Business Case: Aerofit - Descriptive Statistics & Probability

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.

1. Perform descriptive analytics **to create a customer profile** for each AeroFit treadmill product by developing appropriate tables and charts.
2. 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:

aerofit_treadmill.csv

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 weel
Income:	Annual income (in $)
Fitness:	Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 is the excell shape.
Miles:	The average number of miles the customer expects to walk/run each week

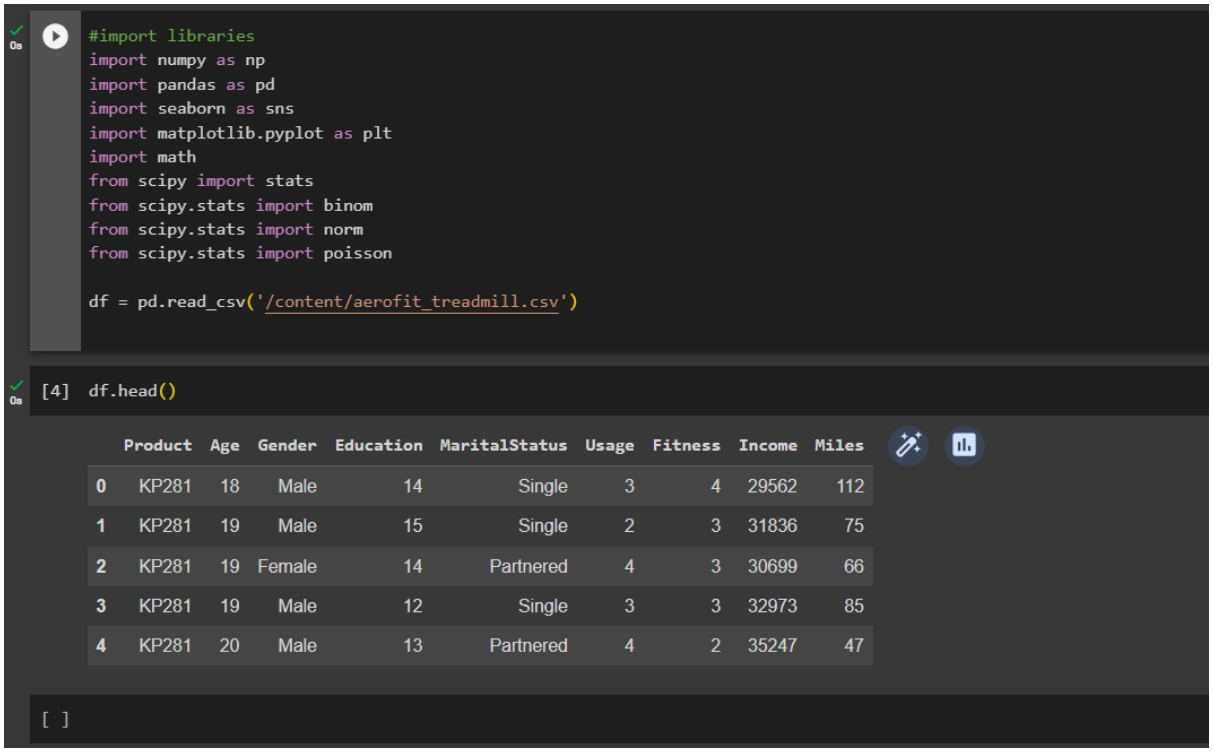
Product Portfolio:

- 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.

1.Defining Problem Statement and Analysing basic metrics (10 Points)

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

```
#import libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import math
from scipy import stats
from scipy.stats import binom
from scipy.stats import norm
from scipy.stats import poisson

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



```
#import libraries
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import math
from scipy import stats
from scipy.stats import binom
from scipy.stats import norm
from scipy.stats import poisson

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

```
[4] 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

```
[ ]
```

```
[5] df
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47
...	...	...	...	...	...	...	...	...	...
175	KP781	40	Male	21	Single	6	5	83416	200
176	KP781	42	Male	18	Single	5	4	89641	200
177	KP781	45	Male	16	Single	5	5	90886	160
178	KP781	47	Male	18	Partnered	4	5	104581	120
179	KP781	48	Male	18	Partnered	4	5	95508	180

180 rows × 9 columns

To find lenghth of the dataset use

len(df)

```
[6] len(df)
180
```

To find diferent data types in the dataset use:

df.dtypes

```
[7] df.dtypes
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

The following line displays a summary of the DataFrame, including descriptive statistics for each column.

df.describe(include="all")

```
[8]  df.describe(include="all")
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

The following lines count the occurrences of each unique value in the specified columns.

df["Age"].value_counts()

```
[9]  df["Age"].value_counts()
```

```
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
Name: Age, dtype: int64
```

```
[10] df["Gender"].value_counts()
```

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

```
[11] df["MaritalStatus"].value_counts()
```

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

```
[12] df["Usage"].value_counts()

3    69
4    52
2    33
5    17
6     7
7     2
Name: Usage, dtype: int64
```

```
[13] df["Fitness"].value_counts()

3    97
5    31
2    26
4    24
1     2
Name: Fitness, dtype: int64
```

```
[14] df["Product"].value_counts()

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

2.Non-Graphical Analysis: Value counts and unique attributes (10 Points)

```
[15] #unique value
     df.nunique()

Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

The line returns the number of unique values in each column of the DataFrame.

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

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

The line loop prints the name of each column in the DataFrame along with the number of unique values in that column.

```
#missing value
df.isna().sum()
```

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

```
[ ]
```

It return the dimensions and total size of the DataFrame.

```
[18] df.shape
```

```
(180, 9)
```

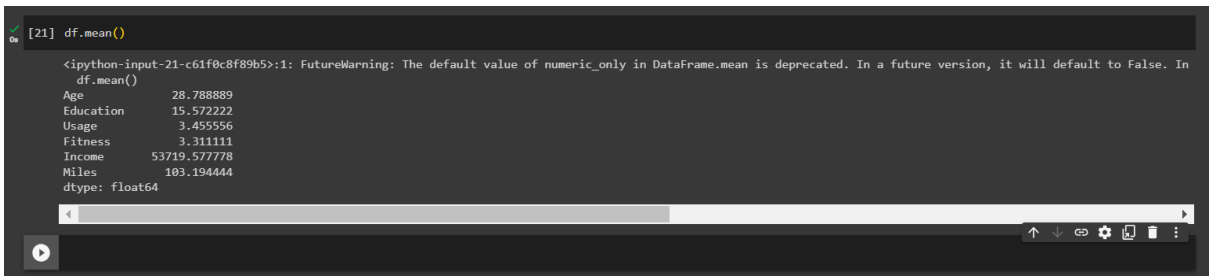
```
[19] df.size
```

```
1620
```

3.Visual Analysis - Univariate & Bivariate (30 Points)

1. **For continuous variable(s): Distplot, countplot, histogram for univariate analysis (10 Points)**
2. **For categorical variable(s): Boxplot (10 Points)**
3. **For correlation: Heatmaps, Pairplots(10 Points)**

```
plt.figure(figsize = (10, 7))
df.boxplot()
```



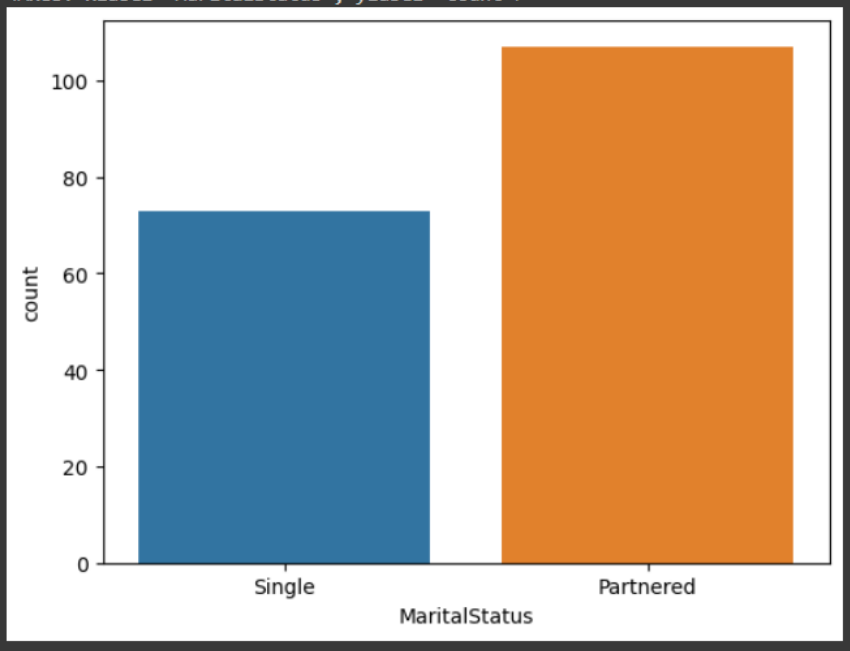
```
[22] df.median()

<ipython-input-22-6d467abf240d>:1: FutureWarning: The default value of numeric_only in DataFrame.median is deprecated. In a future version, it will default to False. I
  df.median()
Age            26.0
Education      16.0
Usage           3.0
Fitness         3.0
Income      50596.5
Miles          94.0
dtype: float64
```

```
[23] df.mode()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	25	Male	16	Partnered	3	3	45480	85

It return the mean, median and mode values for each column in the DataFrame.

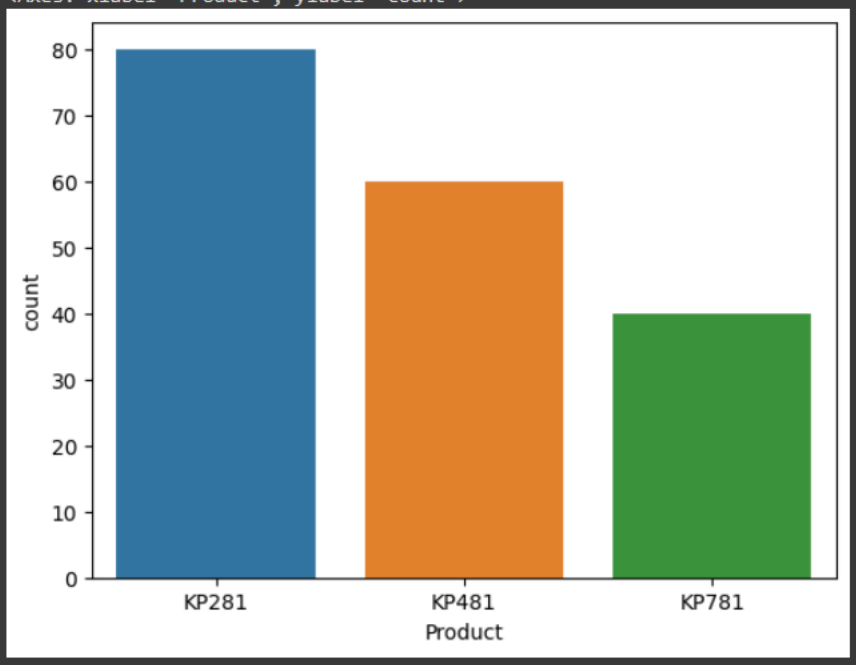
```
[24] sns.countplot(data=df, x='MaritalStatus')

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



It create count plots of the "MaritalStatus", "Product", and "Gender" columns of the DataFrame using Seaborn.

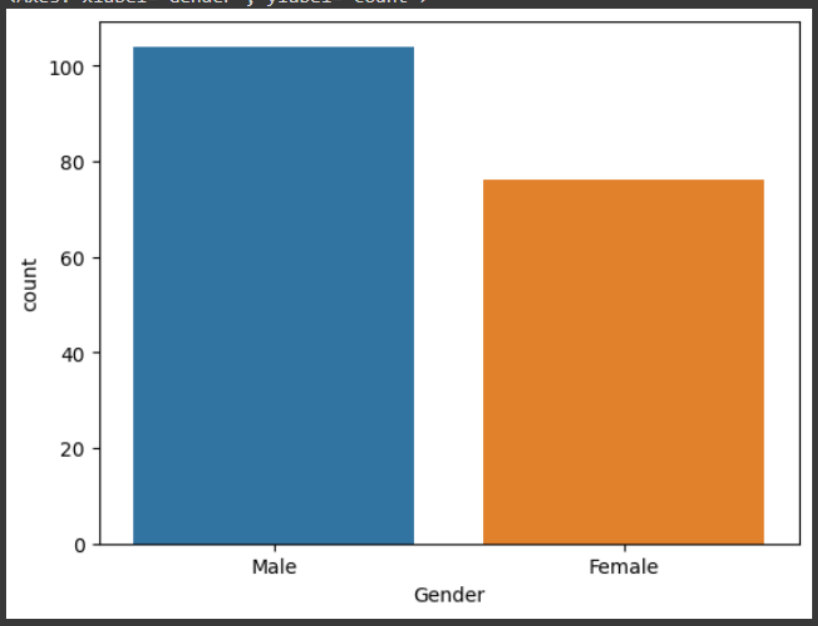
```
[25] sns.countplot(data=df, x='Product')
```

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

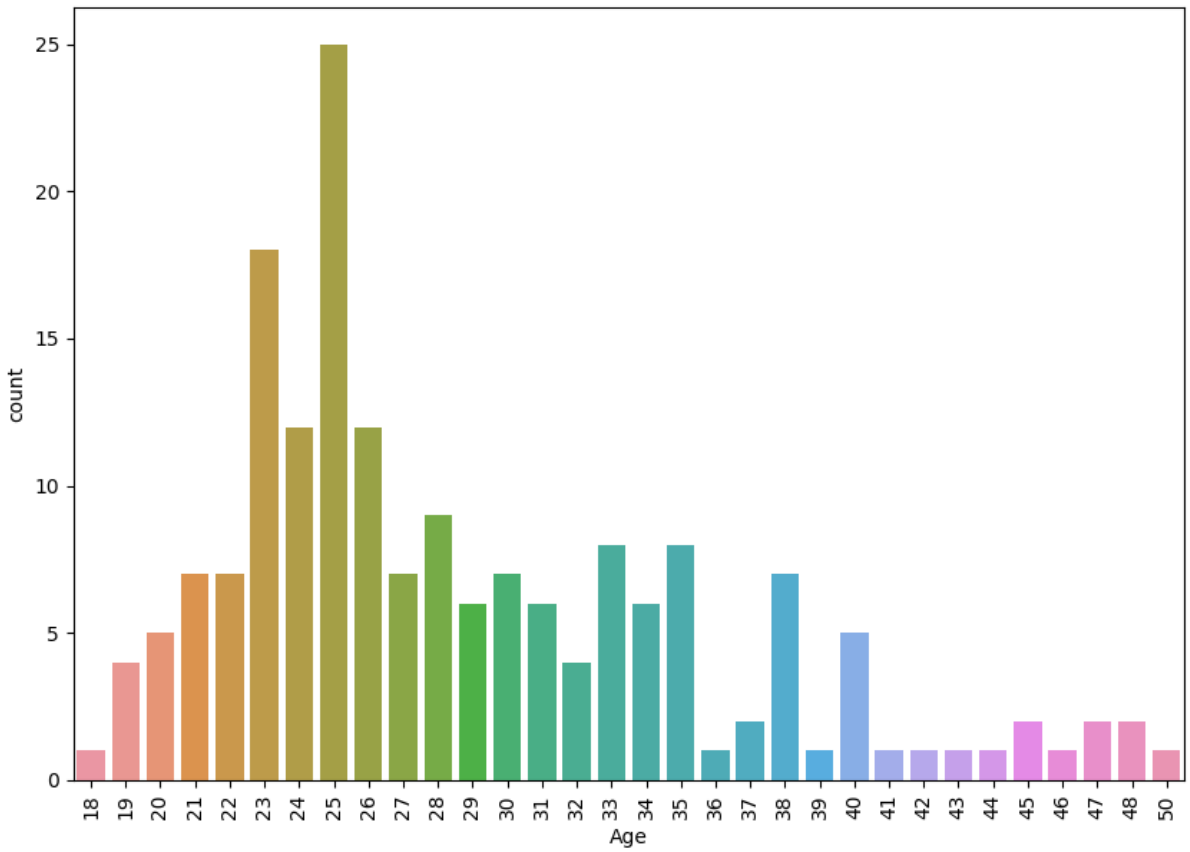



```
sns.countplot(data=df, x='Gender')
```

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

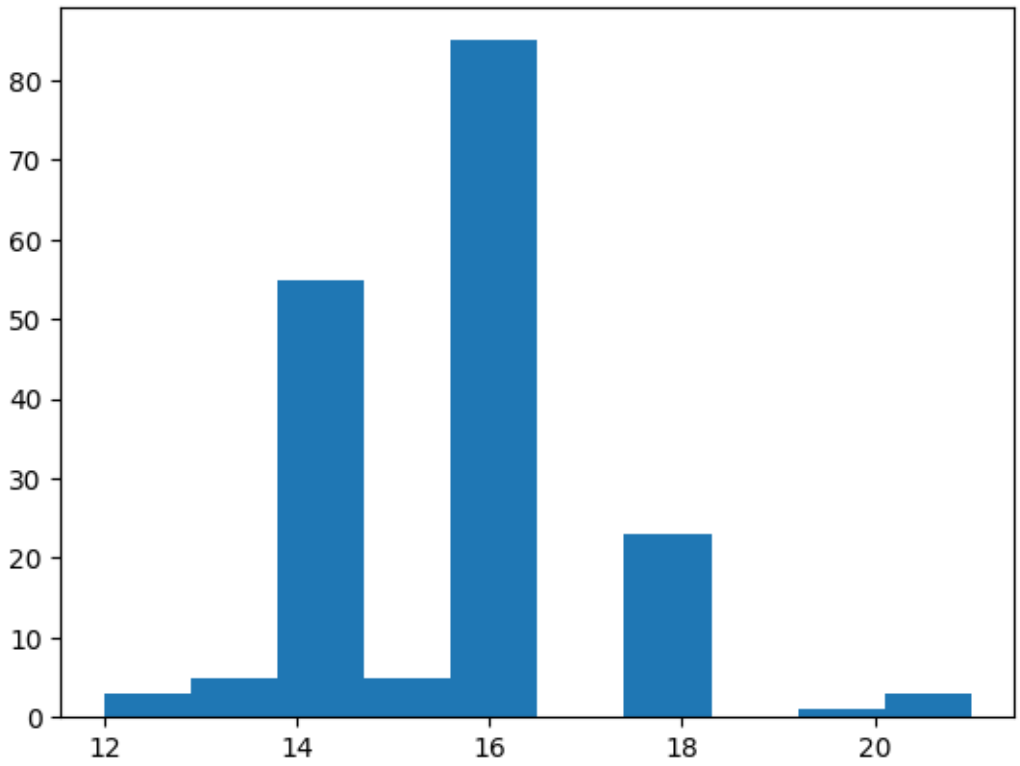


```
plt.figure(figsize = (10, 7))
sns.countplot(data=df, x='Age')
plt.xticks(rotation=90)
plt.show()
```

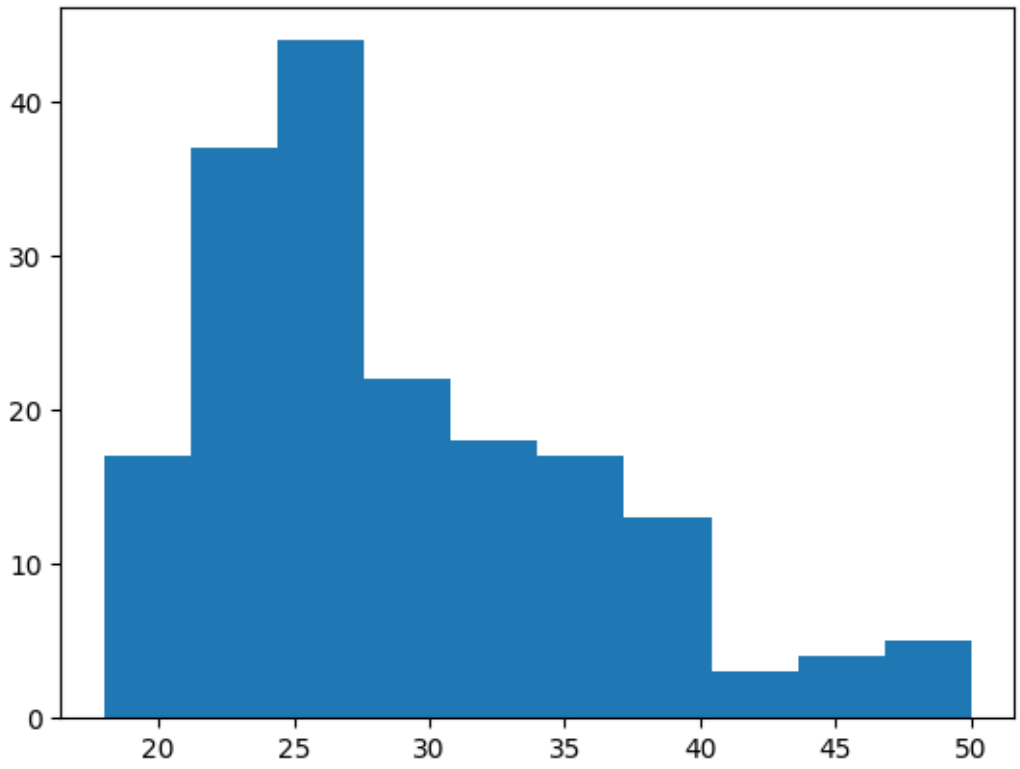


It create histograms of the "Education", "Age", "Product", "Usage", and "Fitness" columns of the DataFrame using Matplotlib.

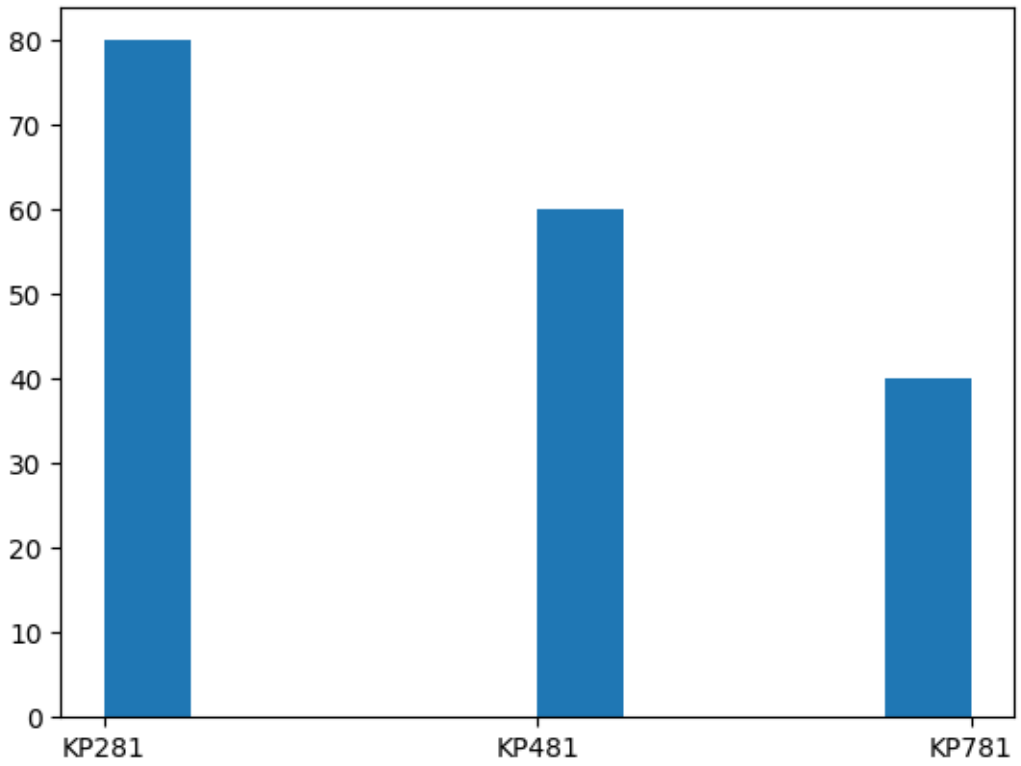
```
plt.hist(df["Education"])
plt.show()
```



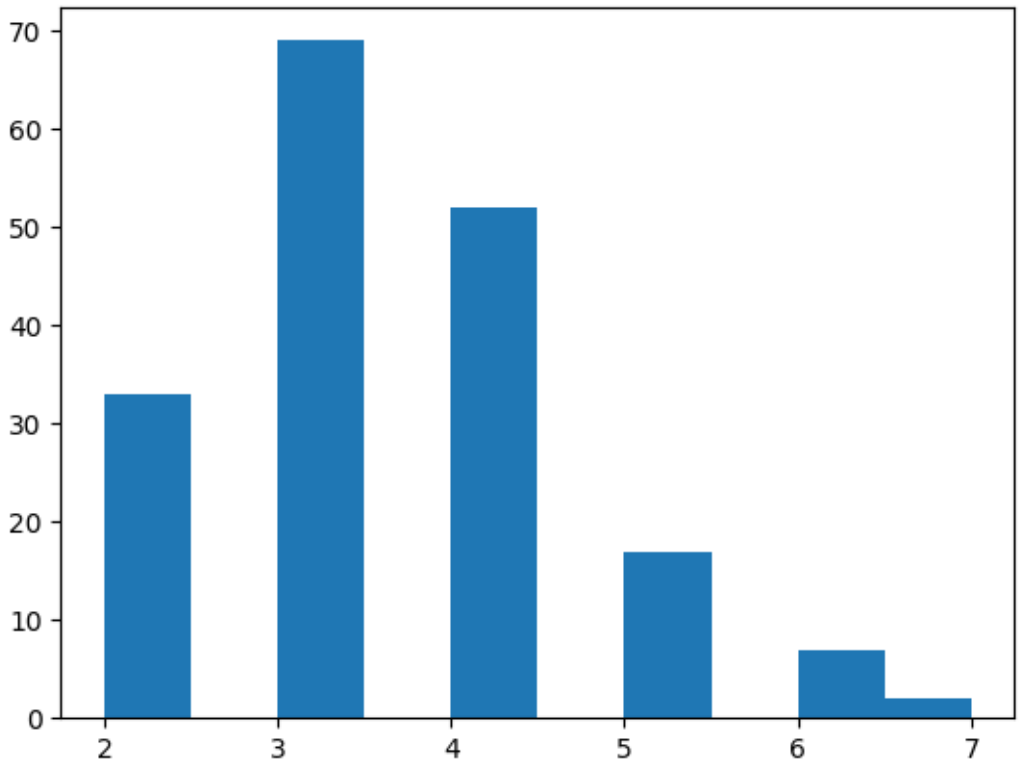

```
plt.hist(df["Age"])
plt.show()
```



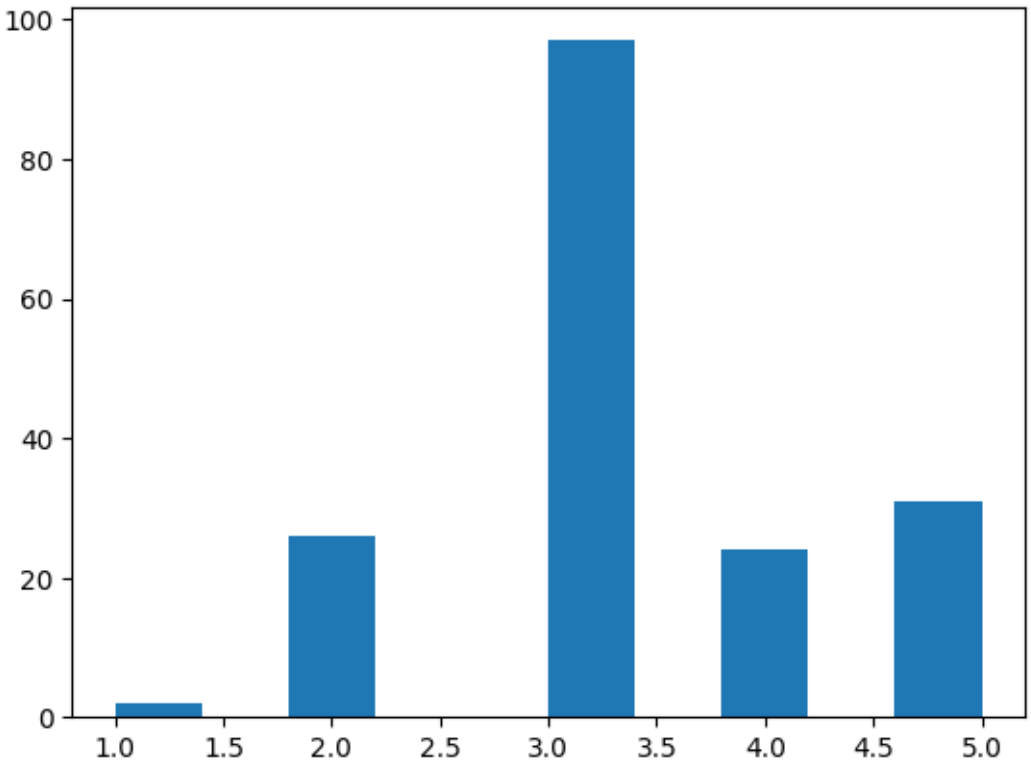

```
plt.hist(df["Product"])
plt.show()
```



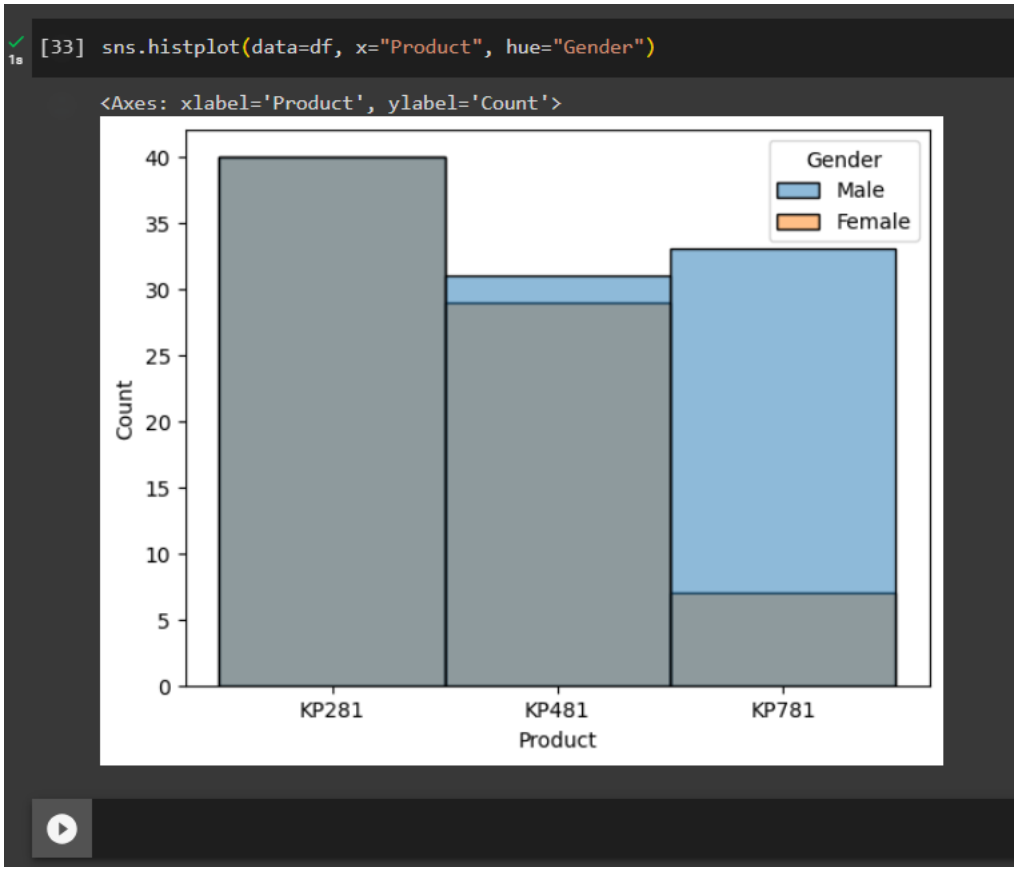

```
plt.hist(df["Usage"])
plt.show()
```




```
plt.hist(df["Fitness"])
plt.show()
```

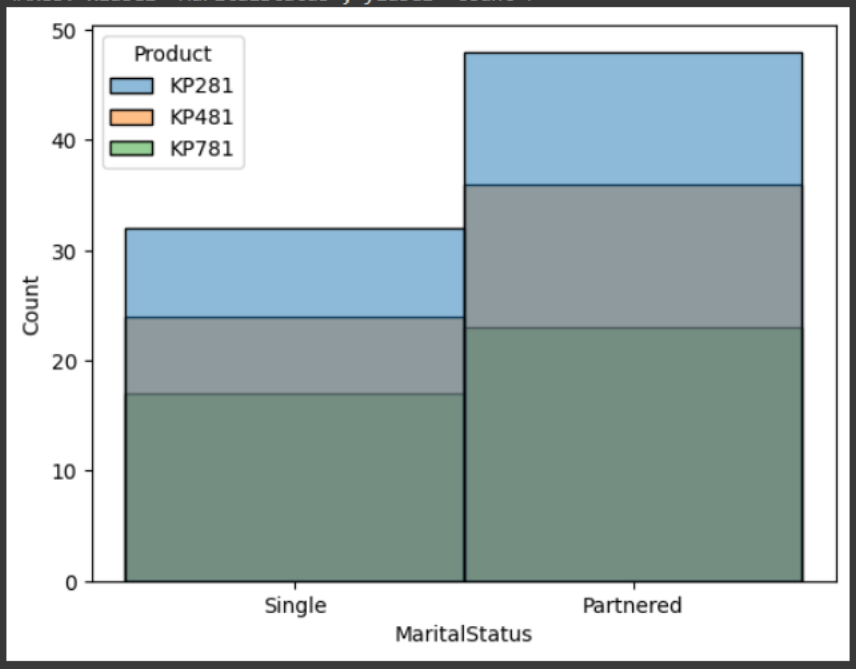


Bivariate analysis : It create a histogram for Product and Gender columns of the DataFrame using Seaborn.



```
[34] sns.histplot(data=df, x="MaritalStatus", hue="Product")
```

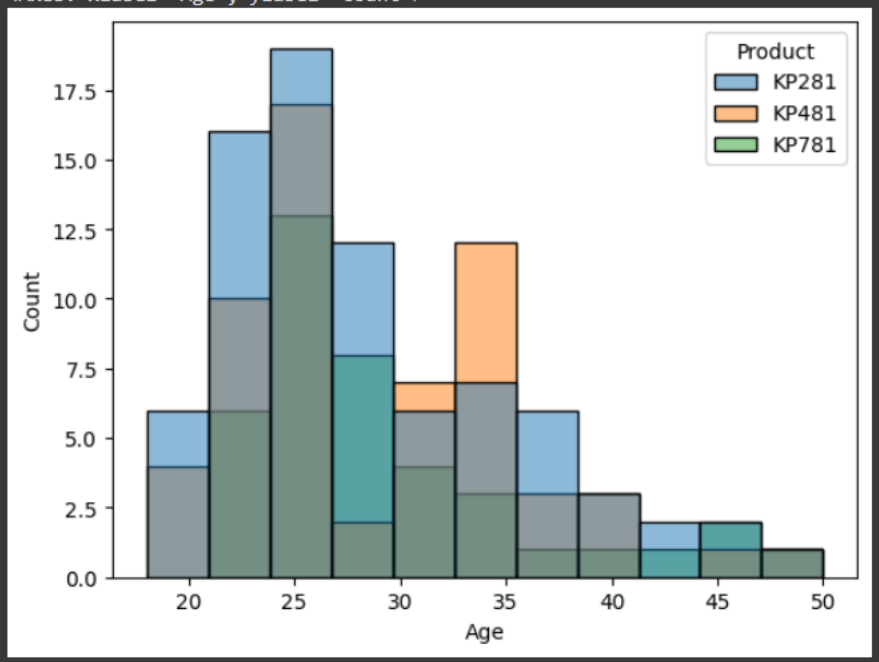
```
<Axes: xlabel='MaritalStatus', ylabel='Count'>
```



It create a histogram for MaritalStatus and Product columns of the DataFrame using Seaborn.

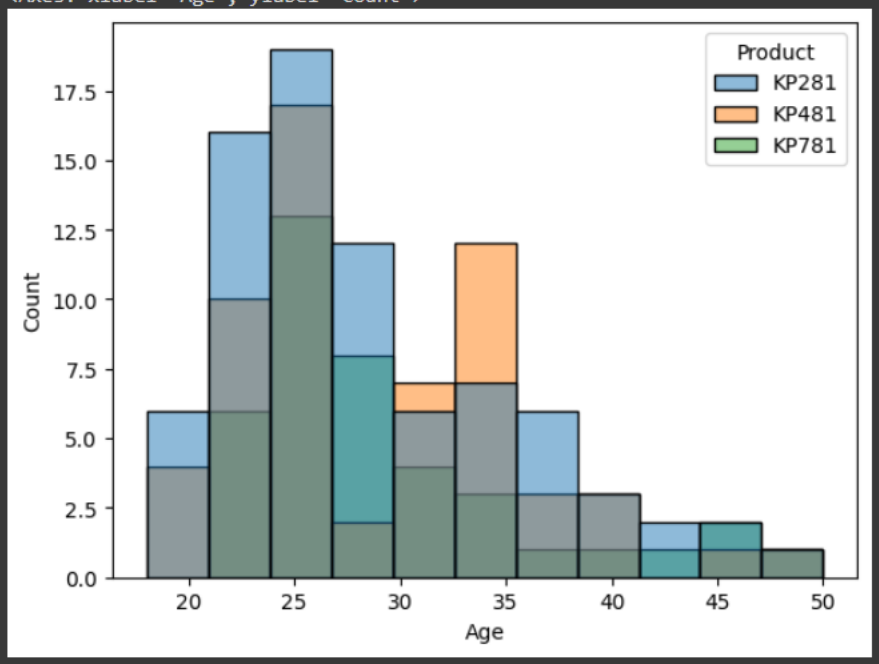
```
[35] sns.histplot(data=df, x="Age", hue="Product")
```

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



```
[35] sns.histplot(data=df, x="Age", hue="Product")
```

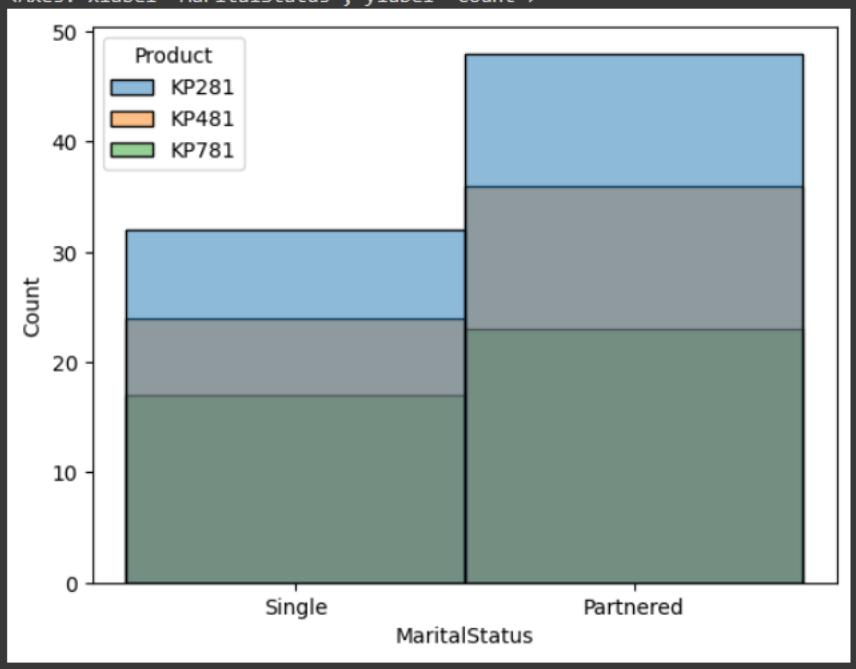
```
<Axes: xlabel='Age', ylabel='Count'>
```



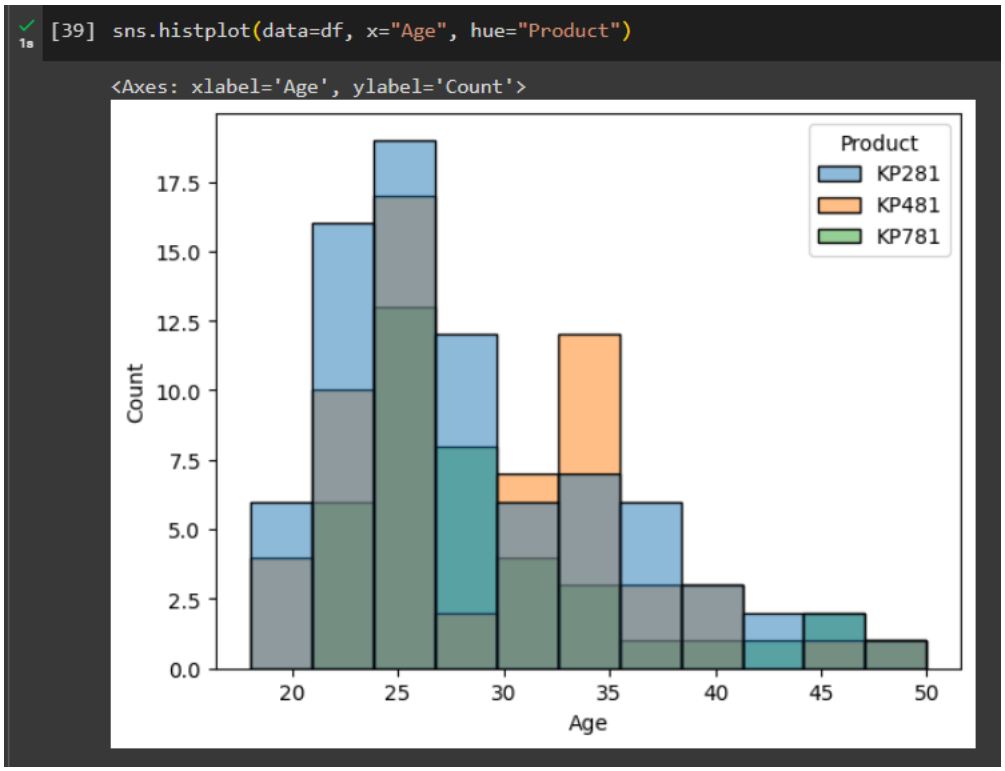
It create a histogram for Age and Product columns of the DataFrame using Seaborn

```
[38] sns.histplot(data=df, x="MaritalStatus", hue="Product")
```

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



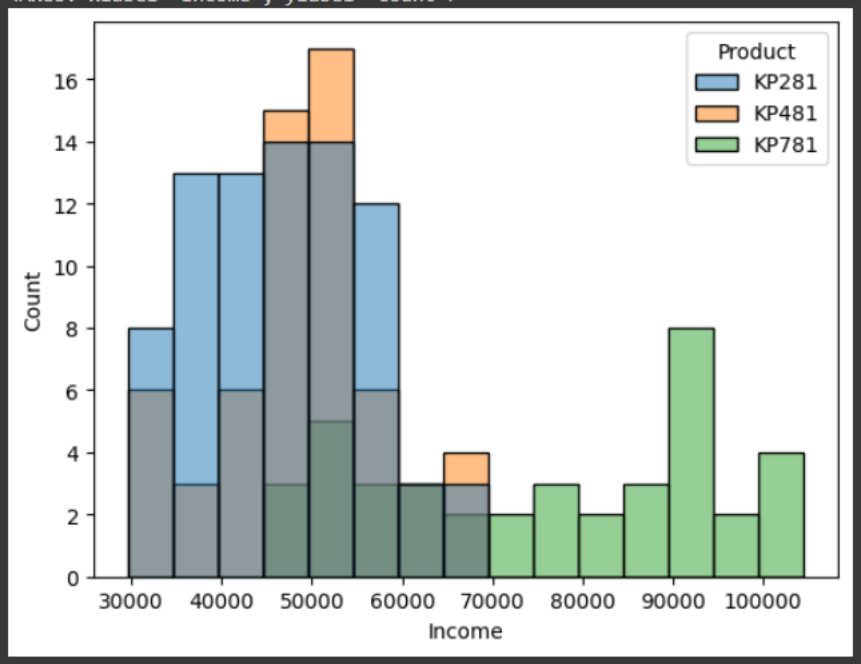
It create a histogram for MaritalStatus and Product columns of the DataFrame using Seaborn.



It create a histogram for Age and Product columns of the DataFrame using Seaborn

```
[40] sns.histplot(data=df, x="Income", hue="Product")
```

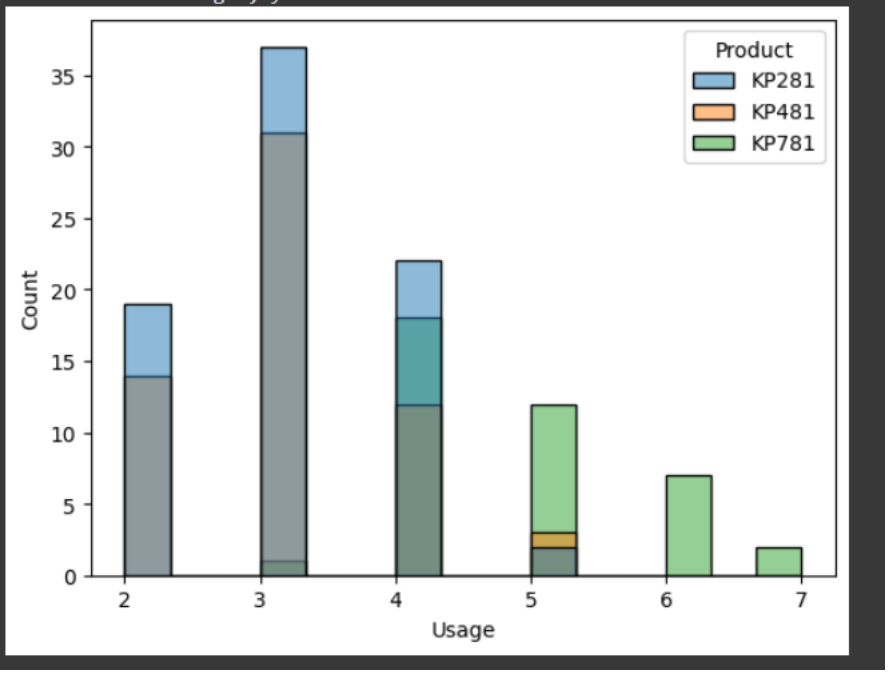
```
<Axes: xlabel='Income', ylabel='Count'>
```



It create a histogram for Income and Product columns of the DataFrame using Seaborn.

```
sns.histplot(data=df, x="Usage", hue="Product")
```

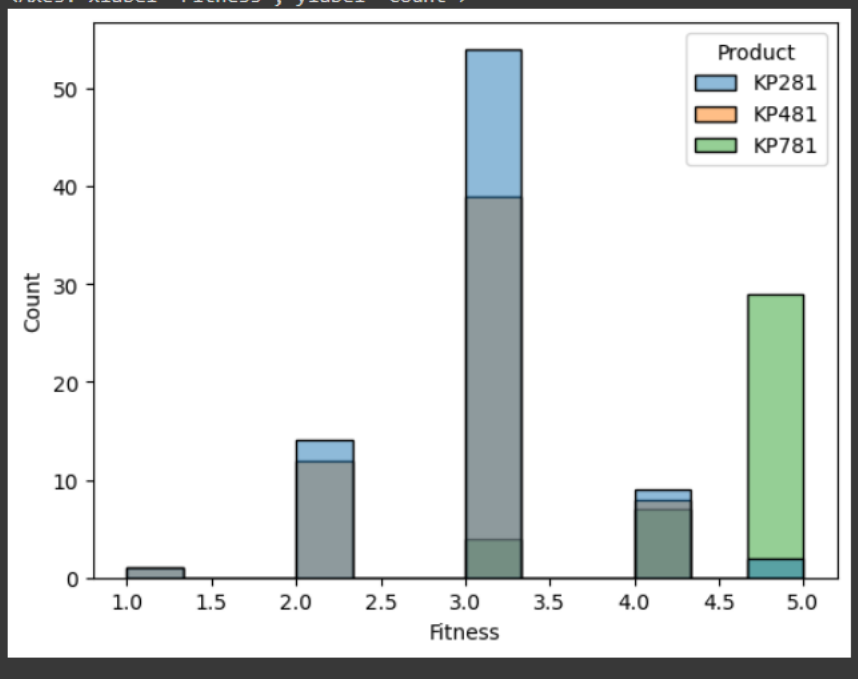
```
<Axes: xlabel='Usage', ylabel='Count'>
```



It create a histogram for Usage and Product columns of the DataFrame using Seaborn.

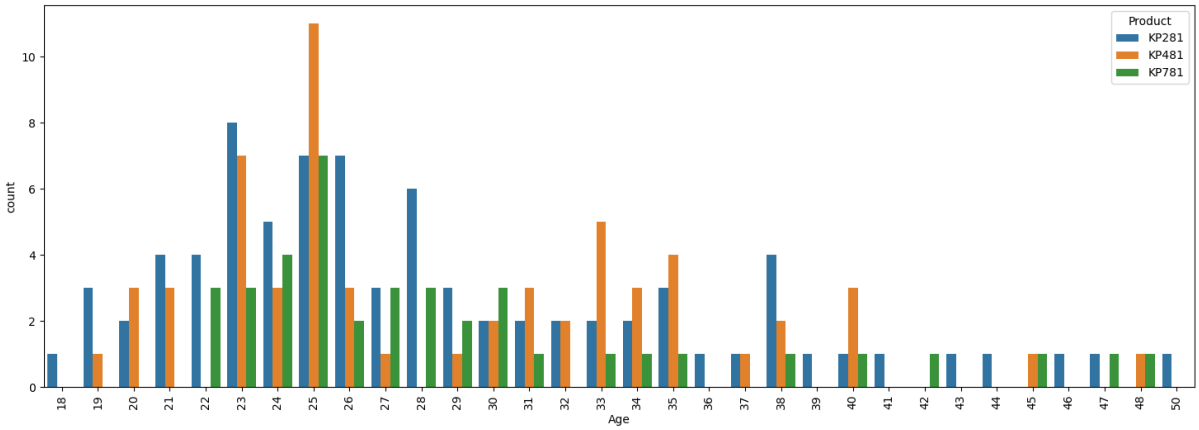
```
[42] sns.histplot(data=df, x="Fitness", hue="Product")
```

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



It create a histogram for Fitness and Product columns of the DataFrame using Seaborn.

```
plt.figure(figsize = (18,6))
sns.countplot(data=df, x='Age',hue="Product")
plt.xticks(rotation=90)
plt.show()
```



It create a Countplot for Fitness and Product columns of the DataFrame using Seaborn.

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

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

It creates a crosstab between Gender and Product using Pandas . A crosstab is used to ompute a simple cross tabulation of two (or more) factors.

```
pd.crosstab(df["Age"],df["Product"])
```

Age	KP281	KP481	KP781
18	1	0	0
19	3	1	0
20	2	3	0
21	4	3	0
22	4	0	3
23	8	7	3
24	5	3	4
25	7	11	7
26	7	3	2
27	3	1	3
28	6	0	3
29	3	1	2
30	2	2	3
31	2	3	1
32	2	2	0
33	2	5	1
34	2	3	1
35	3	4	1
36	1	0	0
37	1	1	0
38	4	2	1
39	1	0	0
40	1	3	1
41	1	0	0
42	0	0	1
43	1	0	0
44	1	0	0
45	0	1	1
46	1	0	0
47	1	0	1

```
#It creates a crosstab between MaritalStatus and Product using Pandas
pd.crosstab(df["MaritalStatus"],df["Product"])
```

MaritalStatus	KP281	KP481	KP781
Partnered	48	36	23
Single	32	24	17

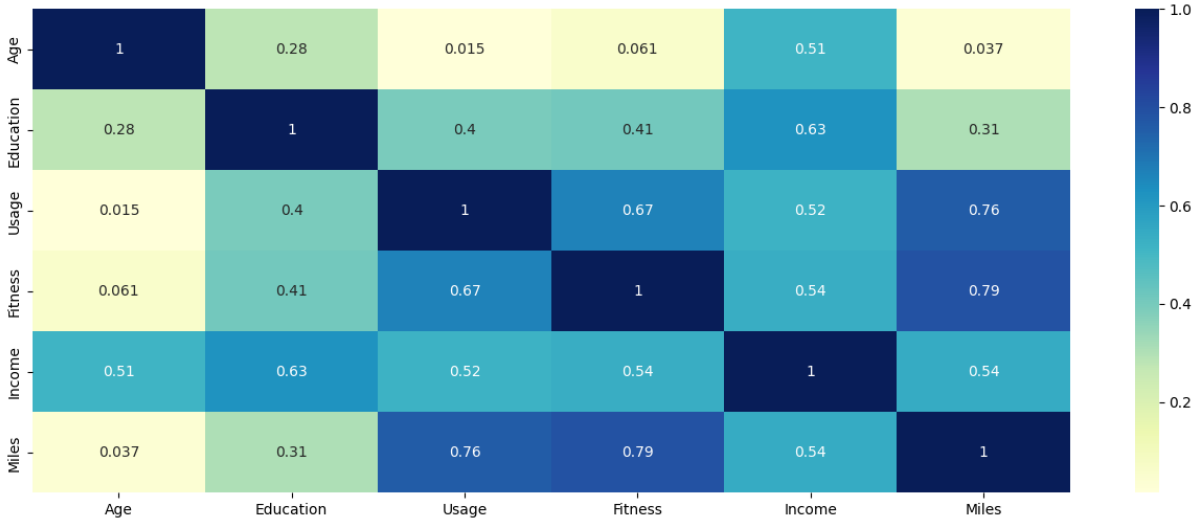
```
[ ]
```

```
#It creates a crosstab between Incomeand Product using Pandas
pd.crosstab(df["Income"],df["Product"])
```

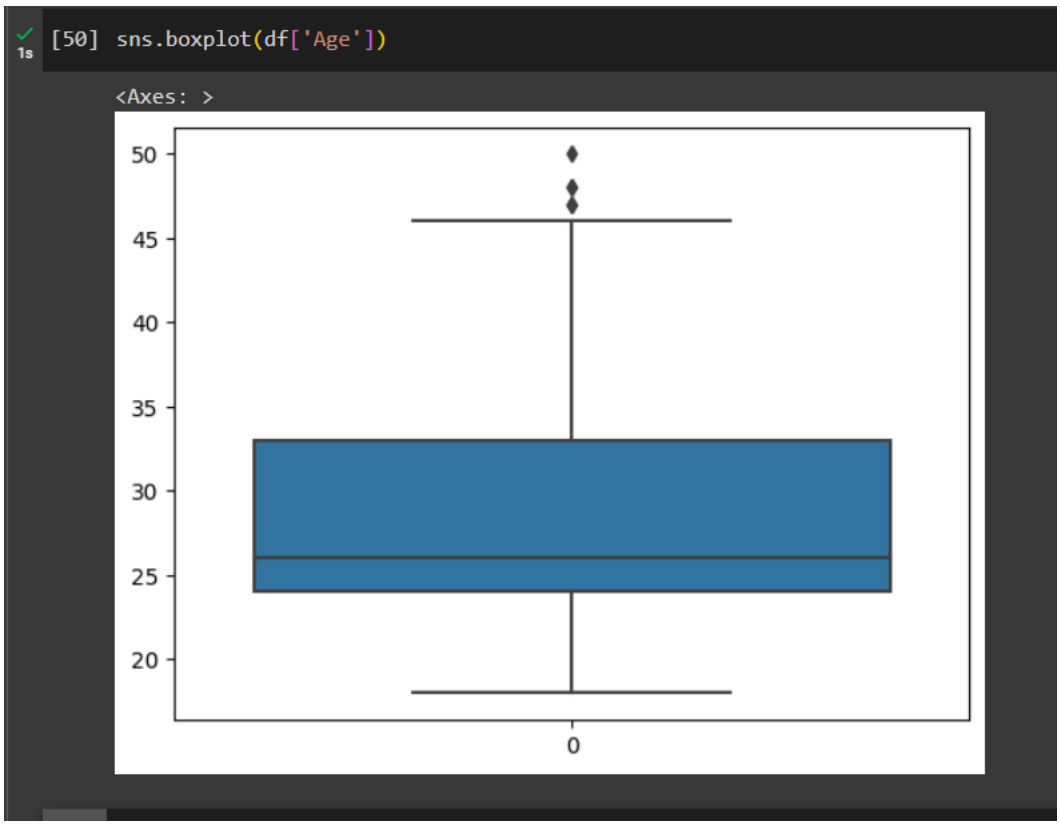
Income	KP281	KP481	KP781
29562	1	0	0
30699	1	0	0
31836	1	1	0
32973	3	2	0
34110	2	3	0
...	...	...	...
95508	0	0	1
95866	0	0	1
99601	0	0	1
103336	0	0	1
104581	0	0	2

62 rows × 3 columns

```
plt.figure(figsize=(16, 6))
sns.heatmap(df.corr(),cmap="YlGnBu", annot=True)
plt.show()
```



A heatmap is drawn using Searborn on all continuous values. Using a heatmap to visualise a confusion matrix, time-series movements, temperature changes, correlation matrix and SHAP interaction values.

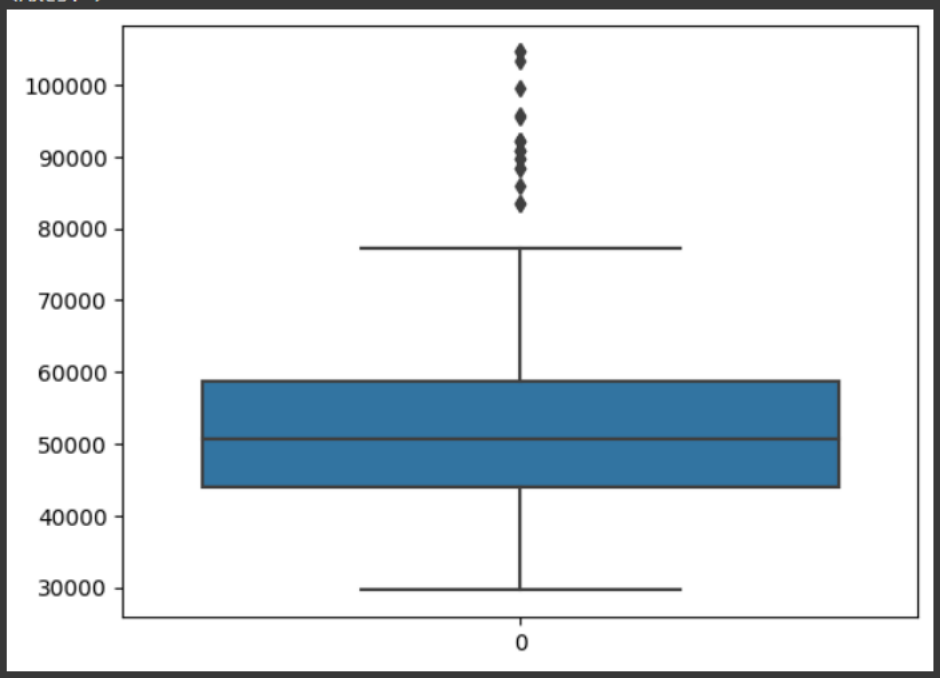


```
np.where(df['Age']>46)
```

```
(array([ 78,  79, 139, 178, 179]),)
```

```
[52] sns.boxplot(df['Income'])
```

```
<Axes: >
```

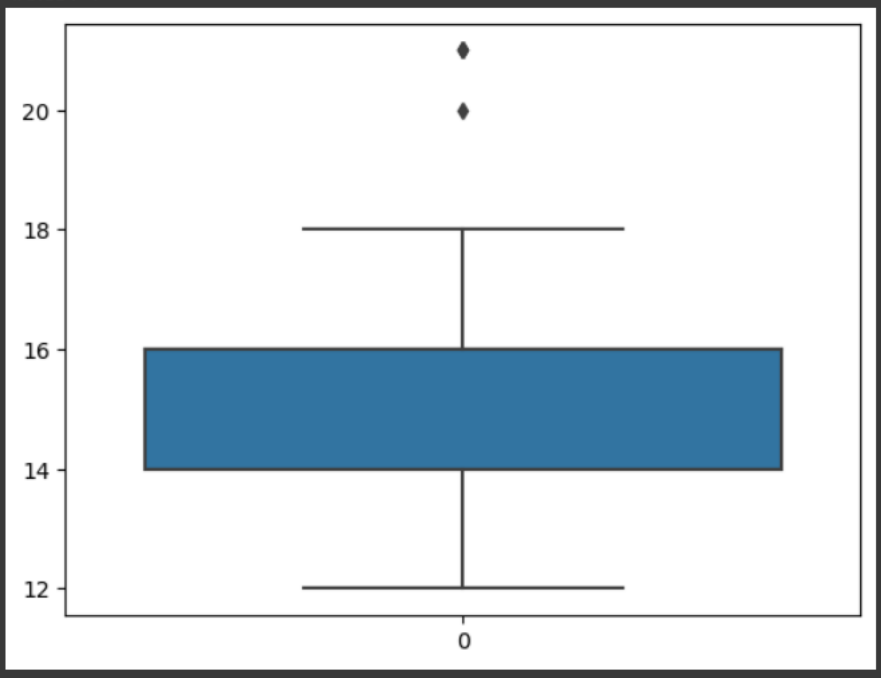


```
[53] np.where(df['Income']>80000)
```

```
(array([159, 160, 161, 162, 164, 166, 167, 168, 169, 170, 171, 172, 173,
        174, 175, 176, 177, 178, 179]),)
```

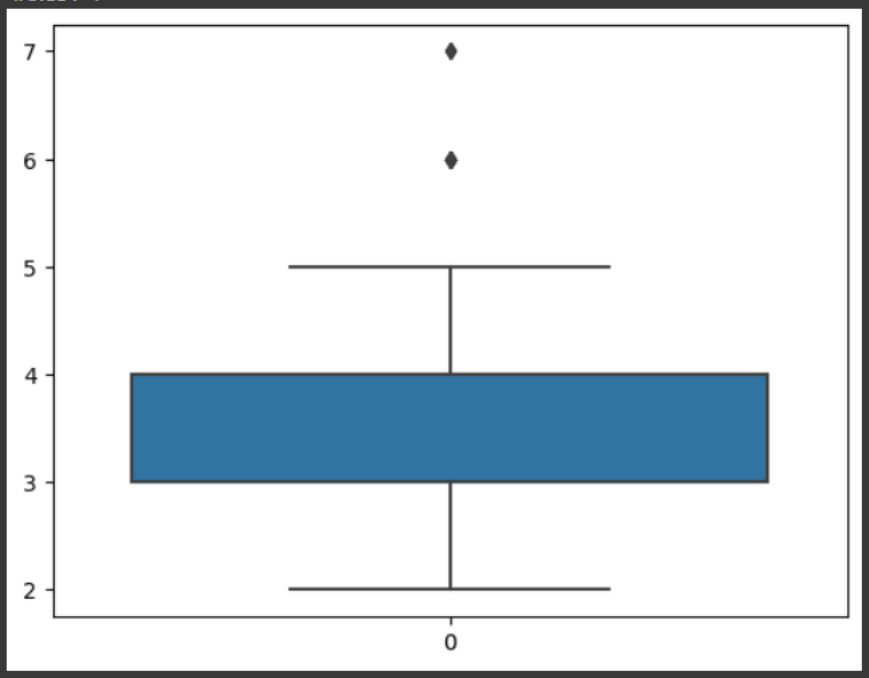
```
[54] sns.boxplot(df['Education'])
```

```
<Axes: >
```



```
sns.boxplot(df['Usage'])
```

```
<Axes: >
```



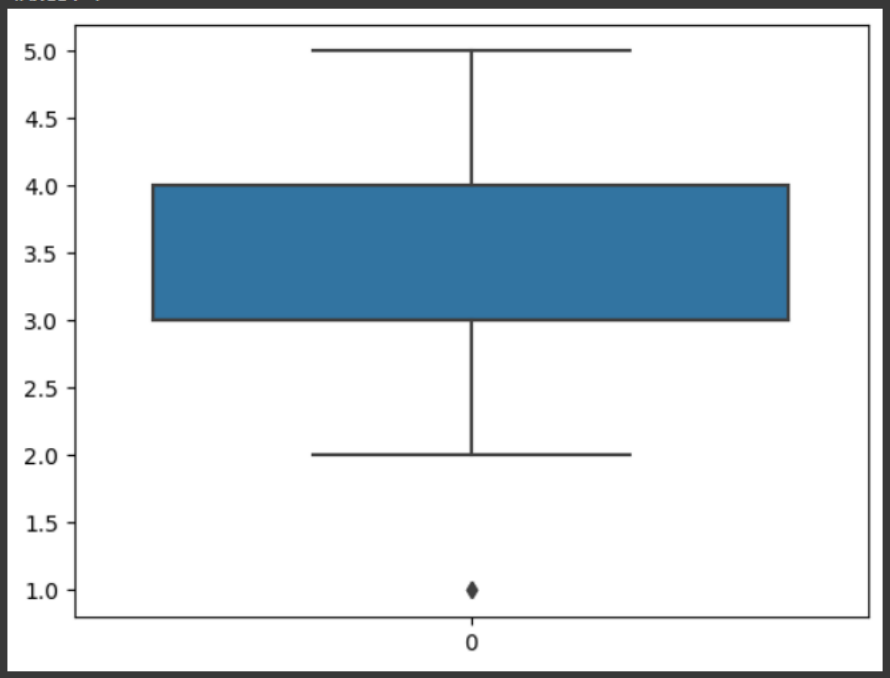

```
[ ]
```

```
[57] np.where(df["Usage"]>5)
```

```
(array([154, 155, 162, 163, 164, 166, 167, 170, 175]),)
```

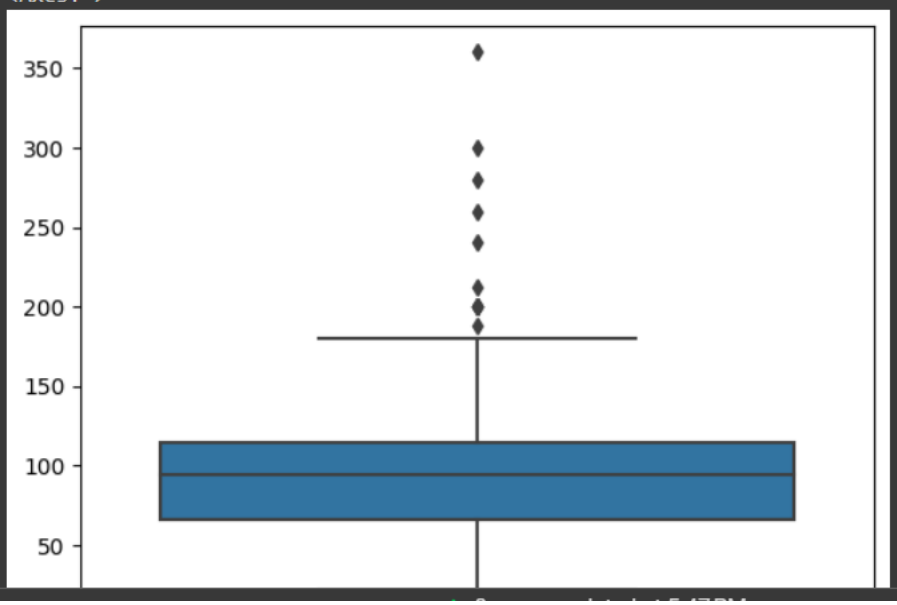
```
[58] sns.boxplot(df['Fitness'])
```

```
<Axes: >
```



```
[59] np.where(df["Fitness"]<2)

     (array([ 14, 117]),)
```

```
[60] sns.boxplot(df['Miles'])

     <Axes: >
```




```
[61] np.where(df["Miles"]>180)

     (array([ 23,  84, 142, 148, 152, 155, 166, 167, 170, 171, 173, 175, 176]),)
```

- **Since most of the customers who bought KP781 are male we can say that it is best suited for male not for female**
- **Cutomers who are in age between 25-30 are buying KP781 treadmill more, so it is adviced that people belong to 30+ and below 25 are not recommended to buy this treadmill.**
- **Customers who are less educated shouldn't buy KP781 treadmill.**
- **Customers who are not using treadmill less than 4 times a week shouldn't buy this treadmill.**
- **Customers with fitness less than 3 shouldn't buy KP781 treadmill.**
- **Customers who don't walk/run greater than 120 miles per week shouldn't nuy KP781 treadmill.**
- **There are no missing values in the data.**
- **There are 3 unique products in the dataset.**
- **KP281 is the most frequent product.**
- **Minimum & Maximum age of the person is 18 & 50, mean is 28.79 and 75% of persons have age less than or equal to 33.**
- **Most of the people are having 16 years of education i.e. 75% of persons are having education <= 16 years.**
- **Out of 180 data points, 104's gender is Male and rest are the female.**
- **Standard deviation for Income & Miles is very high. These variables might have the outliers in it.**

- **Females planning to use treadmill 3-4 times a week, are more likely to buy KP481 product**

4. Missing Value & Outlier Detection (10 Points)

```
[65] df.isna()
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
175	False	False	False	False	False	False	False	False	False
176	False	False	False	False	False	False	False	False	False
177	False	False	False	False	False	False	False	False	False
178	False	False	False	False	False	False	False	False	False
179	False	False	False	False	False	False	False	False	False

180 rows × 9 columns

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

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

```
df[df['Miles']>225]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
155	KP781	25	Male	18	Partnered	6	5	75946	240
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
170	KP781	31	Male	16	Partnered	6	5	89641	260
173	KP781	35	Male	16	Partnered	4	5	92131	360

```
[ ]
```

```
[68] df[df['Income']<30000]
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112

```
#Removing outliers
df= df[ ~(df['Miles']>225) ]
df = df[ ~(df['Income']<30000) ]
df.reset_index(drop=True, inplace=True)
df
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	19	Male	15	Single	2	3	31836	75
1	KP281	19	Female	14	Partnered	4	3	30699	66
2	KP281	19	Male	12	Single	3	3	32973	85
3	KP281	20	Male	13	Partnered	4	2	35247	47
4	KP281	20	Female	14	Partnered	3	3	32973	66
...	...	...	...	...	...	...	...	...	...
169	KP781	40	Male	21	Single	6	5	83416	200
170	KP781	42	Male	18	Single	5	4	89641	200
171	KP781	45	Male	16	Single	5	5	90886	160
172	KP781	47	Male	18	Partnered	4	5	104581	120
173	KP781	48	Male	18	Partnered	4	5	95508	180

174 rows × 9 columns

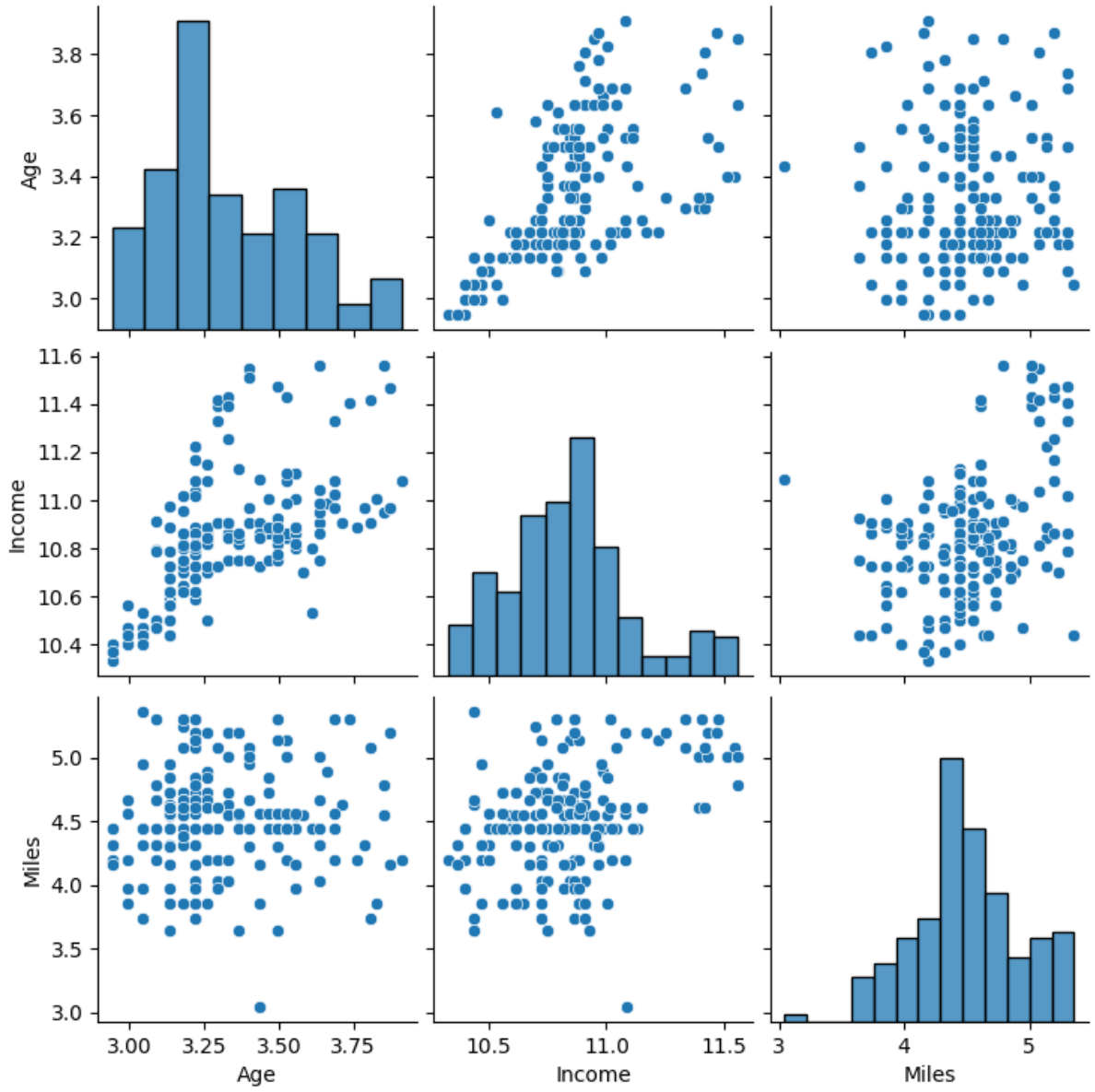
5. Business Insights based on Non-Graphical and Visual Analysis (10 Points)

1. Comments on the range of attributes
2. Comments on the distribution of the variables and relationship between them
3. Comments for each univariate and bivariate plot

```
plt.figure(figsize=(20,10))
sns.pairplot(df)
plt.show()
```

```
transformed_df = pd.DataFrame()
```

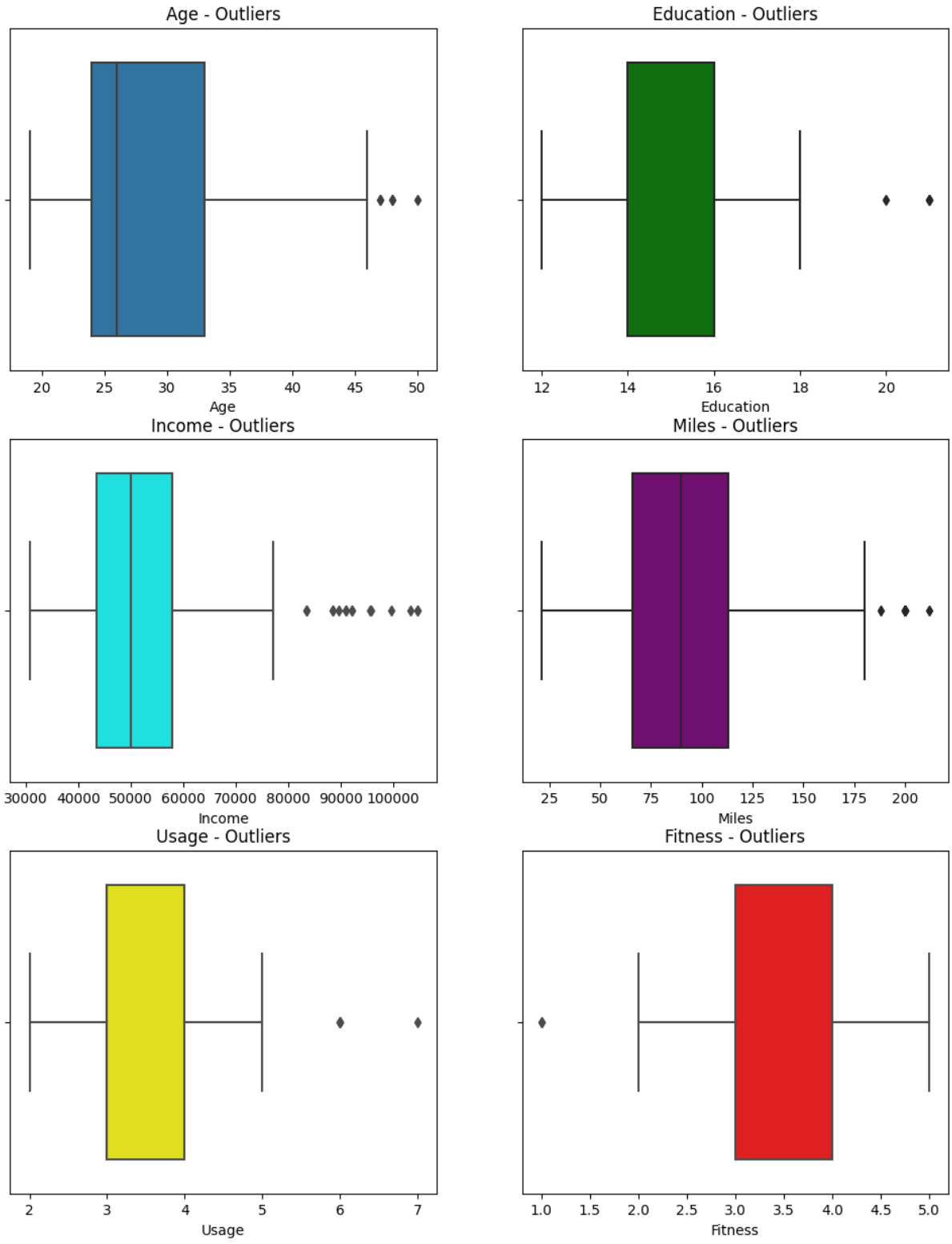
```
transformed_df['Age'] = np.log(df['Age'])
transformed_df['Income'] = np.log(df['Income'])
transformed_df['Miles'] = np.log(df['Miles'])

plt.figure(figsize=(20,10))
sns.pairplot(transformed_df)
plt.show()
```



```
df.describe().loc[['mean','50%']]
```

	Age	Education	Usage	Fitness	Income	Miles
mean	28.816092	15.568966	3.390805	3.258621	52904.896552	97.833333
50%	26.000000	16.000000	3.000000	3.000000	50028.000000	89.500000

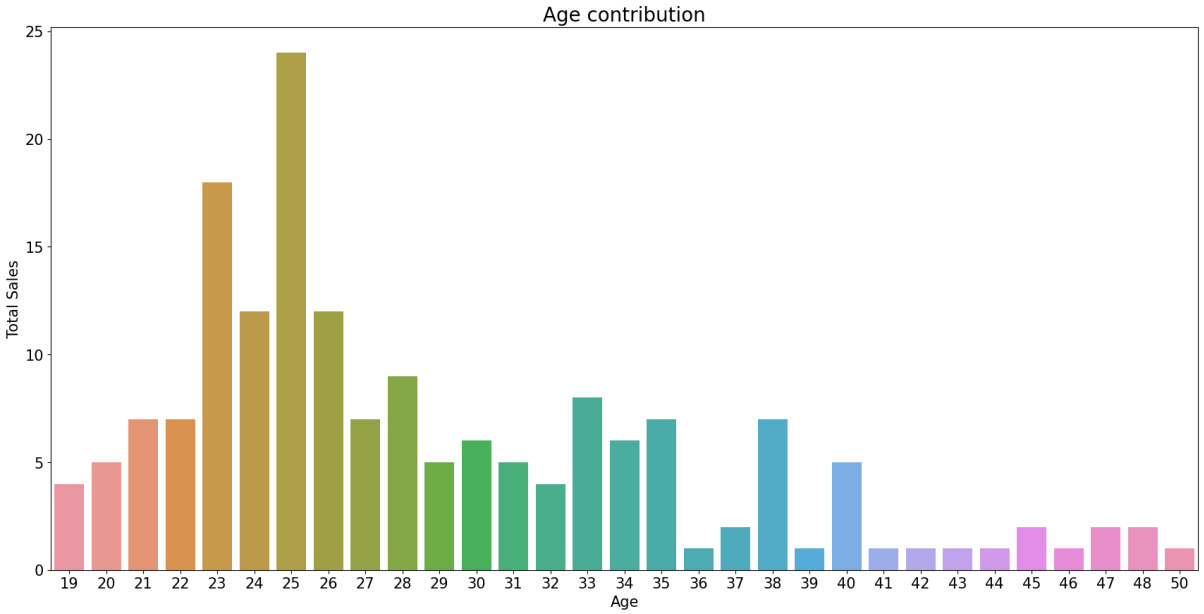
```
fig, ax = plt.subplots(nrows=3, ncols=2, figsize=(12,15))
sns.boxplot(data=df, x='Age',
ax=ax[0,0]);                        ax[0,0].set_title('Age - Outliers')
sns.boxplot(data=df, x='Education', ax=ax[0,1], color='green');
ax[0,1].set_title('Education - Outliers')
sns.boxplot(data=df, x='Income', ax=ax[1,0],
color='cyan');      ax[1,0].set_title('Income - Outliers')
sns.boxplot(data=df, x='Miles', ax=ax[1,1],
color='purple');    ax[1,1].set_title('Miles - Outliers')
sns.boxplot(data=df, x='Usage', ax=ax[2,0],
color='yellow');    ax[2,0].set_title('Usage - Outliers')
sns.boxplot(data=df, x='Fitness', ax=ax[2,1],
color='red');      ax[2,1].set_title('Fitness - Outliers')
```



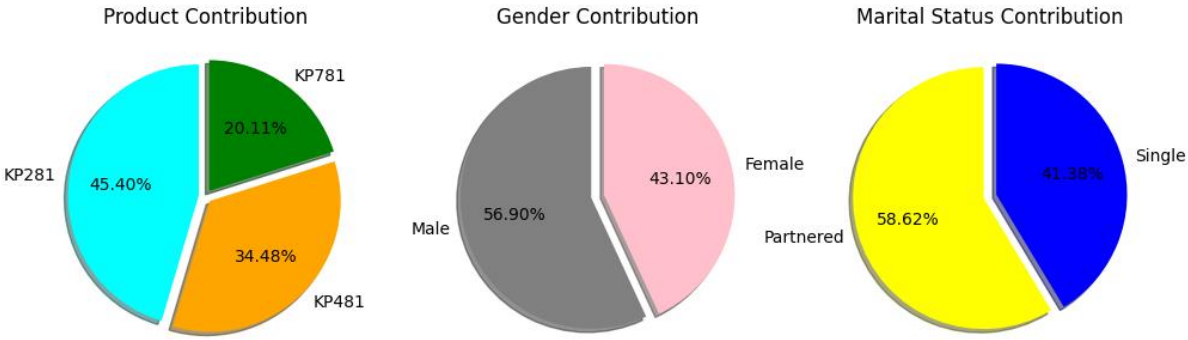
5.Business Insights based on Non-Graphical and Visual Analysis (10 Points)

1. **Comments on the range of attributes**
2. **Comments on the distribution of the variables and relationship between them**
3. **Comments for each univariate and bivariate plot**

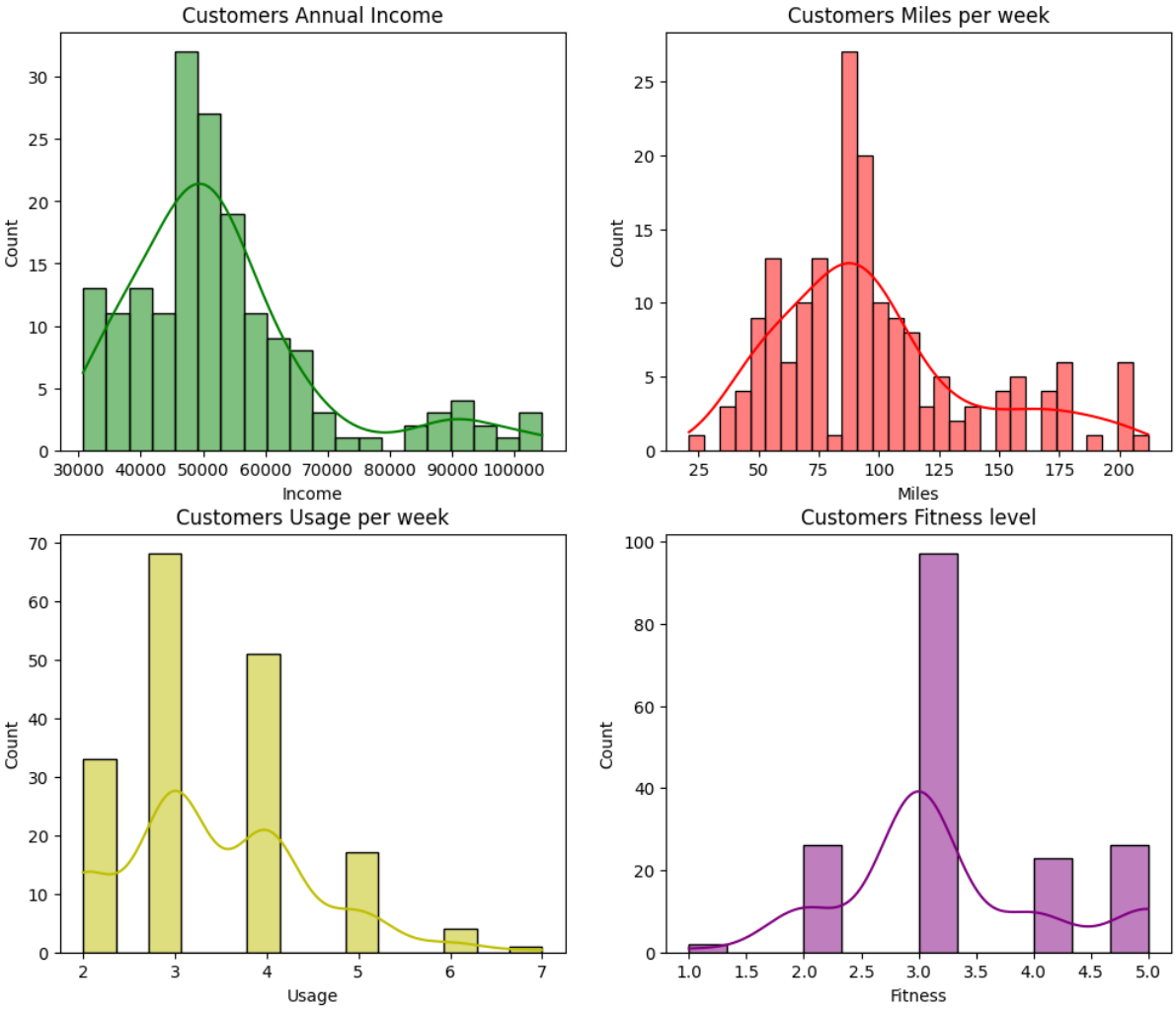
```
2. plt.figure(figsize=(21,10))
3. sns.countplot(data=df, x='Age')
4. plt.ylabel('Total Sales', fontsize=15); plt.title('Age
   contribution', fontsize=20); plt.xlabel("Age", fontsize=15)
5. plt.xticks(fontsize=15); plt.yticks(fontsize=15); plt.show()
```




```
fig= plt.figure(figsize=(12,5))
a1 = fig.add_subplot(131)
a1.pie(x=df['Product'].value_counts(),
        startangle=90, shadow=True, explode=[0.05,0.05,0.05],
        autopct='%1.2f%%', colors=['cyan','orange','green'],
labels=df['Product'].value_counts().index)
a1.set_title('Product Contribution')
a2= fig.add_subplot(132)
a2.pie(x=df['Gender'].value_counts(),
        startangle=90, shadow=True, explode=[0.05,0.05],
        autopct='%1.2f%%', colors=['Grey','Pink'],
labels=df['Gender'].value_counts().index)
a2.set_title('Gender Contribution')
a2= fig.add_subplot(133)
a2.pie(x=df['MaritalStatus'].value_counts(),
        startangle=90, shadow=True, explode=[0.05,0.05],
        autopct='%1.2f%%', colors=['yellow','blue'],
labels=df['MaritalStatus'].value_counts().index)
a2.set_title('Marital Status Contribution')
plt.show()
```

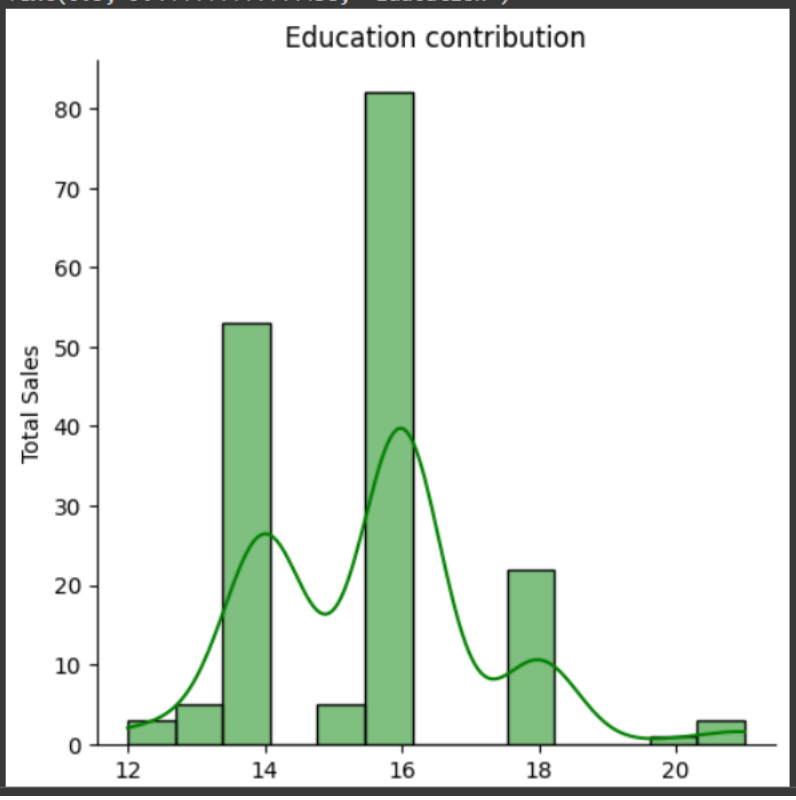



```
fig,ax = plt.subplots(nrows=2, ncols=2, figsize=(12,10))
sns.histplot(df['Income'], kde=True, bins=20, ax=ax[0,0], color='g');
ax[0,0].set_title("Customers Annual Income")
sns.histplot(df['Miles'], kde=True, bins=30, ax=ax[0,1], color='r');
ax[0,1].set_title("Customers Miles per week")
sns.histplot(df['Usage'], kde=True, ax=ax[1,0], color='y');
ax[1,0].set_title("Customers Usage per week")
sns.histplot(df['Fitness'], kde=True, ax=ax[1,1], color='purple');
ax[1,1].set_title("Customers Fitness level")
```



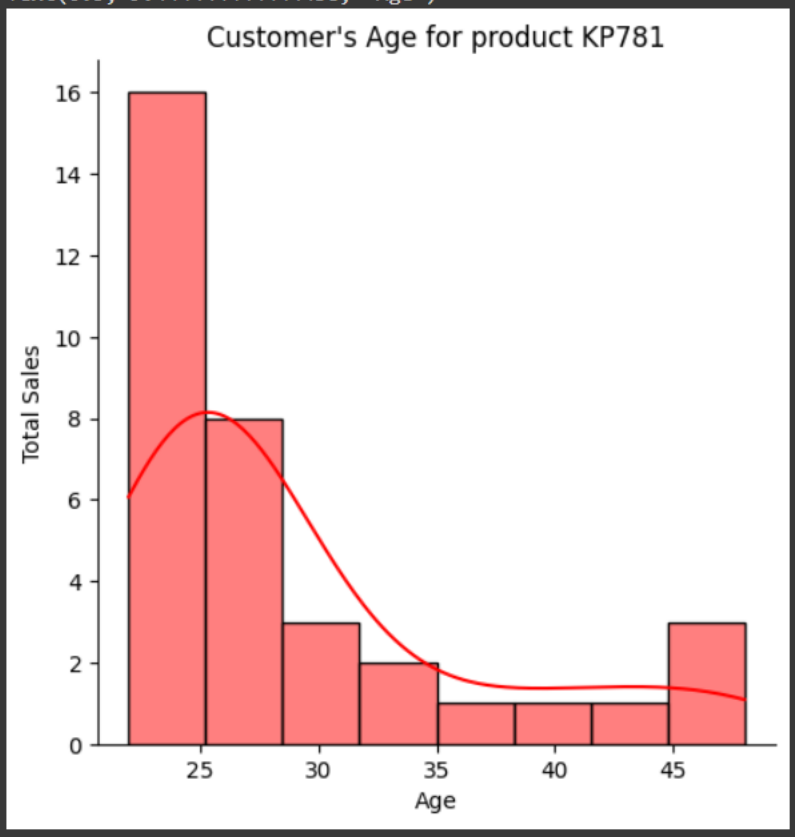
```
sns.displot(x=df['Education'], kde=True, color='g')
plt.ylabel('Total Sales'); plt.title('Education contribution'); plt.xlabel("Education")
```

```
Text(0.5, 9.444444444444438, 'Education')
```

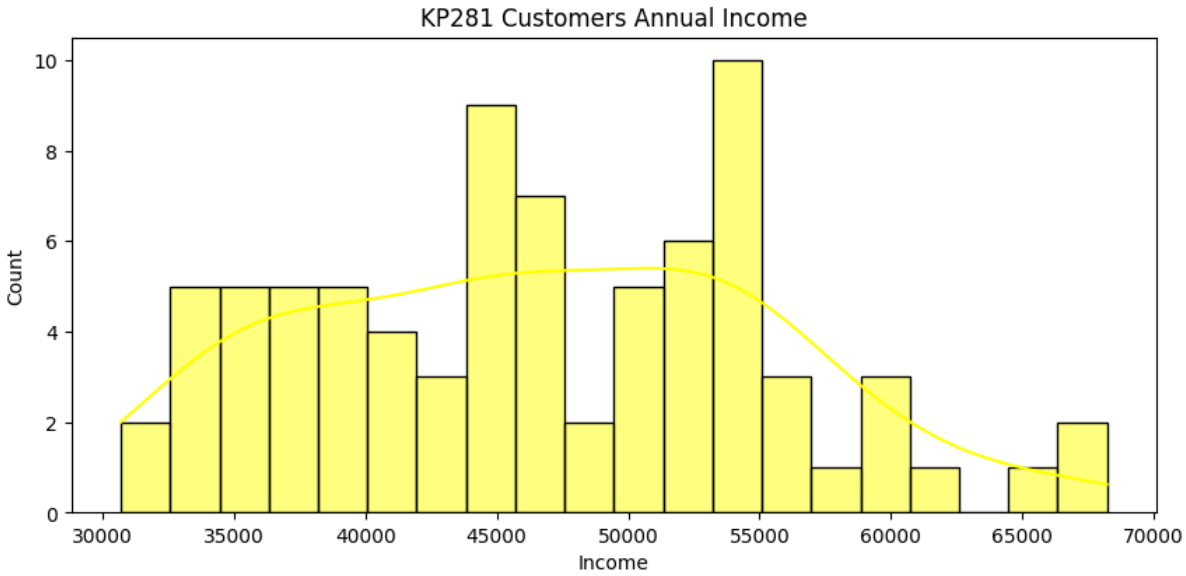


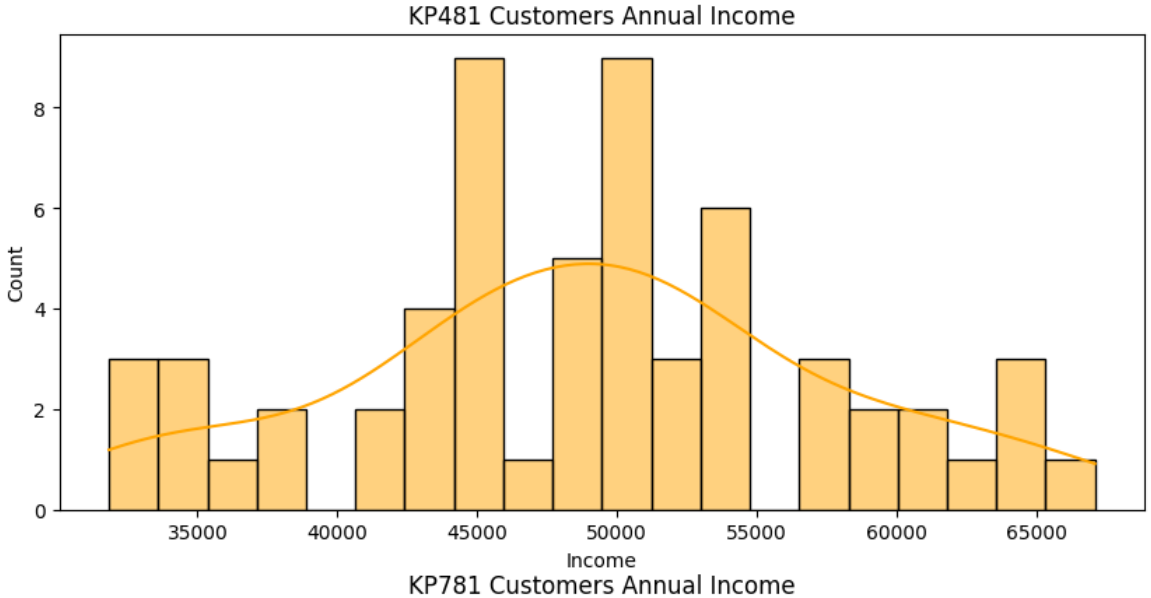
```
[82] sns.displot(x=df[df['Product']=='KP781']['Age'], kde=True, color='r')
     plt.ylabel('Total Sales'); plt.title("Customer's Age for product KP781"); plt.xlabel("Age")
```

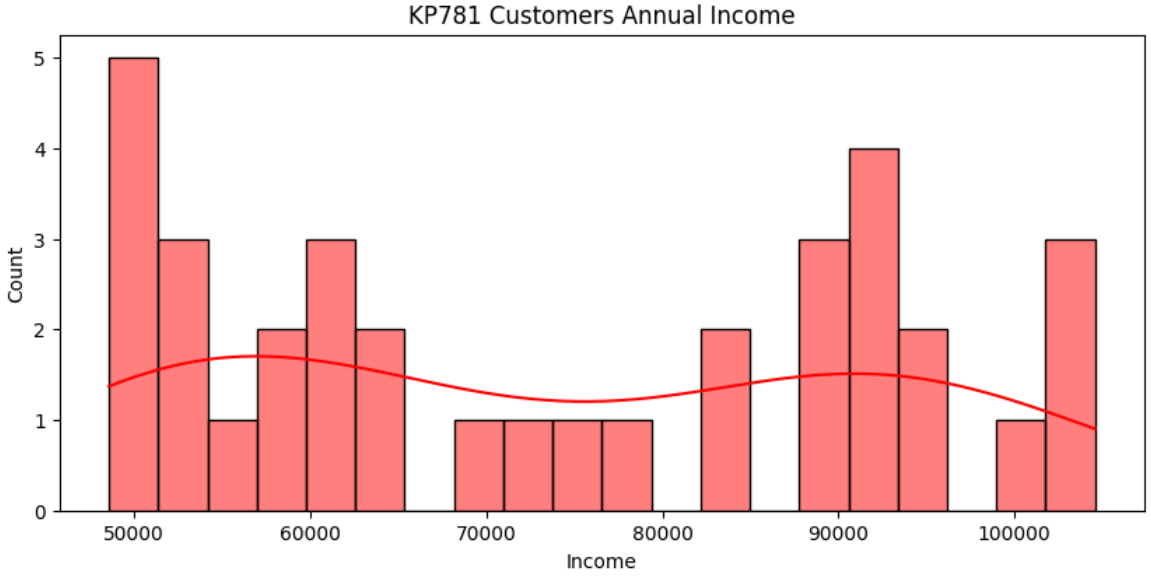
```
Text(0.5, 9.444444444444438, 'Age')
```




```
p1 = df[df['Product']=='KP281']
p2 = df[df['Product']=='KP481']
p3 = df[df['Product']=='KP781']
fig,ax = plt.subplots(nrows=3, ncols=1, figsize=(10,15))
sns.histplot(p1['Income'], kde=True, bins=20, ax=ax[0],
color='yellow'); ax[0].set_title("KP281 Customers Annual Income")
sns.histplot(p2['Income'], kde=True, bins=20, ax=ax[1],
color='orange'); ax[1].set_title("KP481 Customers Annual Income")
sns.histplot(p3['Income'], kde=True, bins=20, ax=ax[2], color='red');
ax[2].set_title("KP781 Customers Annual Income")
```

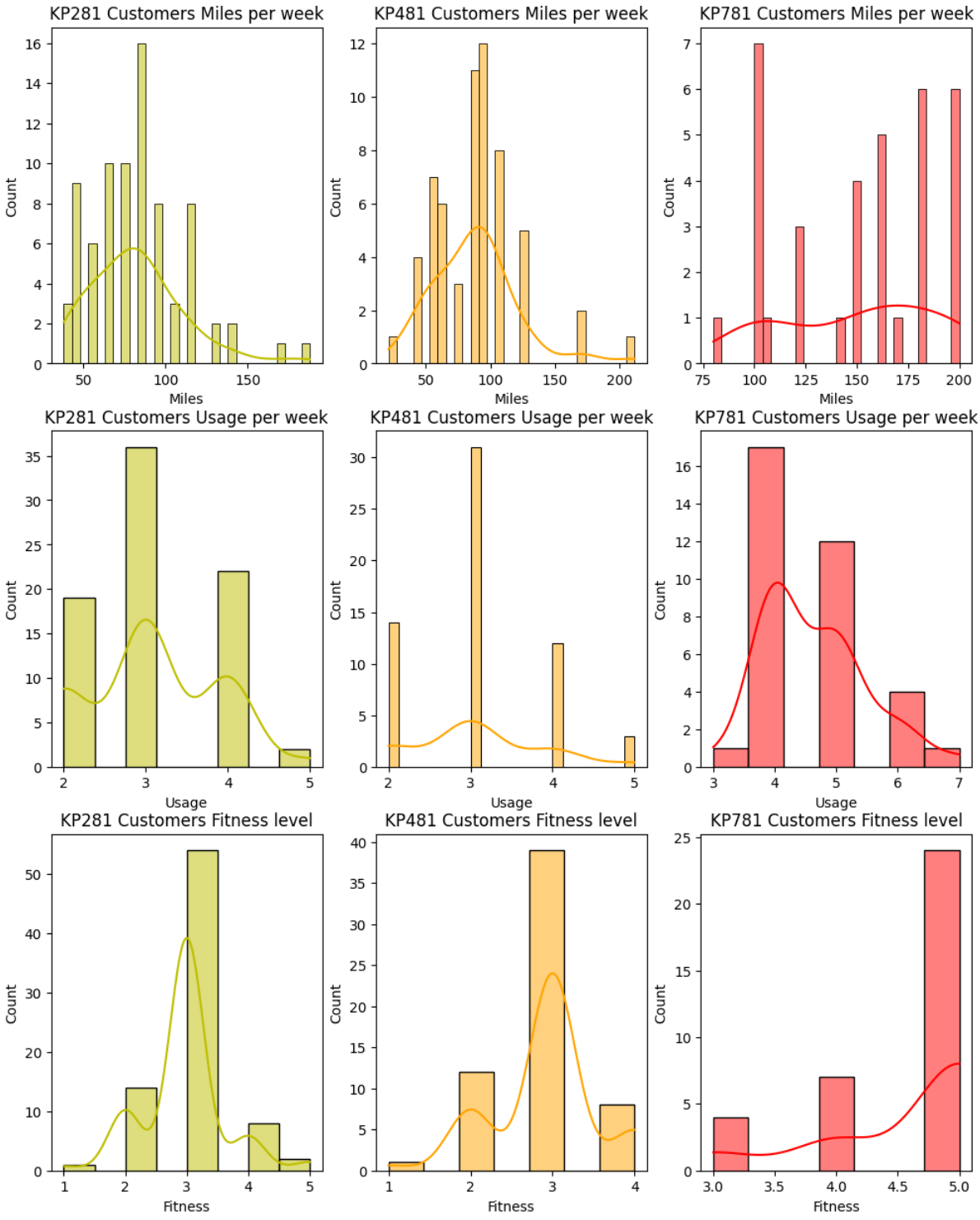




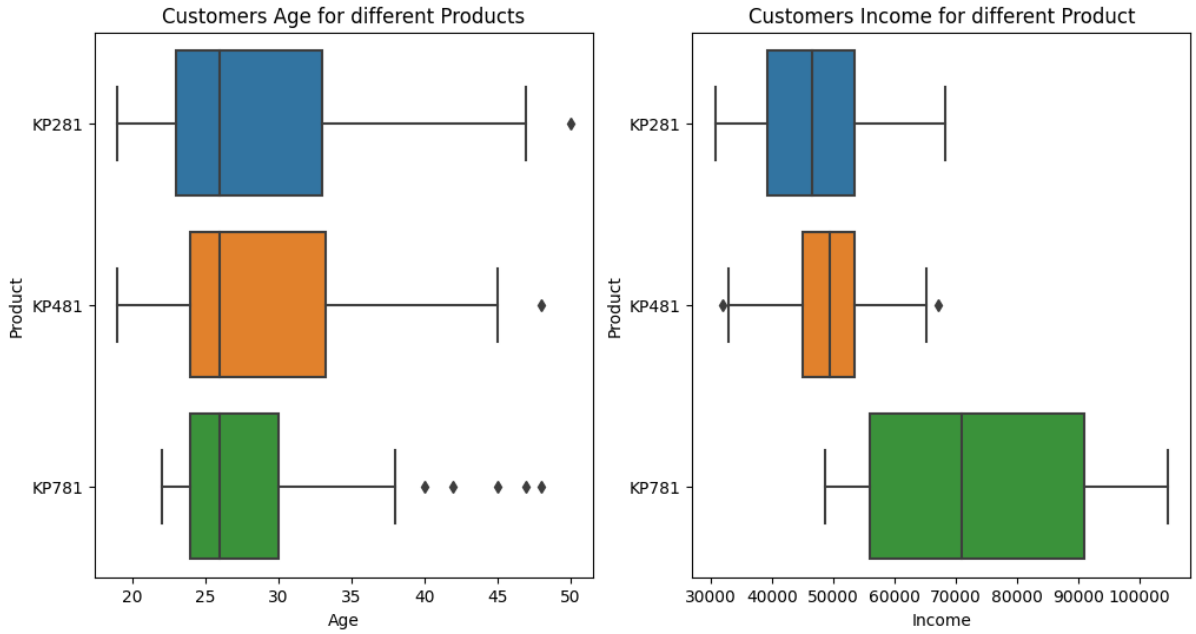



```
fig,ax = plt.subplots(nrows=3, ncols=3, figsize=(12,15))
```

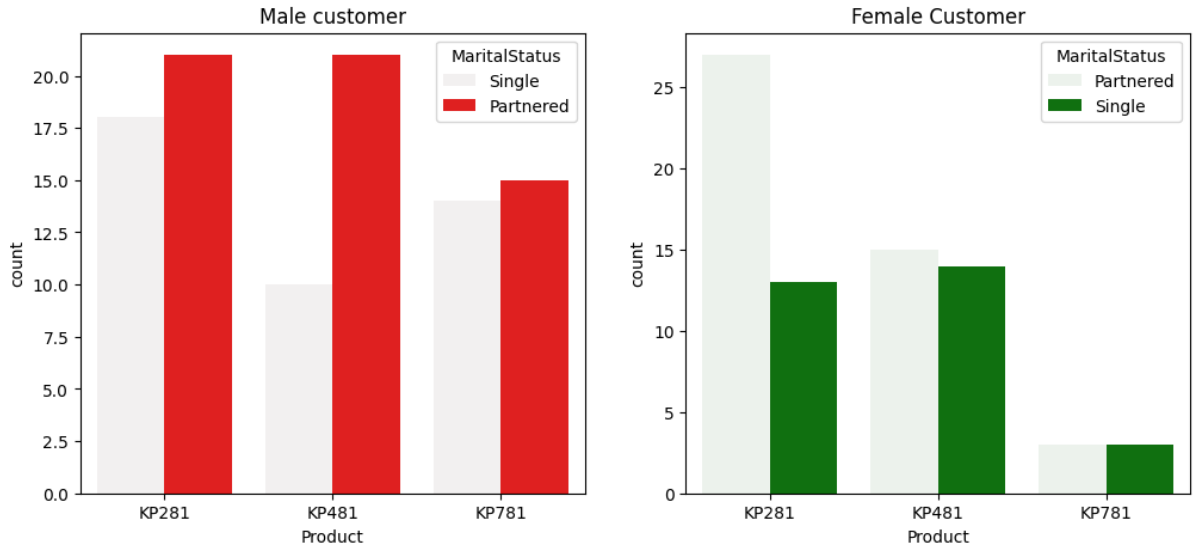
```
sns.histplot(p1['Miles'], kde=True, bins=30, ax=ax[0,0], color='y');
ax[0,0].set_title("KP281 Customers Miles per week")
sns.histplot(p2['Miles'], kde=True, bins=30, ax=ax[0,1],
color='orange'); ax[0,1].set_title("KP481 Customers Miles per week")
sns.histplot(p3['Miles'], kde=True, bins=30, ax=ax[0,2], color='r');
ax[0,2].set_title("KP781 Customers Miles per week")

sns.histplot(p1['Usage'], kde=True, ax=ax[1,0], color='y');
ax[1,0].set_title("KP281 Customers Usage per week")
sns.histplot(p2['Usage'], kde=True, ax=ax[1,1], color='orange');
ax[1,1].set_title("KP481 Customers Usage per week")
sns.histplot(p3['Usage'], kde=True, ax=ax[1,2], color='r');
ax[1,2].set_title("KP781 Customers Usage per week")

sns.histplot(p1['Fitness'], kde=True, ax=ax[2,0], color='y');
ax[2,0].set_title("KP281 Customers Fitness level")
sns.histplot(p2['Fitness'], kde=True, ax=ax[2,1], color='orange');
ax[2,1].set_title("KP481 Customers Fitness level")
sns.histplot(p3['Fitness'], kde=True, ax=ax[2,2], color='r');
ax[2,2].set_title("KP781 Customers Fitness level")
```




```
fig,ax = plt.subplots(nrows=1, ncols=2, figsize=(12,6))
sns.boxplot(data=df, x='Age', y='Product', ax=ax[0]);
ax[0].set_title('Customers Age for different Products')
sns.boxplot(data=df, x='Income', y='Product', ax=ax[1]);
ax[1].set_title('Customers Income for different Product')
```

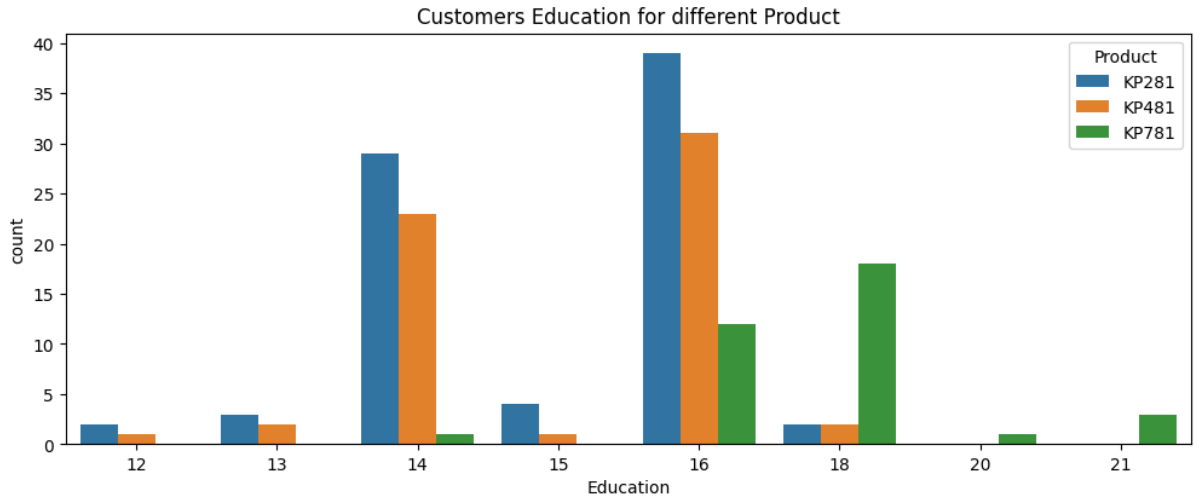


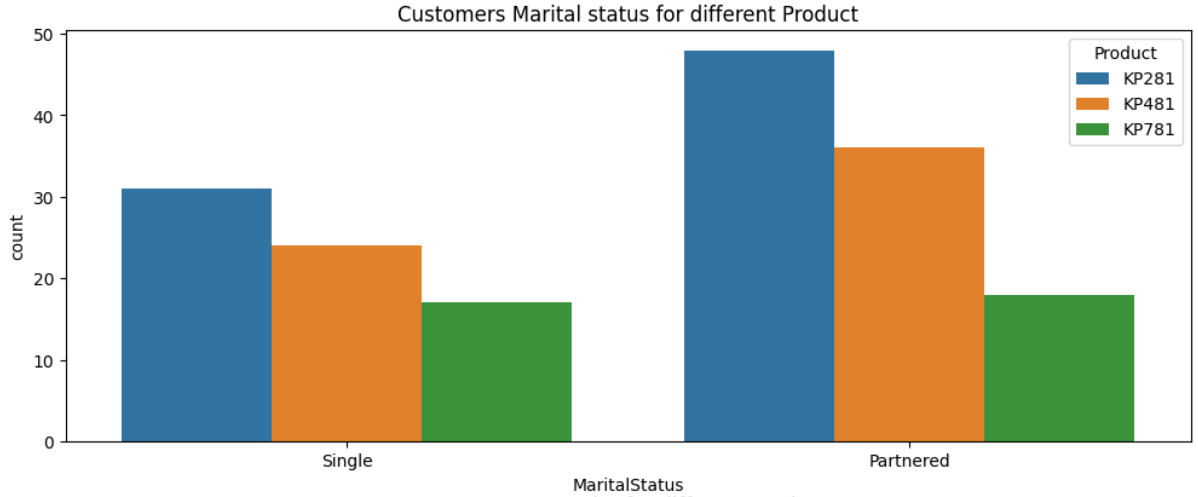

```
dff=df[df['Gender']=='Female']
dfm=df[df['Gender']=='Male']
fig,ax=plt.subplots(nrows=1, ncols=2, figsize=(12,5))
sns.countplot(data=dfm, x='Product',
hue='MaritalStatus',ax=ax[0],color='r'); ax[0].set_title('Male
customer')
sns.countplot(data=dff, x='Product',
hue='MaritalStatus',ax=ax[1],color='g'); ax[1].set_title('Female
Customer')
```

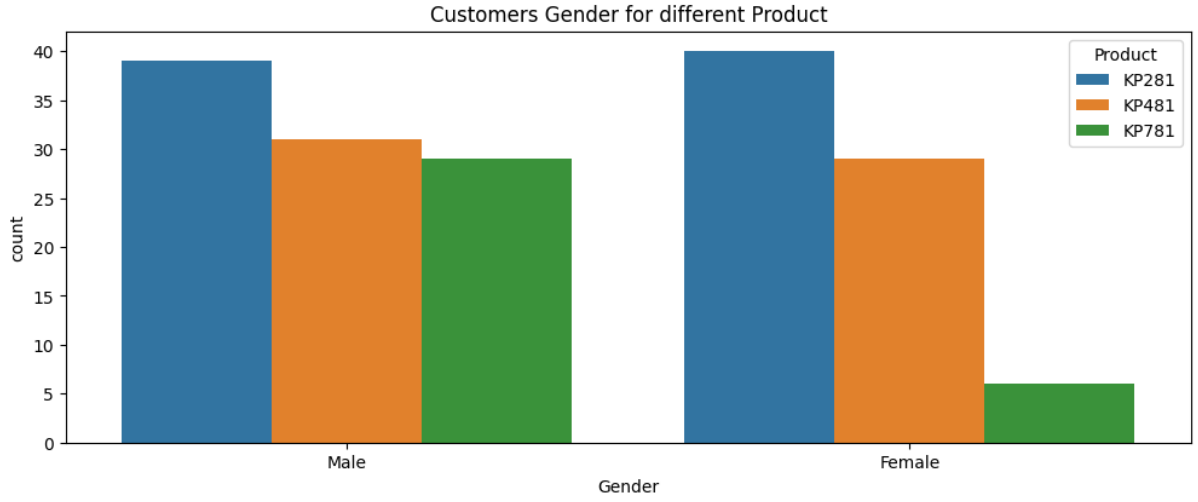



```
fig,ax = plt.subplots(nrows=3, ncols=1, figsize=(12,15))
sns.countplot(data=df, x='Education', hue='Product', ax=ax[0]);
ax[0].set_title('Customers Education for different Product')
sns.countplot(data=df, x='MaritalStatus', hue='Product', ax=ax[1])
ax[1].set_title('Customers Marital status for different Product')
```

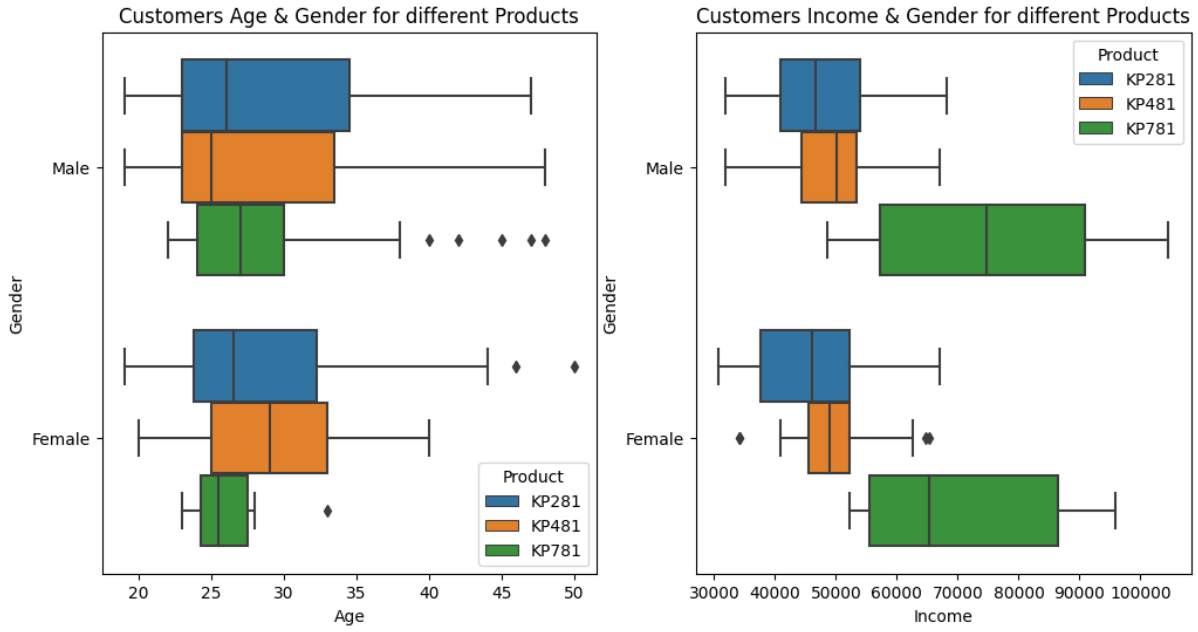
```
sns.countplot(data=df, x='Gender', hue='Product', ax=ax[2]);
ax[2].set_title('Customers Gender for different Product')
```





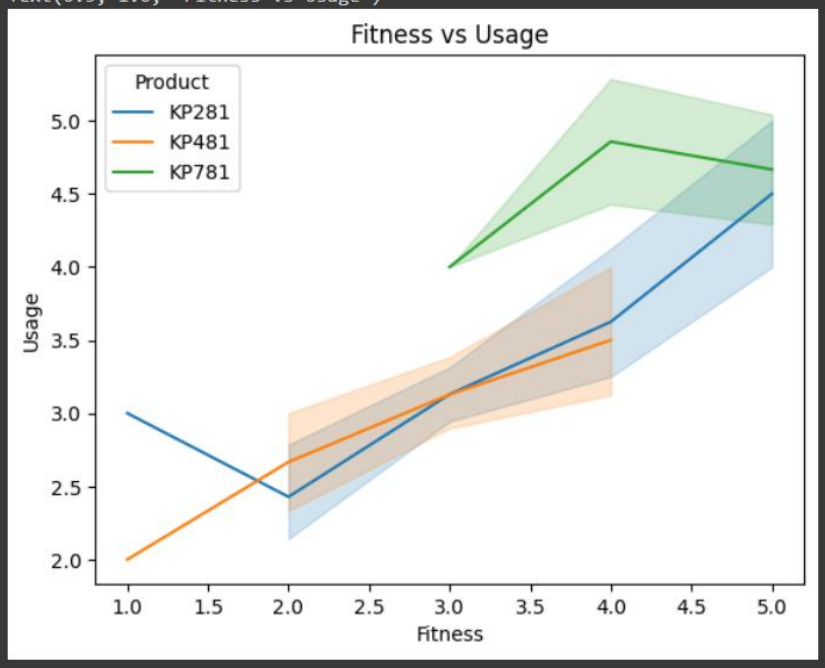



```
fig,ax = plt.subplots(nrows=1, ncols=2, figsize=(12,6))
sns.boxplot(data=df, x='Age', hue='Product', y='Gender', ax=ax[0])
ax[0].set_title('Customers Age & Gender for different Products')
sns.boxplot(data=df, x='Income', hue='Product', y='Gender', ax=ax[1])
ax[1].set_title('Customers Income & Gender for different Products')
```

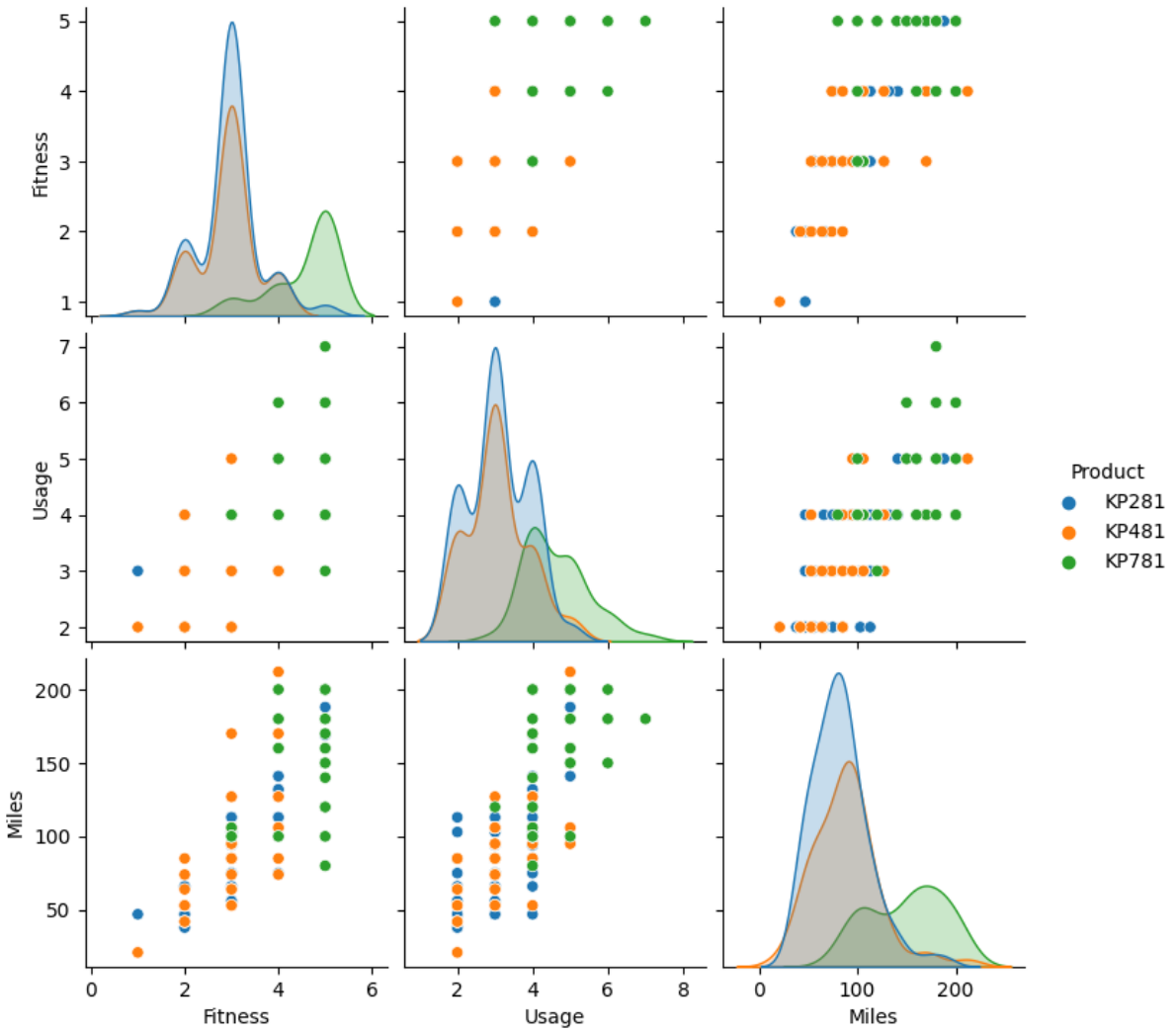



```
[89] sns.lineplot(data=df, x='Fitness', y='Usage', hue='Product'); plt.title('Fitness vs Usage')
```

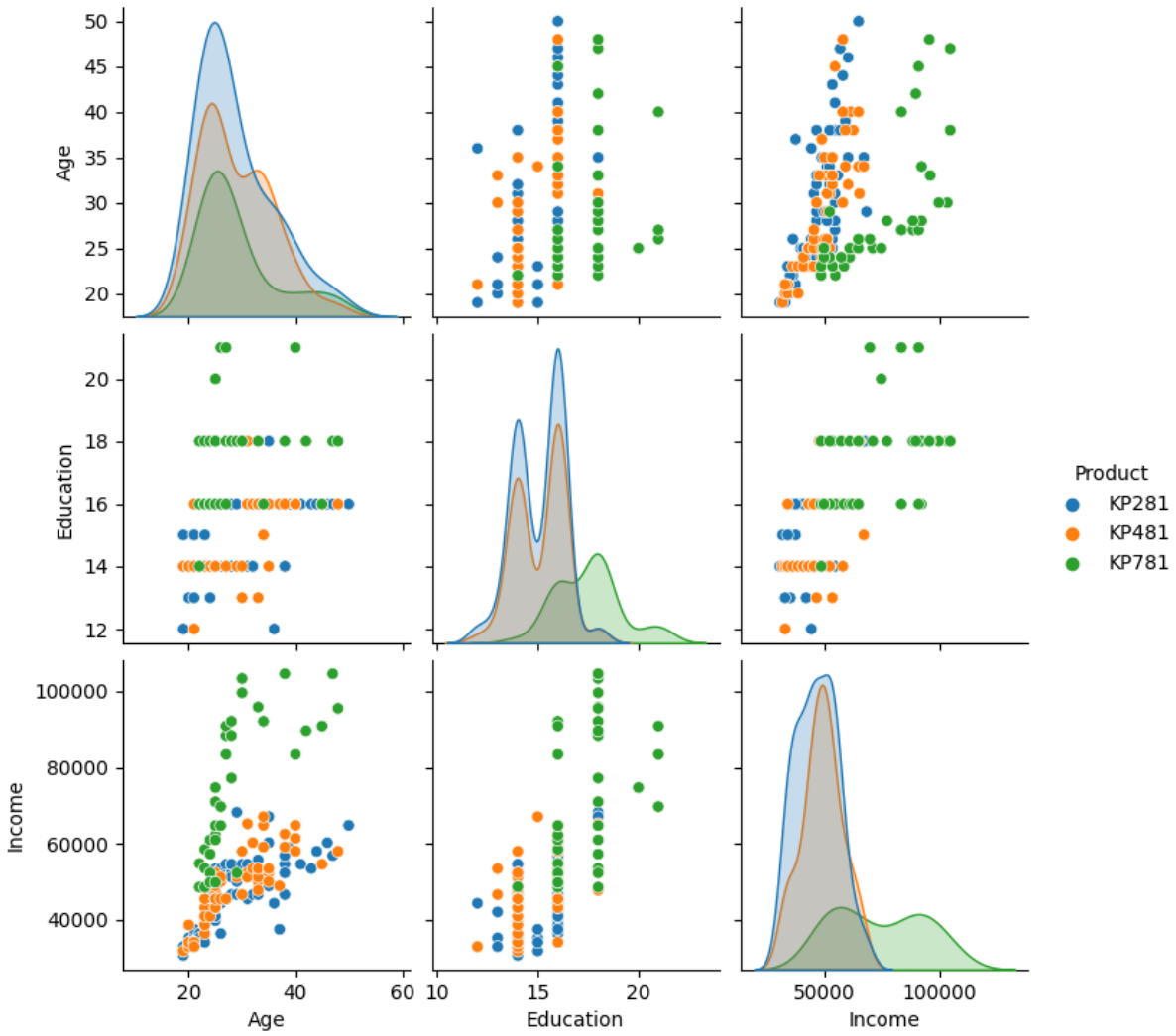
```
Text(0.5, 1.0, 'Fitness vs Usage')
```




```
#Visual Analysis - Correlation
sns.pairplot(data=df[['Fitness','Usage','Miles','Product']],
hue='Product')
```




```
sns.pairplot(data=df[['Product','Age','Education','Income','Gender']],
hue='Product')
```


Age
Education
Income
Product
KP281
KP481
KP781

Business Insights:

1. **Product `KP281` brings in the highest revenue, `KP481` and `KP781` come next in line respectively**
2. **Majority of the customers are in the age group of `22-33` years**
3. **~60-40% distribution of the `male` and `female` product buyers**
4. **Majority of the buyers spend 14, 16, 18 years on their education**
5. **~60-40% distribution of the `single` and `partnered` product buyers**
6. **Most of the users use the treadmill `3-4` times a week**
7. **Most of the users rate themselves `average` in terms of their fitness levels**
8. **Majority of the users earn between $35000 and $60000 annually**
9. **Majority of the users set target miles expected to be walked/ran between 53 and 132 miles**
10. **Insights from product-based study:**
    A. **Relationship with `Age`: no major insights**
    B. **Relationship with `Gender`:**
        - **Very few female customers buy KP781 product(priced at 2500 dollats); could be cost-related reasons**
    C. **Relationship with `Education`:**
        - **Highly educated customers prefer product KP781; they could be more aware of the product's typical features and its usage**
    D. **Relationship with `MaritalStatus`: no major insights**
    E. **Relationship with `Usage`:**
        - **product KP781 is used more compared to others products KP281 and KP481**
        - **this product is also prefered by highly-educated customers; this means highly-educated customers tend to exercise more**
    F. **Relationship with `Fitness`:**
        - **since highly-educated customer prefer product KP781 because they exercise more; their fitness levels are generally on high scale**
    G. **Relationship with `Income` and `Miles`:**
        - **product KP781 is prefered by high-income earning individuals**
        - **since highly-educated customer prefer product KP781 because they exercise more; their fitness levels are generally on high scale, the number of target miles they set are also higher**

6.Recommendations (10 Points) - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand

1. **A better, high-end, premium product for highly-educated, high income and active customers to increase revenue.**
2. **Campaigns to promote KP781 product for females specially**

3. **Since KP281 and KP481 also brings in significant revenue and is prefered by young & learnings individuals, added features and specialized discounts could help boost sales.**