aerofit-project

July 30, 2023

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

[ ]:
```
mydata = pd.read_csv('original_aerofit_treadmill.csv')
mydata
```

[ ]:
```
    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]
```

[ ]:
```
mydata.head(10)
```

[ ]:

	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	KP281	20	Female	14	Partnered	3	3	32973	66
6	KP281	21	Female	14	Partnered	3	3	35247	75
7	KP281	21	Male	13	Single	3	3	32973	85
8	KP281	21	Male	15	Single	5	4	35247	141
9	KP281	21	Female	15	Partnered	2	3	37521	85

[ ]:
```
mydata.shape
```

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

[ ]:
```
mydata.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

[ ]:
```
mydata.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
```

[ ]:
```
print('below are the unique values for categorical columns')
print("product :", mydata['Product'].unique())
print('Gender:', mydata['Gender'].unique())
print('Education:', mydata['Education'].unique())
print('MaritalStatus:', mydata['MaritalStatus'].unique())
print('Usage:', mydata['Usage'].unique())
print('Fitness:', mydata['Fitness'].unique())
```

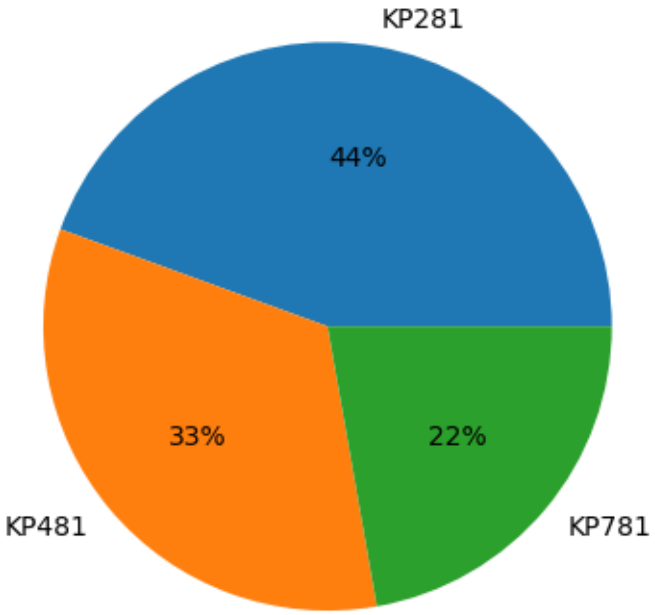
```
below are the unique values for categorical columns
product : ['KP281' 'KP481' 'KP781']
Gender: ['Male' 'Female']
Education: [14 15 12 13 16 18 20 21]
MaritalStatus: ['Single' 'Partnered']
Usage: [3 2 4 5 6 7]
Fitness: [4 3 2 1 5]
```

[ ]:
```
mydata['Product'].value_counts()
```

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

[ ]:
```
labels=  mydata['Product'].unique()
plt.pie(mydata['Product'].value_counts(), labels = labels, autopct='%.0f%%')
```

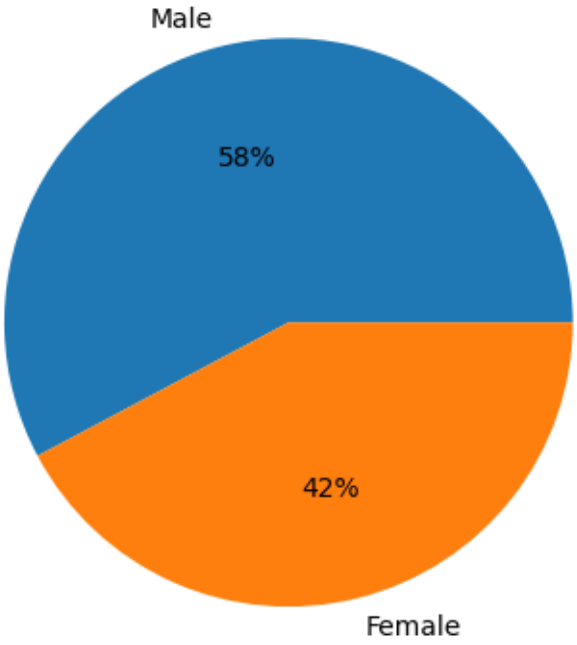
[ ]:
```
([<matplotlib.patches.Wedge at 0x7fa45c4f80a0>,
  <matplotlib.patches.Wedge at 0x7fa45c6c7f70>,
  <matplotlib.patches.Wedge at 0x7fa45c4f8d90>],
 [Text(0.19101298416420226, 1.083288530300532, 'KP281'),
  Text(-0.8426488506529132, -0.7070664144854603, 'KP481'),
  Text(0.8426489499534077, -0.7070662961437348, 'KP781')],
 [Text(0.10418890045320121, 0.5908846528911992, '44%'),
  Text(-0.4596266458106798, -0.38567258971934193, '33%'),
  Text(0.45962669997458594, -0.3856725251693099, '22%')])
```



We found that 44% people bought KP281

[ ]:
```
labels=  mydata['Gender'].unique()
plt.pie(mydata['Gender'].value_counts(), labels = labels, autopct='%.0f%%')
```

[ ]:
```
([<matplotlib.patches.Wedge at 0x7fa45c52aec0>,
  <matplotlib.patches.Wedge at 0x7fa45c52add0>],
 [Text(-0.2661141695451412, 1.0673252778639228, 'Male'),
  Text(0.26611426947534417, -1.0673252529485115, 'Female')],
 [Text(-0.14515318338825883, 0.5821774242894124, '58%'),
  Text(0.14515323789564225, -0.582177410699188, '42%')])
```

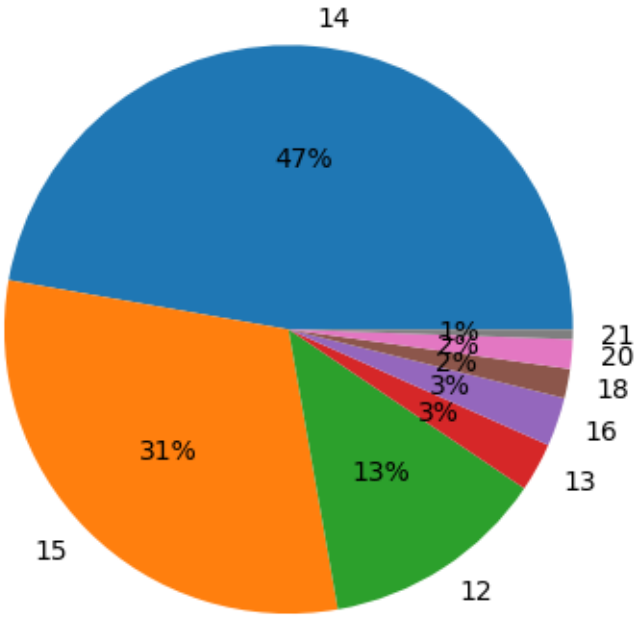


males usage are highest compare to females

```
[ ]: labels=  mydata['Education'].unique()
     plt.pie(mydata['Education'].value_counts(), labels = labels, autopct='%.0f%%')
```

```
[ ]: ([<matplotlib.patches.Wedge at 0x7fa45c5858d0>,
       <matplotlib.patches.Wedge at 0x7fa45c5857e0>,
       <matplotlib.patches.Wedge at 0x7fa45c5865c0>,
       <matplotlib.patches.Wedge at 0x7fa45c586c50>,
       <matplotlib.patches.Wedge at 0x7fa45c5872e0>,
       <matplotlib.patches.Wedge at 0x7fa45c587970>,
       <matplotlib.patches.Wedge at 0x7fa45c5c4040>,
       <matplotlib.patches.Wedge at 0x7fa45c5c46d0>],
      [Text(0.09587136262132485, 1.0958141639115322, '14'),
       Text(-0.7778175321297253, -0.7778173864806726, '15'),
       Text(0.5991028636588824, -0.9225376733530868, '12'),
       Text(0.9620816471323494, -0.5332906376930929, '13'),
       Text(1.0400704129946297, -0.35812502846517263, '16'),
       Text(1.07978989068398, -0.20988995206221503, '18'),
       Text(1.0958141633505245, -0.09587136903367063, '20'),
       Text(1.0998324638213786, -0.01919769581476827, '21')],
      [Text(0.05229347052072264, 0.5977168166790175, '47%'),
       Text(-0.4242641084343956, -0.4242640289894577, '31%'),
       Text(0.3267833801775722, -0.5032023672835018, '13%'),
```

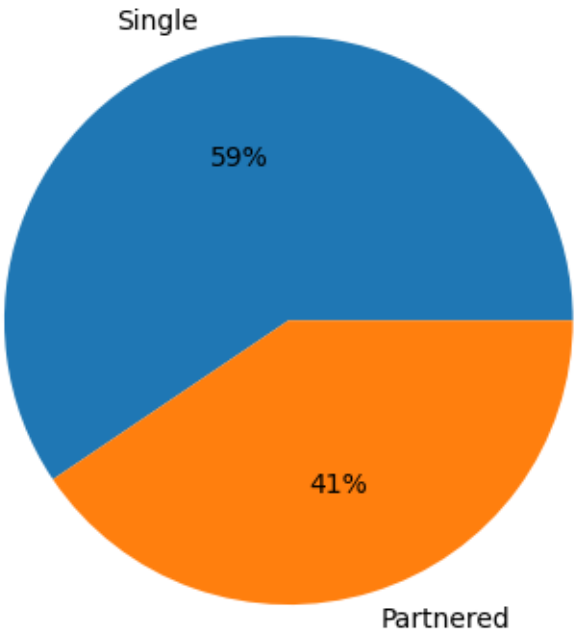
```
Text(0.524771807526736, -0.2908858023780506, '3%'),
Text(0.567311134360707, -0.19534092461736688, '3%'),
Text(0.5889763040094436, -0.11448542839757182, '2%'),
Text(0.5977168163730132, -0.052293474018365795, '2%'),
Text(0.5999086166298427, -0.010471470444419055, '1%')])
```



14 years of educated uses 47% of treadmill which is highest, whereas 21 years of educated uses 1% of treadmill

[ ]:
```
labels=  mydata['MaritalStatus'].unique()
plt.pie(mydata['MaritalStatus'].value_counts(), labels = labels, autopct='%.
  ↪0f%%')
```

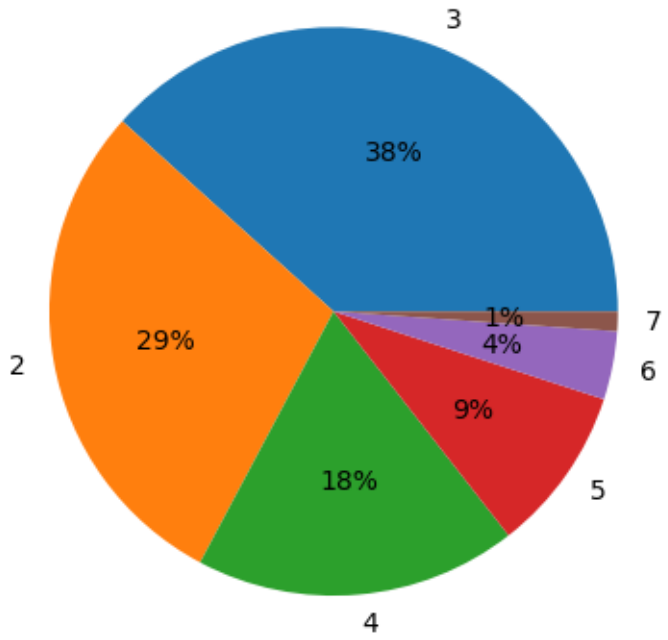
[ ]:
```
([<matplotlib.patches.Wedge at 0x7fa45c3f7a00>,
  <matplotlib.patches.Wedge at 0x7fa45c3f7910>],
 [Text(-0.3216089058361209, 1.0519352221914111, 'Single'),
  Text(0.32160890583612123, -1.051935222191411, 'Partnered')],
 [Text(-0.175423039546975, 0.5737828484680423, '59%'),
  Text(0.17542303954697522, -0.5737828484680423, '41%')])
```



the marital people contribute less in fitness activity as compare to single.

[ ]:
```
labels=  mydata['Usage'].unique()
plt.pie(mydata['Usage'].value_counts(), labels = labels, autopct='%.0f%%')
```

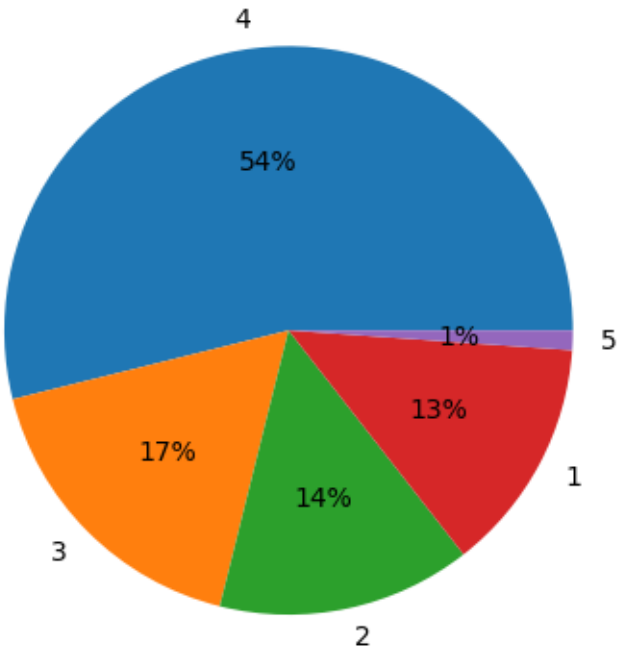
[ ]:
```
([<matplotlib.patches.Wedge at 0x7fa45c446770>,
  <matplotlib.patches.Wedge at 0x7fa45c446680>,
  <matplotlib.patches.Wedge at 0x7fa45c447220>,
  <matplotlib.patches.Wedge at 0x7fa45c4478b0>,
  <matplotlib.patches.Wedge at 0x7fa45c447f40>,
  <matplotlib.patches.Wedge at 0x7fa45c488610>],
 [Text(0.39420477013954197, 1.0269384593047586, '3'),
  Text(-1.083288530300532, -0.19101298416420204, '2'),
  Text(0.09587136262132409, -1.0958141639115324, '4'),
  Text(0.9010672864585332, -0.6309340260869245, '5'),
  Text(1.0797899164763236, -0.20988981937210427, '6'),
  Text(1.0993299126089058, -0.038389363674011634, '7')],
 [Text(0.2150207837124774, 0.5601482505298682, '38%'),
  Text(-0.5908846528911992, -0.1041889004532011, '29%'),
  Text(0.052293470520722224, -0.5977168166790177, '18%'),
  Text(0.49149124715919984, -0.34414583241104973, '9%'),
  Text(0.5889763180779946, -0.11448535602114777, '4%'),
  Text(0.5996344977866759, -0.02093965291309725, '1%')])
```



very few people uses treadmill whole week whereas they prefered 3 days for treadmill

[ ]:
```
labels=  mydata['Fitness'].unique()
plt.pie(mydata['Fitness'].value_counts(), labels = labels, autopct='%.0f%%')
```

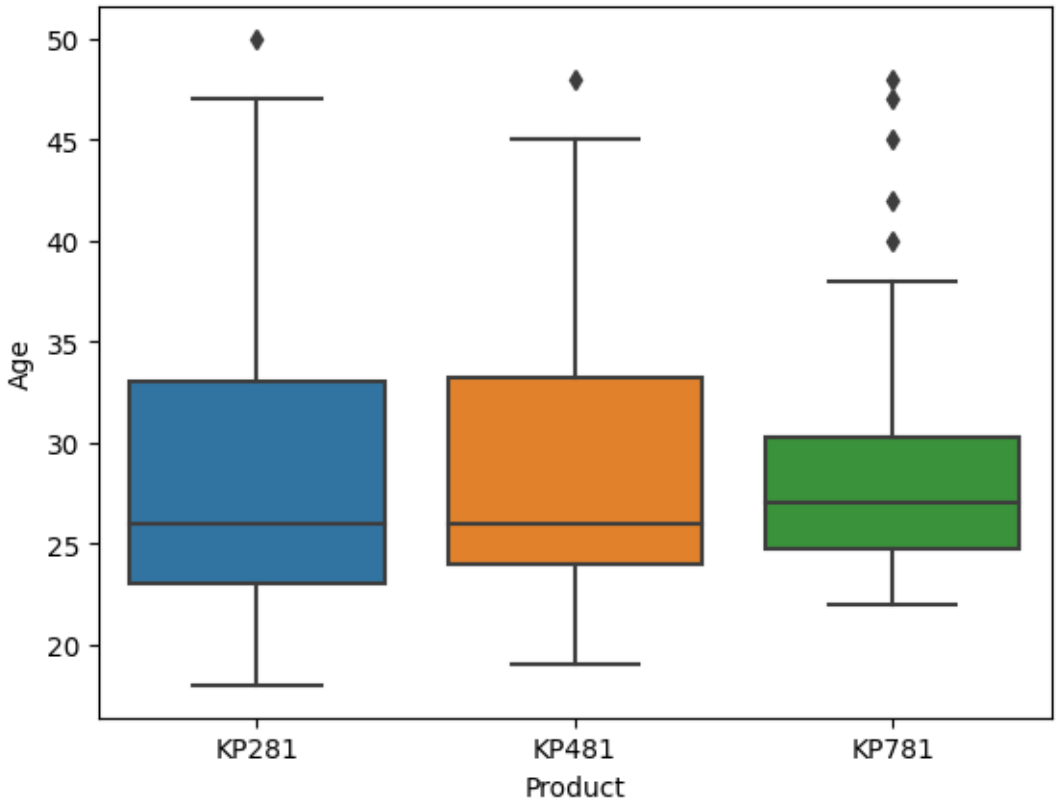
[ ]:
```
([<matplotlib.patches.Wedge at 0x7fa45c4b2650>,
  <matplotlib.patches.Wedge at 0x7fa45c4b3550>,
  <matplotlib.patches.Wedge at 0x7fa45c2ec250>,
  <matplotlib.patches.Wedge at 0x7fa45c2ec8e0>,
  <matplotlib.patches.Wedge at 0x7fa45c2ecf70>],
 [Text(-0.13405621868419756, 1.0918007740572888, '4'),
  Text(-0.7778175321297253, -0.7778173864806726, '3'),
  Text(0.22870279609825295, -1.0759623743685656, '2'),
  Text(0.9712423430119265, -0.5164187362409536, '1'),
  Text(1.0993299099131995, -0.03838944086903273, '5')],
 [Text(-0.07312157382774412, 0.5955276949403393, '54%'),
  Text(-0.4242641084343956, -0.4242640289894577, '17%'),
  Text(0.12474697968995614, -0.5868885678373993, '14%'),
  Text(0.529768550733778, -0.2816829470405201, '13%'),
  Text(0.5996344963162905, -0.020939695019472397, '1%')])
```



Only 1% folks gives 5 star rating for fitness, on the other side 54% of folks gives 4 star rating.

[ ]:
```
sns.boxplot(data= mydata, x='Product', y='Age')
```

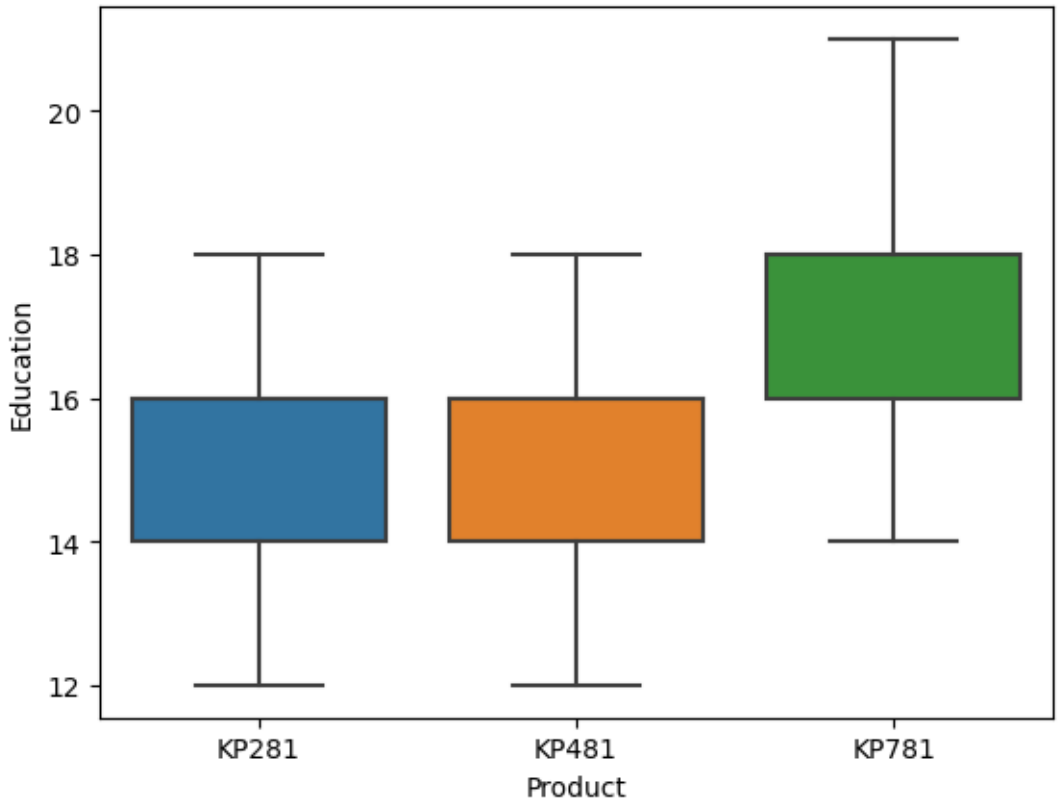
[ ]:
```
<Axes: xlabel='Product', ylabel='Age'>
```



the usage of KP281 and KP481 having same age group where 27 years old people uses KP781

[ ]:
```
sns.boxplot(data= mydata, x='Product', y='Education')
```

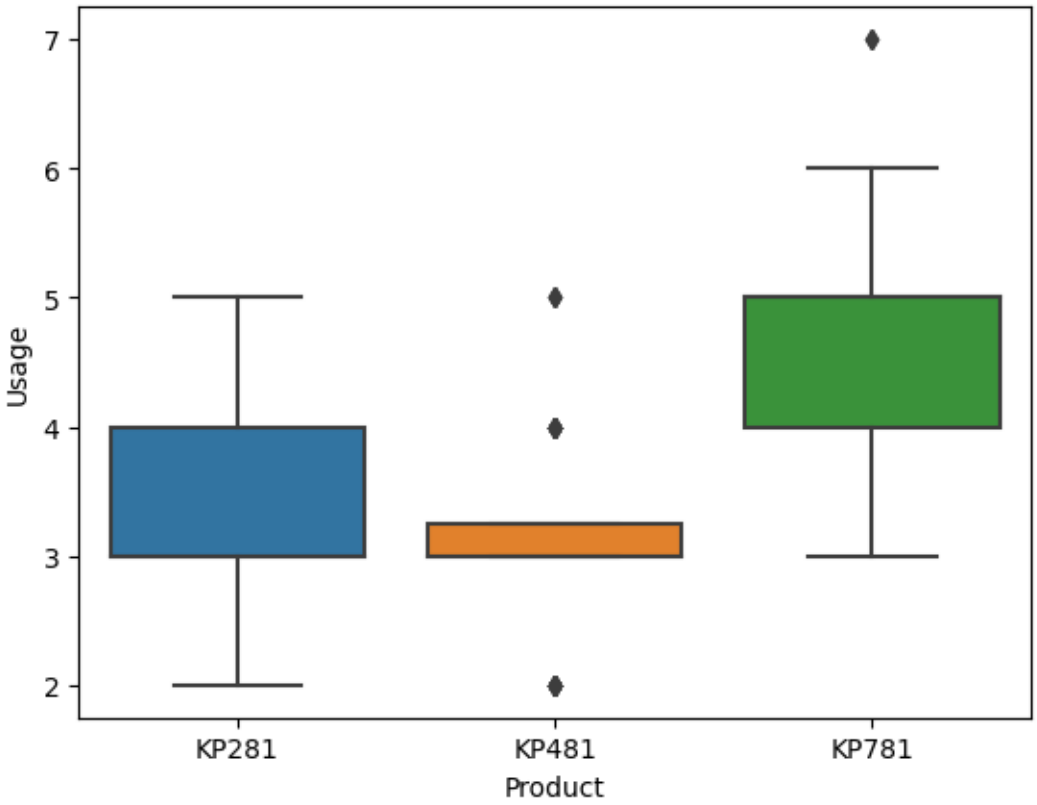
[ ]:
```
<Axes: xlabel='Product', ylabel='Education'>
```



educated people preffers KP781 treadmill

[ ]:
```
sns.boxplot(data= mydata, x='Product', y='Usage')
```

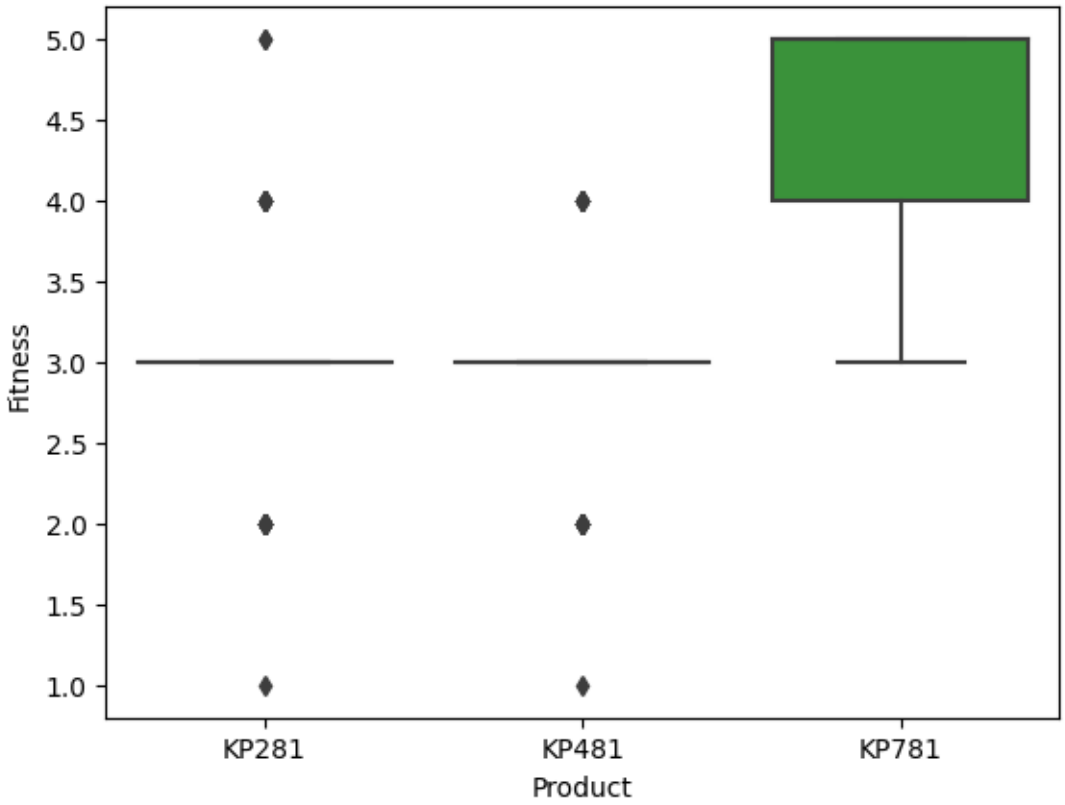
[ ]:
```
<Axes: xlabel='Product', ylabel='Usage'>
```



among the above three treadmills KP781 having highest usage

[ ]:
```
sns.boxplot(data= mydata, x='Product', y='Fitness')
```

[ ]:
```
<Axes: xlabel='Product', ylabel='Fitness'>
```



[ ]:
```
mydata.isnull().sum()
```

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

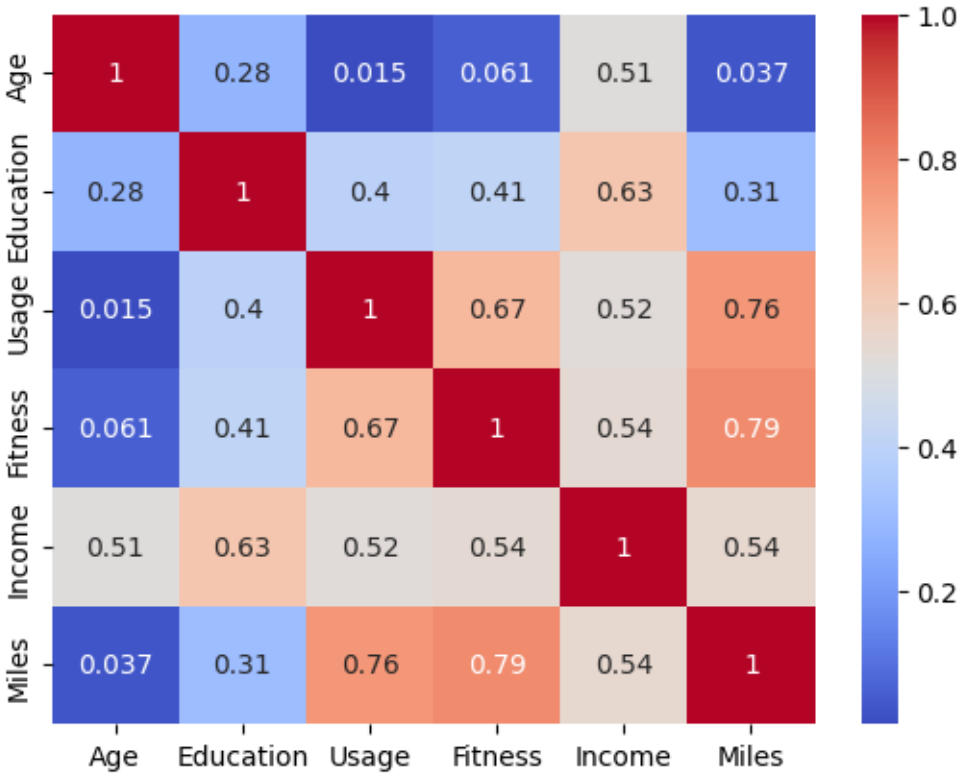
there is no null value in any colums

[ ]:
```
mydata_matrix= mydata.corr()
sns.heatmap(mydata_matrix, cmap='coolwarm', annot=True)
```

```
<ipython-input-50-8f4168fbeb85>: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.
```

```
mydata_matrix= mydata.corr()
```

[ ]: <Axes: >



1. Above the heatmap strongly correlates ages and income
2. highest usage correlates with miles
3. education correlates with income

#PROBABILITY

[ ]:
```
pd.crosstab(index=mydata['Product'], columns=mydata['MaritalStatus'],␣
↪margins=True)
```

[ ]:
```
MaritalStatus  Partnered  Single  All
Product
KP281                 48      32   80
KP481                 36      24   60
KP781                 23      17   40
All                  107      73  180
```

[59]:
```
print('probability of Partnered buying KP281 is ', 48/80)
print('probability of Partnered buying KP481 is ', 36/60)
print('probability of Partnered buying KP781 is ', 23/40)
```

```
probability of Partnered buying KP281 is  0.6
probability of Partnered buying KP481 is  0.6
probability of Partnered buying KP781 is  0.575
```

[60]:
```
pd.crosstab(index=mydata['Product'], columns=mydata['Usage'], margins=True)
```

[60]:

Usage	2	3	4	5	6	7	All
Product							
KP281	19	37	22	2	0	0	80
KP481	14	31	12	3	0	0	60
KP781	0	1	18	12	7	2	40
All	33	69	52	17	7	2	180

[68]:
```
print('probability of Highest Usage of KP281 is for usage = 3 is', round(37/
 ↪80,2))
print('probability of Highest Usage of KP481 is for usage = 3 is', round(31/
 ↪60,2))
print('probability of Highest Usage of KP781 is for usage = 4 is', round(18/
 ↪40,2))
```

```
probability of Highest Usage of KP281 is for usage = 3 is 0.46
probability of Highest Usage of KP481 is for usage = 3 is 0.52
probability of Highest Usage of KP781 is for usage = 4 is 0.45
```

[69]:
```
pd.crosstab(index=mydata['Product'], columns=mydata['Fitness'], margins=True)
```

[69]:

Fitness	1	2	3	4	5	All
Product						
KP281	1	14	54	9	2	80
KP481	1	12	39	8	0	60
KP781	0	0	4	7	29	40
All	2	26	97	24	31	180

[73]:
```
print('probability of Highest fitness of KP281 is 3 =', round(54/80,2))
print('probability of Highest fitness of KP481 is 3 =', round(39/60,2))
print('probability of Highest fitness of KP781 is 5 =', round(29/40,2))
```

```
probability of Highest fitness of KP281 is 3 = 0.68
probability of Highest fitness of KP481 is 3 = 0.65
probability of Highest fitness of KP781 is 5 = 0.72
```

[74]:
```
pd.crosstab(index=mydata['Product'], columns=mydata['Gender'], margins=True)
```

[74]:

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

```
All          76   104  180
```

[79]:
```
print('probability of Males using KP281 is ', round(40/80,2))
print('probability of females using KP281 is ', round(40/80,2))

print('probability of Males using KP481 is ', round(31/60,2))
print('probability of females using KP481 is ', round(29/60,2))

print('probability of Males using KP781 is ', round(33/40,2))
print('probability of females using KP781 is ', round(7/40,2))
```

```
probability of Males using KP281 is  0.5
probability of females using KP281 is  0.5
probability of Males using KP481 is  0.52
probability of females using KP481 is  0.48
probability of Males using KP781 is  0.82
probability of females using KP781 is  0.17
```

#BUSINESS INSIGHTS 1. We found that majority of people bought KP281 treadmill 2. The price of KP281 treadmill is also less compare to other two treadmills 3. Females are less active in terms of fitness specifically in treadmill. 4. The marital people contribute less in fitness activity as compare to single. 5. Only 1% folks gives 5 star rating for fitness, on the other side 54% of folks gives 4 star rating. 6. KP781 treadmill is most effective treadmill in terms of fitness

#RECOMMENDATION 1. Specifically giving discounts for females for maximum sales of treadmill. 2. The KP781 is have high effectivness, so we can reduce its price on some special occasion to sell more. 3. Giving discounts for adults and late adults, to attract them for buying treadmills. 4.Correlations of data miles and fitness are strongly matches, that indicates more to run more to be fit, so we can recommend treadmill fitness.

#CUSTEMER PROFILING

1.Product Purchased: KP281

Age: 26 Gender: male=0.5 female= 0.5 Education: 14-16 MaritalStatus: 0.6 Usage: 0.46 Fitness: 0.68

2.Product Purchased: KP481

Age: 26 Gender: male =0.5 female = 0.48 Education: 14-16 MaritalStatus: 0.6 Usage: 0.52 Fitness: 0.65

3.Product Purchased: KP781

Age: 27 Gender: male= 0.82 female= 0.17 Education: 16-18 MaritalStatus: 0.57 Usage: 0.45 Fitness: 0.72