In [50]:
```
import pandas as pd
import seaborn as sns
import numpy as np
from matplotlib import pyplot as plt
```

In [51]:
```
aerofit = pd.read_csv('Desktop/aerofit_treadmill.txt')
```

In [52]:
```
aerofit
```

Out[52]:

	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

In [53]:
```
aerofit.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: 10.6+ KB
```

In [54]:
```
aerofit.describe()
```

Out[54]:

	Age	Education	Usage	Fitness	Income	Miles
count	180.000000	180.000000	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	3.455556	3.311111	53719.577778	103.194444
std	6.943498	1.617055	1.084797	0.958869	16506.684226	51.863605
min	18.000000	12.000000	2.000000	1.000000	29562.000000	21.000000
25%	24.000000	14.000000	3.000000	3.000000	44058.750000	66.000000
50%	26.000000	16.000000	3.000000	3.000000	50596.500000	94.000000
75%	33.000000	16.000000	4.000000	4.000000	58668.000000	114.750000
max	50.000000	21.000000	7.000000	5.000000	104581.000000	360.000000

In [55]:
```
aerofit.notna().sum()
```

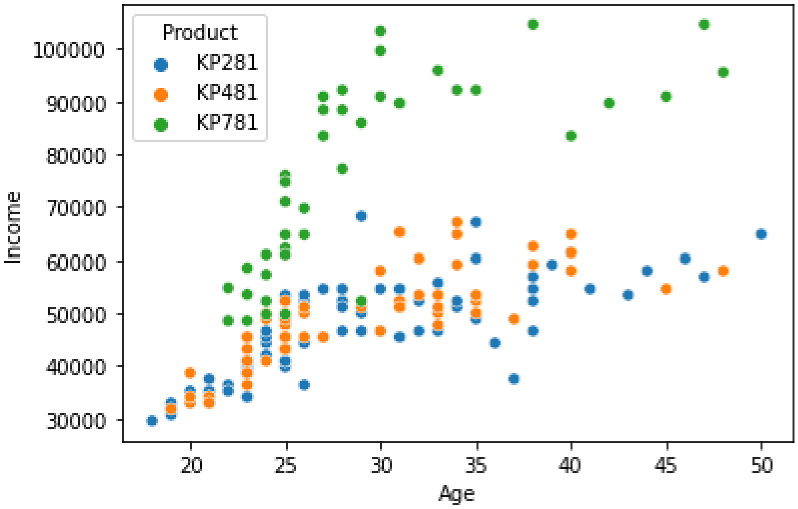
Out[55]:
```
Product          180
Age              180
Gender           180
Education        180
MaritalStatus    180
Usage            180
Fitness          180
Income           180
Miles            180
dtype: int64
```

In [56]:
```
aerofit.nunique()
```

Out[56]:
```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

In [57]:
```
sns.scatterplot(data=aerofit,x='Age',y='Income',hue='Product')
```

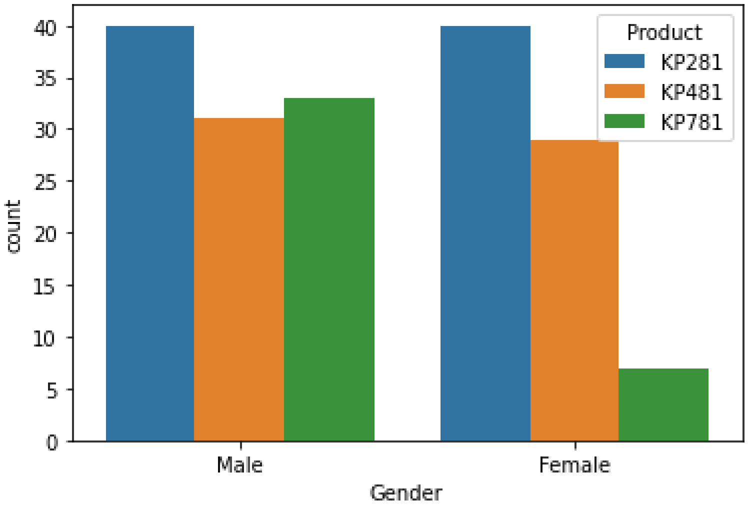
Out[57]:
```
<AxesSubplot:xlabel='Age', ylabel='Income'>
```



In [58]:
```
#KP281 is mostly bought by people with income>50000
#KP481 and 781 is mostly bought by people with income<50000
```

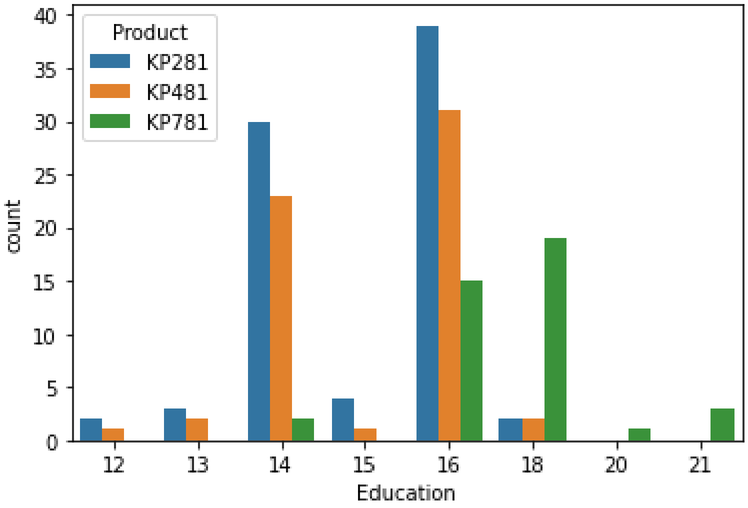
In [59]:
```
sns.countplot(data=aerofit,x='Gender',hue='Product')
#Females have not bought 781, for it target base should be male
```

Out[59]:
```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```



```
In [60]: sns.countplot(data=aerofit,x='Education',hue='Product')
```

```
Out[60]: <AxesSubplot:xlabel='Education', ylabel='count'>
```

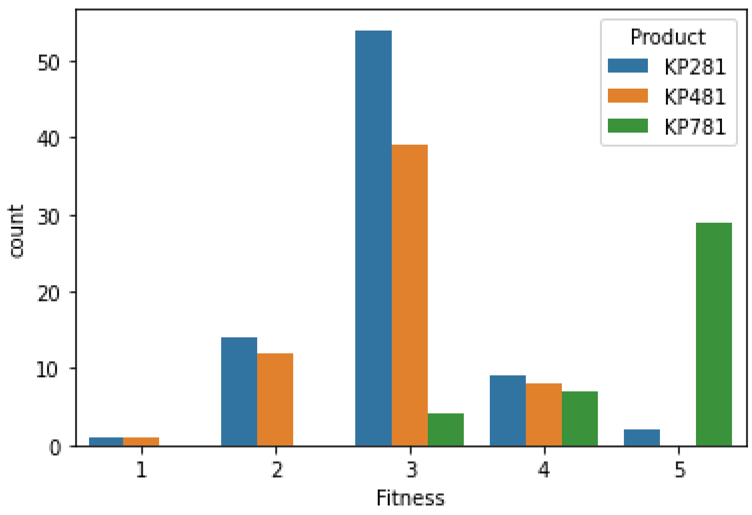


In [134]:
```
print(aerofit.value_counts('Product'))
print(aerofit.value_counts('Gender'))
print(aerofit.value_counts('MaritalStatus'))
print(aerofit.value_counts('Usage'))
print(aerofit.value_counts('Fitness'))
#most sevalue_countsct is 281
```

```
Product
KP281    80
KP481    60
KP781    40
dtype: int64
Gender
Male      104
Female     76
dtype: int64
MaritalStatus
Partnered    107
Single        73
dtype: int64
Usage
3    69
4    52
2    33
5    17
6     7
7     2
dtype: int64
Fitness
3    97
5    31
2    26
4    24
1     2
dtype: int64
```

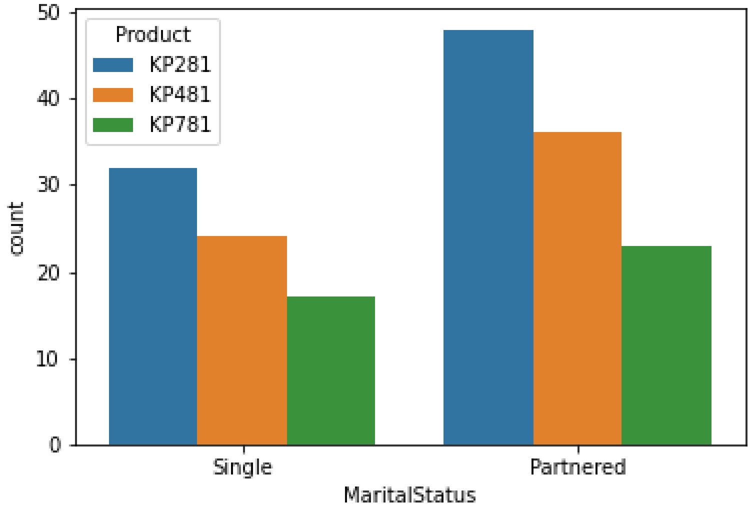
In [62]:
```
sns.countplot(data=aerofit,x='Fitness',hue='Product')
```

Out[62]:
```
<AxesSubplot:xlabel='Fitness', ylabel='count'>
```



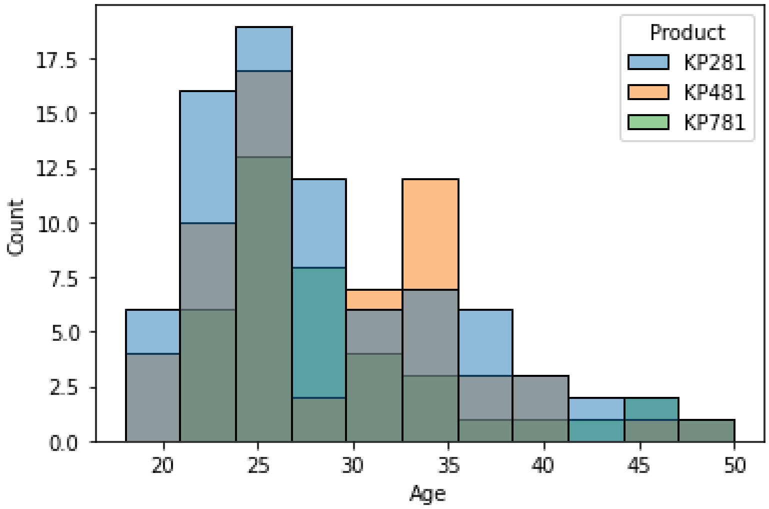
```
In [63]: sns.countplot(data=aerofit,x='MaritalStatus',hue='Product')
```

```
Out[63]: <AxesSubplot:xlabel='MaritalStatus', ylabel='count'>
```



```
In [64]: sns.histplot(data=aerofit,x='Age',hue='Product')
```

```
Out[64]: <AxesSubplot:xlabel='Age', ylabel='Count'>
```



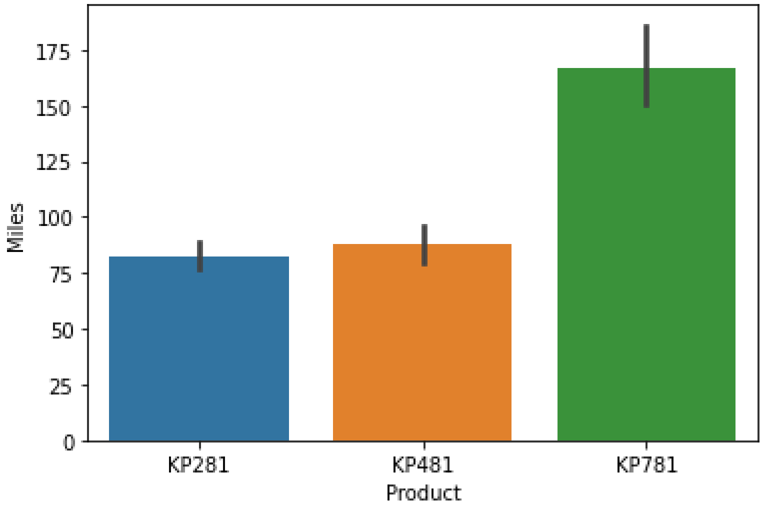
```
In [65]: pd.crosstab(columns=aerofit['Product'],index=1)
```

Out[65]:

Product	KP281	KP481	KP781
row_0			
1	80	60	40

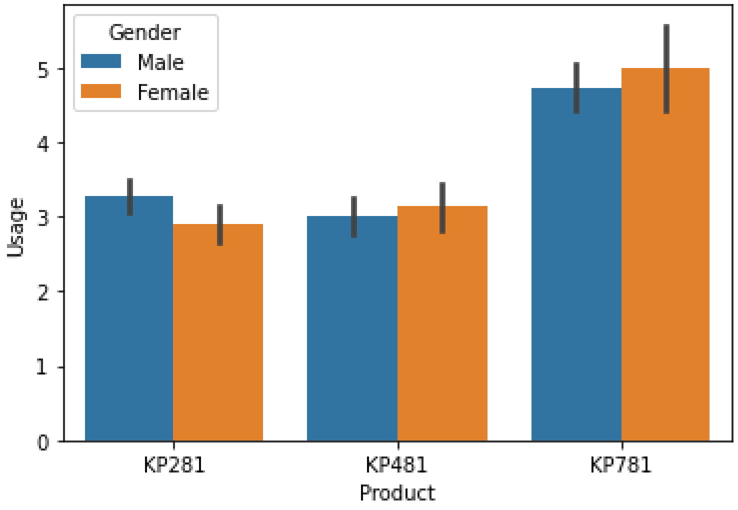
```
In [66]: sns.barplot(data=aerofit,x='Product',y='Miles')
```

```
Out[66]: <AxesSubplot:xlabel='Product', ylabel='Miles'>
```



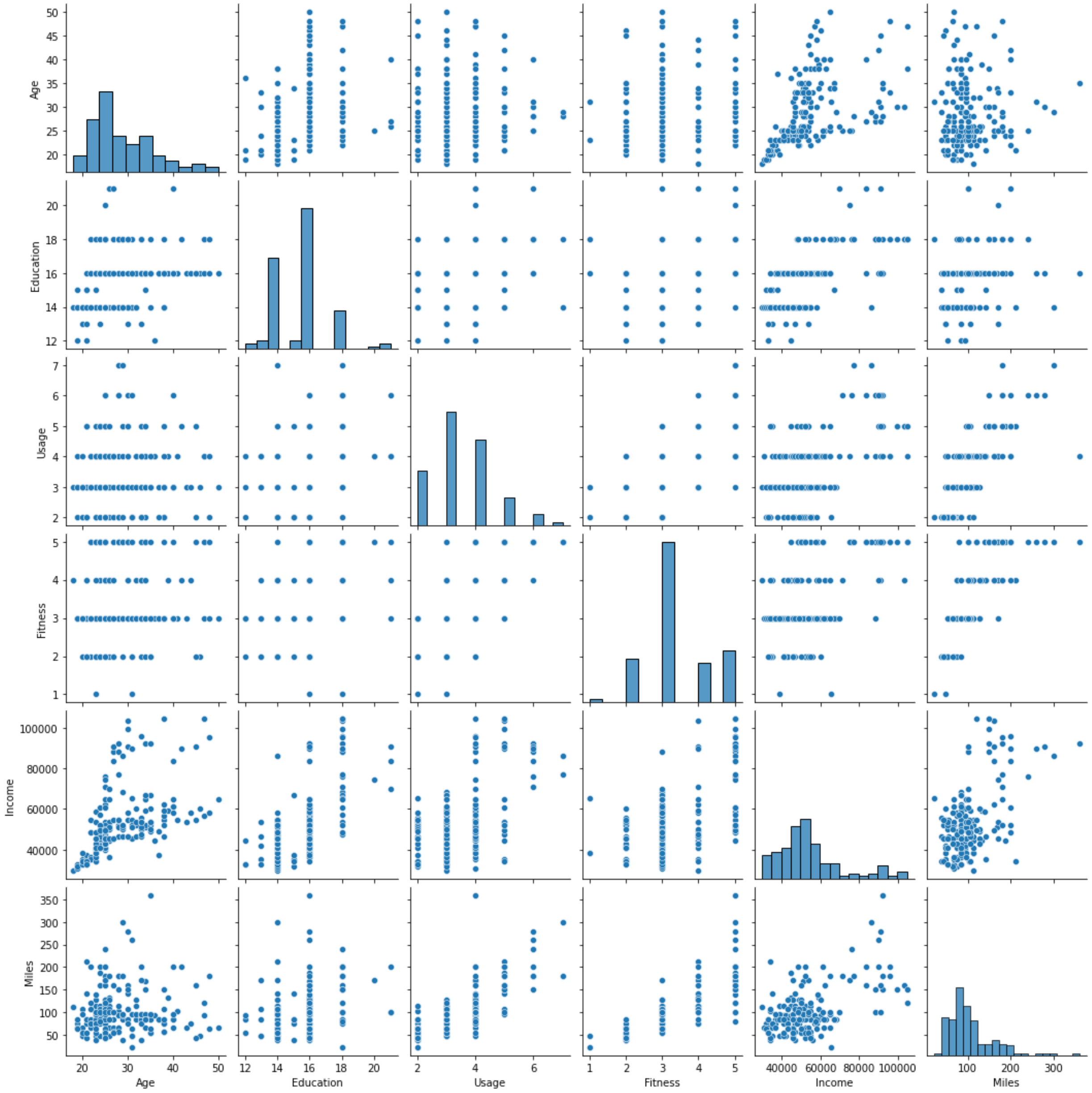
```
In [67]: sns.barplot(data=aerofit,x='Product',y='Usage',hue='Gender')
```

```
Out[67]: <AxesSubplot:xlabel='Product', ylabel='Usage'>
```



In [68]:
```
sns.pairplot(data=aerofit)
```

Out[68]:
```
<seaborn.axisgrid.PairGrid at 0x14cf2cd0>
```



In [ ]:

In [69]:

```
#two-way contingency tables

pd.crosstab([aerofit.Age, aerofit.Gender], aerofit.Product, margins = False)
```

Out[69]:

	Product	**KP281**	**KP481**	**KP781**
Age	**Gender**			
18	**Male**	1	0	0
19	**Female**	1	0	0
	Male	2	1	0
20	**Female**	1	1	0
	Male	1	2	0
21	**Female**	2	1	0
	Male	2	2	0
22	**Female**	3	0	0
	Male	1	0	3
23	**Female**	3	3	1
	Male	5	4	2
24	**Female**	3	2	1
	Male	2	1	3
25	**Female**	4	5	1
	Male	3	6	6
26	**Female**	3	2	1
	Male	4	1	1
27	**Female**	2	0	0
	Male	1	1	3
28	**Female**	4	0	1
	Male	2	0	2
29	**Female**	2	1	0
	Male	1	0	2
30	**Female**	0	2	1
	Male	2	0	2
31	**Female**	1	2	0
	Male	1	1	1
32	**Female**	1	0	0
	Male	1	2	0
33	**Female**	2	3	1
	Male	0	2	0

	Product	KP281	KP481	KP781
Age	Gender			
34	Female	1	1	0
	Male	1	2	1
35	Female	2	2	0
	Male	1	2	1
36	Male	1	0	0
37	Female	1	1	0
38	Female	1	1	0
	Male	3	1	1
39	Male	1	0	0
40	Female	0	2	0
	Male	1	1	1
41	Male	1	0	0
42	Male	0	0	1
43	Male	1	0	0
44	Female	1	0	0
45	Male	0	1	1
46	Female	1	0	0
47	Male	1	0	1
48	Male	0	1	1
50	Female	1	0	0

In [70]:

```
aerofit.isnull().sum()
#No null values
```

Out[70]:

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

In [81]:

```
prob1 = pd.crosstab([aerofit.MaritalStatus], aerofit.Product, margins = False)
prob1
```

Out[81]:

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

In [93]:

```
#probablity
print(prob1.loc[['Single'][0]]/prob1.loc['Single'].sum())
print(prob1.loc[['Partnered'][0]]/prob1.loc['Partnered'].sum())
```

```
Product
KP281    0.438356
KP481    0.328767
KP781    0.232877
Name: Single, dtype: float64
Product
KP281    0.448598
KP481    0.336449
KP781    0.214953
Name: Partnered, dtype: float64
```

In [97]:
```
prob2=pd.crosstab([aerofit.Usage, aerofit.Fitness], aerofit.Product, margins = Fa
prob2
```

Out[97]:

	Product	KP281	KP481	KP781
Usage	**Fitness**			
2	**1**	0	1	0
	2	9	5	0
	3	10	8	0
3	**1**	1	0	0
	2	4	6	0
	3	27	20	0
	4	5	5	0
	5	0	0	1
4	**2**	1	1	0
	3	17	9	4
	4	3	2	2
	5	1	0	12
5	**3**	0	2	0
	4	1	1	4
	5	1	0	8
6	**4**	0	0	1
	5	0	0	6
7	**5**	0	0	2

In [99]:

```
#probablity
print(prob2.loc[[2][0]]/prob2.loc[2].sum())
print(prob2.loc[[3][0]]/prob2.loc[3].sum())
print(prob2.loc[[4][0]]/prob2.loc[4].sum())
print(prob2.loc[[5][0]]/prob2.loc[5].sum())
print(prob2.loc[[6][0]]/prob2.loc[6].sum())
print(prob2.loc[[7][0]]/prob2.loc[7].sum())
```

```
Product     KP281     KP481  KP781
Fitness
1        0.000000  0.071429    NaN
2        0.473684  0.357143    NaN
3        0.526316  0.571429    NaN
Product     KP281     KP481  KP781
Fitness
1        0.027027  0.000000    0.0
2        0.108108  0.193548    0.0
3        0.729730  0.645161    0.0
4        0.135135  0.161290    0.0
5        0.000000  0.000000    1.0
Product     KP281     KP481     KP781
Fitness
2        0.045455  0.083333  0.000000
3        0.772727  0.750000  0.222222
4        0.136364  0.166667  0.111111
5        0.045455  0.000000  0.666667
Product  KP281     KP481     KP781
Fitness
3          0.0  0.666667  0.000000
4          0.5  0.333333  0.333333
5          0.5  0.000000  0.666667
Product  KP281  KP481     KP781
Fitness
4          NaN    NaN  0.142857
5          NaN    NaN  0.857143
Product  KP281  KP481  KP781
Fitness
5          NaN    NaN    1.0
```

In [103]:

```
prob3=pd.crosstab([aerofit.Education], aerofit.Product, margins = False)
prob3
```

Out[103]:

Product	KP281	KP481	KP781
Education			
12	2	1	0
13	3	2	0
14	30	23	2
15	4	1	0
16	39	31	15
18	2	2	19
20	0	0	1
21	0	0	3

In [106]:

```
print(prob3.loc[[12][0]]/prob3.loc[12].sum())
print(prob3.loc[[13][0]]/prob3.loc[13].sum())
print(prob3.loc[[14][0]]/prob3.loc[14].sum())
print(prob3.loc[[15][0]]/prob3.loc[15].sum())
print(prob3.loc[[16][0]]/prob3.loc[16].sum())
print(prob3.loc[[18][0]]/prob3.loc[18].sum())
print(prob3.loc[[20][0]]/prob3.loc[20].sum())
print(prob3.loc[[21][0]]/prob3.loc[21].sum())
```

```
Product
KP281    0.666667
KP481    0.333333
KP781    0.000000
Name: 12, dtype: float64
Product
KP281    0.6
KP481    0.4
KP781    0.0
Name: 13, dtype: float64
Product
KP281    0.545455
KP481    0.418182
KP781    0.036364
Name: 14, dtype: float64
Product
KP281    0.8
KP481    0.2
KP781    0.0
Name: 15, dtype: float64
Product
KP281    0.458824
KP481    0.364706
KP781    0.176471
Name: 16, dtype: float64
Product
KP281    0.086957
KP481    0.086957
KP781    0.826087
Name: 18, dtype: float64
Product
KP281    0.0
KP481    0.0
KP781    1.0
Name: 20, dtype: float64
Product
KP281    0.0
KP481    0.0
KP781    1.0
Name: 21, dtype: float64
```

In [108]:

```
prob4=pd.crosstab([aerofit.Gender], aerofit.Product, margins = False)
prob4
```

Out[108]:

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

In [124]:

```
print(prob4.loc[['Female'][0]]/prob4.loc['Female'].sum())
print(prob4.loc[['Male'][0]]/prob4.loc['Male'].sum())
```

```
Product
KP281    0.526316
KP481    0.381579
KP781    0.092105
Name: Female, dtype: float64
Product
KP281    0.384615
KP481    0.298077
KP781    0.317308
Name: Male, dtype: float64
```

In [73]:

```
#convert categorical into numerical values
```

In [74]:

```
#Creating new df
aerofit1=aerofit
```

In [75]:

```
aerofit1
```

Out[75]:

	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

In [76]:
```
#aerofit1['Product']=aerofit1['Product'].map({'KP281':0,'KP481':1,'KP781':2})
#aerofit1['Gender']=aerofit1['Gender'].map({'Male':0,'Female':1})
#aerofit1['MaritalStatus']=aerofit1['MaritalStatus'].map({'Single':0,'Partnered':
```

In [77]:
```
aerofit1
```

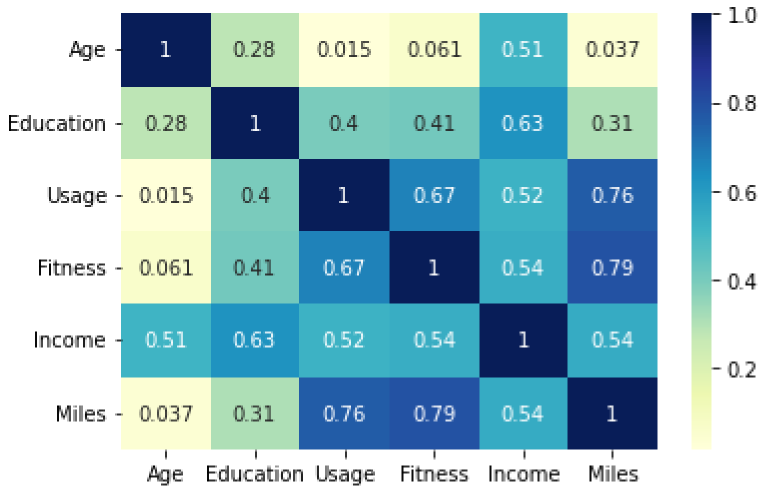
Out[77]:

	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

In [78]:
```
sns.heatmap(aerofit1.corr(), cmap="YlGnBu", annot=True)
```

Out[78]:
```
<AxesSubplot:>
```



```
In [79]: aerofit1.corr()
```

Out[79]:

	Age	Education	Usage	Fitness	Income	Miles
Age	1.000000	0.280496	0.015064	0.061105	0.513414	0.036618
Education	0.280496	1.000000	0.395155	0.410581	0.625827	0.307284
Usage	0.015064	0.395155	1.000000	0.668606	0.519537	0.759130
Fitness	0.061105	0.410581	0.668606	1.000000	0.535005	0.785702
Income	0.513414	0.625827	0.519537	0.535005	1.000000	0.543473
Miles	0.036618	0.307284	0.759130	0.785702	0.543473	1.000000

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```