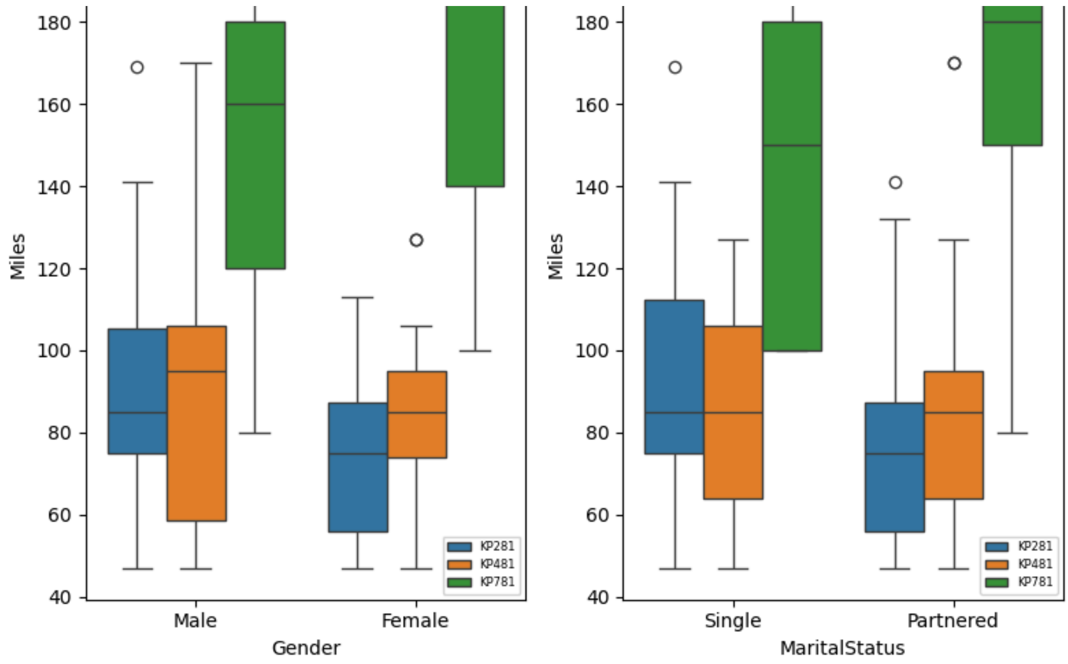
```
plt.figure(figsize=(8,8))
plt.subplot(2,2,1)
plt.xlabel('Age_group')
plt.ylabel('Count')
plt.title('Age_group')
sns.countplot(data=aerofit_data,x='Product',hue='Age_group')

plt.subplot(2,2,2)
plt.xlabel('Fitness_group')
plt.ylabel('Count')
plt.title('Fitness_group')
sns.countplot(data=aerofit_data,x='Product',hue='Fitness_group')

plt.subplot(2,1,2)
plt.xlabel('Education_group')
plt.ylabel('Count')
plt.title('Education_group')
#plt.legend(loc='lower right')
sns.countplot(data=aerofit_data,y='Product',hue='Education_group')


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




```
#Boxplot for all numerical columns with product.

for i in numeric_columns:
  plt.figure(figsize=(8,6))
  plt.subplot(1,2,1)
  plt.ylabel(i)
  plt.xlabel('Gender')
  plt.title(i)
  sns.boxplot(data=aerofit_data,y=i,x='Gender',hue='Product')
  plt.legend(loc='lower right',prop={'size': 6})

  plt.subplot(1,2,2)
  plt.ylabel(i)
  plt.xlabel('MaritalStatus')
  plt.title(i)
  sns.boxplot(data=aerofit_data,y=i,x='MaritalStatus',hue='Product')
  plt.legend(loc='lower right',prop={'size': 6})

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


Age
Age
Gender
Male
Female
MaritalStatus
Single
Partnered
Education
Usage
KP281
KP481
KP781


Fitness
Income
Miles
Gender
MaritalStatus
Male
Female
Single
Partnered
KP281
KP481
KP781



Observations and Insights

- Lower age group(18-26 years) people are high across the three products. Very yong generation shown interest to do excercises and prfer our products. And then medium age(26-34) > above medium age(34-42) > high age(42-50) group people bought our products.
- We could observe that the average fitness(2-3 score) people are very high in the products KP281 and KP481. Mostly good finess(3-5 score) people uses the product KP781.

- Level-2 education group(15-18) people uses all three products similarly. But high education level-3(18-21) pleple uses only KP781 product only. And level-1 people are mostly using KP281.

Product KP281:

- No.of male users are more when compared to female users and there are some high aged femals(outliers) uses this product. Married users are more compared to un-married people.
- Male people have have their education in between 12-18 years. Same way female users for this product have 14-18 years education background. Similarly married people having education background of 12-18 years and un-married people have 13-18 years.
- More no.of male people uses the trendmill than female and maritual status not effected to this product.
- Fitness is not good for male and female users also not good but they are better than male people. same way married and un-marired also.
- Average and low income (30k-70k) people uses this product mostly male and female / single and married are equilly uses.
- Males walk/ran more distance than female, married and un-married walks almost same distance.

Product KP481:

- No.of male users are more when compared to female users this product. Married users are more compared to un-married people.
- Male people have have their education in between 12-16 years. Same way female users for this product have 15-18 years education background. Similarly married people having education background of 12-18 years and un-married people have 13-18 years.
- All female uses the trendmill but very very few male people uses this product.
- Fitness is bad for male and female users and married and un-married people as well.
- Average and low income (30k-70k) people uses this product mostly male and female / single and married are equilly uses.
- Males walk/ran more distance than female, married and un-married walks almost same distance.

Product KP781:

- No.of male users are more when compared to female users this product. Married and un-married users are almost same.
- Male people have have their education in between 14-21 years. Same way female users for this product have 16-18 years education background. Similarly married people having education background of 14-21 years and un-married people have 14-21 years.
- Mostly high education background people bought this product.
- Both male and female people uses almost equally and very well. Married people uses very well compared to un-maried people.
- Fitness is very good for male and female users but female users are bit better than male users. same way married and un-marired also.
- Average and low income (55k-1L +) people uses this product mostly male and female / single and married are equilly uses.
- Females walk/ran little more distance than male, married ran/walk 100 to 200 miles and un-married walks 80 to 260 and some others walked more than this also(outliers).

Recomendations:

- KP718 is good product and gives excellent result. Mostly all well education backround people bought this product and they planed, utilized tredmill equipment, all the days uses mosly, and they workouted consistenly then get the very good results(They are maintaining fit body and make their health wealthy). So promote this product to good education background people and get the good rests and make more profit.

- Target the people how got more income and explain about this product and give some days free trail who is interested.
- Put more focus on young aged people(21-28 years) and who are just married(less than 5-7 years) more interested to use our products.
- Show the regular utilization of educated people and their succesive results to mid level educated people. And mostly concentrate of just married couples, seems to be they are more as per our data.

-- **Final Page -- CASE STUDY DONE --Final Page** --

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.