my-business-case-aerofit

April 19, 2024

[3]:
```python
# calling the necessary python lybrary and modules
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
from scipy.stats import poisson
```

[4]:
```python
from google.colab import files
files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving aerofit_treadmill.csv to aerofit_treadmill.csv
```

[4]:

```
{'aerofit_treadmill.csv': b'Product,Age,Gender,Education,MaritalStatus,Usage,Fit
ness,Income,Miles\nKP281,18,Male,14,Single,3,4,29562,112\nKP281,19,Male,15,Singl
e,2,3,31836,75\nKP281,19,Female,14,Partnered,4,3,30699,66\nKP281,19,Male,12,Sing
le,3,3,32973,85\nKP281,20,Male,13,Partnered,4,2,35247,47\nKP281,20,Female,14,Par
tnered,3,3,32973,66\nKP281,21,Female,14,Partnered,3,3,35247,75\nKP281,21,Male,13
,Single,3,3,32973,85\nKP281,21,Male,15,Single,5,4,35247,141\nKP281,21,Female,15,
Partnered,2,3,37521,85\nKP281,22,Male,14,Single,3,3,36384,85\nKP281,22,Female,14
,Partnered,3,2,35247,66\nKP281,22,Female,16,Single,4,3,36384,75\nKP281,22,Female
,14,Single,3,3,35247,75\nKP281,23,Male,16,Partnered,3,1,38658,47\nKP281,23,Male,
16,Partnered,3,3,40932,75\nKP281,23,Female,14,Single,2,3,34110,103\nKP281,23,Mal
e,16,Partnered,4,3,39795,94\nKP281,23,Female,16,Single,4,3,38658,113\nKP281,23,F
emale,15,Partnered,2,2,34110,38\nKP281,23,Male,14,Single,4,3,38658,113\nKP281,23
,Male,16,Single,4,3,40932,94\nKP281,24,Female,16,Single,4,3,42069,94\nKP281,24,F
emale,16,Partnered,5,5,44343,188\nKP281,24,Male,14,Single,2,3,45480,113\nKP281,2
4,Male,13,Partnered,3,2,42069,47\nKP281,24,Female,16,Single,4,3,46617,75\nKP281,
25,Female,14,Partnered,3,3,48891,75\nKP281,25,Male,14,Partnered,2,3,45480,56\nKP
281,25,Female,14,Partnered,2,2,53439,47\nKP281,25,Female,14,Partnered,3,3,39795,
85\nKP281,25,Male,16,Single,3,4,40932,113\nKP281,25,Female,16,Partnered,2,2,4093
2,47\nKP281,25,Male,16,Single,3,3,43206,85\nKP281,26,Female,14,Partnered,3,4,443
43,113\nKP281,26,Female,16,Partnered,4,3,52302,113\nKP281,26,Male,16,Partnered,2
,2,53439,47\nKP281,26,Male,16,Partnered,3,3,51165,85\nKP281,26,Female,16,Single,
3,3,36384,66\nKP281,26,Male,16,Partnered,4,4,44343,132\nKP281,26,Male,16,Single,
3,3,50028,85\nKP281,27,Female,14,Partnered,3,2,45480,66\nKP281,27,Male,16,Single
```

,4,3,54576,85\nKP281,27,Female,14,Partnered,2,3,45480,56\nKP281,28,Female,14,Par
tnered,2,3,46617,56\nKP281,28,Female,16,Partnered,2,3,52302,66\nKP281,28,Male,14
,Single,3,3,52302,103\nKP281,28,Female,14,Partnered,3,3,54576,94\nKP281,28,Male,
14,Single,4,3,54576,113\nKP281,28,Female,16,Partnered,3,3,51165,56\nKP281,29,Mal
e,18,Partnered,3,3,68220,85\nKP281,29,Female,14,Partnered,2,2,46617,38\nKP281,29
,Female,16,Partnered,4,3,50028,94\nKP281,30,Male,14,Partnered,4,4,46617,141\nKP2
81,30,Male,14,Single,3,3,54576,85\nKP281,31,Male,14,Partnered,2,2,54576,47\nKP28
1,31,Female,14,Single,2,2,45480,47\nKP281,32,Female,14,Single,3,4,46617,113\nKP2
81,32,Male,14,Partnered,4,3,52302,85\nKP281,33,Female,16,Single,2,2,55713,38\nKP
281,33,Female,16,Partnered,3,3,46617,85\nKP281,34,Male,16,Single,4,5,51165,169\n
KP281,34,Female,16,Single,2,2,52302,66\nKP281,35,Male,16,Partnered,4,3,48891,85\
nKP281,35,Female,16,Partnered,3,3,60261,94\nKP281,35,Female,18,Single,3,3,67083,
85\nKP281,36,Male,12,Single,4,3,44343,94\nKP281,37,Female,16,Partnered,3,3,37521
,85\nKP281,38,Male,16,Partnered,3,3,46617,75\nKP281,38,Female,14,Partnered,2,3,5
4576,56\nKP281,38,Male,14,Single,2,3,52302,56\nKP281,38,Male,16,Partnered,3,3,56
850,75\nKP281,39,Male,16,Partnered,4,4,59124,132\nKP281,40,Male,16,Partnered,3,3
,61398,66\nKP281,41,Male,16,Partnered,4,3,54576,103\nKP281,43,Male,16,Partnered,
3,3,53439,66\nKP281,44,Female,16,Single,3,4,57987,75\nKP281,46,Female,16,Partner
ed,3,2,60261,47\nKP281,47,Male,16,Partnered,4,3,56850,94\nKP281,50,Female,16,Par
tnered,3,3,64809,66\nKP481,19,Male,14,Single,3,3,31836,64\nKP481,20,Male,14,Sing
le,2,3,32973,53\nKP481,20,Female,14,Partnered,3,3,34110,106\nKP481,20,Male,14,Si
ngle,3,3,38658,95\nKP481,21,Female,14,Partnered,5,4,34110,212\nKP481,21,Male,16,
Partnered,2,2,34110,42\nKP481,21,Male,12,Partnered,2,2,32973,53\nKP481,23,Male,1
4,Partnered,3,3,36384,95\nKP481,23,Male,14,Partnered,3,3,38658,85\nKP481,23,Fema
le,16,Single,3,3,45480,95\nKP481,23,Male,16,Partnered,4,3,45480,127\nKP481,23,Fe
male,16,Partnered,3,2,43206,74\nKP481,23,Female,14,Single,3,2,40932,53\nKP481,23
,Male,16,Partnered,3,3,45480,64\nKP481,24,Female,14,Single,3,2,40932,85\nKP481,2
4,Male,14,Single,3,4,48891,106\nKP481,24,Female,16,Single,3,3,50028,106\nKP481,2
5,Female,14,Partnered,2,3,45480,85\nKP481,25,Female,14,Single,3,4,43206,127\nKP4
81,25,Male,16,Partnered,2,2,52302,42\nKP481,25,Female,14,Partnered,5,3,47754,106
\nKP481,25,Male,14,Single,3,3,45480,95\nKP481,25,Female,14,Single,2,3,43206,64\n
KP481,25,Male,14,Partnered,4,3,45480,170\nKP481,25,Male,14,Partnered,3,4,43206,1
06\nKP481,25,Male,16,Partnered,2,3,50028,53\nKP481,25,Female,14,Single,2,2,45480
,42\nKP481,25,Male,14,Single,4,3,48891,127\nKP481,26,Female,16,Partnered,4,3,454
80,85\nKP481,26,Female,16,Single,4,4,50028,127\nKP481,26,Male,16,Single,4,3,5116
5,106\nKP481,27,Male,14,Single,4,2,45480,53\nKP481,29,Female,14,Partnered,3,3,51
165,95\nKP481,30,Female,14,Single,3,3,57987,74\nKP481,30,Female,13,Single,4,3,46
617,106\nKP481,31,Male,16,Partnered,3,3,52302,95\nKP481,31,Female,16,Partnered,2
,3,51165,64\nKP481,31,Female,18,Single,2,1,65220,21\nKP481,32,Male,16,Single,4,3
,60261,127\nKP481,32,Male,16,Partnered,3,3,53439,95\nKP481,33,Male,13,Partnered,
4,4,53439,170\nKP481,33,Female,16,Partnered,2,3,50028,85\nKP481,33,Male,16,Partn
ered,3,3,51165,95\nKP481,33,Female,16,Partnered,5,3,53439,95\nKP481,33,Female,18
,Single,3,4,47754,74\nKP481,34,Female,16,Partnered,4,3,64809,95\nKP481,34,Male,1
6,Partnered,3,4,59124,85\nKP481,34,Male,15,Single,3,3,67083,85\nKP481,35,Female,
14,Partnered,3,2,52302,53\nKP481,35,Male,16,Partnered,3,2,53439,53\nKP481,35,Fem
ale,16,Single,3,2,50028,64\nKP481,35,Male,16,Partnered,3,3,53439,95\nKP481,37,Fe
male,16,Partnered,2,3,48891,85\nKP481,38,Female,16,Partnered,4,3,62535,85\nKP481

```
,38,Male,16,Partnered,3,3,59124,106\nKP481,40,Female,16,Partnered,3,3,61398,85\n
KP481,40,Female,16,Single,3,3,57987,85\nKP481,40,Male,16,Partnered,3,3,64809,95\
nKP481,45,Male,16,Partnered,2,2,54576,42\nKP481,48,Male,16,Partnered,2,3,57987,6
4\nKP781,22,Male,14,Single,4,3,48658,106\nKP781,22,Male,16,Single,3,5,54781,120\
nKP781,22,Male,18,Single,4,5,48556,200\nKP781,23,Male,16,Single,4,5,58516,140\nK
P781,23,Female,18,Single,5,4,53536,100\nKP781,23,Male,16,Single,4,5,48556,100\nK
P781,24,Male,16,Single,4,5,61006,100\nKP781,24,Male,18,Partnered,4,5,57271,80\nK
P781,24,Female,16,Single,5,5,52291,200\nKP781,24,Male,16,Single,5,5,49801,160\nK
P781,25,Male,16,Partnered,4,5,49801,120\nKP781,25,Male,16,Partnered,4,4,62251,16
0\nKP781,25,Female,18,Partnered,5,5,61006,200\nKP781,25,Male,18,Partnered,4,3,64
741,100\nKP781,25,Male,18,Partnered,6,4,70966,180\nKP781,25,Male,18,Partnered,6,
5,75946,240\nKP781,25,Male,20,Partnered,4,5,74701,170\nKP781,26,Female,21,Single
,4,3,69721,100\nKP781,26,Male,16,Partnered,5,4,64741,180\nKP781,27,Male,16,Partn
ered,4,5,83416,160\nKP781,27,Male,18,Single,4,3,88396,100\nKP781,27,Male,21,Part
nered,4,4,90886,100\nKP781,28,Female,18,Partnered,6,5,92131,180\nKP781,28,Male,1
8,Partnered,7,5,77191,180\nKP781,28,Male,18,Single,6,5,88396,150\nKP781,29,Male,
18,Single,5,5,52290,180\nKP781,29,Male,14,Partnered,7,5,85906,300\nKP781,30,Fema
le,16,Partnered,6,5,90886,280\nKP781,30,Male,18,Partnered,5,4,103336,160\nKP781,
30,Male,18,Partnered,5,5,99601,150\nKP781,31,Male,16,Partnered,6,5,89641,260\nKP
781,33,Female,18,Partnered,4,5,95866,200\nKP781,34,Male,16,Single,5,5,92131,150\
nKP781,35,Male,16,Partnered,4,5,92131,360\nKP781,38,Male,18,Partnered,5,5,104581
,150\nKP781,40,Male,21,Single,6,5,83416,200\nKP781,42,Male,18,Single,5,4,89641,2
00\nKP781,45,Male,16,Single,5,5,90886,160\nKP781,47,Male,18,Partnered,4,5,104581
,120\nKP781,48,Male,18,Partnered,4,5,95508,180\n'}
```

[5]:
```
# read the dataset in pandas
fit = pd.read_csv('aerofit_treadmill.csv')
fit
```

[5]:
```
    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]
```

[4]:
```
# get the information about the data set
fit.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
```

[5]:
```
# checking the numbers of null values of every columns in our dataset
fit.isna().sum()
```

[5]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

[6]:
```
# checking the numbers of unique values of every columns in the dataset
for i in fit.columns:
  print(i,":",fit[i].nunique())
```

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

[7]:
```
# chacking the no of rows and columns
fit.shape
```

[7]:
```
(180, 9)
```

[8]:
```
fit.head()
```

[8]:

	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

[23]:
```
# getting the names of the unique products
fit['Product'].unique()
```

[23]:
```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

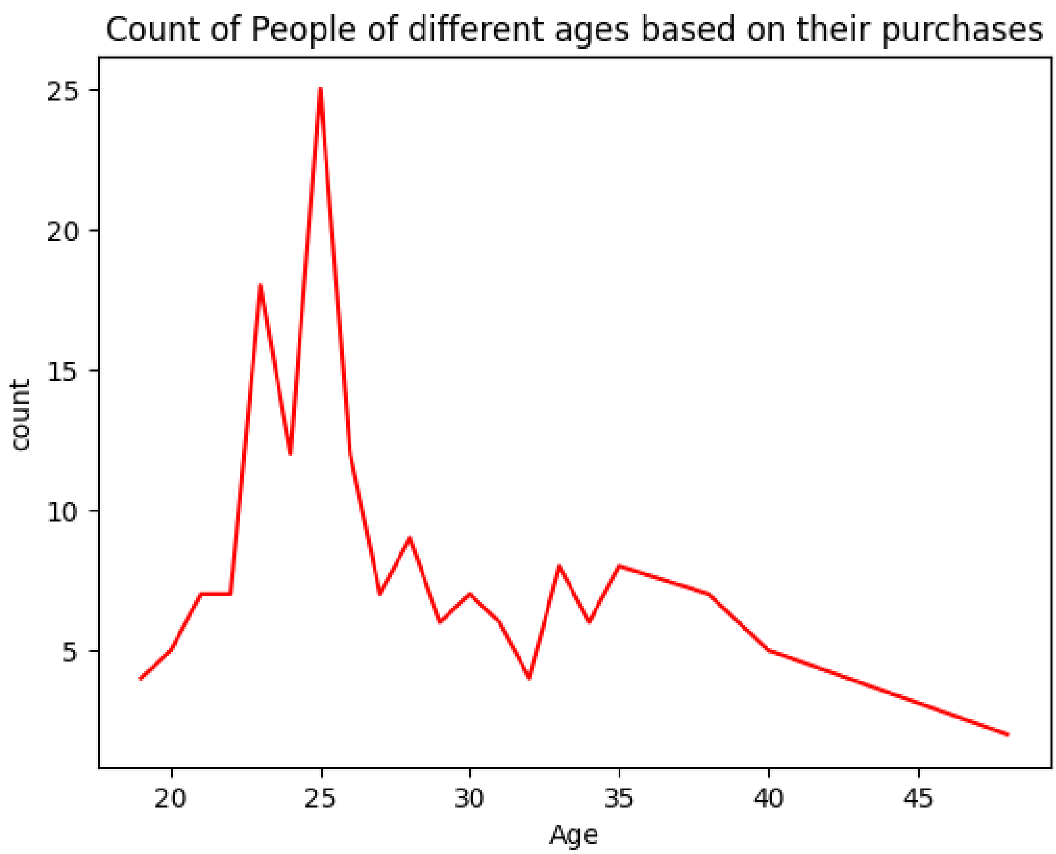
[28]:
```
# getting the over view of top 20 people of what ages are buying these products
fit_age = fit.value_counts('Age').head(20).reset_index()
fit_age
```

[28]:

	Age	count
0	25	25
1	23	18
2	24	12
3	26	12
4	28	9
5	35	8
6	33	8
7	30	7
8	38	7
9	21	7
10	22	7
11	27	7
12	31	6

```
13   34      6
14   29      6
15   20      5
16   40      5
17   32      4
18   19      4
19   48      2
```

[40]:
```
sns.lineplot(data= fit_age, x = 'Age', y  = 'count', color = 'Red')
plt.title('Count of People of different ages based on their purchases')
plt.show()
```



Above plot shows that the people in the range of 23 to 26 ages buyinng these products most

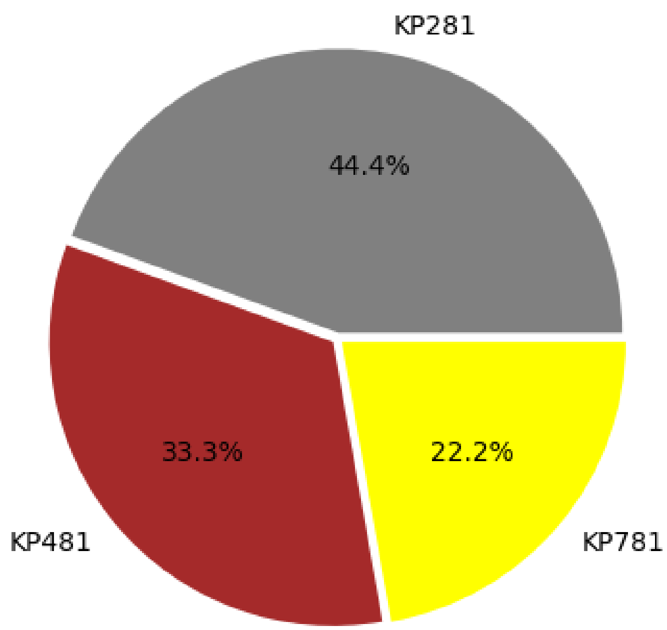
[43]:
```
# get the cout of specific product sold
fit_count = fit.value_counts(['Product']).reset_index()
fit_count
```

[43]:

	Product	count
0	KP281	80

```
1   KP481     60
2   KP781     40
```

[57]:
```
plt.pie(fit_count['count'], labels=fit_count['Product'], explode = (0.02,0.02,0.
↪02), colors = ['gray','brown', 'yellow'], autopct = '%.1f%%')
plt.show()
```



Above chart shows that the KP281 has the highest sales.

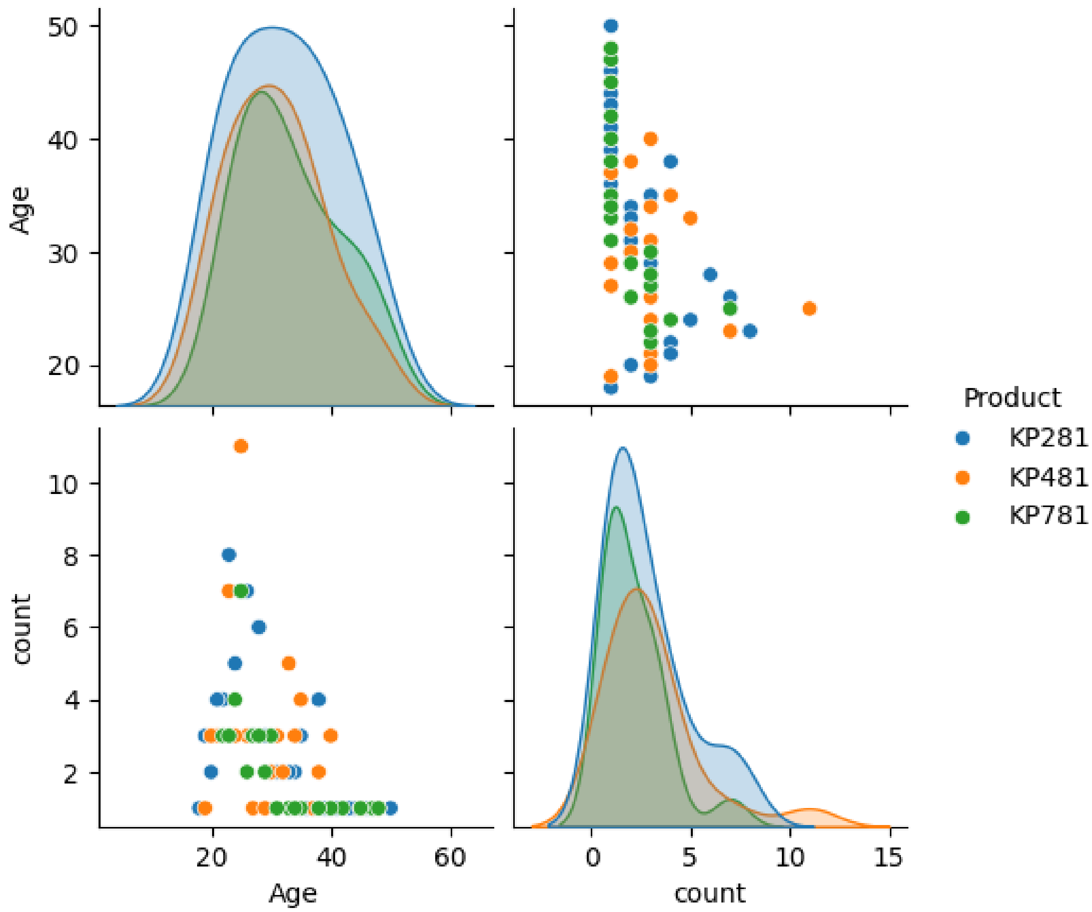
[74]:
```
fit_1 = fit.groupby('Product').value_counts(['Age']).reset_index()
fit_1
```

[74]:

	Product	Age	count
0	KP281	23	8
1	KP281	25	7
2	KP281	26	7
3	KP281	28	6
4	KP281	24	5
..	…	…	…
63	KP781	40	1
64	KP781	38	1
65	KP781	35	1
66	KP781	34	1

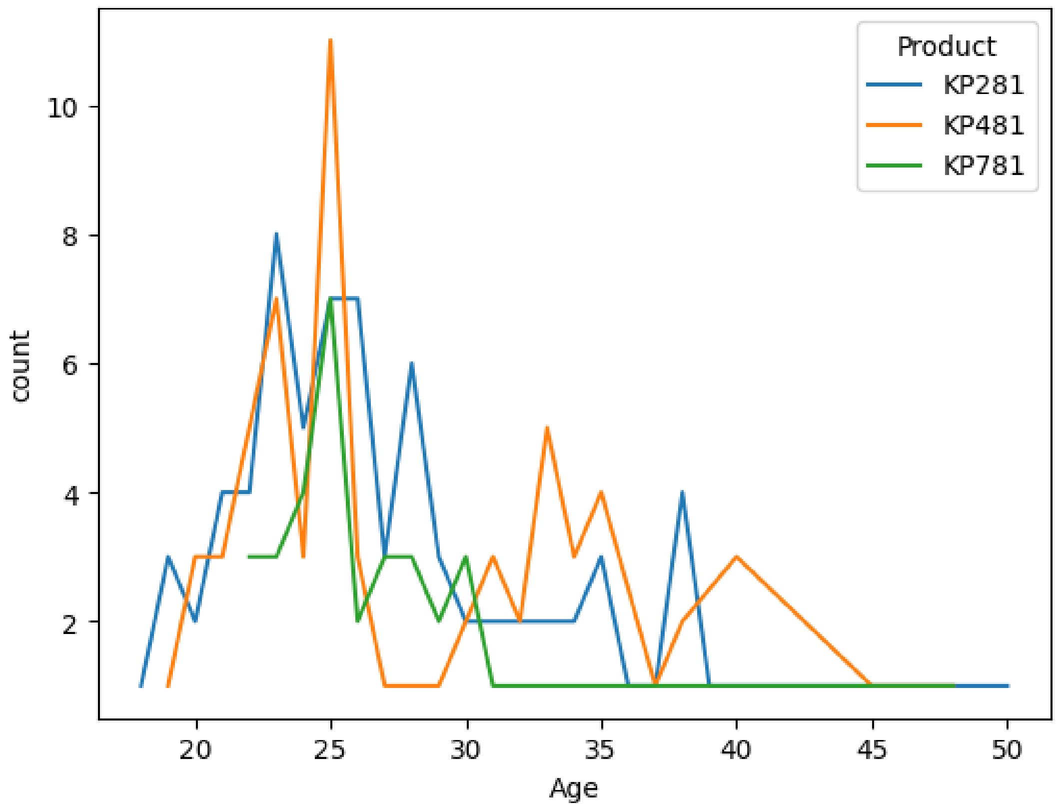
```
67    KP781    48       1

[68 rows x 3 columns]
```

[69]:
```
sns.pairplot(data=fit_1, hue='Product')
plt.show()
```



Above chart shows 4 different representation of what age buoght the paroducts.

[75]:
```
sns.lineplot(data=fit_1, x = 'Age', y = 'count', hue = 'Product')
plt.show()
```



Above chart shows that the 25 years old people are buying KP481 product more that other products, but the main customers are in the age of 20 to 40 years old

[78]:
```
fit.head()
```

[78]:

	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

[80]:
```
# to get the marginal probability and conditional probability
pd.crosstab(index= fit.Product, columns = fit.Gender, margins = True)
```

[80]:

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

[81]:
```
# Marginal probability of coustomer being Female
76/180
```

[81]: 0.4222222222222222

[82]:
```
# Marginal probability of coustomer being Male
104/180
```

[82]: 0.5777777777777777

[83]:
```
# Marginal probability of coustomer being Female and product bought KP281
40/180
```

[83]: 0.2222222222222222

[84]:
```
# Marginal probability of coustomer being Female and product bought KP481
29/180
```

[84]: 0.16111111111111112

[85]:
```
# Marginal probability of coustomer being Female and product bought KP781
7/180
```

[85]: 0.03888888888888889

[86]:
```
# Marginal probability of coustomer being Male and product bought KP281
40/180
```

[86]: 0.2222222222222222

[87]:
```
# Marginal probability of coustomer being Male and product bought KP481
31/180
```

[87]: 0.17222222222222222

[88]:
```
# Marginal probability of coustomer being Male and product bought KP781
33/180
```

[88]: 0.18333333333333332

[94]:
```
# Conditional probability of coustomer being Male given product bought KP281
40/80
```

[94]: 0.5

[95]:
```
# Conditional probability of coustomer being FeMale given product bought KP281
40/80
```

[95]: 0.5

[96]:
```
# Conditional probability of coustomer being Male given product bought KP481
31/60
```

[96]: 0.5166666666666667

[97]:
```
# Conditional probability of coustomer being Female given product bought KP481
29/60
```

[97]: 0.48333333333333334

[98]:
```
# Conditional probability of coustomer being Male given product bought KP781
33/40
```

[98]: 0.825

[99]:
```
# Conditional probability of coustomer being Female given product bought KP781
7/40
```

[99]: 0.175

above analisys shows that 1. probability of coustomer being Male is higher than coustomer being Female 2. probability of product being bought by Male and Female is same 3. probability of coustomer being Male given product bought KP781 is very higher than the coustomer being Female

[102]:
```
# to get the marginal probability and conditional probability
pd.crosstab(index= fit.MaritalStatus, columns = fit.Gender, margins = True)
```

[102]:

Gender	Female	Male	All
MaritalStatus			
Partnered	46	61	107
Single	30	43	73
All	76	104	180

[104]:
```
# Marginal probability of coustomer being Partnered
107/180
```

[104]: 0.5944444444444444

[105]:
```
# Marginal probability of coustomer being Single
73/180
```

[105]: 0.40555555555555556

[106]:
```
# Marginal probability of coustomer being Partnered and Male
61/180
```

[106]: 0.3388888888888889

[107]:
```
# Marginal probability of coustomer being Partnered and Female
46/180
```

[107]: 0.25555555555555554

[108]:
```
# Marginal probability of coustomer being Single and Male
43/180
```

[108]: 0.2388888888888889

[111]:
```
# Marginal probability of coustomer being Single and Female
30/180
```

[111]: 0.16666666666666666

[112]:
```
# conditinal probability of customer being partnered given customer is a Male
61/140
```

[112]: 0.4357142857142857

[113]:
```
# conditinal probability of customer being partnered given customer is a Female
46/76
```

[113]: 0.6052631578947368

[114]:
```
# conditinal probability of customer being single given customer is a Male
46/104
```

[114]: 0.4423076923076923

[115]:
```
# conditinal probability of customer being single given customer is a Female
30/76
```

[115]: 0.39473684210526316

Above analysis shows that 1. probabilty of customer being partnered is higher than customer being single. 2. probabilty of customer being male partnered is higher than female partnered customer

[127]:
```
'''
To get
count, mean, std diviation, min, 25%, 50%, 75% and Max
of every single numerical column

'''
fit.describe()
```

[127]:
```
                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
```

[40]:
```
# calculate the median of columns
fit['Age'].median().astype(int)
```

[40]: 26

[130]:
```
fit['Age'].describe()
```

[130]:
```
count    180.000000
mean      28.788889
std        6.943498
min       18.000000
25%       24.000000
50%       26.000000
75%       33.000000
max       50.000000
Name: Age, dtype: float64
```

1. avarage age in the database is **28.79**
2. Minimun age in the dataset is **18**.

3. This value **25%** indicates that 25% of all the values present in the ‘Age’ series is less than **24**
4. This value **50%** indicates that 50% of all the values present in the ‘Age’ series is less than **26**
5. This value **75%** indicates that 75% of all the values present in the ‘Age’ series is less than **33**
6. Maximum age in the dataset is **50**

[132]:
```
#IQR = Inter Quertile Range  = value(75%)- value(25%)
IQR = 33-24
```

```
IQR
```

[132]: 9

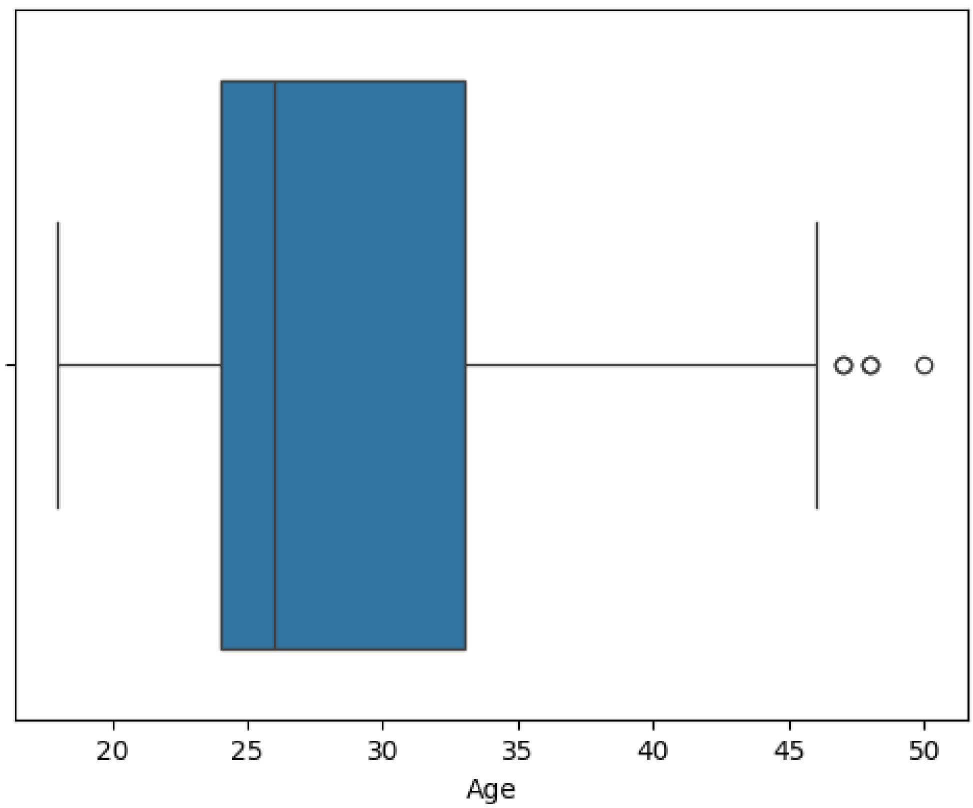
IQR = 9, which means that middle 50% of the data lies in the range of 9.

[134]:
```
# max age - min age
50-18
```

[134]: 32

On the other hand, the normal range is very high i.e. 32
We can observe one thing that there is Outlier present in the dataset
This is why the normal range is getting affected by the outlier.

[131]:
```
# BOX plot,  This is the graphical representation of above analysis
sns.boxplot(data=fit['Age'], orient="h")
plt.show()
```



All the values outside the whisker are considered “Outlier”

[140]:
```
fit
```

[140]:
```
     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]
```

[41]:
```python
# calculate the median of columns
fit['Education'].median().astype(int)
```

[41]:
```
16
```

[135]:
```python
fit['Education'].describe()
```

[135]:
```
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
```

avarage education in the database is 15.57
Minimun education in the dataset is 12.
This value 25% indicates that 25% of all the values present in the 'education' series is less than 14

This value 50% indicates that 50% of all the values present in the 'education' series is less than 16
This value 75% indicates that 75% of all the values present in the 'education' series is less than 16
Maximum education in the dataset is 21

[142]:
```
#IQR = Inter Quertile Range  = value(75%)- value(25%)
IQR = 16-14
IQR
```

[142]: 2

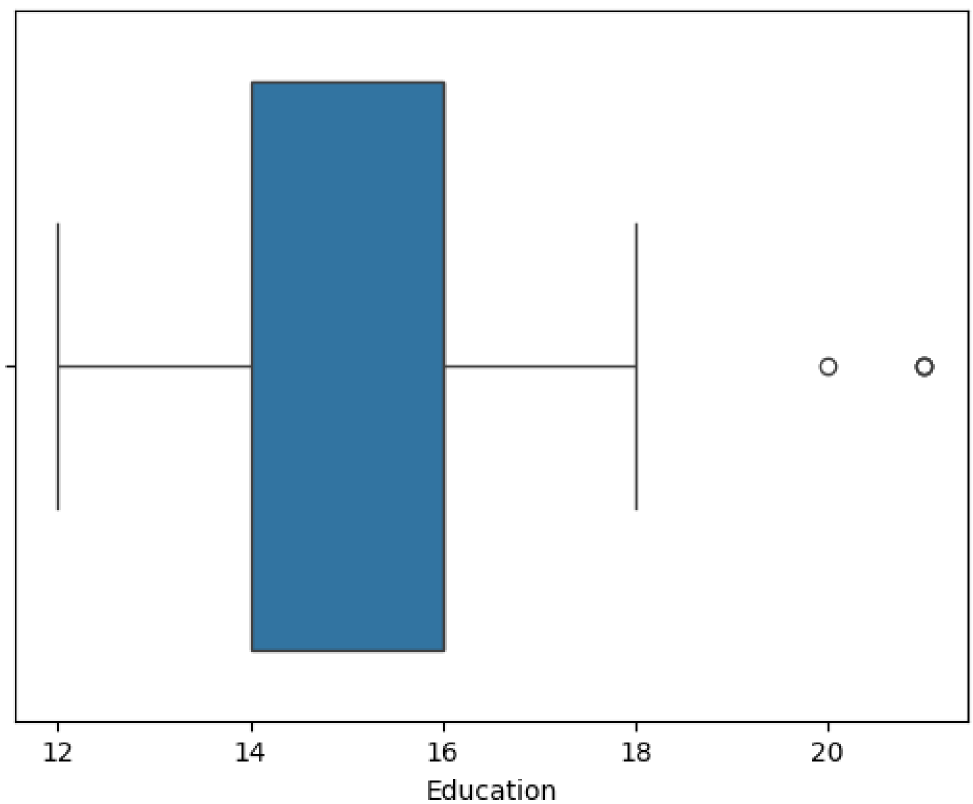
IQR = 2, which means that middle 50% of the data lies in the range of 2.

[143]:
```
# max education - min education
21-12
```

[143]: 9

the normal range is high i.e. 9 We can observe one thing that there is Outlier present in the dataset This is why the normal range is getting affected by the outlier.

[144]:
```
# BOX plot,  This is the graphical representation of above analysis
sns.boxplot(data=fit['Education'], orient="h")
plt.show()
```



All the values outside the whisker are considered "Outlier"

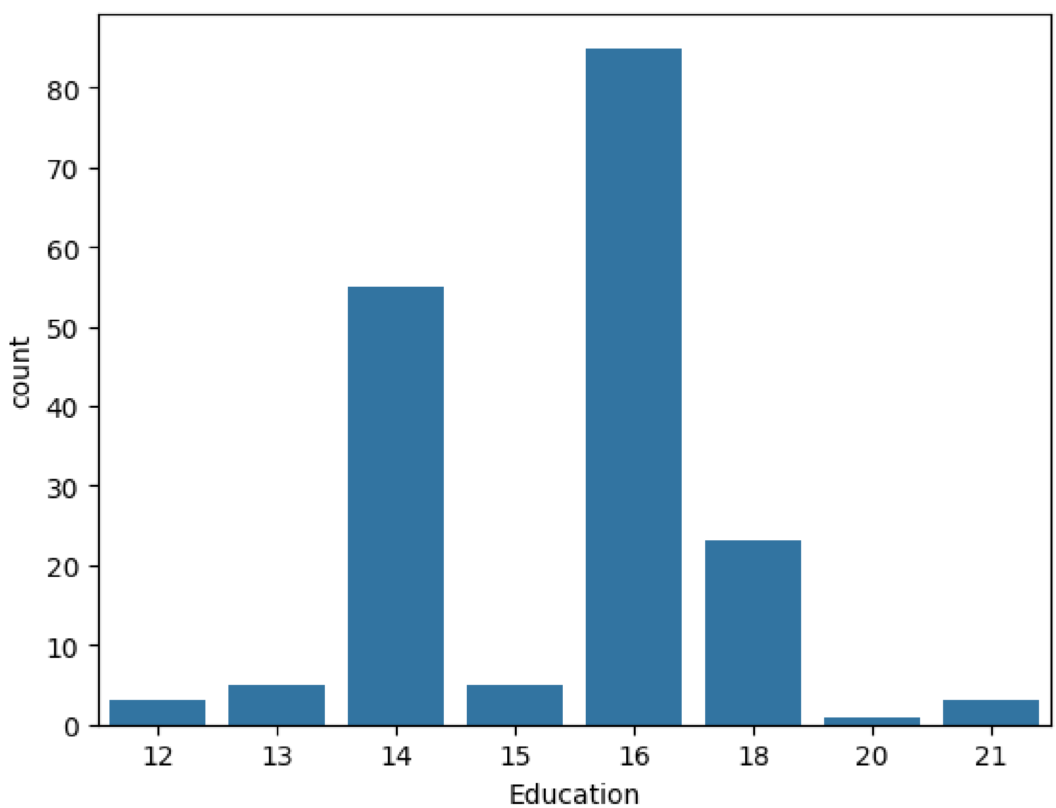
[147]:
```
# getting the over view of people with how many years of education are buying these products
fit_edu = fit.value_counts('Education').reset_index()
fit_edu
```

[147]:

	Education	count
0	16	85
1	14	55
2	18	23
3	13	5
4	15	5
5	12	3
6	21	3
7	20	1

[149]:
```
sns.barplot(data =fit_edu, x = 'Education', y = 'count')
```

[149]: <Axes: xlabel='Education', ylabel='count'>



above chart shows that the people with 15 and 14 years of education bought the product most.

[42]:
```python
# calculate the median of columns
fit['Usage'].median().astype(int)
```

[42]: 3

[151]:
```python
fit['Usage'].describe()
```

[151]:
```
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
```

avarage Usage in the database is 3.455
Minimun Usage in the dataset is 2.
This value 25% indicates that 25% of all the values present in the 'Usage' series is less than 3
This value 50% indicates that 50% of all the values present in the 'Usage' series is less than 3
This value 75% indicates that 75% of all the values present in the 'Usage' series is less than 4
Maximum Usage in the dataset is 7

[152]:
```python
#IQR = Inter Quertile Range  = value(75%)- value(25%)
IQR = 4-3
IQR
```

[152]: 1

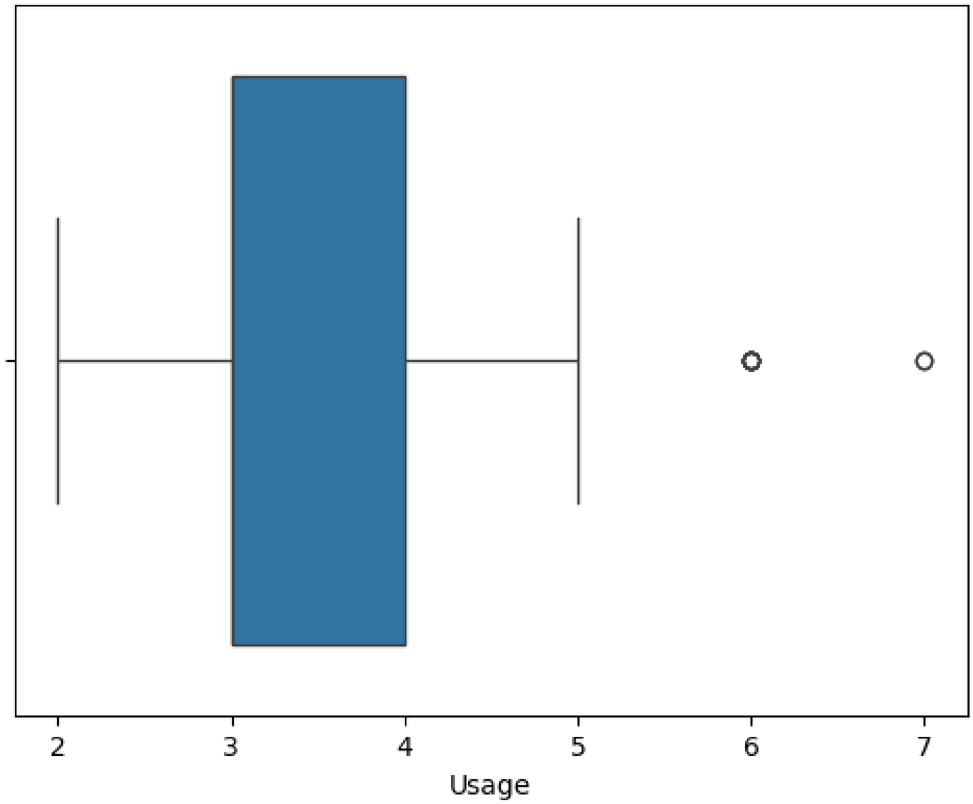
IQR = 1, which means that middle 50% of the data lies in the range of 1.

[153]:
```python
# max Usage - min Usage
7-2
```

[153]: 5

the normal range is high i.e. 5 We can observe one thing that there is Outlier present in the dataset This is why the normal range is getting affected by the outlier.

[154]:
```python
# BOX plot,  This is the graphical representation of above analysis
sns.boxplot(data=fit['Usage'], orient="h")
plt.show()
```



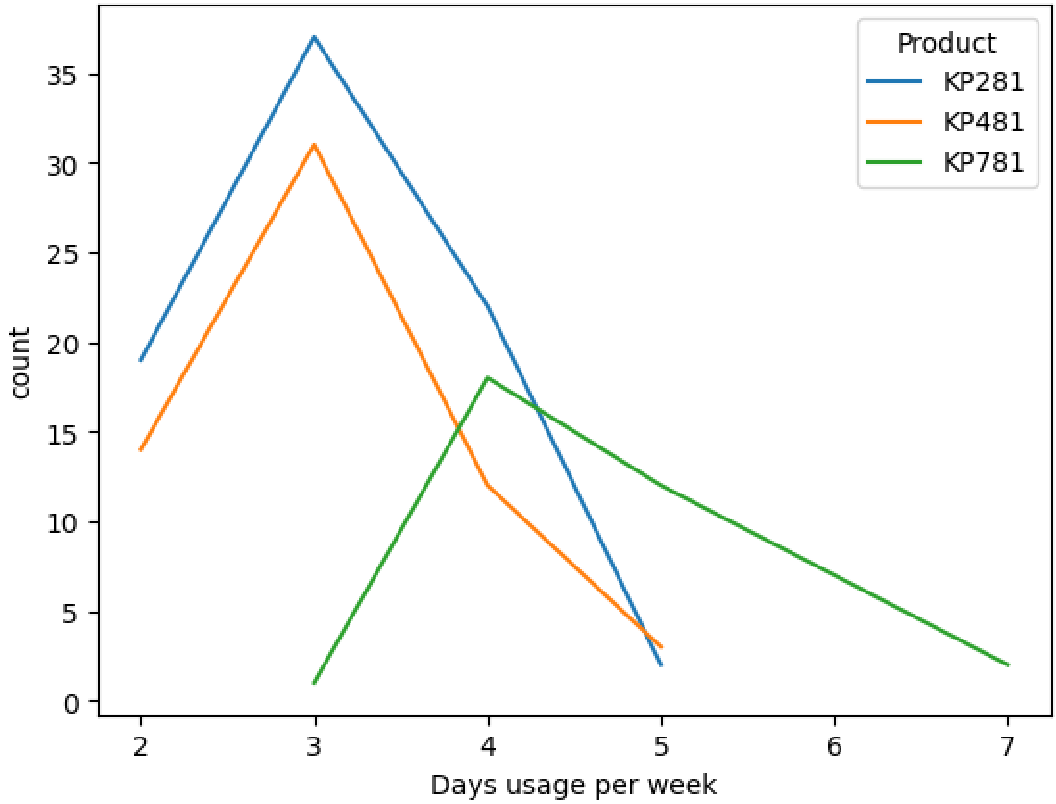
All the values outside the whisker are considered “Outlier”

[51]:
```
fit_use= fit.groupby('Product').value_counts(['Usage']).reset_index()
fit_use
```

[51]:

	Product	Usage	count
0	KP281	3	37
1	KP281	4	22
2	KP281	2	19
3	KP281	5	2
4	KP481	3	31
5	KP481	2	14
6	KP481	4	12
7	KP481	5	3
8	KP781	4	18
9	KP781	5	12
10	KP781	6	7
11	KP781	7	2
12	KP781	3	1

[48]:
```
sns.lineplot(data = fit_use, x = 'Usage', y = 'count', hue = 'Product')
plt.xlabel('Days usage per week')
plt.show()
```



Above chart shows us that most people like to use these product 2to 5 days per week.
and less people use KP781 product 3 to 7 days per week

[50]:
```
# to get the marginal probability and conditional probability
pd.crosstab(index = fit_use.Product, columns = fit_use.Usage, margins = True)
```

[50]:

Usage	2	3	4	5	6	7	All
Product							
KP281	1	1	1	1	0	0	4
KP481	1	1	1	1	0	0	4
KP781	0	1	1	1	1	1	5
All	2	3	3	3	1	1	13

[19]:
```
# probablity of customer will use the product 2 times in a week
2/13
```

[19]: 0.15384615384615385

```
[20]: # probablity of customer will use the product 3 times in a week
3/13
```

```
[20]: 0.23076923076923078
```

```
[23]: # probablity of customer will use the product 4 times in a week
3/13
```

```
[23]: 0.23076923076923078
```

```
[24]: # probablity of customer will use the product 5 times in a week
3/13
```

```
[24]: 0.23076923076923078
```

```
[25]: # probablity of customer will use the product 6 times in a week
1/13
```

```
[25]: 0.07692307692307693
```

```
[26]: # probablity of customer will use the product 6 times in a week
1/13
```

```
[26]: 0.07692307692307693
```

Above analysis showa that the probability of customer will use the product 3 times , 4 times, 5 times in a week is same.

```
[155]: fit
```

```
[155]:     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]
```

[52]:
```
fit['Fitness'].describe()
```

[52]:
```
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
```

[53]:
```
# calculate the median of columns
fit['Fitness'].median().astype(int)
```

[53]:
```
3
```

avarage Fitness in the database is 3.311
Minimun Fitness in the dataset is 1
This value 25% indicates that 25% of all the values present in the 'Fitness' series is less than 3
This value 50% indicates that 50% of all the values present in the 'Fitness' series is less than 3
This value 75% indicates that 75% of all the values present in the 'Fitness' series is less than 4
Maximum Fitness in the dataset is 5

[54]:
```
#IQR = Inter Quertile Range  = value(75%)- value(25%)
IQR = 4-3
IQR
```

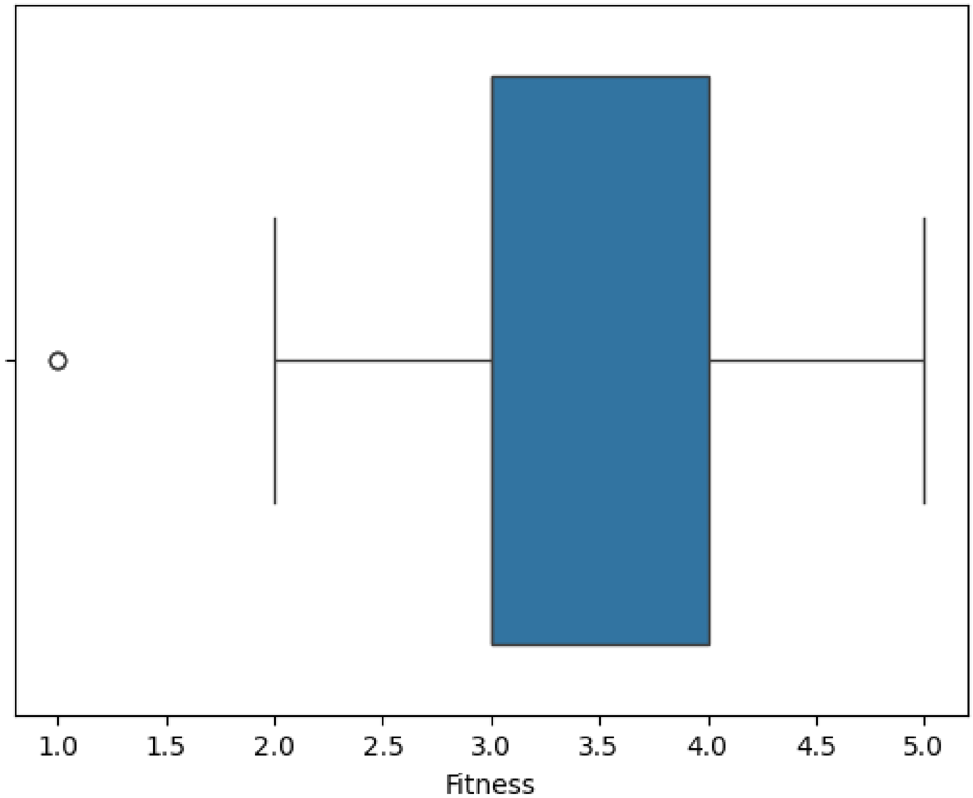
[54]:
```
1
```

IQR = 1, which means that middle 50% of the data lies in the range of 1.

[55]:
```
# Max fitness - min fitness
5-1
```

[55]:
```
4
```

[56]:
```python
# BOX plot,  This is the graphical representation of above analysis
sns.boxplot(data=fit['Fitness'], orient="h")
plt.show()
```



[61]:
```python
# calculate the median of columns
fit['Income'].median().astype(int)
```

[61]: 50596

[60]:
```python
fit['Income'].describe()
```

[60]:
```
count       180.000000
mean      53719.577778
std       16506.684226
min       29562.000000
25%       44058.750000
50%       50596.500000
75%       58668.000000
max      104581.000000
Name: Income, dtype: float64
```

avarage Income in the database is 53719.577
Minimun Income in the dataset is 29662

This value 25% indicates that 25% of all the values present in the 'Income' series is less than 44058.75
This value 50% indicates that 50% of all the values present in the 'Income' series is less than 50596.5
This value 75% indicatesthat 75% of all the values present in the 'Income' series is less than 58668.
Maximum Income in the dataset is 104581

[62]:
```
#IQR = Inter Quertile Range  = value(75%)- value(25%)
IQR =104581-44058.75
IQR
```

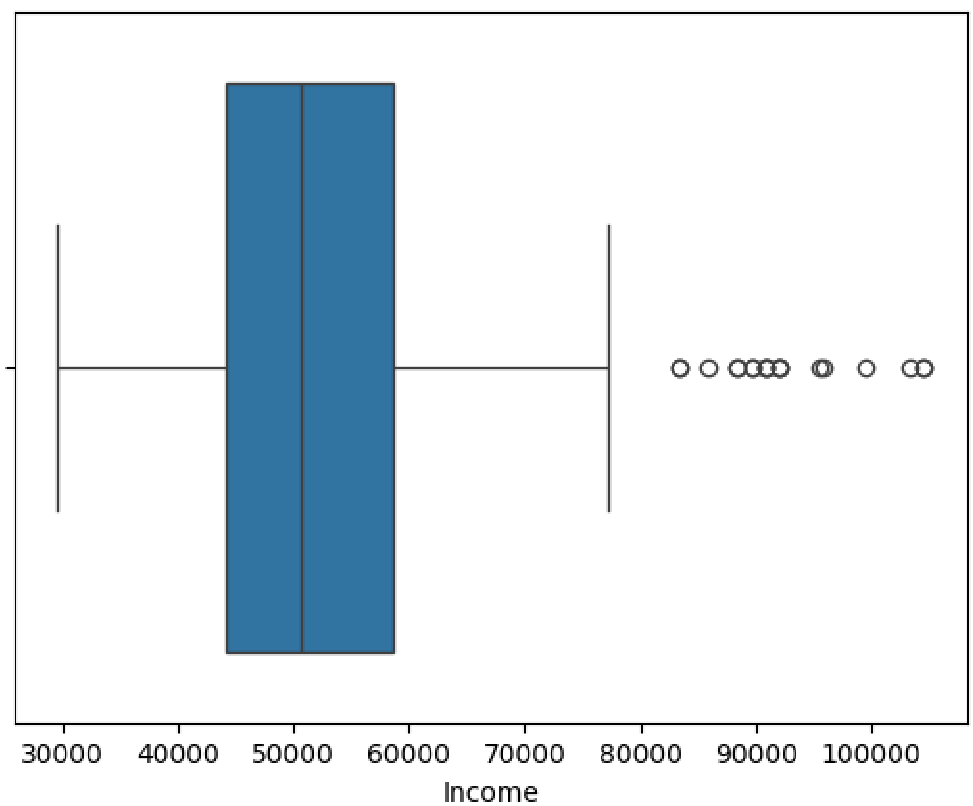
[62]:
```
60522.25
```

IQR = 60522.25, which means that middle 50% of the data lies in the range of 60522.25.

[63]:
```
# Max income - min income
104581-29662
```

[63]:
```
74919
```

[64]:
```
# BOX plot,  This is the graphical representation of above analysis
sns.boxplot(data=fit['Income'], orient="h")
plt.show()
```



All the values outside the whisker are considered "Outlier"

[66]:
```
fit['Miles'].describe()
```

[66]:
```
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
```

avarage Miles in the database is 103.19 Minimun Miles in the dataset is 21

This value 25% indicates that 25% of all the values present in the 'Miles' series is less than 66 This value 50% indicates that 50% of all the values present in the 'Miles' series is less than 94
This value 75% indicatesthat 75% of all the values present in the 'Miles' series is less than 114.75. Maximum Miles in the dataset is 360

[67]:
```
#IQR = Inter Quertile Range  = value(75%)- value(25%)
IQR =114.75-66
IQR
```

[67]:
```
48.75
```

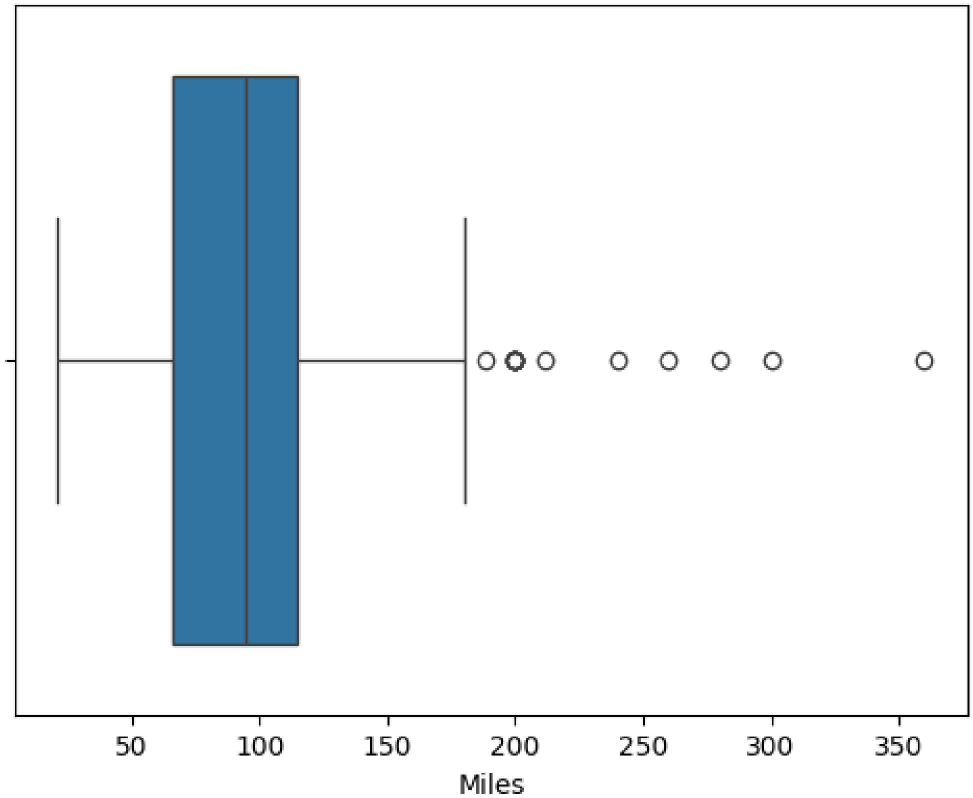
IQR = 48.75, which means that middle 50% of the data lies in the range of 48.75

[68]:
```
# max miles - min miles
360-21
```

[68]:
```
339
```

the normal range is high i.e. 339 We can observe one thing that there is Outlier present in the dataset This is why the normal range is getting affected by the outlier.

[69]:
```
# BOX plot,  This is the graphical representation of above analysis
sns.boxplot(data=fit['Miles'], orient="h")
plt.show()
```



All the values outside the whisker are considered “Outlier”

RECOMENDATION :-
1. Product KP281 is on demand, recomended to increase production 2. people in the range of 23 to 26 ages buyinng these products most 3. Male coustomer is higher than female 4. people with 15 and 14 years of education bought the product most 5. customer being partnered is higher than customer being single.

```
[ ]:
```