#Aerofit BusinessCase

```
# importing libraries -
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# reading the data file -
df=pd.read_csv('aerofit_treadmill.csv')
df.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
```

```
df.shape
```

```
(180, 9)
```

```
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
```

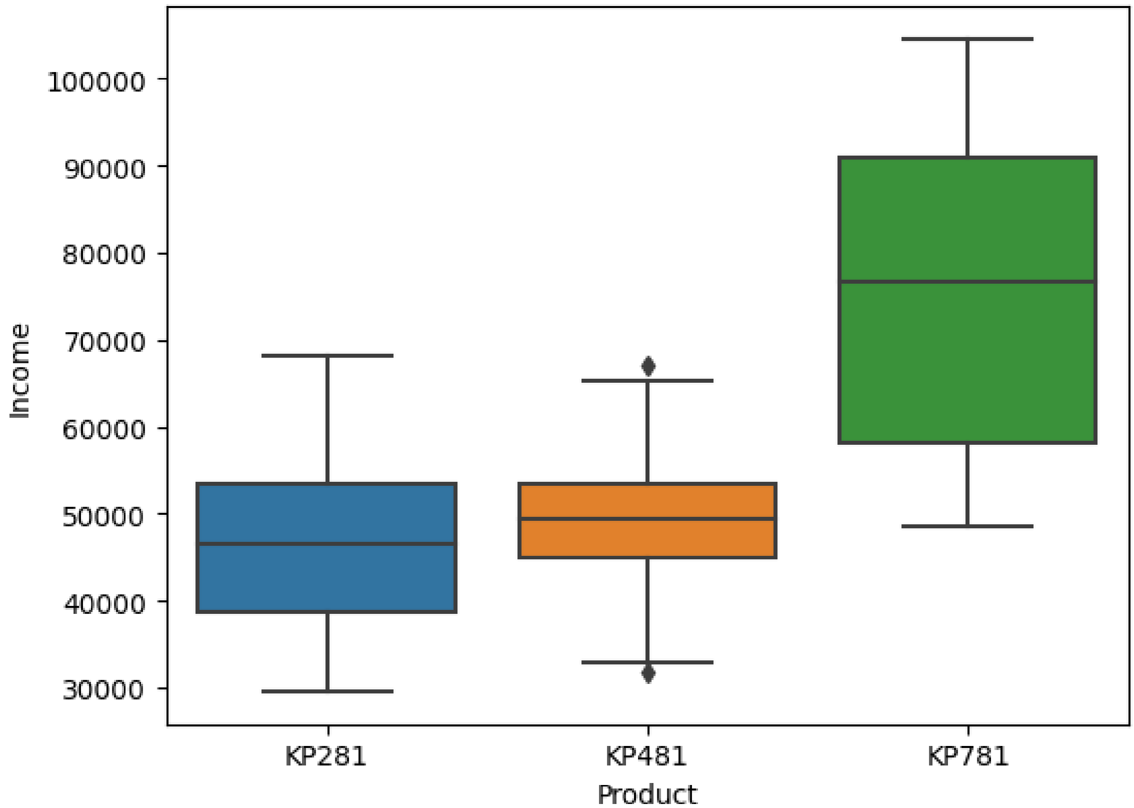
Let's check the statistical information of the data

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

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



The mean from above analysis lies inline with the median of the products KP281,KP481. Seems like the KP781 buyer's income could act as outliers.

#checking if there are any null values

```
df.isna().sum()
```

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

We can clearly see that there are no Null values in every column.

Non-Graphical Analysis

```
df['Age'].value_counts(normalize=True) * 100
```

```
25    13.888889
23    10.000000
24     6.666667
26     6.666667
28     5.000000
35     4.444444
33     4.444444
30     3.888889
38     3.888889
21     3.888889
22     3.888889
27     3.888889
31     3.333333
34     3.333333
29     3.333333
20     2.777778
40     2.777778
32     2.222222
19     2.222222
48     1.111111
37     1.111111
45     1.111111
47     1.111111
46     0.555556
50     0.555556
18     0.555556
44     0.555556
43     0.555556
41     0.555556
39     0.555556
36     0.555556
42     0.555556
Name: Age, dtype: float64
```

From the data, we can say that the top buyers are in the age group of 23-30

```
df['Gender'].value_counts(normalize=True) * 100

Male      57.777778
Female    42.222222
Name: Gender, dtype: float64
```

Male buyers are almost 15% greater than Female buyers.

```
df['MaritalStatus'].value_counts(normalize=True) * 100

Partnered    59.444444
Single       40.555556
Name: MaritalStatus, dtype: float64
```

Seems like , people are focusing more on their fitness after getting Married 🤫

```
df['Product'].value_counts(normalize=True) * 100
```

```
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: Product, dtype: float64
```

Top selling product is KP281

```
df['Age'].unique()
```

```
array([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])
```

```
df['Income'].unique()
```

```
array([ 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])
```

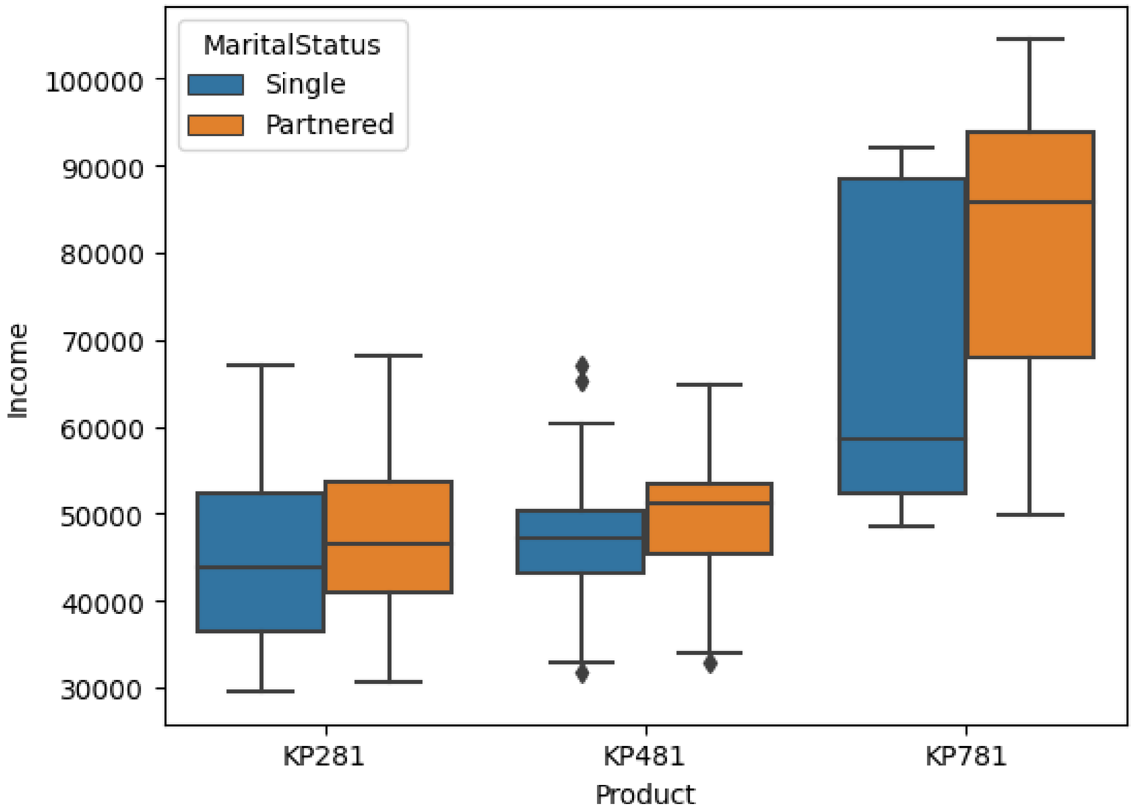
```
df['Education'].unique()
```

```
array([14, 15, 12, 13, 16, 18, 20, 21])
```

Graphical Analysis

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

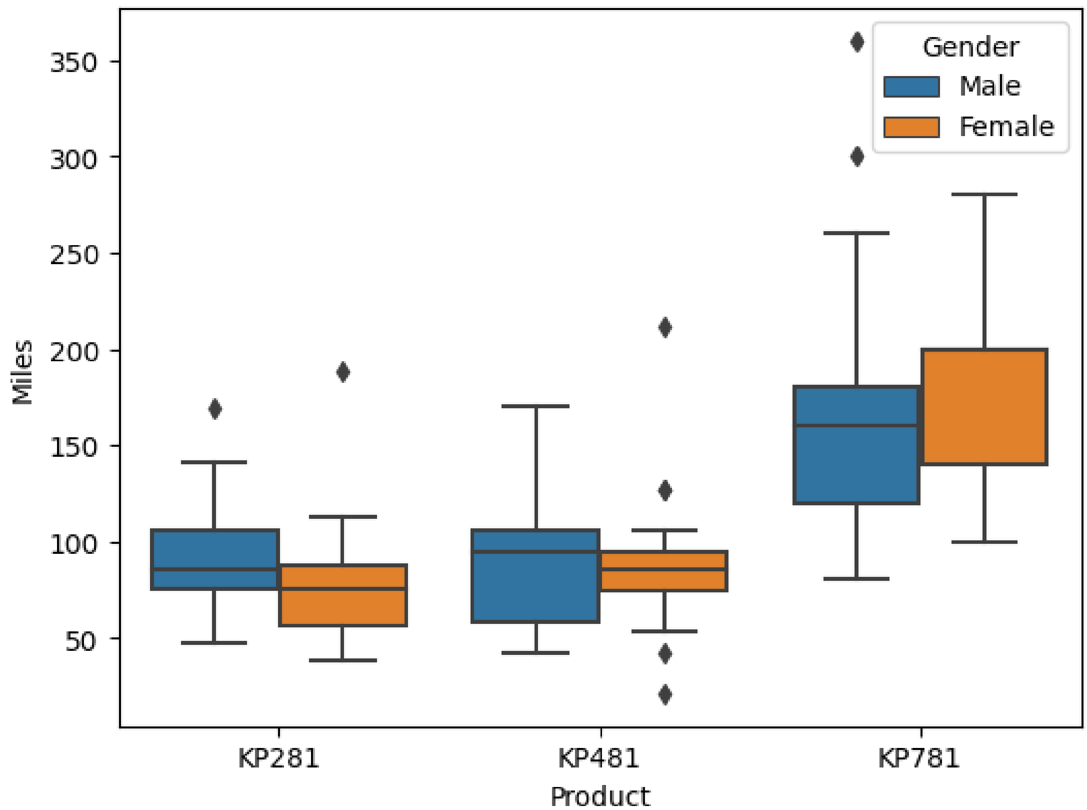
```
<Axes: xlabel='Product', ylabel='Income'>
```



From the above graph, We can say that income of the buyer has the effect on the Product they buy, Suppose buyers with high income prefer more to buy KP781 than others.

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

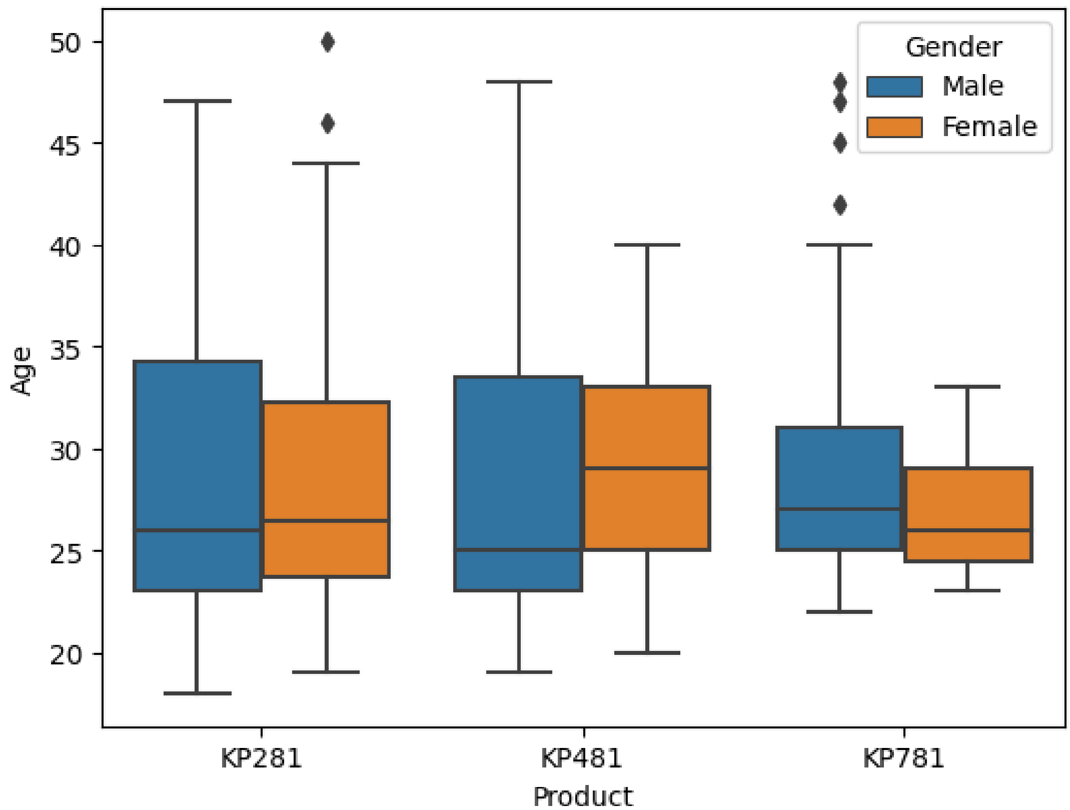
```
<Axes: xlabel='Product', ylabel='Miles'>
```



There is no significant difference for the products KP281 and KP481 in Miles but KP781 buyers tend to make more Miles.

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

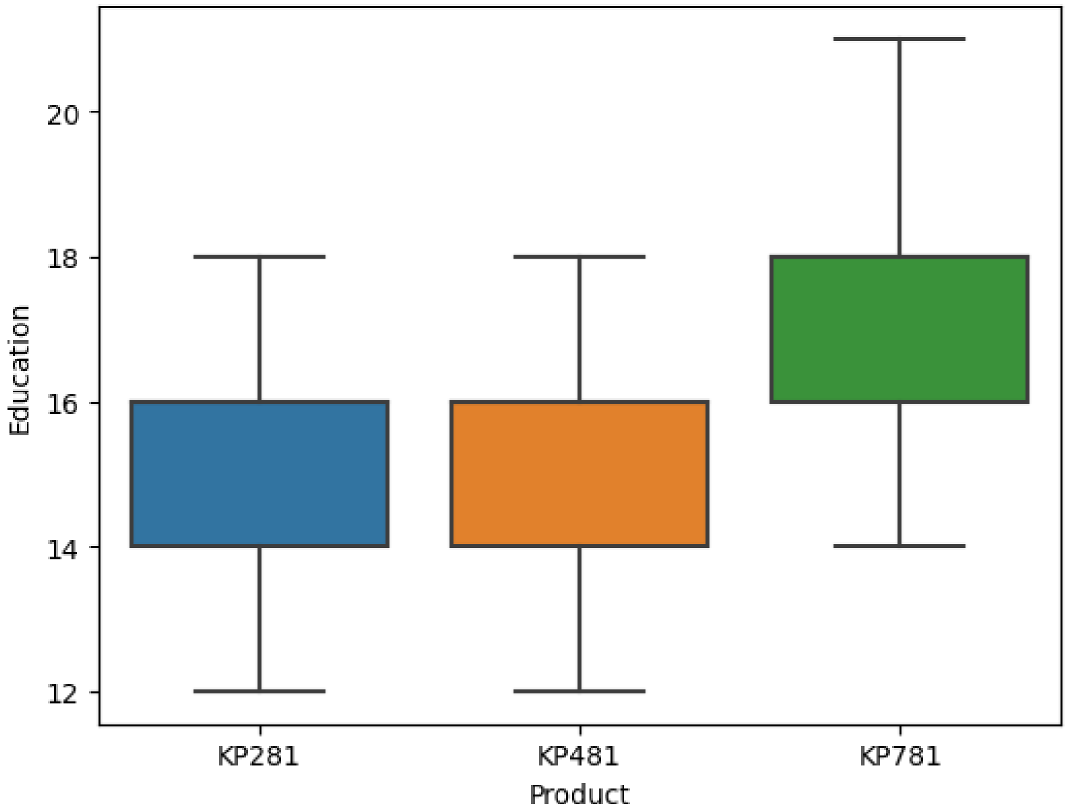
```
<Axes: xlabel='Product', ylabel='Age'>
```



Almost for all the products, The Age difference between Male and Female buyers is not much(<=5).

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

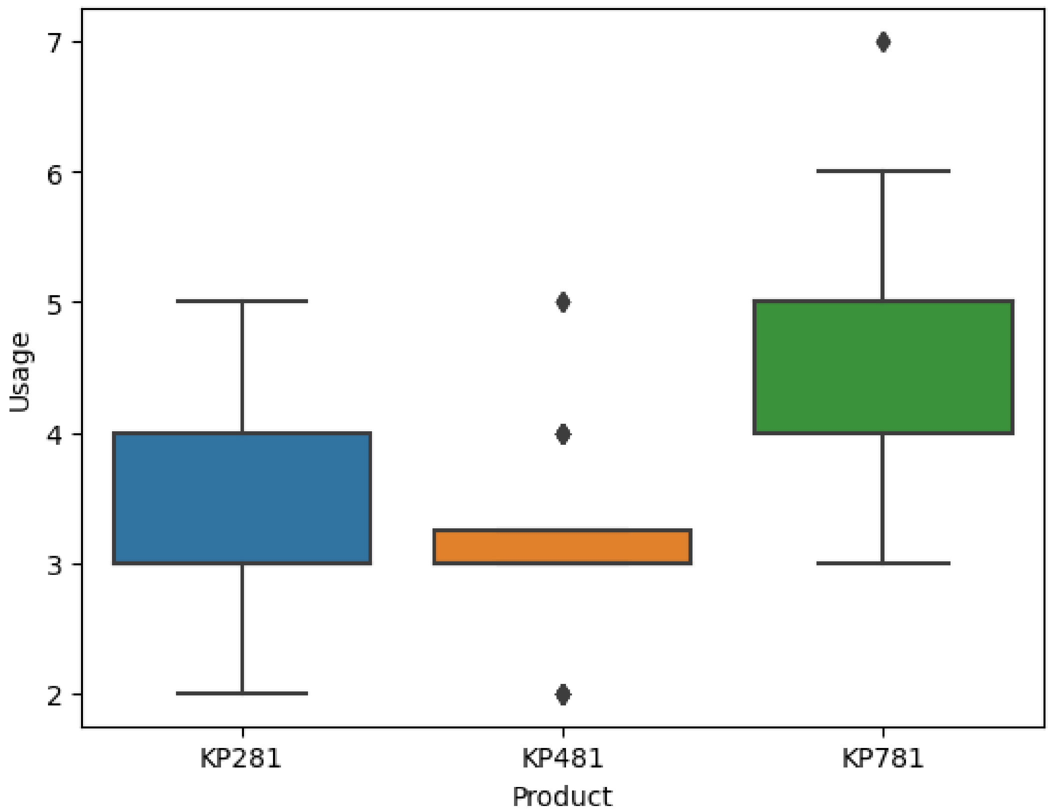
```
<Axes: xlabel='Product', ylabel='Education'>
```



KP781 buyers seems to be more educated than others.

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

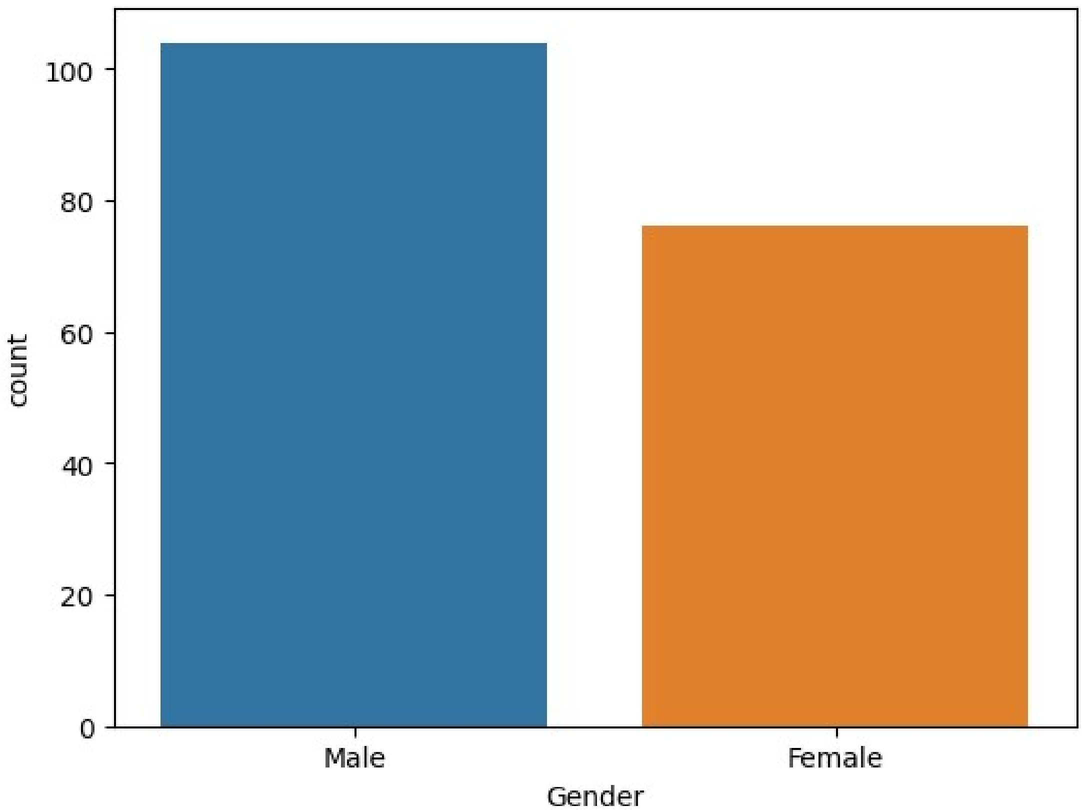
```
<Axes: xlabel='Product', ylabel='Usage'>
```



The Usage of KP481 product is comparitively less than the other two.

```
sns.countplot(data=df, x='Gender')
```

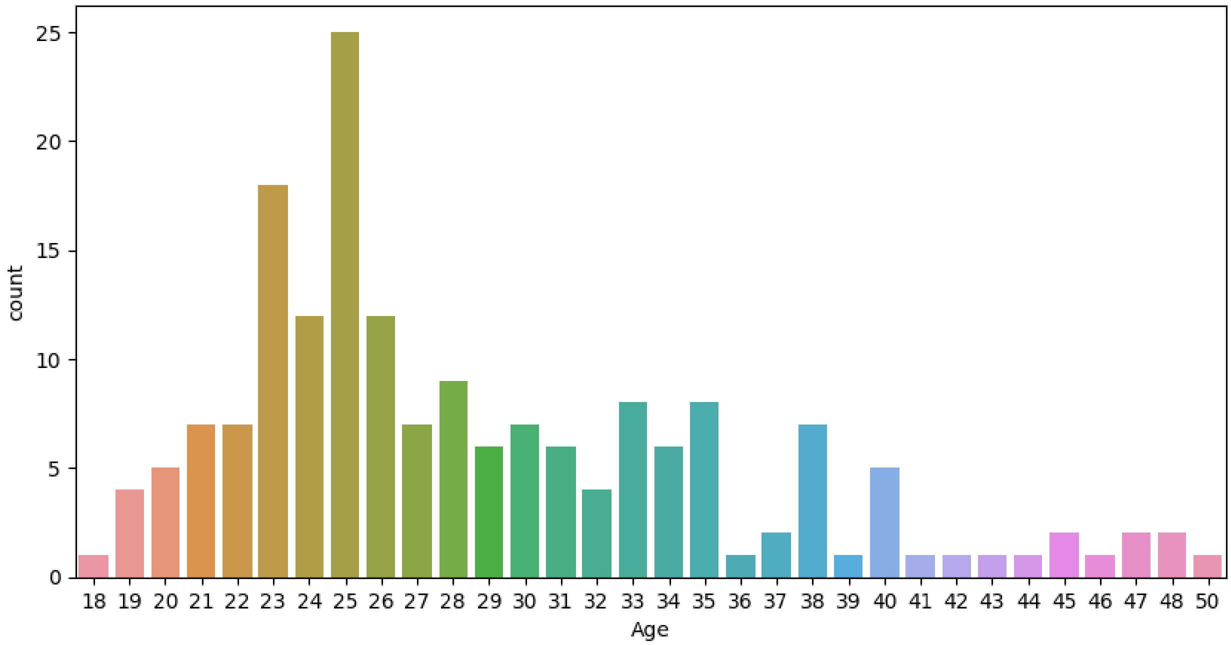
```
<Axes: xlabel='Gender', ylabel='count'>
```



Male buyers are more.

```
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='Age')
```

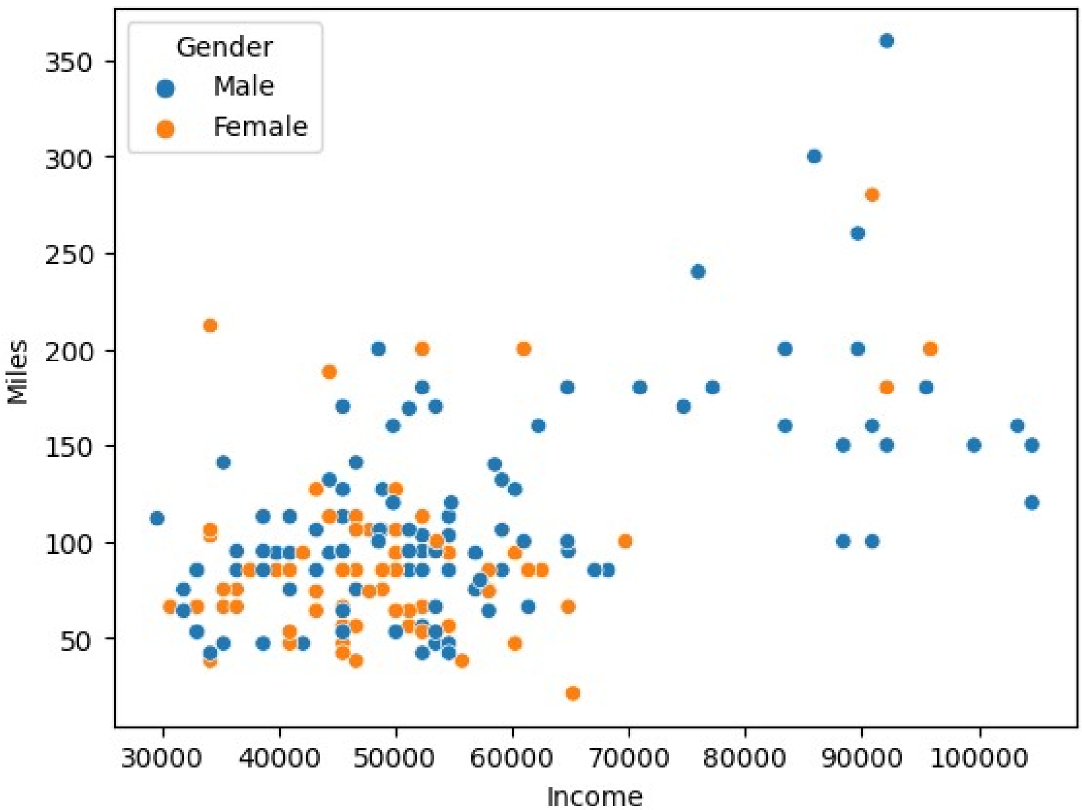
```
<Axes: xlabel='Age', ylabel='count'>
```



The Age group of the buyers are higher in 20's comparitively.

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

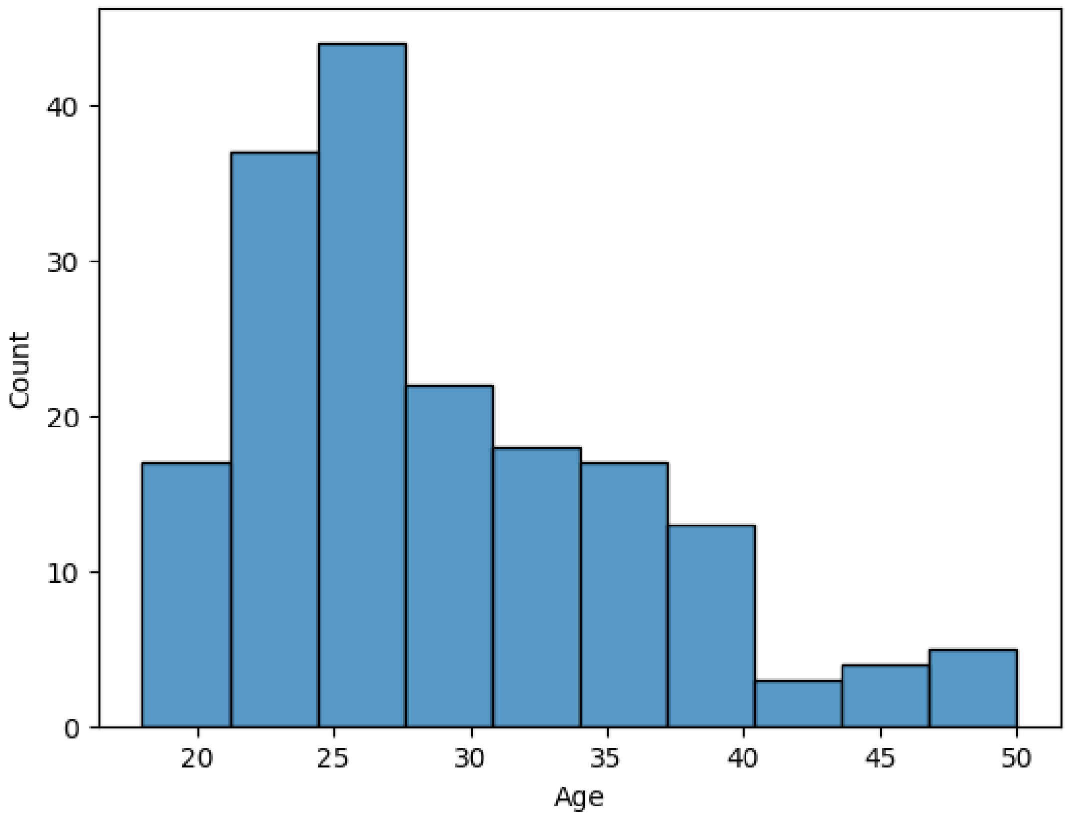
```
<Axes: xlabel='Income', ylabel='Miles'>
```



Most of the buyers income range is between 30000 to 60000.

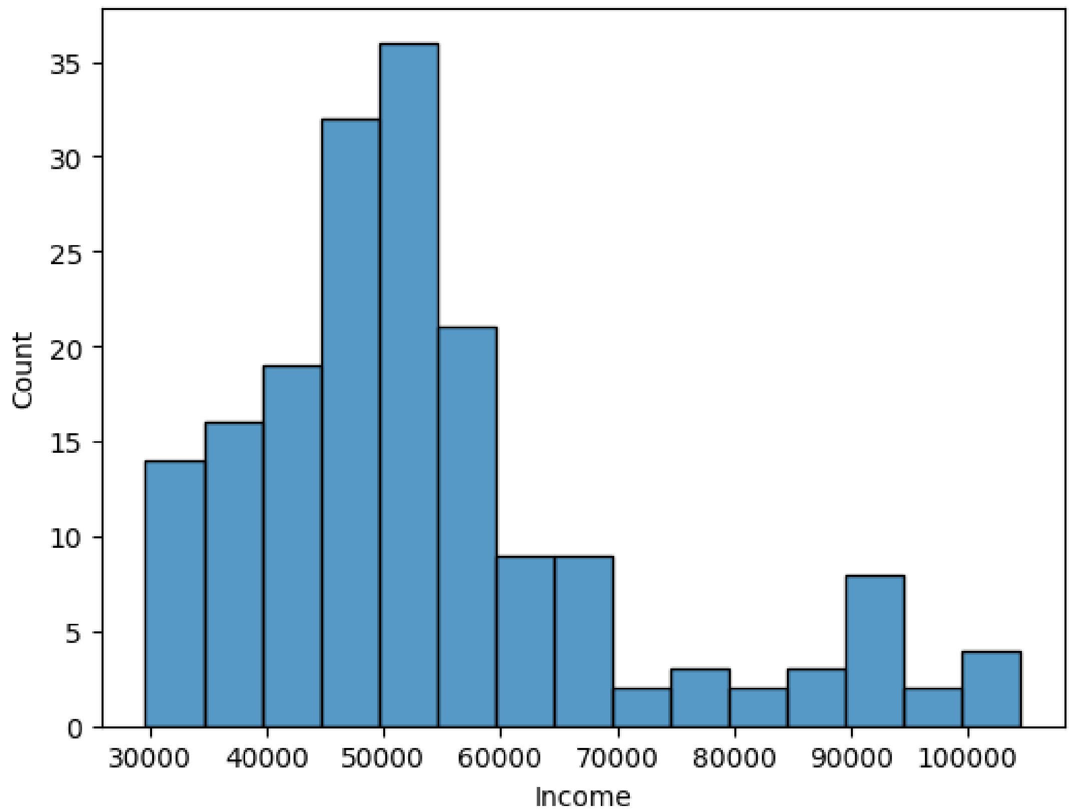
```
sns.histplot(df['Age'],bins=10)
```

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



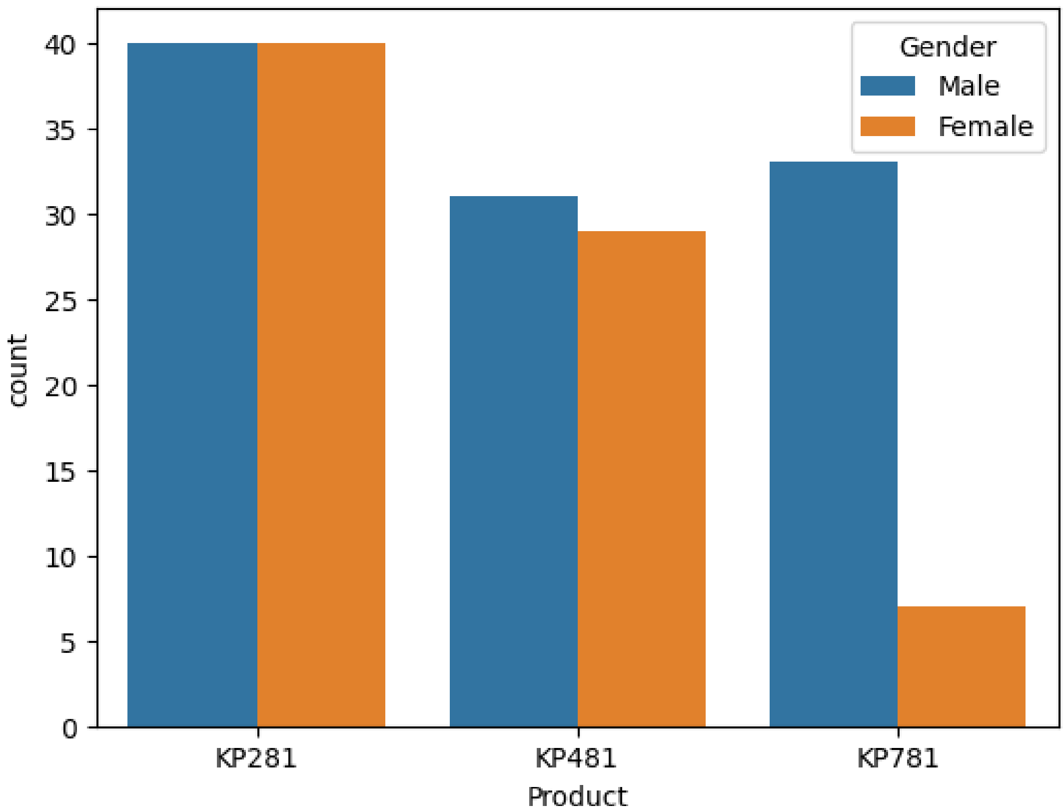

```
sns.histplot(df['Income'],bins=15)
```

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



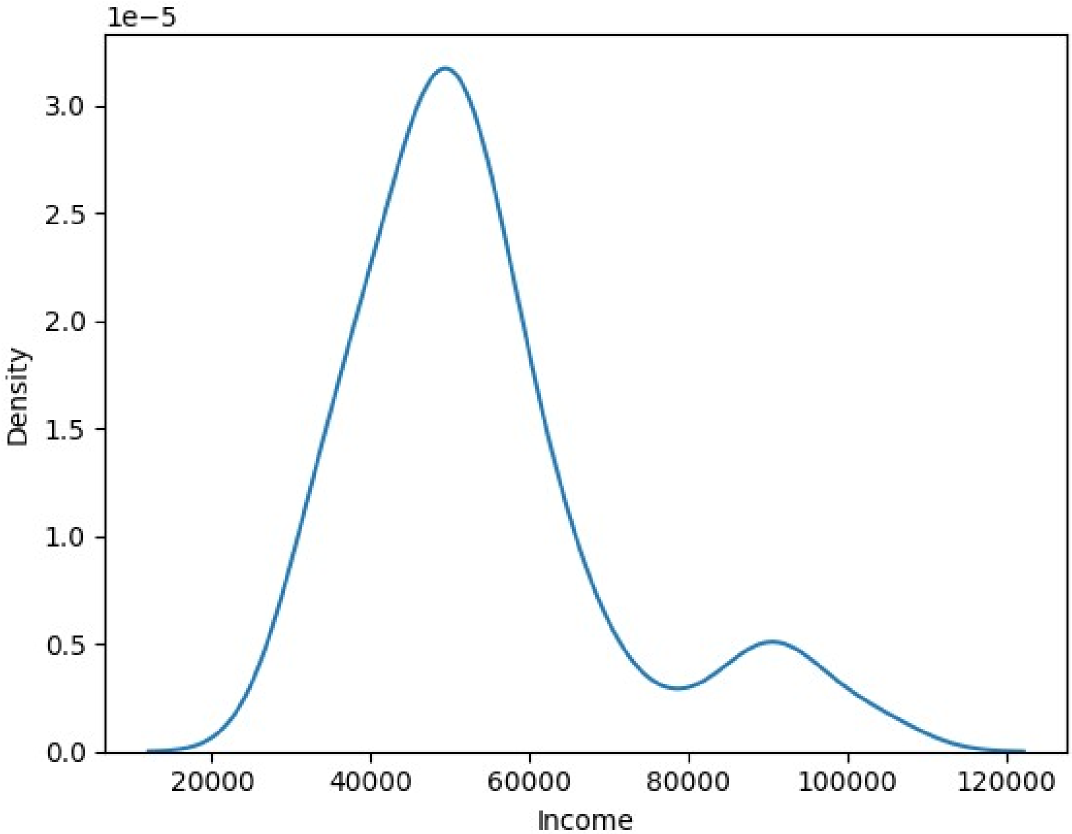

```
sns.countplot(data=df, x='Product',hue='Gender')
```

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



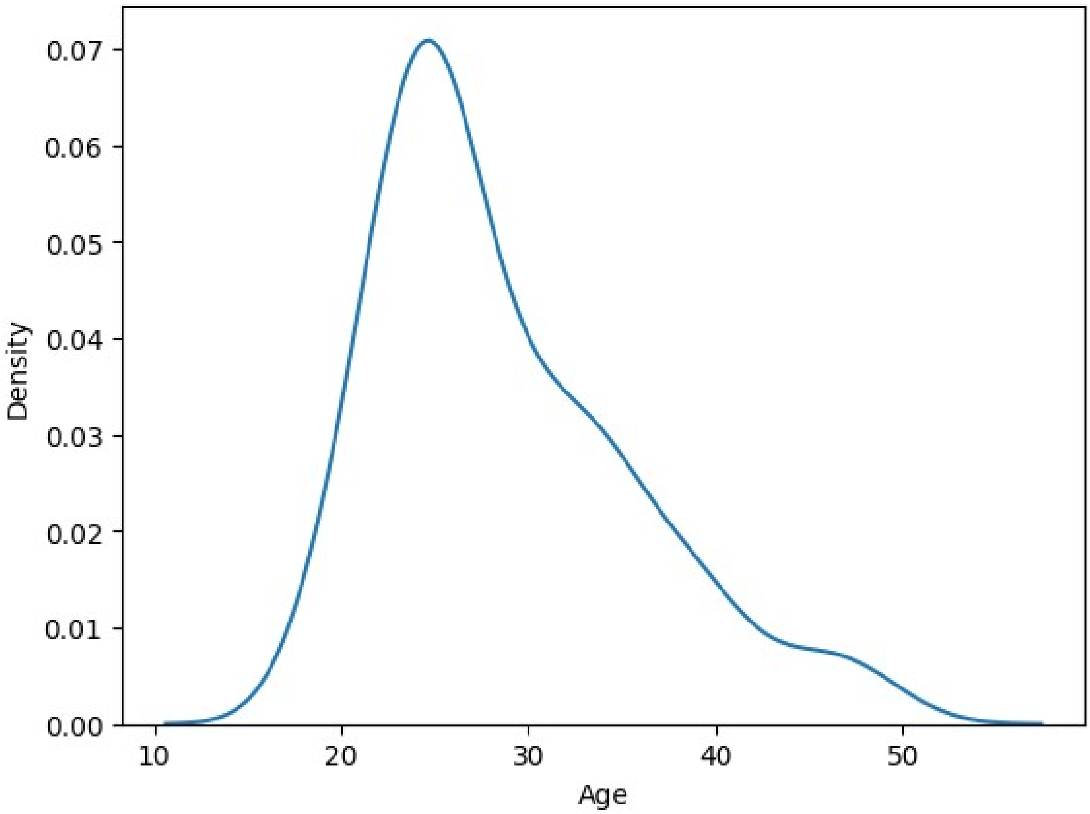
Male buyers are much higher for the product KP781.

```
sns.kdeplot(df['Income'])

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



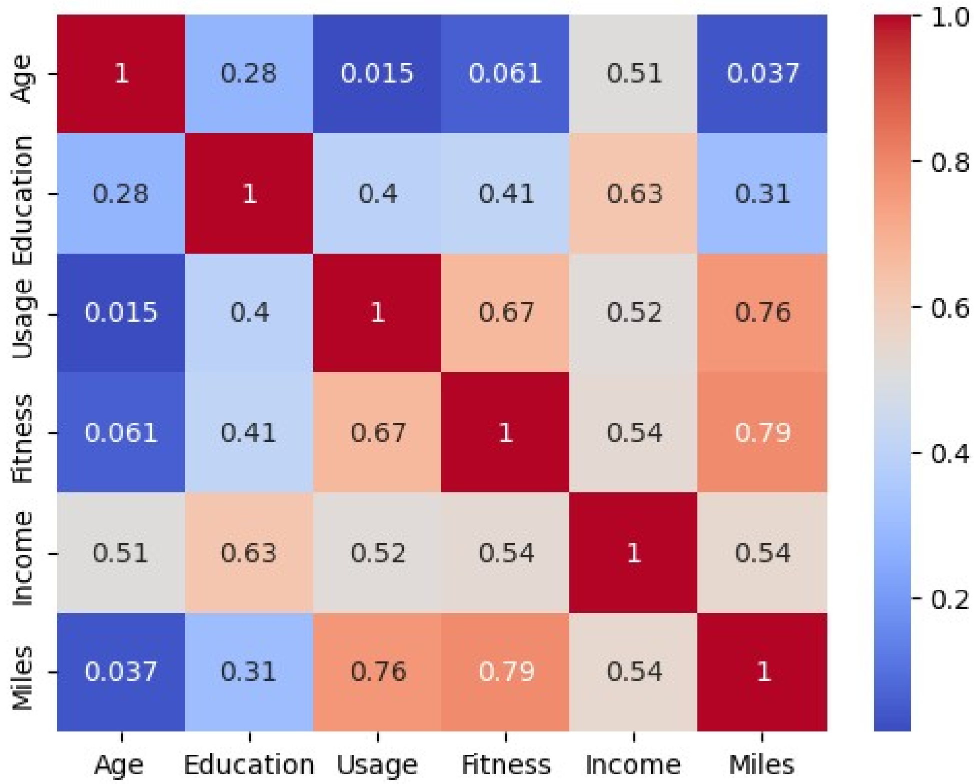

```
sns.kdeplot(df['Age'])
```

```
<Axes: xlabel='Age', ylabel='Density'>
```




```
sns.heatmap(df.corr(), cmap= "coolwarm", annot=True)
plt.show()
```

```
<ipython-input-34-82df116f6821>:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it
will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  sns.heatmap(df.corr(), cmap= "coolwarm", annot=True)
```



Positive correlation is measured on a 0.1 to 1.0 scale. Weak positive correlation would be in the range of 0.1 to 0.3, moderate positive correlation from 0.3 to 0.5, and strong positive correlation from 0.5 to 1.0.

#Contigency table - Product vs Gender

```
pd.crosstab(df['Product'], df['Gender'])

Gender    Female  Male
Product
KP281         40    40
KP481         29    31
KP781          7    33
```

From the table,

Total buyers = 40+40+29+31+7+33 = 180

Total Male buyers = 40+29+7 = 76

Total Male buyers = 40+31+33 = 104

Total buyers of KP281 = 40+40 = 80

Total buyers of KP481 = 29+31 = 60

Total buyers of KP781 = 7+33 = 40

#Marginal Probability

As we know the probability of Female and Male buyers for Product KP281, let's find out what is the probability that atleast one buyer(either Female or Male) buys the product.

```
P_Female_KP281 = 40/80
P_Male_KP281 = 40/80
P_FemaleUMale_KP281 =  P_Female_KP281 + P_Male_KP281 - 0
P_FemaleUMale_KP281
```

```
1.0
```

#Conditional Probability

What is the Probability that the person buys the product KP281, given that the Person is Female?

```
Prob_of_KP281_and_Female = 40/180
Prob_of_Female = 76/180
Prob_of_KP281_given_Female =
(Prob_of_KP281_and_Female)/(Prob_of_Female)
Prob_of_KP281_given_Female
```

```
0.5263157894736842
```

What is the Probability that the person buys the product KP281, given that the Person is Male?

```
Prob_of_KP281_and_Male = 40/180
Prob_of_Male = 104/180
Prob_of_KP281_given_Male = (Prob_of_KP281_and_Male)/(Prob_of_Male)
Prob_of_KP281_given_Male
```

```
0.38461538461538464
```

What is the Probability that the person buys the product KP481, given that the Person is Female?

```
Prob_of_KP481_and_Female = 29/180
Prob_of_Female = 76/180
Prob_of_KP481_given_Female =
(Prob_of_KP481_and_Female)/(Prob_of_Female)
Prob_of_KP481_given_Female
```

```
0.3815789473684211
```

What is the Probability that the person buys the product KP481, given that the Person is Male?

```
Prob_of_KP481_and_Male = 31/180
Prob_of_Male = 104/180
Prob_of_KP481_given_Male = (Prob_of_KP481_and_Male)/(Prob_of_Male)
Prob_of_KP481_given_Male
```

```
0.29807692307692313
```

What is the Probability that the person buys the product KP781, given that the Person is Female?

```
Prob_of_KP781_and_Female = 7/180
Prob_of_Female = 76/180
Prob_of_KP781_given_Female =
(Prob_of_KP781_and_Female)/(Prob_of_Female)
Prob_of_KP781_given_Female

0.09210526315789473
```

What is the Probability that the person buys the product KP781, given that the Person is Male?

```
Prob_of_KP781_and_Male = 33/180
Prob_of_Male = 104/180
Prob_of_KP781_given_Male = (Prob_of_KP781_and_Male)/(Prob_of_Male)
Prob_of_KP781_given_Male

0.3173076923076923
```

Baye's Theorem

What is the Probability that the Person is Male , given that the person buys the product KP481?

```
Prob_of_Male_given_KP481 = (Prob_of_KP481_given_Male *
Prob_of_Male)/((Prob_of_KP481_given_Male * Prob_of_Male)+
(Prob_of_KP481_given_Female * Prob_of_Female))
Prob_of_Male_given_KP481

0.5166666666666667
```

What is the Probability that the Person is Female , given that the person buys the product KP781?

```
Prob_of_Female_given_KP781 = (Prob_of_KP781_given_Female *
Prob_of_Female)/((Prob_of_KP781_given_Male * Prob_of_Male)+
(Prob_of_KP781_given_Female * Prob_of_Female))
Prob_of_Female_given_KP781

0.17500000000000004
```

#Contigency table - Product vs MaritalStatus

```
pd.crosstab(df['Product'], df['MaritalStatus'])

MaritalStatus  Partnered  Single
Product
KP281                 48      32
```

```
KP481                          36        24
KP781                          23        17
```

From the table,

Total buyers of Parterned Status = 48+36+23 = 107

Total buyers of Single Status = 32+24+17 = 73

Total buyers of KP281 = 40+40 = 80

Total buyers of KP481 = 29+31 = 60

Total buyers of KP781 = 23+17 = 40

#Marginal Probability

As we know the probability of Partnered and Single buyers for Product KP781, let's find out what is the probability that atleast one buyer buys the product.

```
P_Partnered_KP781 = 23/40
P_Single_KP781 = 17/40
P_PartneredUSingle_KP781 =  P_Partnered_KP781 + P_Single_KP781 - 0
P_PartneredUSingle_KP781

1.0
```

#Conditional Probability

What is the Probability that the person buys the product KP281, given that the Person is Partnered?

```
Prob_of_KP281_and_Partnered = 48/180
Prob_of_Partnered = 107/180
Prob_of_KP281_given_Partnered =
(Prob_of_KP281_and_Partnered)/(Prob_of_Partnered)
Prob_of_KP281_given_Partnered

0.4485981308411215
```

What is the Probability that the person buys the product KP281, given that the Person is Single?

```
Prob_of_KP281_and_Single = 32/180
Prob_of_Single = 73/180
Prob_of_KP281_given_Single =
(Prob_of_KP281_and_Single)/(Prob_of_Single)
Prob_of_KP281_given_Single

0.4383561643835617
```

What is the Probability that the person buys the product KP481, given that the Person is Partnered?

```
Prob_of_KP481_and_Partnered = 36/180
Prob_of_Partnered = 107/180
Prob_of_KP481_given_Partnered =
(Prob_of_KP481_and_Partnered)/(Prob_of_Partnered)
Prob_of_KP481_given_Partnered
```

```
0.33644859813084116
```

What is the Probability that the person buys the product KP481, given that the Person is Single?

```
Prob_of_KP481_and_Single = 24/180
Prob_of_Single = 73/180
Prob_of_KP481_given_Single =
(Prob_of_KP481_and_Single)/(Prob_of_Single)
Prob_of_KP481_given_Single
```

```
0.3287671232876712
```

What is the Probability that the person buys the product KP781, given that the Person is Partnered?

```
Prob_of_KP781_and_Partnered = 23/180
Prob_of_Partnered = 107/180
Prob_of_KP781_given_Partnered =
(Prob_of_KP781_and_Partnered)/(Prob_of_Partnered)
Prob_of_KP781_given_Partnered
```

```
0.21495327102803738
```

What is the Probability that the person buys the product KP781, given that the Person is Single?

```
Prob_of_KP781_and_Single = 17/180
Prob_of_Single = 73/180
Prob_of_KP781_given_Single =
(Prob_of_KP781_and_Single)/(Prob_of_Single)
Prob_of_KP781_given_Single
```

```
0.2328767123287671
```

Baye's Theoem

What is the Probability that the person's status is Partnered , given that the person buys the product KP781?

```
Prob_of_Partnered_given_KP781 = (Prob_of_KP781_given_Partnered *
Prob_of_Partnered)/((Prob_of_KP781_given_Partnered *
```

```
Prob_of_Partnered)+(Prob_of_KP781_given_Single * Prob_of_Single))
Prob_of_Partnered_given_KP781
```

```
0.575
```

What is the Probability that the person's status is Single , given that the person buys the product KP281?

```
Prob_of_Single_given_KP281 = (Prob_of_KP281_given_Single *
Prob_of_Single)/((Prob_of_KP281_given_Partnered * Prob_of_Partnered)+
(Prob_of_KP281_given_Single * Prob_of_Single))
Prob_of_Single_given_KP281
```

```
0.4
```