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.

Dataset

The company collected the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months. The dataset has the following features:

Product Purchased: KP281, KP481, or KP781

Age: In years

Gender: Male/Female

Education: In years

MaritalStatus: Single or partnered

Usage: The average number of times the customer plans to use the treadmill each week .
Income: Annual income (in $)

Fitness: Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 is the excellent shape.

Miles: The average number of miles the customer expects to walk/run each week

Product Portfolio:

The KP281 is an entry-level treadmill that sells for $1,500.

The KP481 is for mid-level runners that sell for $1,750.

The KP781 treadmill is having advanced features that sell for $2,500.

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

```
from scipy.stats import binom, norm, poisson, geom, expon
import warnings

warnings.filterwarnings("ignore", category=FutureWarning)
```

1.Defining Problem Statement and Analysing basic metrics.

Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), statistical summary

```
Aerofit=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/
public_assets/assets/000/001/125/original/aerofit_treadmill.csv")
Aerofit

    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]
```

```
Aerofit.shape
```

```
(180, 9)
```

```
Aerofit.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
```

```
Aerofit.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
```

```
Aerofit.describe(include="all")
```

```
       Product         Age Gender   Education MaritalStatus
Usage  \
count      180  180.000000    180  180.000000           180
180.000000
unique       3         NaN      2         NaN             2
NaN
top      KP281         NaN   Male         NaN     Partnered
NaN
freq        80         NaN    104         NaN           107
NaN
mean       NaN   28.788889    NaN   15.572222           NaN
3.455556
std        NaN    6.943498    NaN    1.617055           NaN
1.084797
min        NaN   18.000000    NaN   12.000000           NaN
2.000000
25%        NaN   24.000000    NaN   14.000000           NaN
3.000000
50%        NaN   26.000000    NaN   16.000000           NaN
3.000000
75%        NaN   33.000000    NaN   16.000000           NaN
4.000000
max        NaN   50.000000    NaN   21.000000           NaN
7.000000

           Fitness         Income       Miles
count   180.000000     180.000000  180.000000
unique         NaN            NaN         NaN
top            NaN            NaN         NaN
freq           NaN            NaN         NaN
mean      3.311111   53719.577778  103.194444
std       0.958869   16506.684226   51.863605
min       1.000000   29562.000000   21.000000
25%       3.000000   44058.750000   66.000000
50%       3.000000   50596.500000   94.000000
75%       4.000000   58668.000000  114.750000
max       5.000000  104581.000000  360.000000
```

2. Non-Graphical Analysis: Value counts and unique attributes

```
print("Unique Products -", Aerofit["Product"].nunique())
Aerofit["Product"].value_counts().to_frame()
```

```
Unique Products - 3
```

```
         count
Product
KP281       80
KP481       60
KP781       40
```

```python
print("Unique Marital Status - ",Aerofit["MaritalStatus"].nunique())
Aerofit["MaritalStatus"].value_counts().to_frame()
```

```
Unique Marital Status -  2
```

```
               count
MaritalStatus
Partnered        107
Single            73
```

```python
Aerofit["Gender"].value_counts().to_frame()
```

```
        count
Gender
Male      104
Female     76
```

```python
print("Unique Age - ",Aerofit["Age"].nunique())
Aerofit["Age"].value_counts().to_frame()
```

```
Unique Age -  32
```

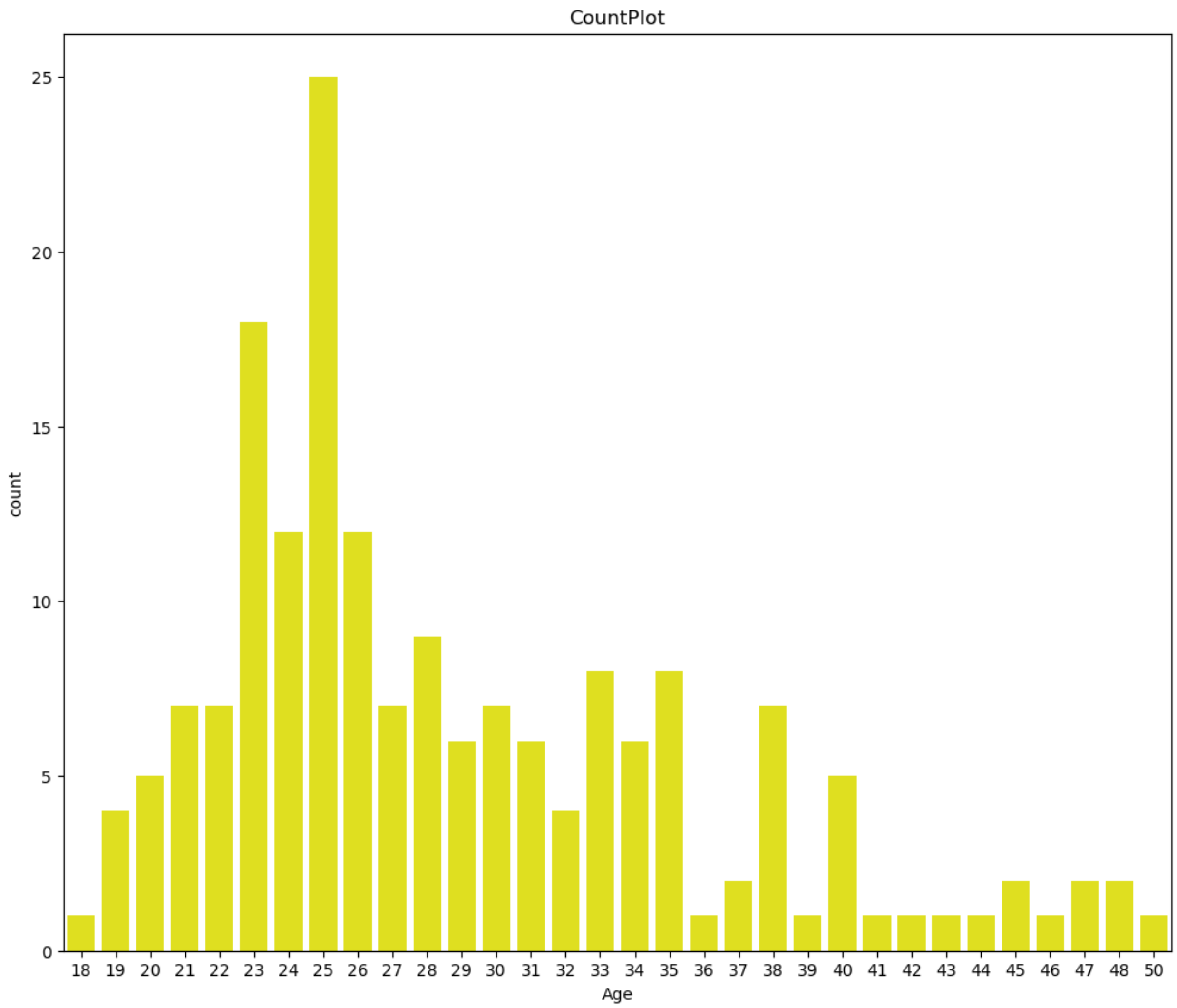
```
     count
Age
25      25
23      18
24      12
26      12
28       9
35       8
33       8
30       7
38       7
21       7
22       7
27       7
31       6
34       6
29       6
20       5
40       5
32       4
19       4
```

```
48    2
37    2
45    2
47    2
46    1
50    1
18    1
44    1
43    1
41    1
39    1
36    1
42    1
```

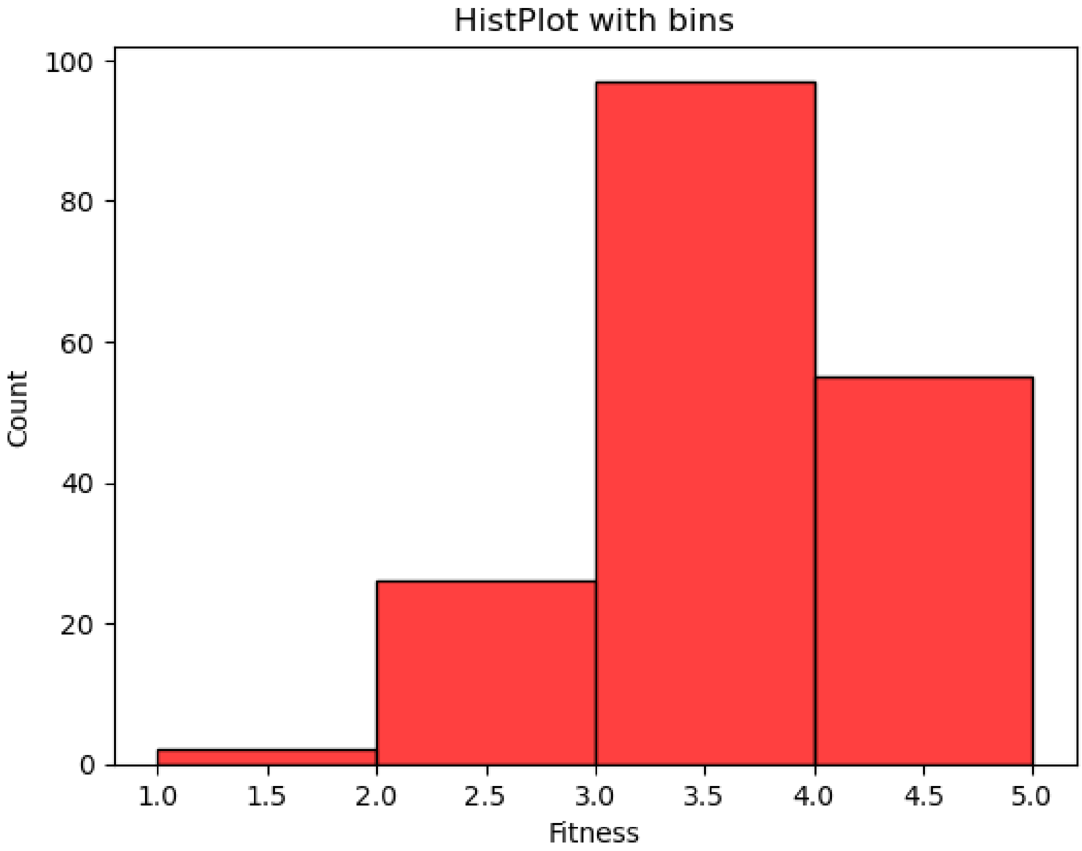
3. Visual Analysis - Univariate & Bivariate

3.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis

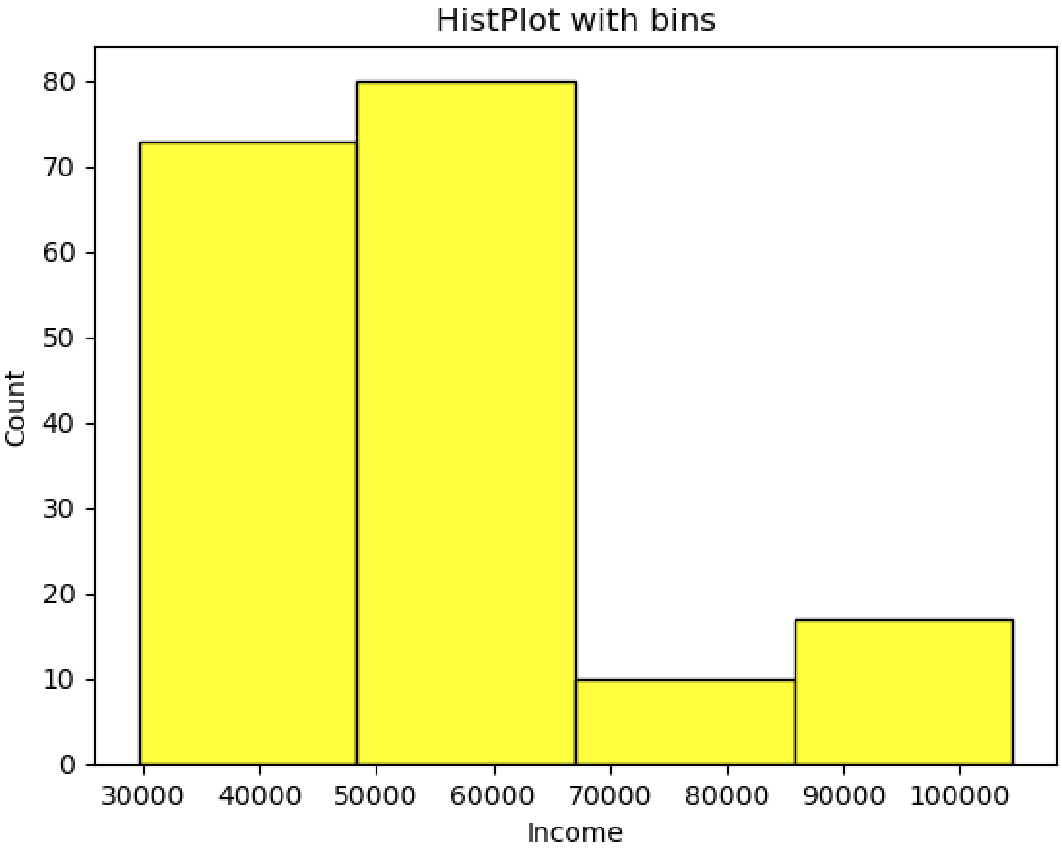
```
#Age
plt.figure(figsize=(12,10))
sns.countplot(x=Aerofit["Age"],color="yellow")
plt.title("CountPlot")
plt.show()
```

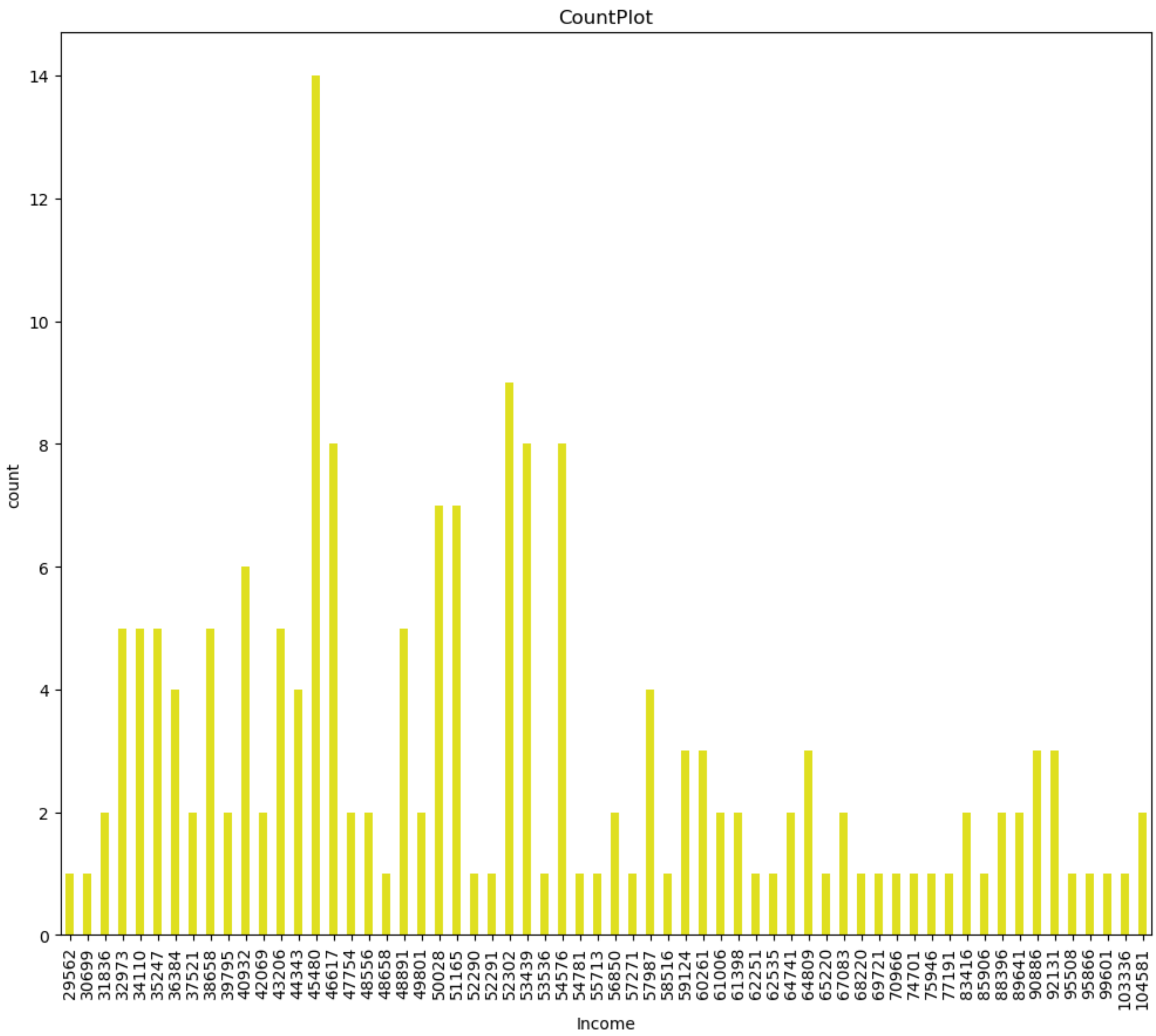


```
# Fitness

sns.histplot(x=Aerofit["Fitness"],color="red",bins=4)
plt.title("HistPlot with bins")
plt.show()
```

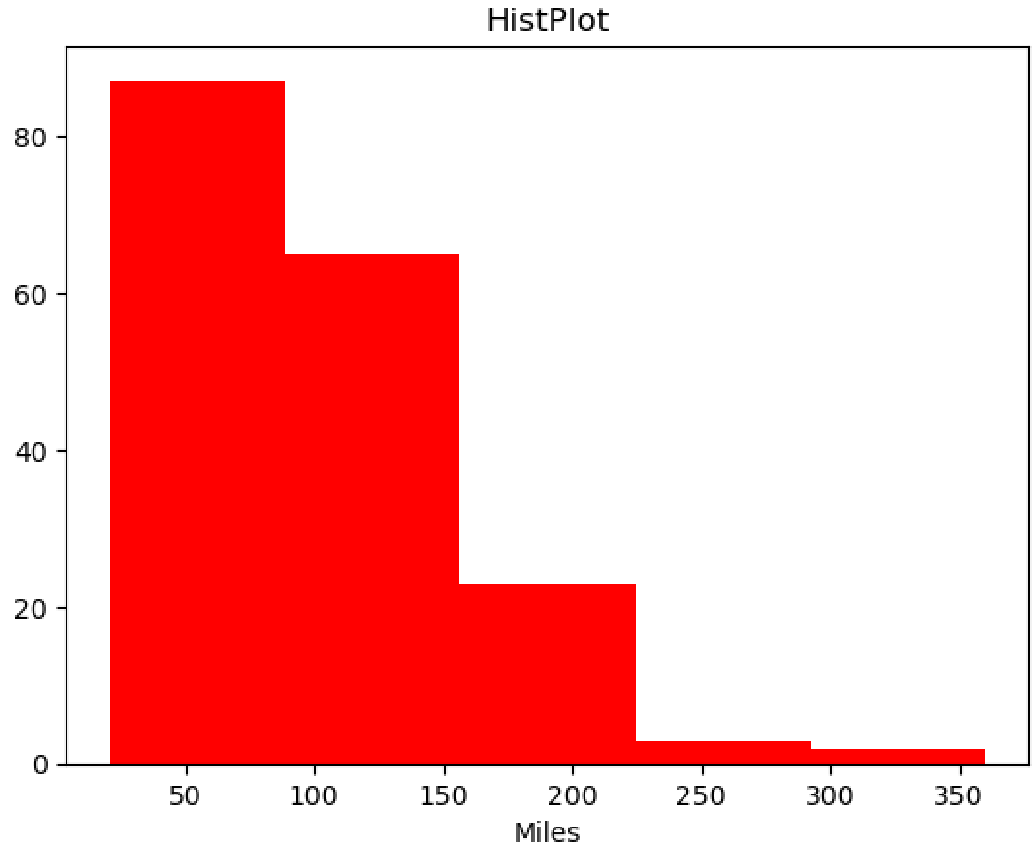


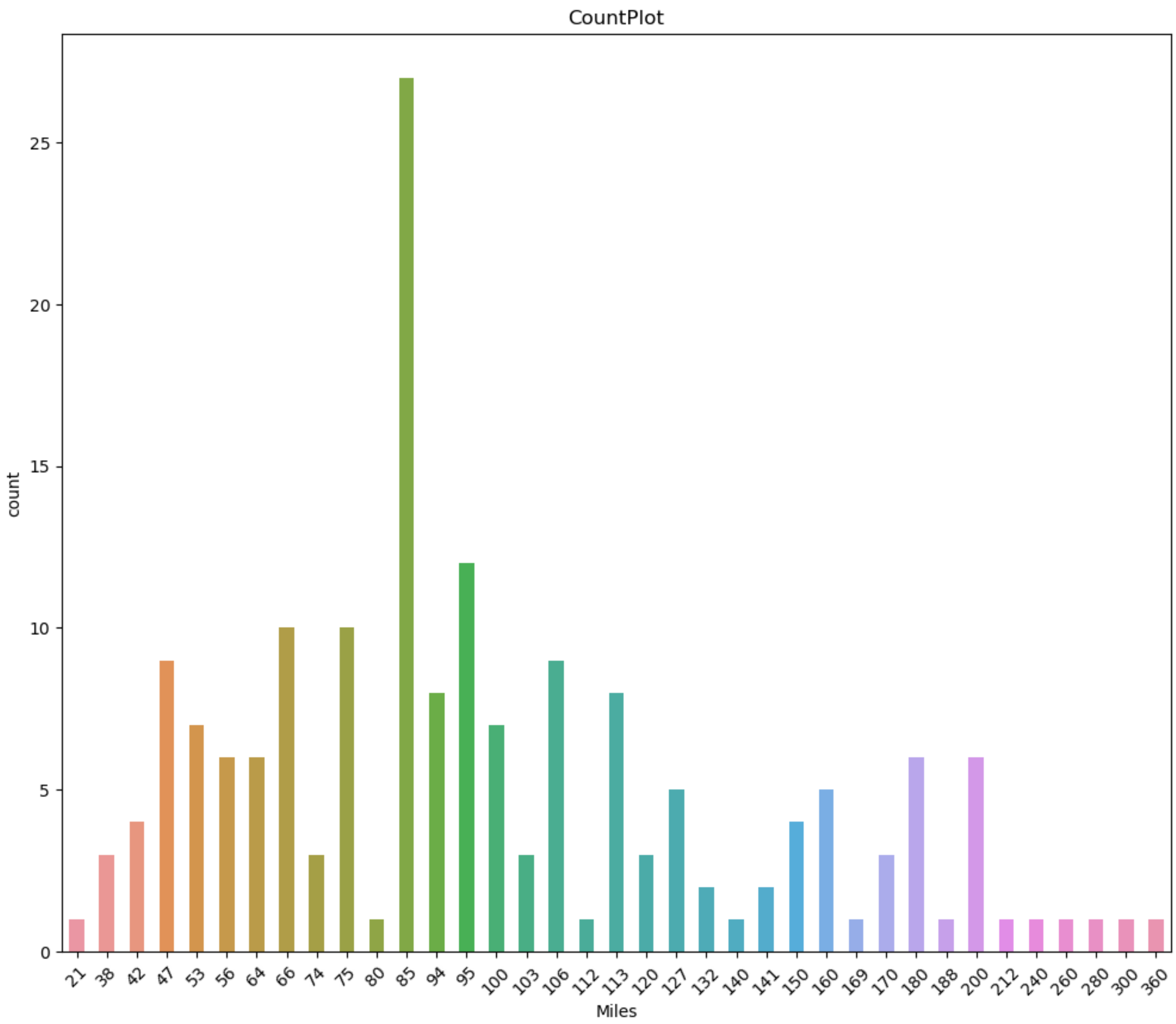
```python
# Income
sns.histplot(x=Aerofit["Income"],color="yellow",bins=4)
plt.title("HistPlot with bins")
plt.show()

plt.figure(figsize=(12,10))
sns.countplot(x=Aerofit["Income"],color = "yellow",width=0.5)
plt.title("CountPlot")
plt.xticks(rotation=90)
plt.show()
```


HistPlot with bins
Count
0
10
20
30
40
50
60
70
80
30000
40000
50000
60000
70000
80000
90000
100000
Income



```
# Miles
# Fitness

plt.hist(x=Aerofit["Miles"],color = "red" ,bins=5)
plt.title("HistPlot")
plt.xlabel("Miles")
plt.show()

plt.figure(figsize=(12,10))
sns.countplot(x=Aerofit["Miles"],width=0.5)
plt.title("CountPlot")
plt.xticks(rotation=45)
plt.show()
```


HistPlot
0
20
40
60
80
50
100
150
200
250
300
350
Miles

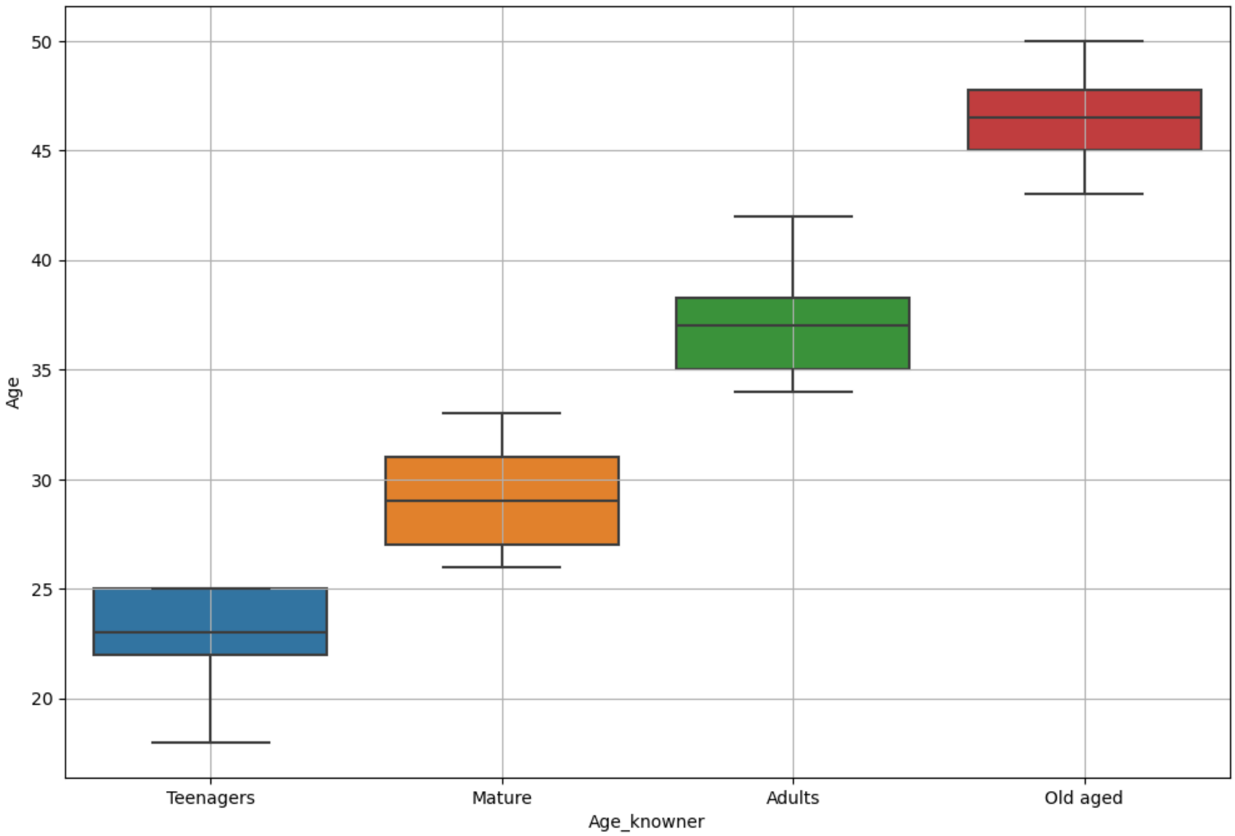


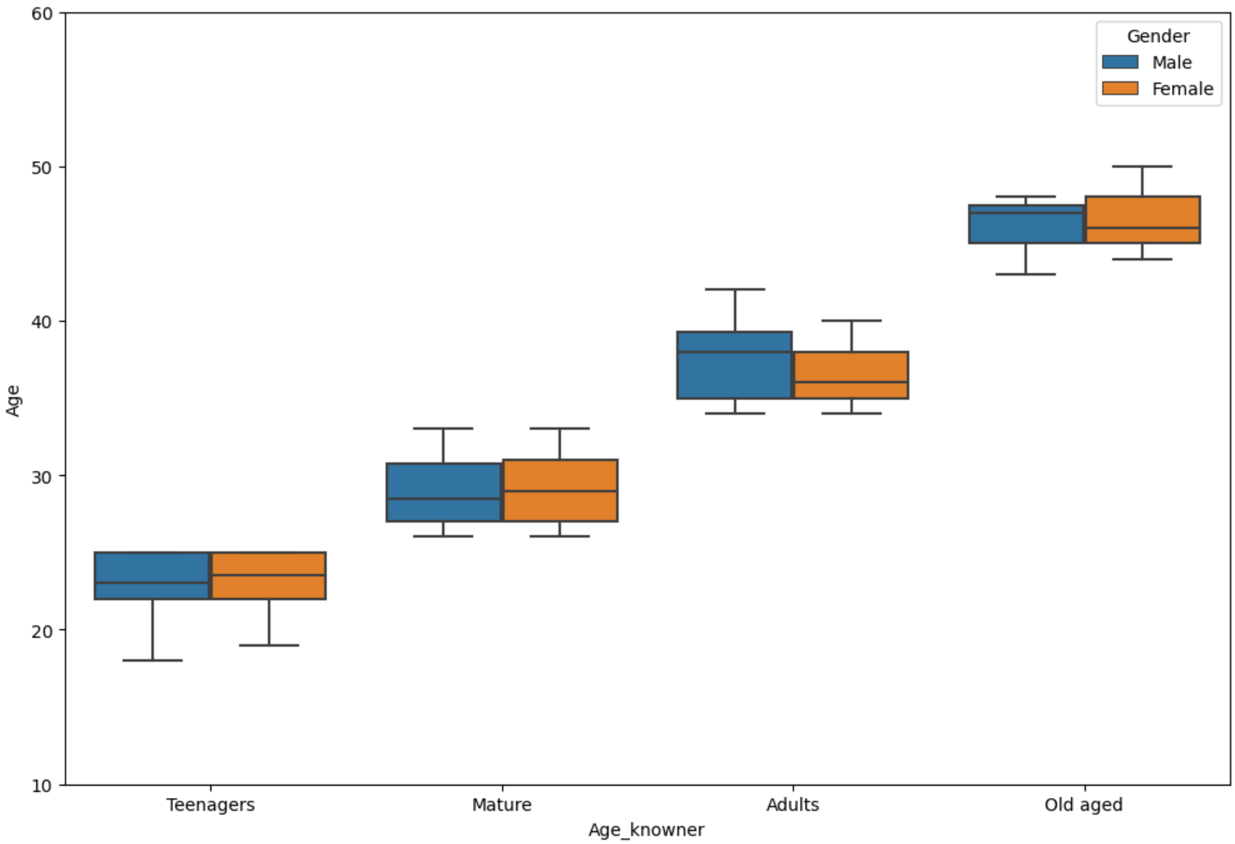
3.2 For categorical variable(s): Boxplot

```
# Modifying Data On the basis of AGE
values=[17,25,33,42,55]
Groups=['Teenagers','Mature','Adults','Old aged']
Aerofit["Age_knowner"]=pd.cut(Aerofit["Age"],labels=Groups,bins=values
)
Aerofit

    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
```

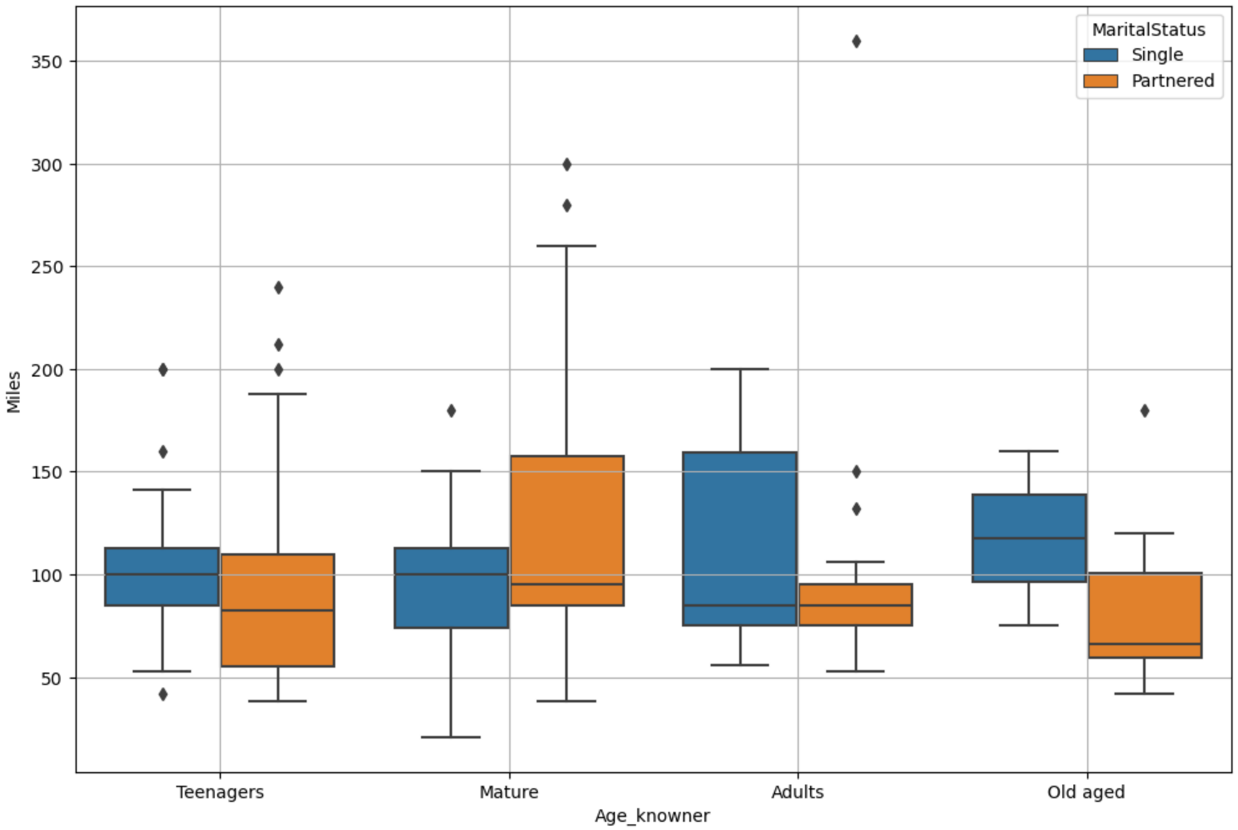
```
32973
4     KP281   20    Male         13     Partnered      4        2
35247
..      ...  ...     ...        ...           ...    ...      ...
...
175   KP781   40    Male         21        Single      6        5
83416
176   KP781   42    Male         18        Single      5        4
89641
177   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 Age_knowner
0      112   Teenagers
1       75   Teenagers
2       66   Teenagers
3       85   Teenagers
4       47   Teenagers
..     ...         ...
175    200      Adults
176    200      Adults
177    160    Old aged
178    120    Old aged
179    180    Old aged

[180 rows x 10 columns]
```

```
# BoxPlot on Age_Knowner

#Bivariate Analysis
plt.figure(figsize=(12,8))
sns.boxplot(data=Aerofit,x="Age_knowner",y="Age")
plt.grid()
plt.show()

# Tri-Variate
plt.figure(figsize=(12,8))
sns.boxplot(data=Aerofit,x="Age_knowner",y="Age",hue="Gender")
plt.ylim(bottom=10,top=60)
plt.show()
```

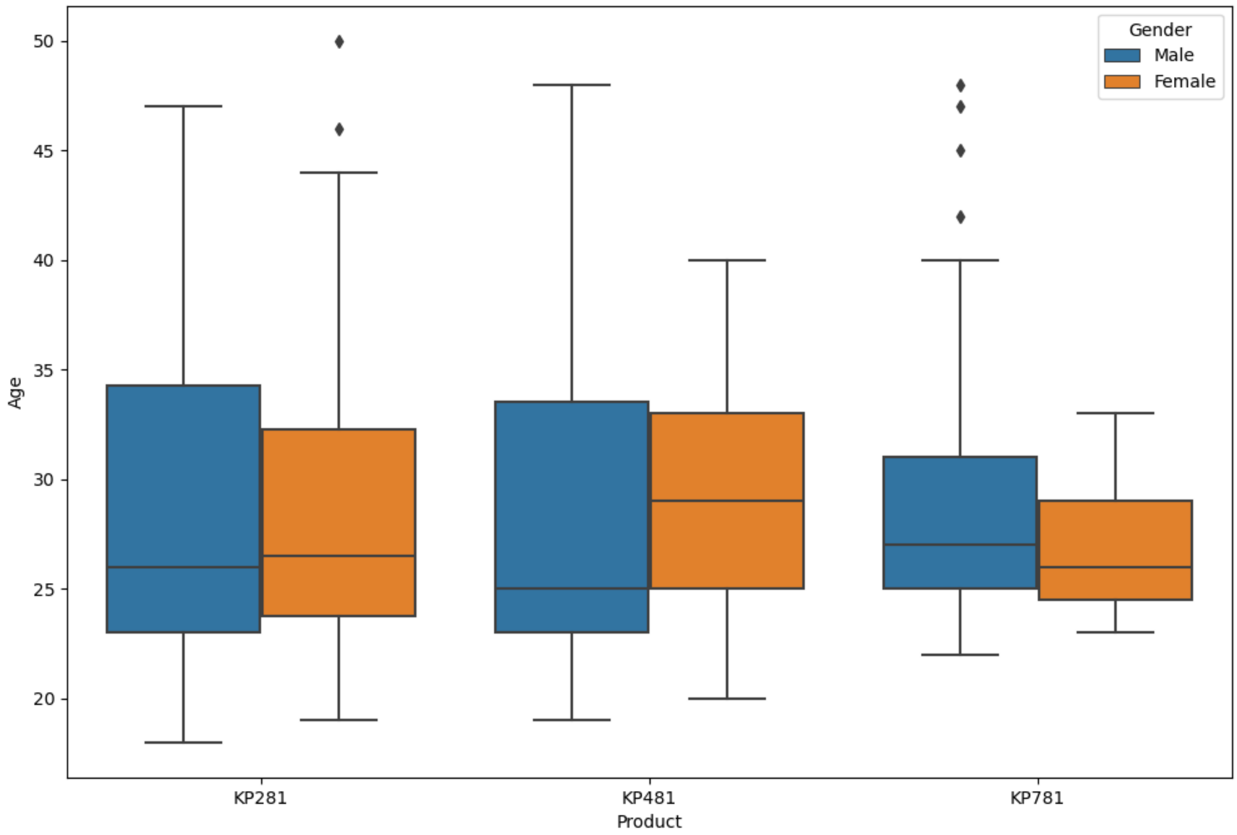

50
45
40
35
30
25
20
Age
Teenagers
Mature
Adults
Old aged
Age_knowner



```
# Marital Status

plt.figure(figsize=(12,8))
sns.boxplot(data=Aerofit,x="Age_knowner",y="Miles",hue="MaritalStatus"
)
plt.grid()
plt.show()
```

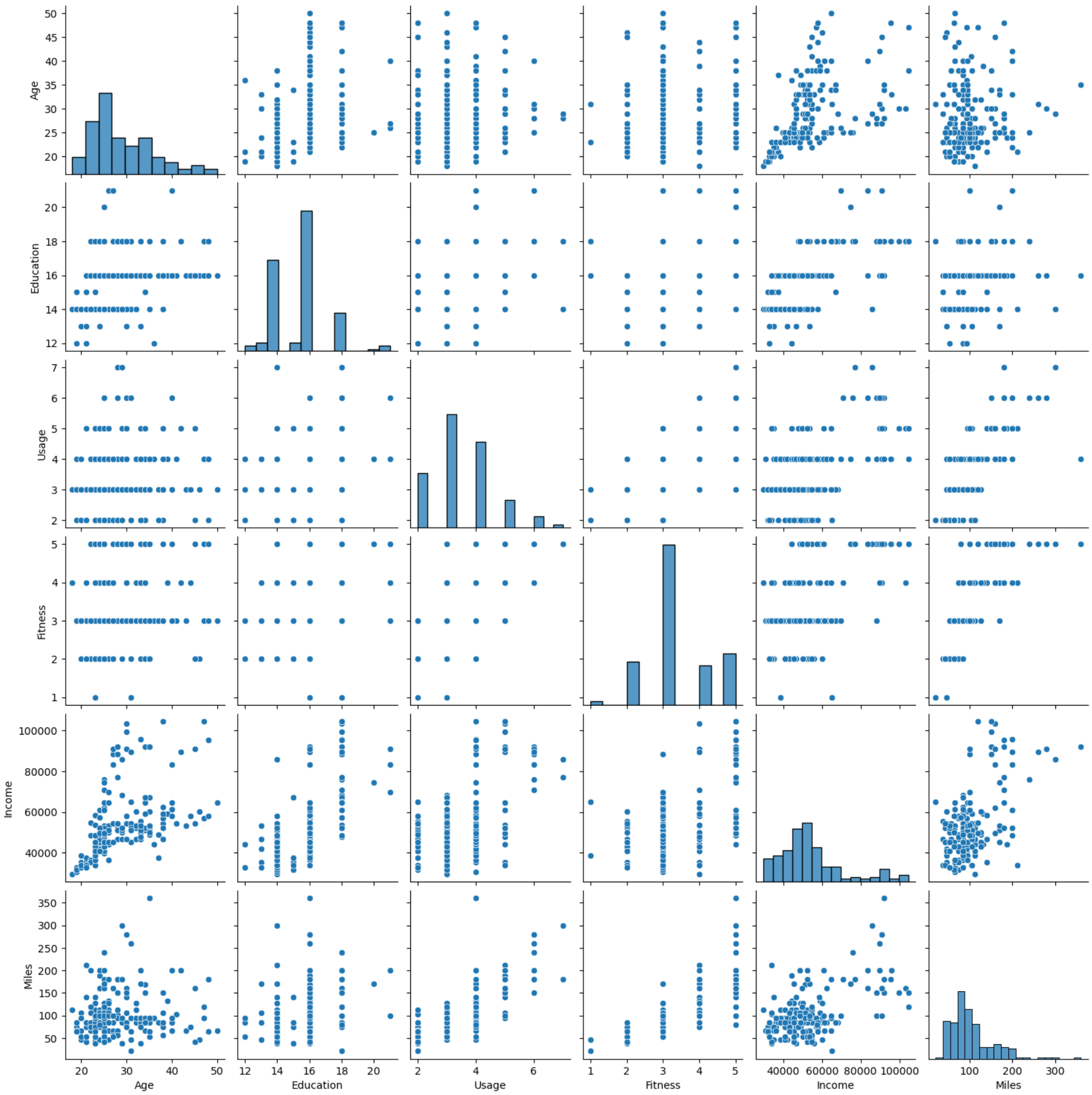


```
# Product

plt.figure(figsize=(12,8))
sns.boxplot(data=Aerofit,x="Product",y="Age",hue="Gender")
plt.show()
```



3.3 For correlation: Heatmaps, Pairplots

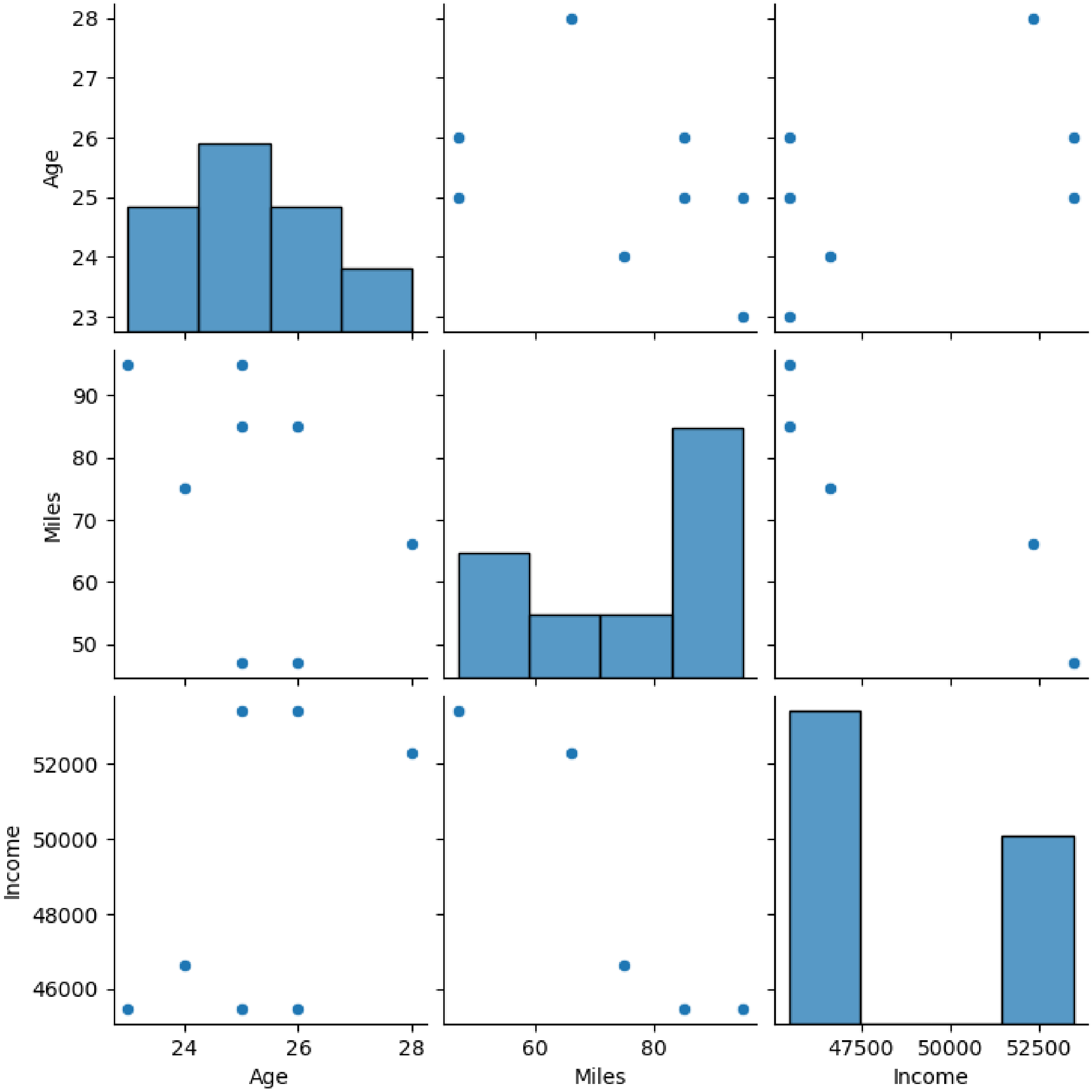
```
sns.pairplot(Aerofit)
plt.show()
```



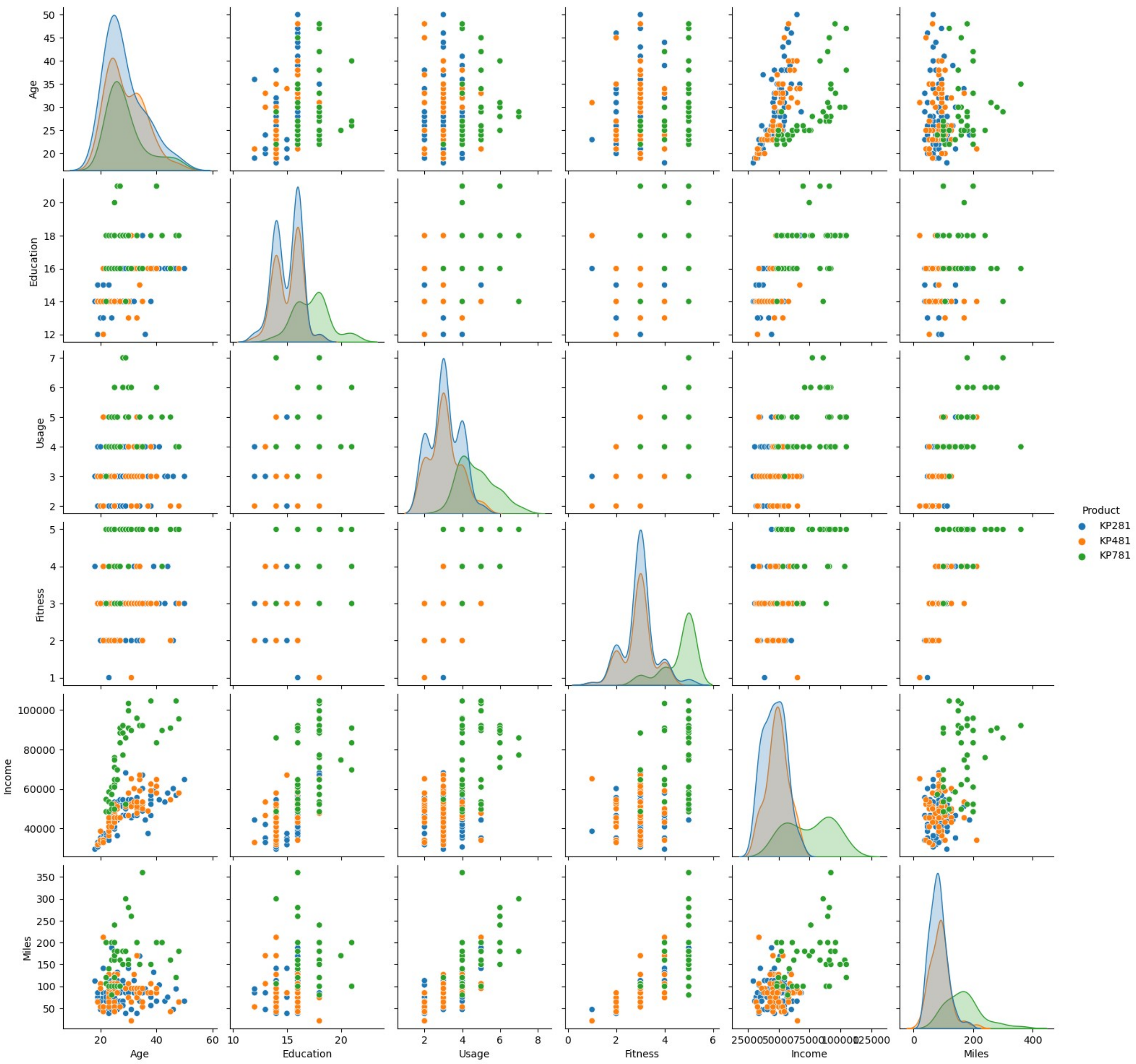
```
# Top 5 Data

Top_5_Age=Aerofit["Age"].value_counts().index[:5]
Top_5_Income=Aerofit["Income"].value_counts().index[:5]
Top_5_Miles=Aerofit["Miles"].value_counts().index[:5]

Top_5_data_based_on_Age_Income_Miles=Aerofit[(Aerofit["Age"].isin(Top_
5_Age)) & (Aerofit["Income"].isin(Top_5_Income)) &
(Aerofit["Miles"].isin(Top_5_Miles))]
Top_5_data_based_on_Age_Income_Miles
```

```
    Product  Age  Gender  Education MaritalStatus  Usage  Fitness
Income  \
26    KP281   24  Female         16        Single      4        3
46617
29    KP281   25  Female         14     Partnered      2        2
53439
36    KP281   26    Male         16     Partnered      2        2
53439
45    KP281   28  Female         16     Partnered      2        3
52302
89    KP481   23  Female         16        Single      3        3
45480
97    KP481   25  Female         14     Partnered      2        3
45480
101   KP481   25    Male         14        Single      3        3
45480
108   KP481   26  Female         16     Partnered      4        3
45480

     Miles Age_knowner
26      75   Teenagers
29      47   Teenagers
36      47      Mature
45      66      Mature
89      95   Teenagers
97      85   Teenagers
101     95   Teenagers
108     85      Mature
```

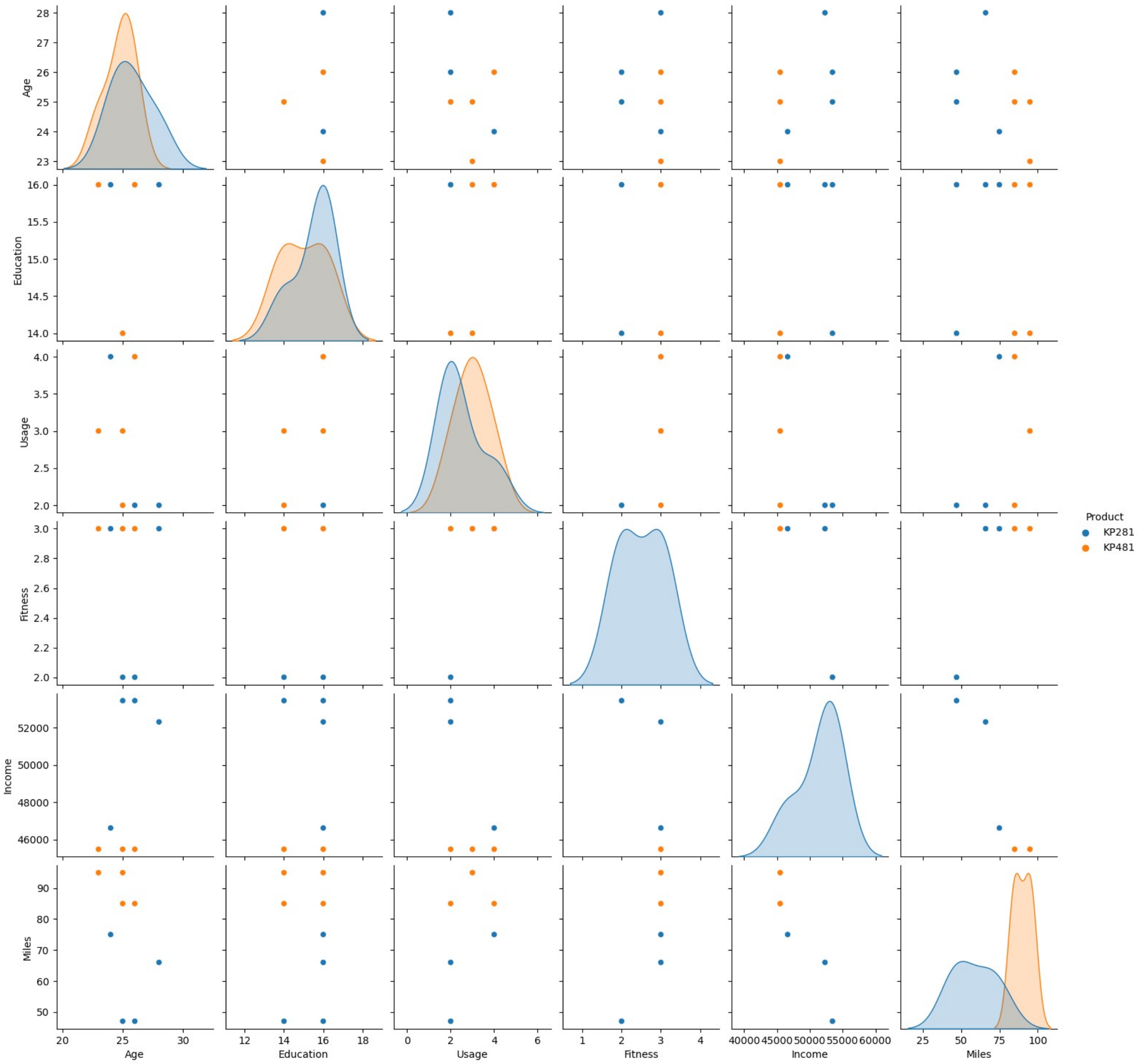
```
sns.pairplot(data=Top_5_data_based_on_Age_Income_Miles[["Age","Miles",
"Income","Product"]])
plt.show()
```



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



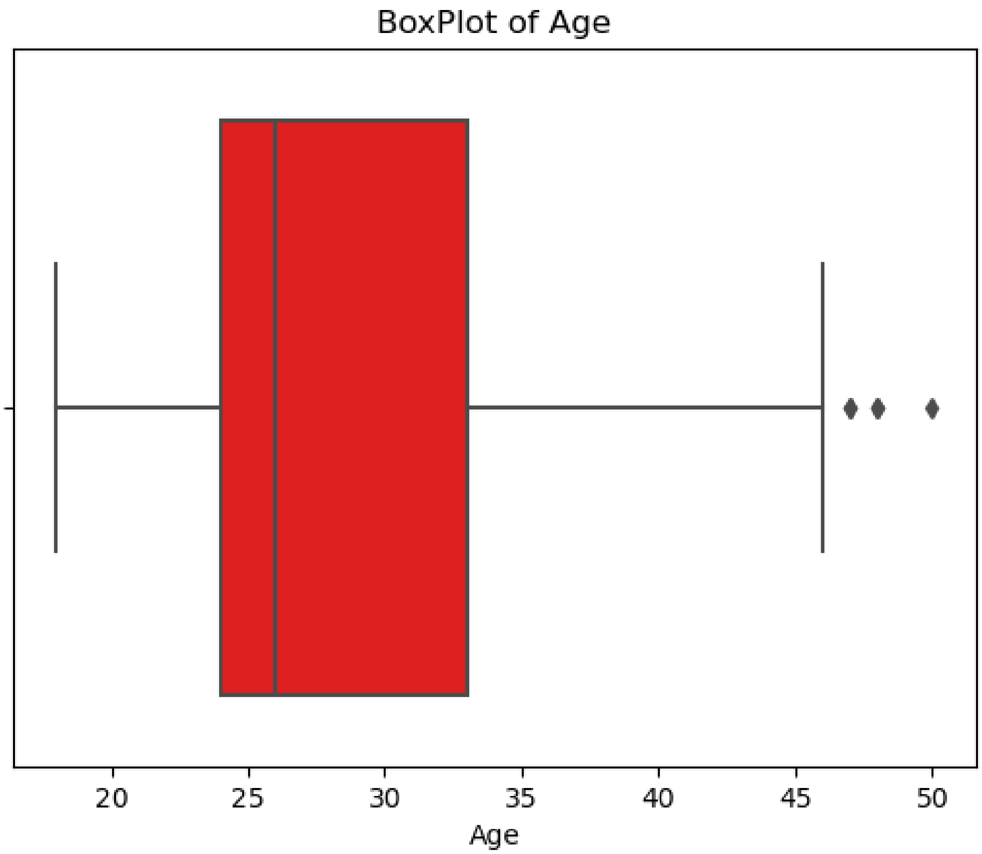
```
sns.pairplot(data=Top_5_data_based_on_Age_Income_Miles,hue="Product")
plt.show()
```



4. Missing Value & Outlier Detection

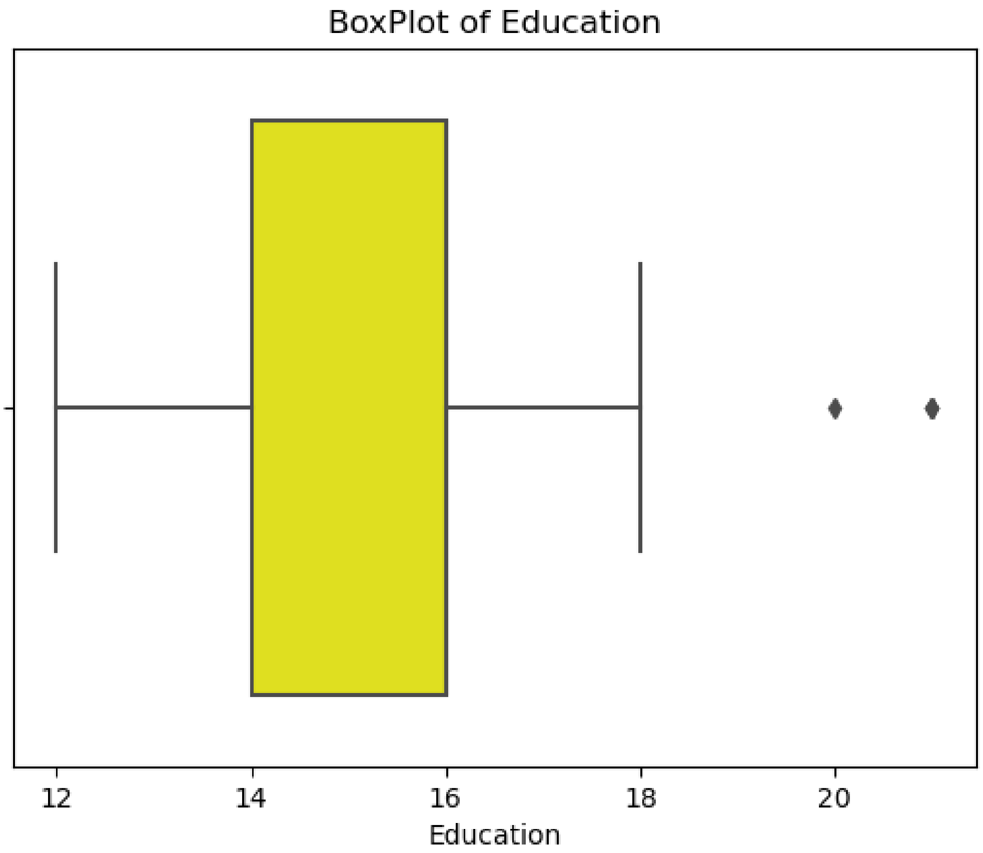
```
# Age - BoxPlot
Age_25=np.percentile(Aerofit["Age"],25)
Age_75=np.percentile(Aerofit["Age"],75)
IQR=Age_75-Age_25
Lower=max(Age_25-(1.5*IQR),0)
Upper=Age_75+(1.5*IQR)

print("People of Age less than ",Lower," and people of age greater
than ",Upper,"are outlier")

sns.boxplot(x=Aerofit["Age"],color="red")
plt.title("BoxPlot of Age")
plt.show()
```

```
People of Age less than  10.5  and people of age greater than  46.5
are outlier
```



```
# Education

Education_25=np.percentile(Aerofit["Education"],25)
Education_75=np.percentile(Aerofit["Education"],75)
IQR=Education_75-Education_25
Lower=max(Education_25-(1.5*IQR),0)
Upper=Education_75+(1.5*IQR)

print("People of Education less than ",Lower," and people of Education
greater than ",Upper,"are outlier")

sns.boxplot(x=Aerofit["Education"],color="yellow")
plt.title("BoxPlot of Education")
plt.show()
```

```
People of Education less than  11.0  and people of Education greater
than  19.0 are outlier
```



```
# to delete rows having missing values
Aerofit.dropna()
```

```
    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 Age_knowner
0      112   Teenagers
1       75   Teenagers
2       66   Teenagers
3       85   Teenagers
4       47   Teenagers
..     ...         ...
175    200      Adults
176    200      Adults
177    160    Old aged
178    120    Old aged
179    180    Old aged

[180 rows x 10 columns]
```

5. Business Insights based on Non-Graphical and Visual Analysis

5.1 Comments on the range of attributes

```
Aerofit.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
```

```python
# Age
a1=Aerofit["Age"].unique()
a2=Aerofit["Age"].nunique()
print("Unique Value of Age : ", a1)
print("Total Unique Age : ", a2)

print("From Aerofit.describe()")
print("minimum age : 18")
print("maximum age : 50")
```

```
Unique Value of Age :  [18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40 41
 43 44 46 47 50 45 48 42]
Total Unique Age :  32
From Aerofit.describe()
minimum age : 18
maximum age : 50
```

```python
# Education
a1=Aerofit["Education"].unique()
a2=Aerofit["Education"].nunique()
print("Unique Value of Education : ", a1)
print("Total Unique Education : ", a2)

print("From Aerofit.describe()")
print("minimum Education : 12")
print("maximum Education : 21")
```

```
Unique Value of Education :  [14 15 12 13 16 18 20 21]
Total Unique Education :  8
From Aerofit.describe()
minimum Education : 12
maximum Education : 21
```

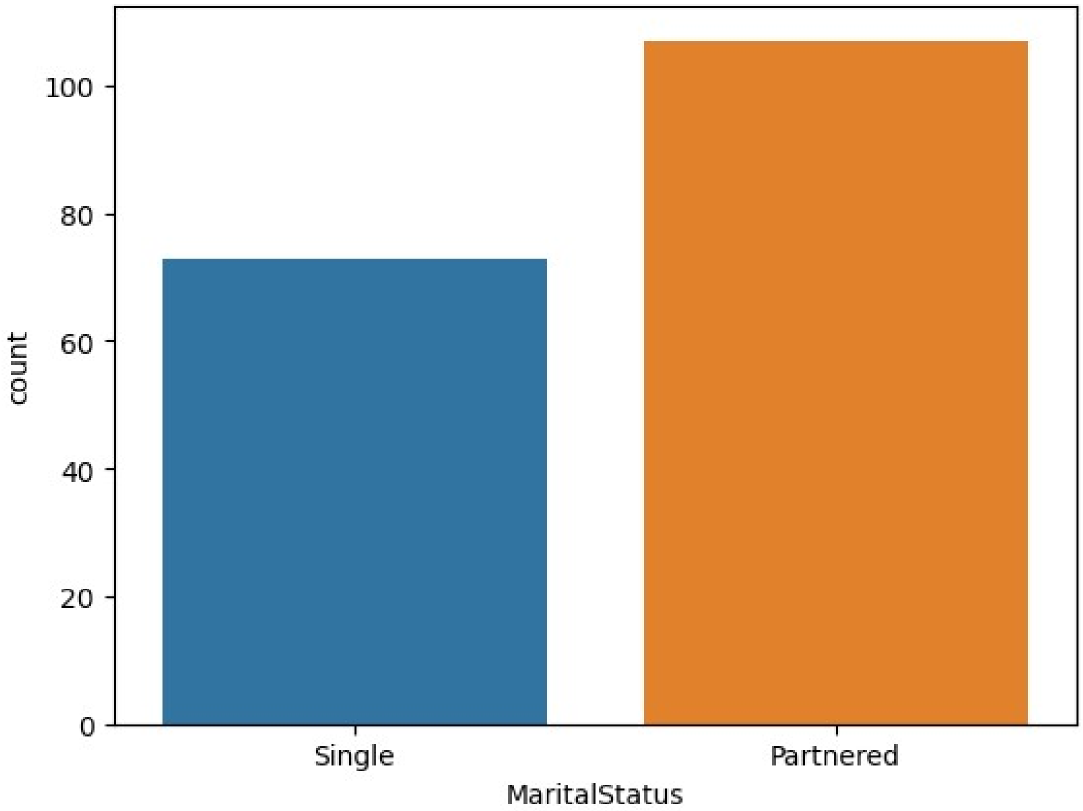
```python
# Income
a1=Aerofit["Income"].unique()
a2=Aerofit["Income"].nunique()
print("Unique Value of Income : ", a1)
print("Total Unique Income : ", a2)
print("From Aerofit.describe()")
print("minimum Income : 29562")
print("maximum Income : 104581")
```

```
Unique Value of Income :  [ 29562  31836  30699  32973  35247  37521
36384  38658  40932  34110
  39795  42069  44343  45480  46617  48891  53439  43206  52302  51165
  50028  54576  68220  55713  60261  67083  56850  59124  61398  57987
  64809  47754  65220  62535  48658  54781  48556  58516  53536  61006
  57271  52291  49801  62251  64741  70966  75946  74701  69721  83416
  88396  90886  92131  77191  52290  85906 103336  99601  89641  95866
 104581  95508]
Total Unique Income :  62
From Aerofit.describe()
minimum Income : 29562
maximum Income : 104581
```

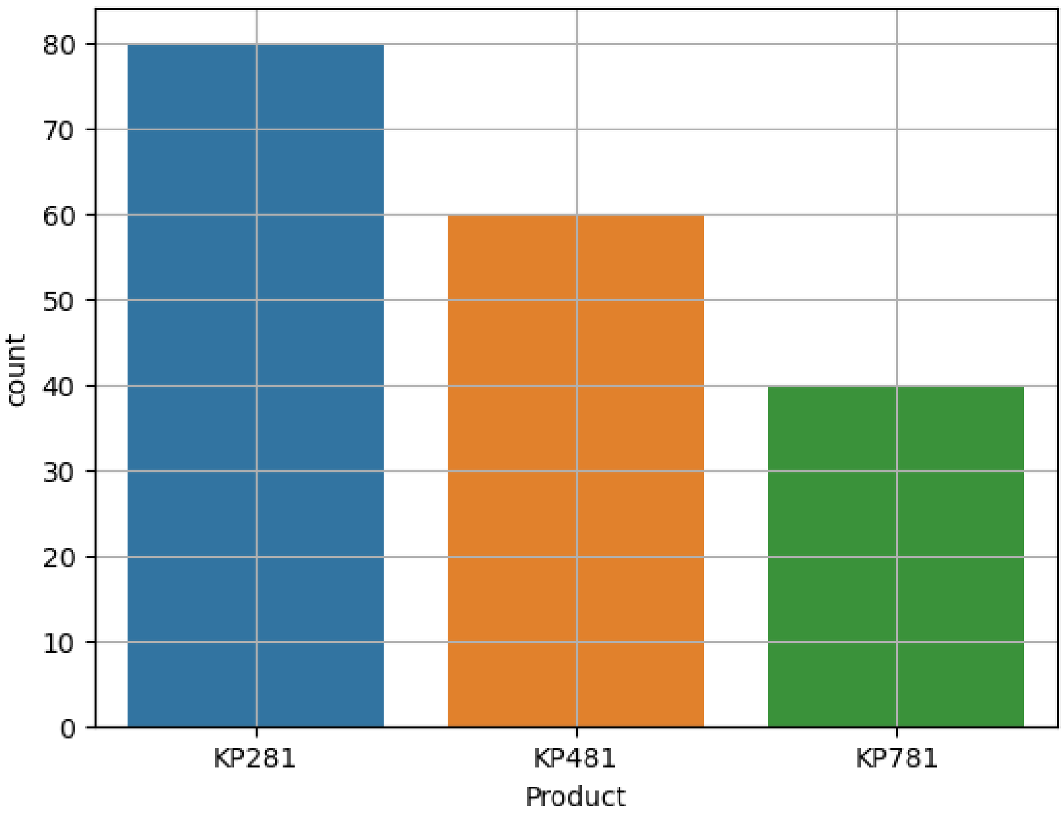
5.2 Comments on the distribution of the variables and relationship between them

5.3 Comments for each univariate and bivariate plot

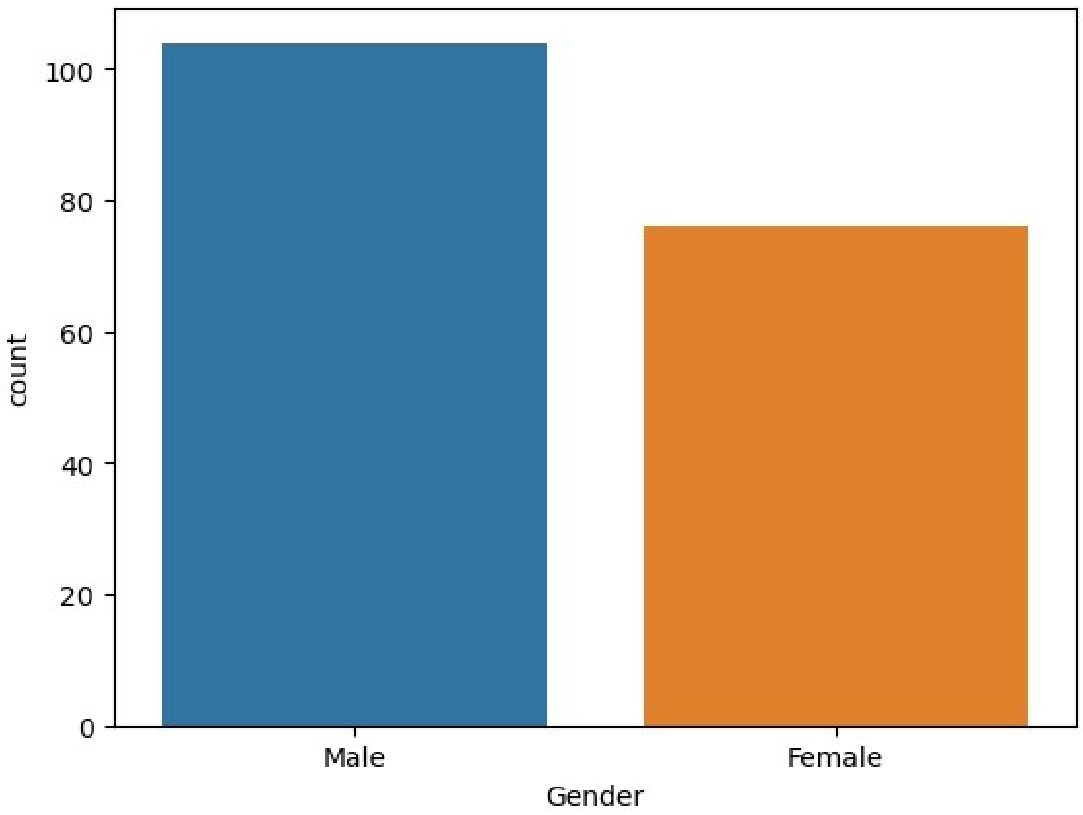
```
# Martial Status
x1=Aerofit["MaritalStatus"].value_counts()
print(x1)
sns.countplot(x=Aerofit["MaritalStatus"])
plt.show()
print("We Observed that Partnered status People mostly purchased a
treadmill from the AeroFit stores during the prior three months")

# Product
x2=Aerofit["Product"].value_counts()
print(x2)
sns.countplot(x=Aerofit["Product"])
plt.grid()
plt.show()
print("Product KP281 is highly purchase treadmil by peoples")

# Gender
x3=Aerofit["Gender"].value_counts()
print(x3)
sns.countplot(x=Aerofit["Gender"])
plt.show()
print("Males Gender is more consious to their healths")

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



```
We Observed that Partnered status People mostly purchased a treadmill
from the AeroFit stores during the prior three months
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
```

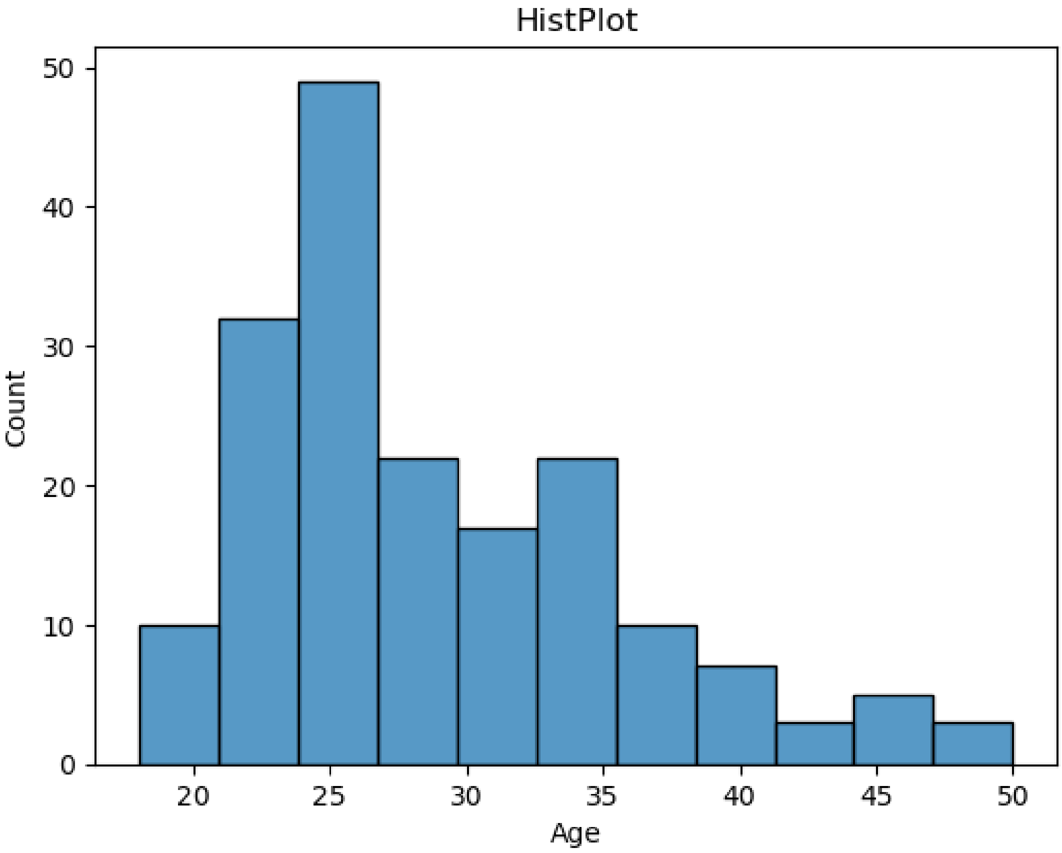


```
Product KP281 is highly purchase treadmil by peoples
Gender
Male      104
Female     76
Name: count, dtype: int64
```

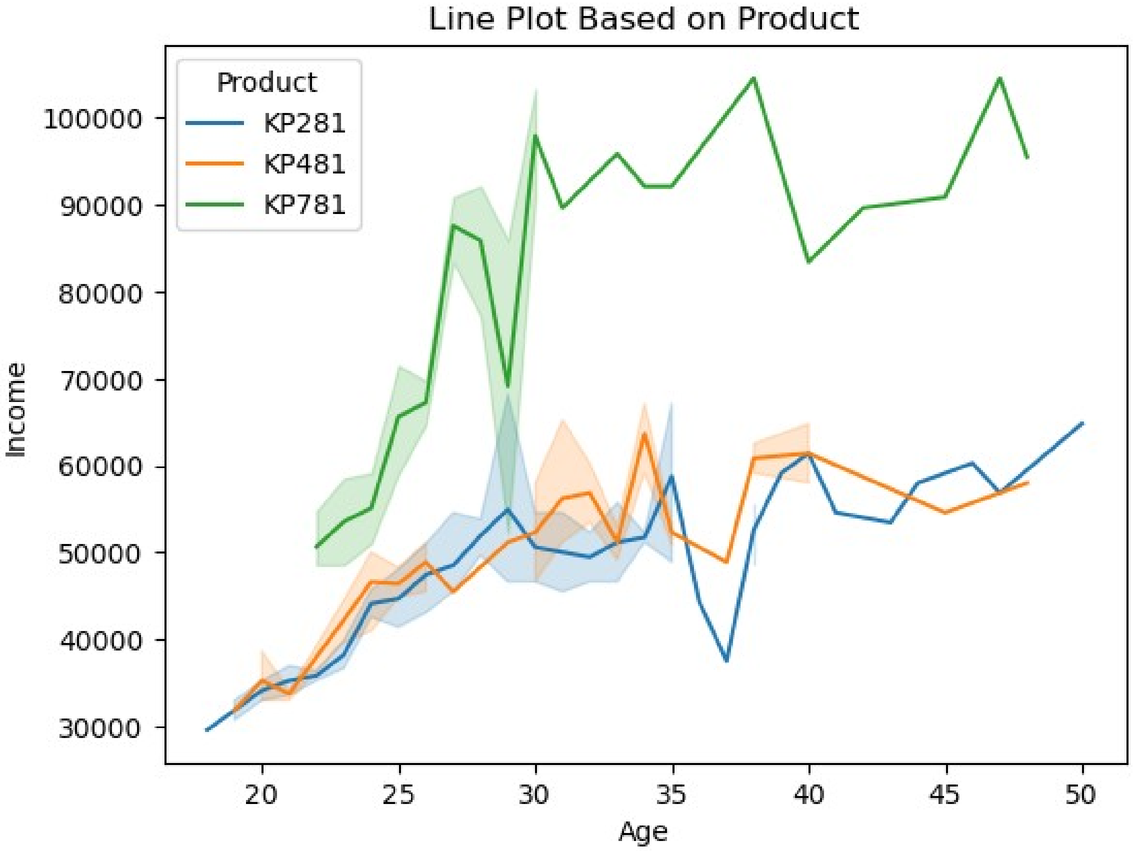


Males Gender is more consious to their healths

```
# GRAPHICAL ANALYSIS

sns.histplot(x=Aerofit["Age"])
plt.title("HistPlot")
plt.show()
print("We notice that age group of 25 people are more use of aerofit
product")
```

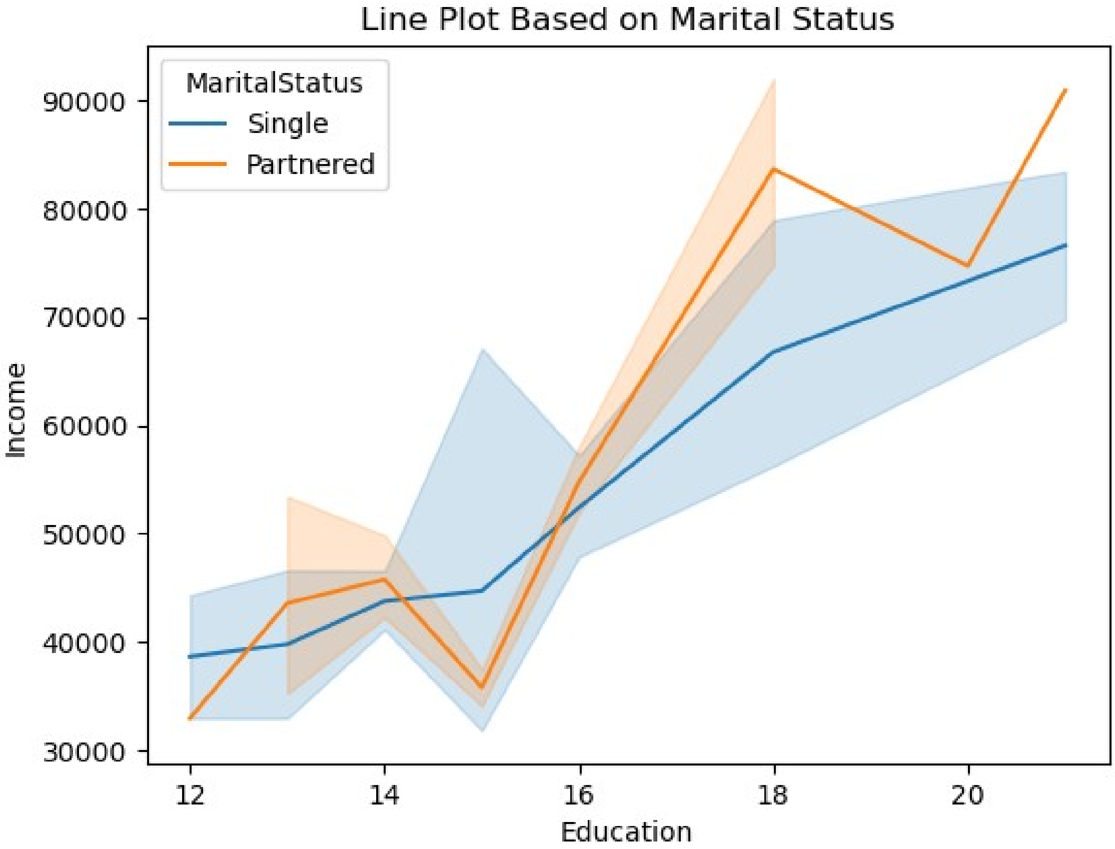


```
We notice that age group of 25 people are more use of aerofit product

# Product

# GRAPHICAL ANALYSIS
sns.lineplot(data=Aerofit,x="Age",y="Income",hue="Product")
plt.title("Line Plot Based on Product")
plt.show()
print("KP781 product is mostly used by all age groups and it is also
noticable that High income people use this mostly this product and it
is also notiocable that KP281 is highly Purchase but people with low
income purchase this mostly")

sns.lineplot(data=Aerofit,x="Education",y="Income",hue="MaritalStatus"
)
plt.title("Line Plot Based on Marital Status")
plt.show()
print("Partnered people and single people both are educated but
incomes of partnered peoples are more")

sns.lineplot(data=Aerofit,x="Age",y="Fitness",hue="Gender",estimator=n
p.max)
plt.title("Line Plot Based on Gender")
plt.show()
print("Male gender is constantly more consious to their health whereas
females are too but they are not constant, they  are fluctating over
age ")



# Non-Graphical Analysis
total_len=Aerofit.shape[0]
len_KP281=Aerofit[Aerofit["Product"]=='KP281'].shape[0]
len_KP481=Aerofit[Aerofit["Product"]=='KP481'].shape[0]
len_KP781=Aerofit[Aerofit["Product"]=='KP781'].shape[0]

Prob_KP281=len_KP281/total_len
Prob_KP481=len_KP481/total_len
Prob_KP781=len_KP781/total_len

print("Probability of use KP281 - ",Prob_KP281)
print("Probability of use KP481 - ",Prob_KP481)
print("Probability of use KP781 - ",Prob_KP781)
```

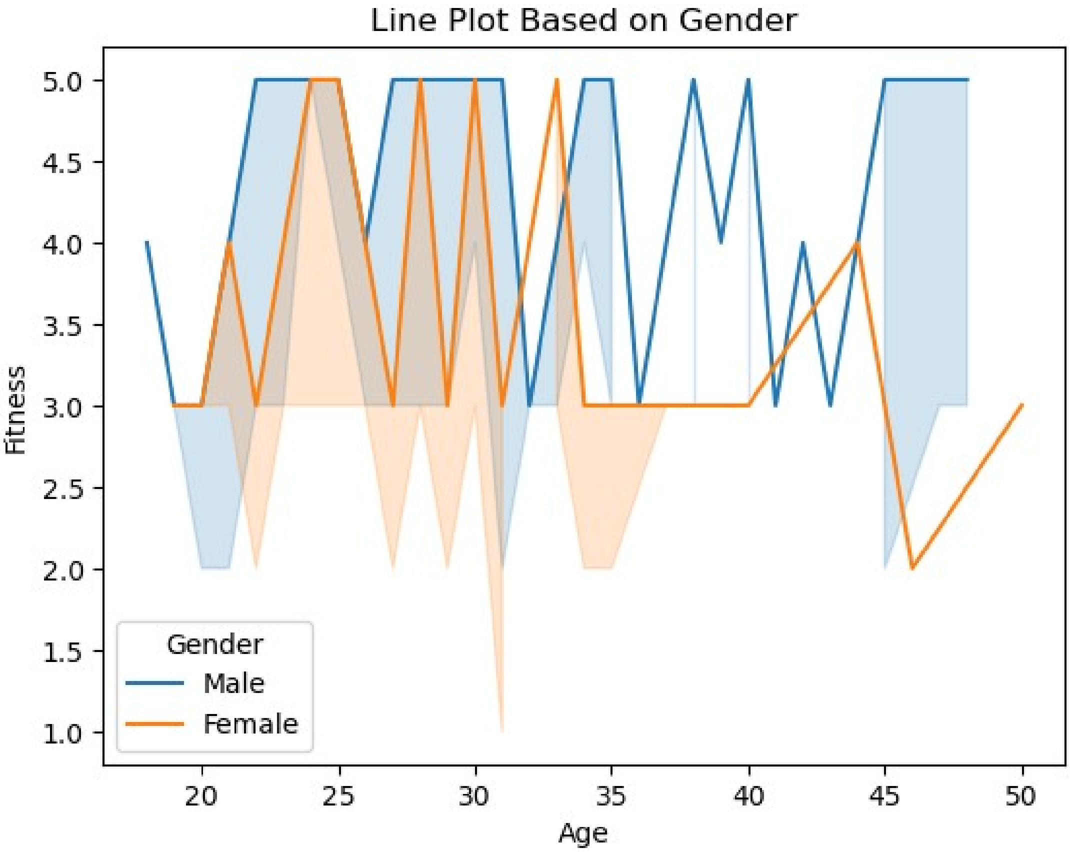


```
KP781 product is mostly used by all age groups and it is also
noticable that High income people use this mostly this product and it
```

is also notiocable that KP281 is highly Purchase but people with low income purchase this mostly



Partnered people and single people both are educated but incomes of partnered peoples are more

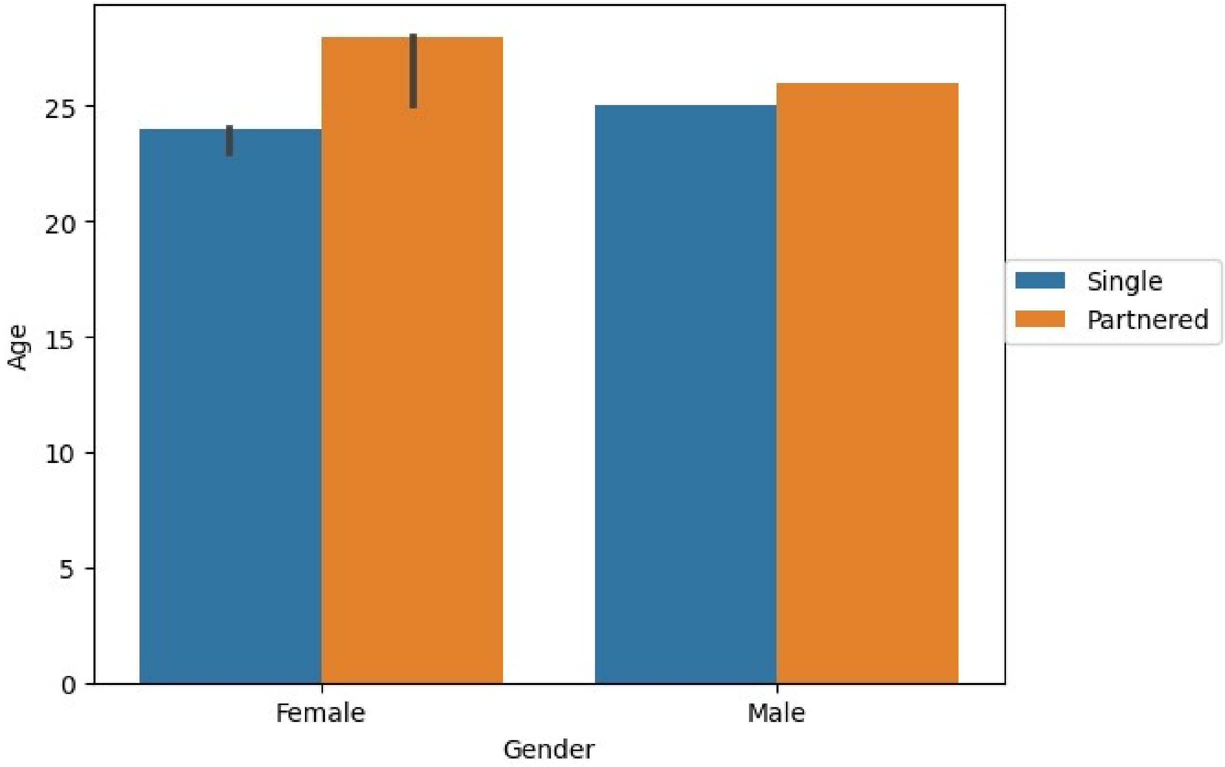


```
Male gender is constantly more consious to their health whereas
females are too but they are not constant, they  are fluctating over
age
Probability of use KP281 -  0.4444444444444444
Probability of use KP481 -  0.3333333333333333
Probability of use KP781 -  0.2222222222222222
```

```
# Non-Graphical Analysis

Top_3_Age=Aerofit["Age"].value_counts().index[:3]
Top_3_Income=Aerofit["Income"].value_counts().index[:3]
Top_3_Miles=Aerofit["Miles"].value_counts().index[:3]

print("Top_3_Age_People more use Aerofit Product : ",Top_3_Age)
print("Top_3_Income_People : ",Top_3_Income)
print("Top_3_Miles_runned by People on Treadmil : ",Top_3_Miles)
```

```
Top_3_Age_People more use Aerofit Product :  Index([25, 23, 24],
dtype='int64', name='Age')
Top_3_Income_People :  Index([45480, 52302, 46617], dtype='int64',
name='Income')
Top_3_Miles_runned by People on Treadmil :  Index([85, 95, 66],
dtype='int64', name='Miles')
```

```
# Graphical analysis on top data
sns.barplot(data=Top_5_data_based_on_Age_Income_Miles,x="Gender",y="Ag
e",hue="MaritalStatus",estimator=np.max)
plt.legend(loc=(1,0.5))
plt.show()

y1=Top_5_data_based_on_Age_Income_Miles["Gender"].value_counts()
y2=Top_5_data_based_on_Age_Income_Miles["MaritalStatus"].value_counts(
)

print("Total Genders in top 5 : ", y1)
print("Marital Status in top 5 : ", y2)
```



```
Total Genders in top 5 :  Gender
Female    6
Male      2
Name: count, dtype: int64
Marital Status in top 5 :  MaritalStatus
Partnered    5
Single       3
Name: count, dtype: int64
```

6. Recommendations - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand

Recommendations for Aerofit Product Strategy:

1. Strategic Pricing Adjustments: Consider revising the pricing strategy for the KP781 treadmill, making it more accessible to high-income individuals. This could potentially boost sales in this demographic without compromising profitability.

2. Targeted Marketing for Single and Female Demographics: Develop marketing campaigns specifically tailored to single individuals and females, highlighting the benefits of Aerofit products. Address the identified lack of awareness within these demographics.

3. Health Awareness Campaigns for 35-50 Age Group: Implement targeted health awareness campaigns focused on the age group between 35-50 years. Emphasize the importance of fitness in maintaining overall well-being and showcase how Aerofit equipment can contribute to a healthy lifestyle.

4. Promoting Female Engagement in Fitness: Create initiatives that make Aerofit products more appealing and engaging for females. This could include workout challenges, exclusive content, or community events to encourage consistent usage.

5. Reward Program for High-Intensity Users: Launch a rewards program for users who consistently run more than 200 miles. Offer discounts, exclusive merchandise, or other incentives to encourage users to surpass this milestone, promoting both fitness and brand loyalty.

6. High-End Product Development: Introduce a premium line of products with advanced features catering to high-income individuals (income > 67K). This will diversify the product range and capture a larger share of the market.

7. Monthly Fitness Band Challenge: Establish a monthly fitness band challenge, rewarding users with only 5 fitness bands. This will motivate users to exercise regularly, fostering a sense of achievement and indirectly promoting Aerofit products.

8. Acknowledgment of User Data Completeness: Leverage the fact that Aerofit users consistently provide complete data. In marketing materials, highlight the user community's commitment and dedication, emphasizing the reliability of the brand and its products.

By implementing these recommendations, Aerofit can refine its product strategy, tap into new market segments, and strengthen its position in the fitness equipment industry.