aerofit-case-study

August 11, 2023

[ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from google.colab import drive
from scipy.stats import norm,binom,geom
```

[ ]:
```
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
df = pd.read_csv('/content/drive/My Drive/Aerofit.csv')
df
```

[ ]:
```
    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]
```

[ ]:
```
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
```

[ ]:
```
df.shape
```

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

[ ]:
```
df.size
```

[ ]:
```
1620
```

[ ]:
```
df.count()
```

[ ]:
```
Product          180
Age              180
Gender           180
Education        180
MaritalStatus    180
Usage            180
Fitness          180
Income           180
Miles            180
dtype: int64
```

[ ]:
```
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.tail()
```

[ ]:
```
    Product  Age Gender  Education MaritalStatus  Usage  Fitness  Income  \
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
175    200
176    200
177    160
178    120
179    180
```

[ ]:
```
df.isnull().sum() # We can see there are no null values
```

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

[ ]:
```
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
```

[ ]:
```
desc=df["Age"].describe()
desc
```

[ ]:
```
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
```

[ ]:
```
Age_25=np.percentile(df["Age"],25) #Q1 Percentile
Age_25
```

[ ]: 24.0

[ ]:
```
Age_50=np.percentile(df["Age"],50) #Q2 Percentile
Age_50
```

[ ]: 26.0

[ ]:
```
Age_75=np.percentile(df["Age"],75)   #Q3 Percentile
Age_75
```

[ ]: 33.0

[ ]:
```
df["Age"].quantile(.25)
```

[ ]: 24.0

[ ]:
```
df["Age"].quantile(.50)
```

[ ]: 26.0

[ ]:
```
df["Age"].quantile(.75)
```

[ ]: 33.0

[ ]:
```
Age_range=df["Age"].max()-df["Age"].min()
Age_range
```

[ ]: 32

[ ]:
```
Age_IQR=Age_75-Age_25
Age_IQR
```

[ ]: 9.0

[ ]:
```
Age_lower_whisker=max((Age_25-(1.5*Age_IQR)),desc["min"])
Age_lower_whisker
```

[ ]: 18.0

[ ]:
```
Age_upper_whisker=min((Age_75+(1.5*Age_IQR)),desc["max"])
Age_upper_whisker
```

[ ]: 46.5

[ ]:
```
Age_outlier1=df.loc[df["Age"]>Age_upper_whisker]
Age_outlier1
```

[ ]:
```
    Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
78    KP281   47    Male         16     Partnered      4        3   56850
79    KP281   50  Female         16     Partnered      3        3   64809
139   KP481   48    Male         16     Partnered      2        3   57987
178   KP781   47    Male         18     Partnered      4        5  104581
179   KP781   48    Male         18     Partnered      4        5   95508

     Miles
78      94
79      66
139     64
178    120
179    180
```

[ ]:
```
Age_outlier2=df.loc[df["Age"]<Age_lower_whisker]
Age_outlier2
```

[ ]:
```
Empty DataFrame
Columns: [Product, Age, Gender, Education, MaritalStatus, Usage, Fitness,
Income, Miles]
```

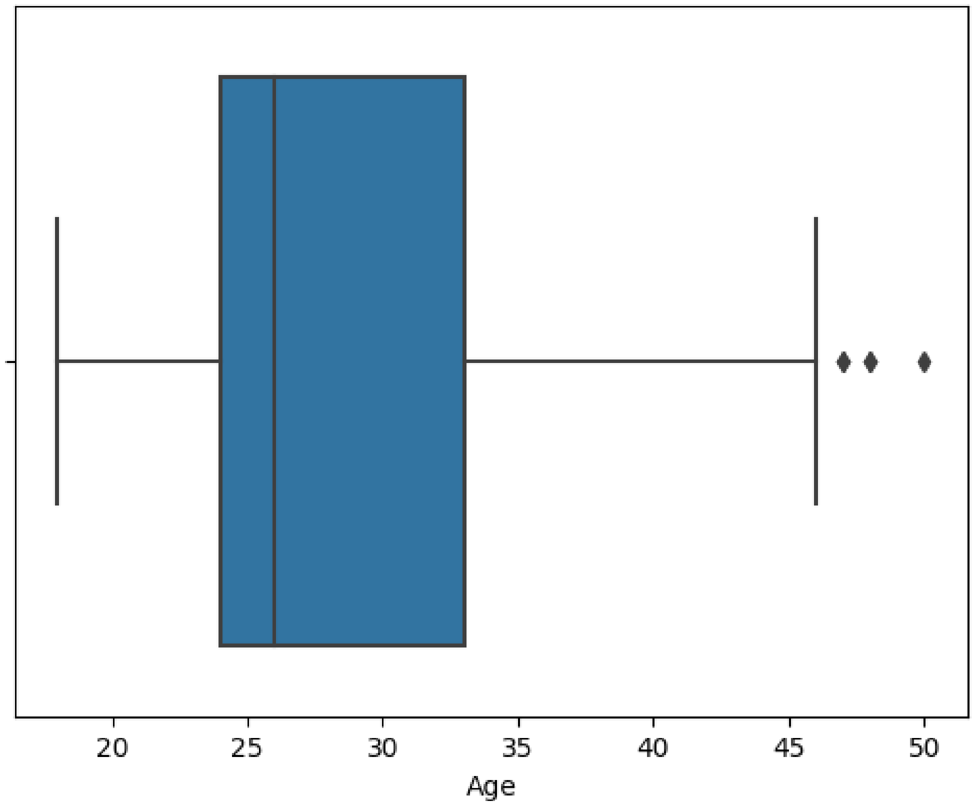
Index: []

[ ]:
```
Age_outlier1.shape[0]
```

[ ]: 5

[ ]:
```
sns.boxplot(x=df["Age"])
```

[ ]: <Axes: xlabel='Age'>



[ ]:
```
df.columns
```

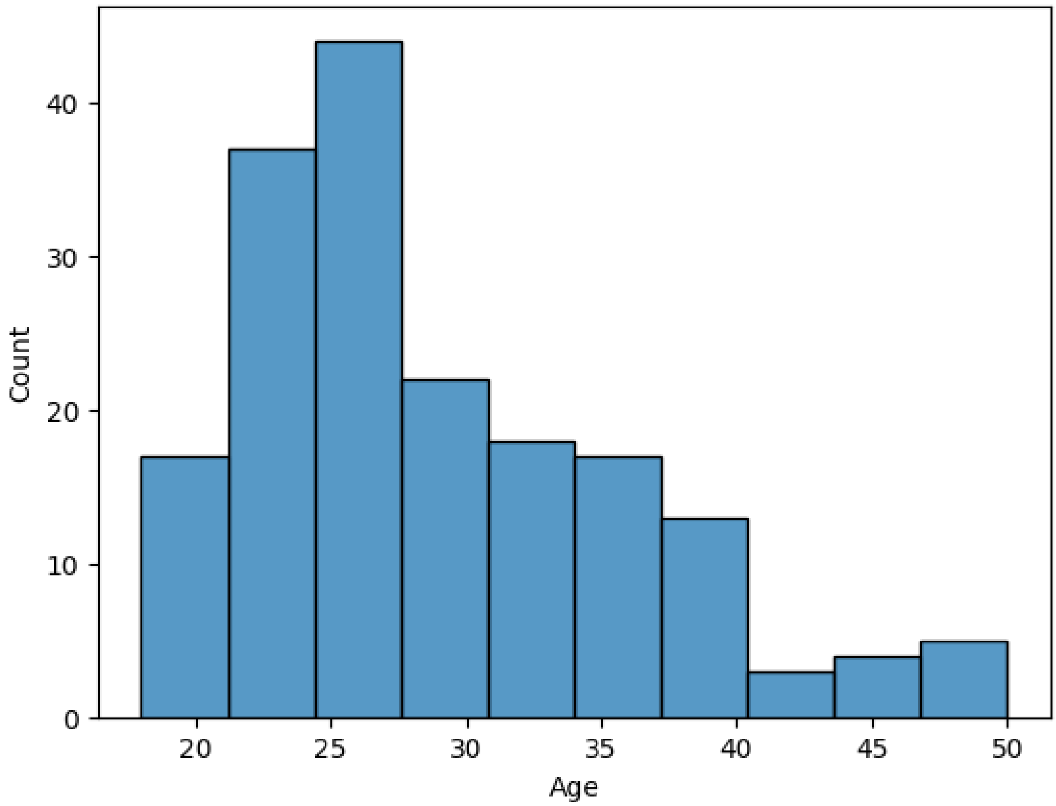
[ ]:
```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

[ ]:
```
np.unique(df["Product"])
```

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

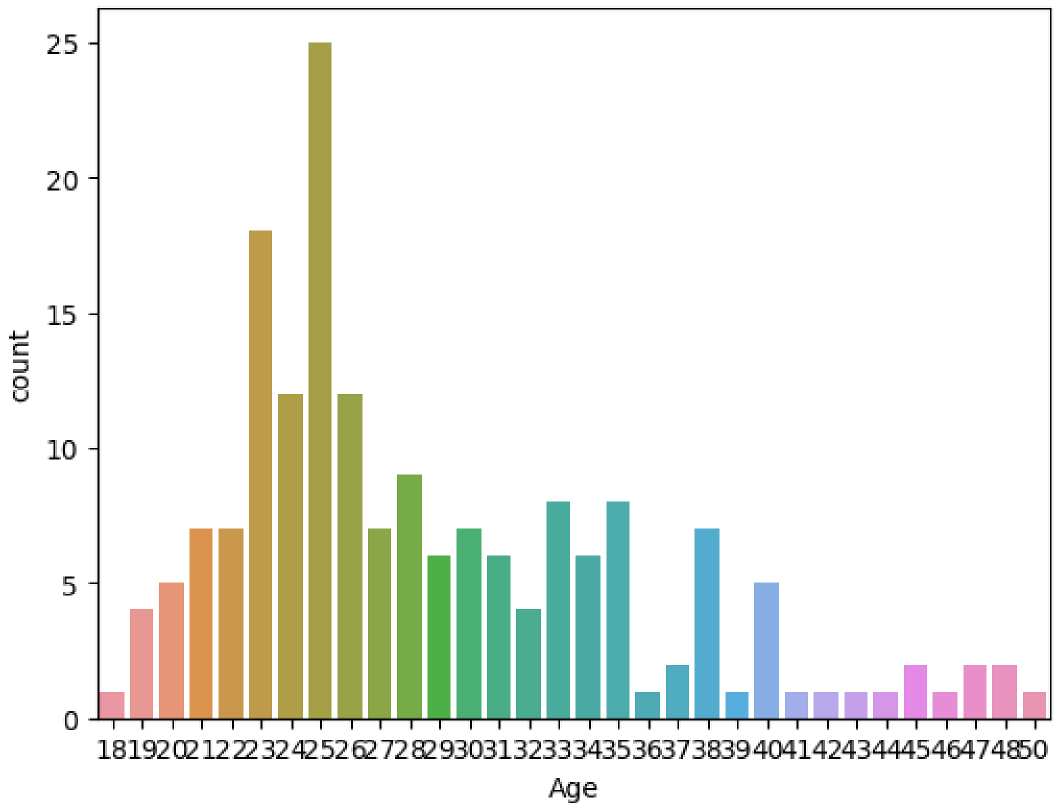
[ ]:
```
sns.histplot(df["Age"], bins=10)
```

[ ]: <Axes: xlabel='Age', ylabel='Count'>



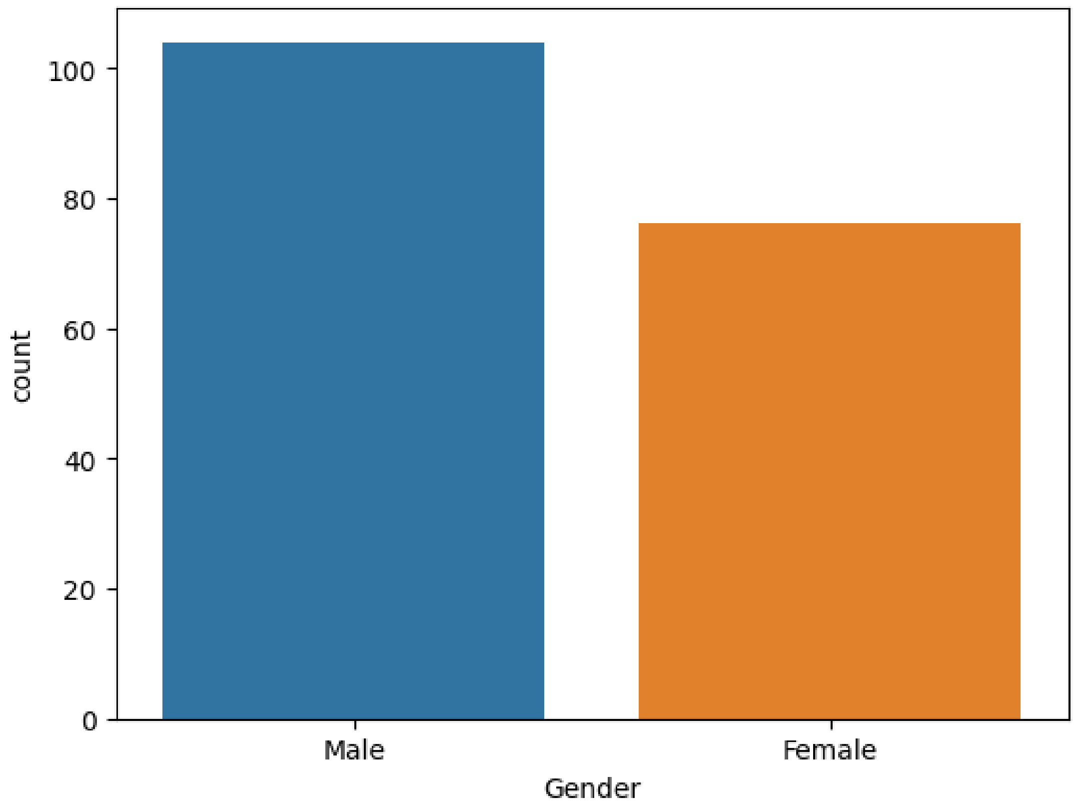

```
[ ]: sns.countplot(x="Age",data=df) #we can see that 23-26 age group people buys␣
     ↪more number of threadmills
```

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



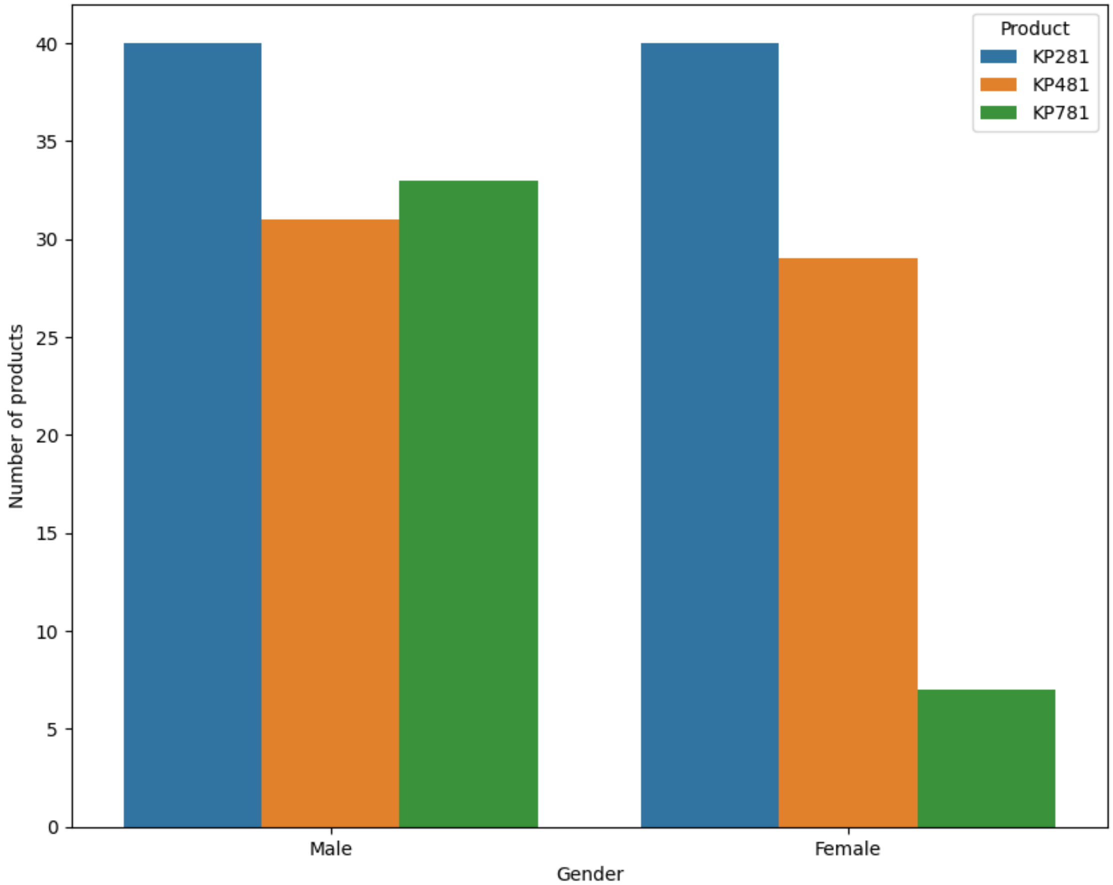

```
[ ]: sns.countplot(x="Gender",data=df) #Female are less interested in fitness when
     ↪compared to male or females are buying less number of threadmills when
     ↪compared to Males
```

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



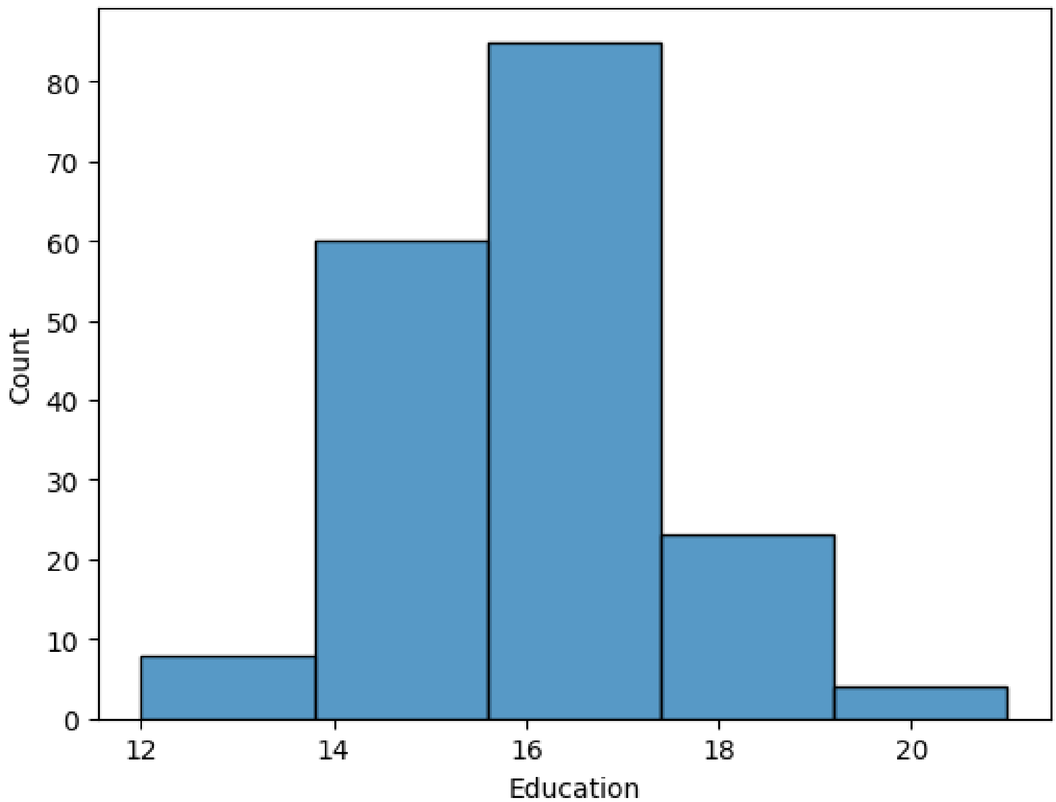

```
[ ]: plt.figure(figsize=(10,8))
     sns.countplot(x='Gender',hue='Product',data=df)  #Men are buying almost all␣
      ↪types of threadmills when compared to female.Females are buying mostly KP281␣
      ↪and buying less KP781.Maybe they do nto want to spend much on fitness/
      ↪thredmill
     plt.ylabel('Number of products')
```

```
[ ]: Text(0, 0.5, 'Number of products')
```




```
[ ]: sns.histplot(df["Education"],bins=5) #who studied average number of years are
     ↪also intersetd in fitness.Who studies much or who studies less are not into
     ↪fitness.
```

```
[ ]: <Axes: xlabel='Education', ylabel='Count'>
```



[ ]:
```
MaritalStatus_vs_Product=df.groupby(["MaritalStatus","Product"])[["Product"]].
 ↪agg("count")#
MaritalStatus_vs_Product.coumns="product_count"
MaritalStatus_vs_Product
#Families tend to buy more products than singles.anyway,both are buying more␣
 ↪products from KP281 as it is cheaper than remaining 2 items
```

[ ]:

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

[ ]:
```
df.columns
```

[ ]:
```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

[ ]:
```
pd.crosstab(index=df["Gender"],columns=df["Product"],margins=True)
```

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

[ ]:
```
##Probabiity of male buying KP281
print(40/104)
```

```
0.38461538461538464
```

[2]:
```
##Probabiity of female buying KP781
print(7/76)
```

```
0.09210526315789473
```

#Probability of KP281 sales out of 180

[ ]:
```
Prob_KP281=80/180
Prob_KP281
```

[ ]:
```
0.4444444444444444
```

[ ]:
```
prob_KP481=60/180
prob_KP481
```

[ ]:
```
0.3333333333333333
```

[ ]:
```
prob_KP781=40/180
prob_KP781
```

[ ]:
```
0.2222222222222222
```

Probability of buying is more for the product KP281

[ ]:
```

```

Income vs Product(KP281,KP481 and KP781)

[ ]:
```
np.unique(df["Income"])
```

[ ]:
```
array([ 29562,  30699,  31836,  32973,  34110,  35247,  36384,  37521,
        38658,  39795,  40932,  42069,  43206,  44343,  45480,  46617,
        47754,  48556,  48658,  48891,  49801,  50028,  51165,  52290,
        52291,  52302,  53439,  53536,  54576,  54781,  55713,  56850,
        57271,  57987,  58516,  59124,  60261,  61006,  61398,  62251,
        62535,  64741,  64809,  65220,  67083,  68220,  69721,  70966,
```

```
        74701,  75946,  77191,  83416,  85906,  88396,  89641,  90886,
        92131,  95508,  95866,  99601, 103336, 104581])
```

[ ]:
```
df["New_income"]=pd.cut(df["Income"],bins=[0,40000,80000,np.
 ↪inf],labels=["low","medium","high"])
df["New_income"].unique()
```

[ ]:
```
['low', 'medium', 'high']
Categories (3, object): ['low' < 'medium' < 'high']
```

[ ]:
```
df
```

[ ]:
```
    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 New_income
0      112        low
1       75        low
2       66        low
3       85        low
4       47        low
..     …          …
175    200       high
176    200       high
177    160       high
178    120       high
179    180       high

[180 rows x 10 columns]
```

[ ]:
```
df2=pd.crosstab(index=df["New_income"],columns=df["Product"],margins=True)
df2
```

[ ]:
```
Product     KP281  KP481  KP781  All
New_income
low            23      9      0   32
medium         57     51     21  129
```

```
high              0       0      19    19
All              80      60      40   180
```

1 Joint Probability

2 #Probability of low and KP281,KP481,KP781

[ ]:
```python
prob_of_low_KP281=23/180
print(prob_of_low_KP281)
prob_of_low_KP481=9/180     #customers having low income has less probability
 ↪to buy products
print(prob_of_low_KP481)
prob_of_low_KP781=0/180
print(prob_of_low_KP781)
```

```
0.12777777777777777
0.05
0.0
```

[ ]:
```python

```

#Probability of medium and KP281,KP481,KP781

[ ]:
```python
prob_of_medium_KP281=57/180
print(prob_of_medium_KP281)
prob_of_medium_KP481=51/180     #probability of customers having medium income
 ↪is more compared to low and high income customers
print(prob_of_medium_KP481)
prob_of_medium_KP781=21/180
print(prob_of_medium_KP781)
```

```
0.31666666666666665
0.2833333333333333
0.11666666666666667
```

3 Probability of High and KP281,KP481,KP781

[ ]:
```python
prob_of_high_KP281=0/180
print(prob_of_high_KP281)
prob_of_high_KP481=0/180     #probability of customers having High income has
 ↪less probability to buy aerofit products when compared to the customers
 ↪having less and medium income
print(prob_of_high_KP481)
prob_of_high_KP781=19/180
print(prob_of_high_KP781)
```

```
0.0
0.0
0.10555555555555556
```

4 Conditional Probability

[ ]:
```
df2
```

[ ]:
```
Product      KP281  KP481  KP781  All
New_income
low             23      9      0   32
medium          57     51     21  129
high             0      0     19   19
All             80     60     40  180
```

[ ]:
```
prob_of_KP281_given_low_income=23/32
print(prob_of_KP281_given_low_income)
prob_of_KP481_given_low_income=9/32           #People having low income tend to␣
↪buy KP281 more and they have zero probability to buy KP781
print(prob_of_KP481_given_low_income)
prob_of_KP781_given_low_income=0/32
print(prob_of_KP781_given_low_income)
```

```
0.71875
0.28125
0.0
```

[ ]:
```
prob_of_KP281_given_medium_income=57/129
print(prob_of_KP281_given_medium_income)
prob_of_KP481_given_medium_income=51/129          #People having medium income␣
↪showing interest to but all three products and but buys KP281 mostly
print(prob_of_KP481_given_medium_income)
prob_of_KP781_given_medium_income=21/129
print(prob_of_KP781_given_medium_income)
```

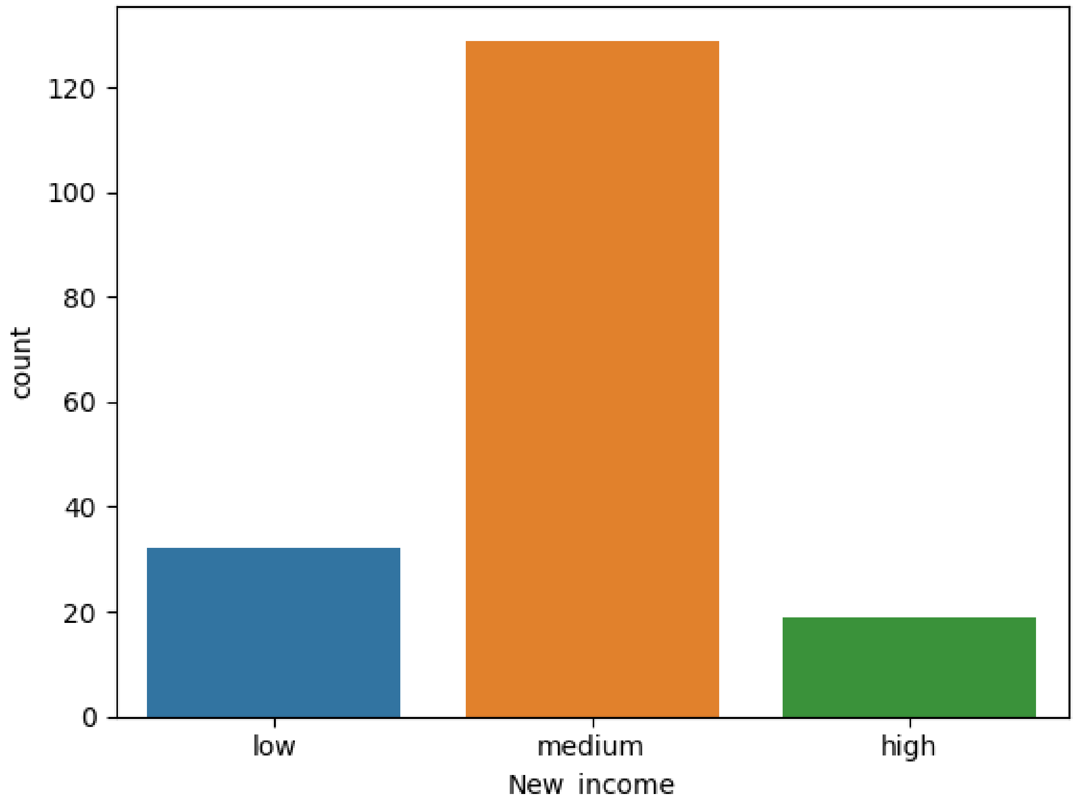
```
0.4418604651162791
0.3953488372093023
0.16279069767441862
```

[ ]:
```
prob_of_KP281_given_high_income=0/19
print(prob_of_KP281_given_high_income)
prob_of_KP481_given_high_income=0/19          #People having high income buys␣
↪KP781 100% and they are not intersted in buying KP481 and KP781
print(prob_of_KP481_given_high_income)
prob_of_KP781_given_high_income=19/19
print(prob_of_KP781_given_high_income)
```

```
0.0
0.0
1.0
```

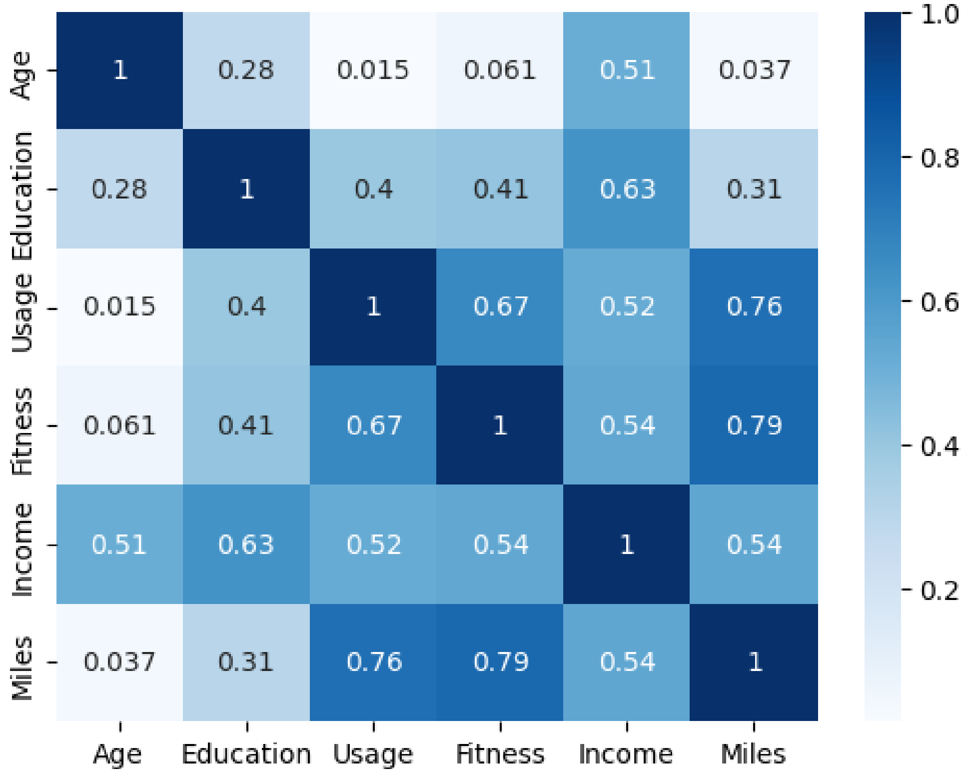
[ ]:
```
sns.countplot(x="New_income",data=df) #Aerofit is getting more customers from␣
↪medium income range
```

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



[ ]:
```
sns.heatmap(df.corr(), cmap= "Blues", annot=True)
plt.show()
```

```
<ipython-input-41-5cc09f5a9639>: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= "Blues", annot=True)
```



From the above heat map we can conclude that there is a strong correlation between

1. Income and Age
2. Education and Income
3. Usage and fitnes
4. fitness and miles

5 Insights

- Cusomers coming to the Aerofit are having mean age “28”.
- 25% of the customer are less than age “24”.
- 50%of the customers are having age less than “26”.
- 75% of the customers are less than the age of “33”.
- Customers having age between 18-50 will come to Aerofit.
- customers use the threadmill “3” times a week in an average.
- Average fitness of customers is “3” out of 5.
- We see that 23-26 age group people buys more number of threadmills.
- Female are less interested in fitness when compared to male or females are buying less number of threadmills when compared to Males.
- Men are buying almost all types of threadmills when compared to females.Females are buying mostly KP281 and buying less KP781.Maybe they do not want to spend much on fitness/threadmill.

- Who studied average(15-17) number of years are much intersetd in fitness.Who studies much or who studies less are not into fitness.
- Families tend to buy more products than singles.anyway,both are buying more products from KP281 as it is cheaper than remaining 2 items.
- Probability of buying KP281 is more when compared to KP481 and KP781.
- KP781 has least probability of buying.
- customers having less and high income has less probability to buy products.
- probability of buying is more for customers having medium income compared to the customers having less and high .
- People having low income tend to buy KP281 more and they have zero probability to buy KP781.
- People having medium income showing interest to but all three products and but buys KP281 mostly.
- Customers having high income are buying only KP781.
- Aerofit is getting more customers having medium range income.
- There is strong correlation between
  1. Income and Age
  2. Education and Income
  3. Usage and fitnes
  4. fitness and miles

6 Recommendations

- Females are showing less interest to buy the products.So we should do some thing which attracts women.For that we should try coming up with new products which will help in women health speacially.

- Partnered customers are buying more number of products than the single customers.So we can say that singles are not able to afford the products.So we should come up with the products where singles also can buy.

- KP281 has more probability of selling than KP481 and KP781.KP781 has least probability of selling.So we should either decrease the price of KP781 like once in a blue moon to attract the customers or We should make it more popular through different media about its features or we should try enhancing its features more.

- Targetting the low income customers we should come up with the renatls of the fitness equipment.So that they can also give a thought to it to buy and improve their fitness/health

- Targetting the high income customers,Here we have 2 cases 1.Customers with no time. *For this we can tie up with the coorporate companies or business administrators to buy our euipment in their offices.So that they also can give some time to health and fitness so to us. 2.Old age customers.* For the old age customers we should come up with very gentle products which are very safe and easy to use.

```
[ ]:
```