In [1]:

```
import pandas as pd
import seaborn as sns
```

In [2]:

```
df=pd.read_csv(r'C:\Users\archa\OneDrive\Desktop\New folder\aerofit.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	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

In [4]:

```
df.info()
# Most of the data in integer datatype
```

```
<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
```

In [5]:

```
df.isnull().sum()/len(df)*100
```

Out[5]:

```
Product          0.0
Age              0.0
Gender           0.0
Education        0.0
MaritalStatus    0.0
Usage            0.0
Fitness          0.0
Income           0.0
Miles            0.0
dtype: float64
```

Observation -As we can clearly see their is no NULL values present

In [ ]:

```

```

In [ ]:

```

```

In [6]:

```
df.columns
```

Out[6]:

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

In [7]:

```
df['age_bins']=pd.cut(x=df['Age'],bins=[0,18,28,38,48,58,68,100],
                     labels=['0-18','18-28','28-38','38-48','48-58','58-68','68-100'])
#  Adding Category i.e adding Age to age_bins
```

In [8]:

```
df
```

Out[8]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	age_bins
0	KP281	18	Male	14	Single	3	4	29562	112	0-18
1	KP281	19	Male	15	Single	2	3	31836	75	18-28
2	KP281	19	Female	14	Partnered	4	3	30699	66	18-28
3	KP281	19	Male	12	Single	3	3	32973	85	18-28
4	KP281	20	Male	13	Partnered	4	2	35247	47	18-28
...	...	...	...	...	...	...	...	...	...	..
175	KP781	40	Male	21	Single	6	5	83416	200	38-48
176	KP781	42	Male	18	Single	5	4	89641	200	38-48
177	KP781	45	Male	16	Single	5	5	90886	160	38-48
178	KP781	47	Male	18	Partnered	4	5	104581	120	38-48
179	KP781	48	Male	18	Partnered	4	5	95508	180	38-48

180 rows × 10 columns

In [ ]:

In [9]:

```
df.describe()
```

Out[9]:

	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 [10]:

```
# Checking diff between mean and median
```

In [ ]:

```

```

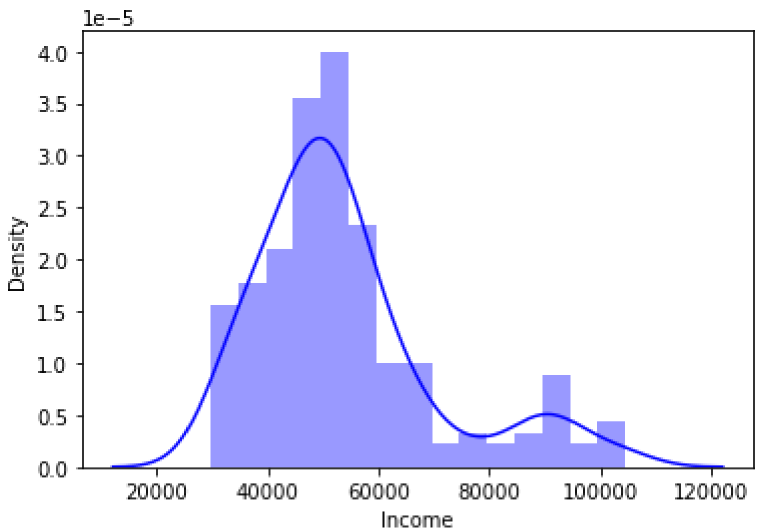
In [11]:

```
# Univeriant graph

sns.distplot(df['Income'],color='blue')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[11]:

```
<AxesSubplot:xlabel='Income', ylabel='Density'>
```



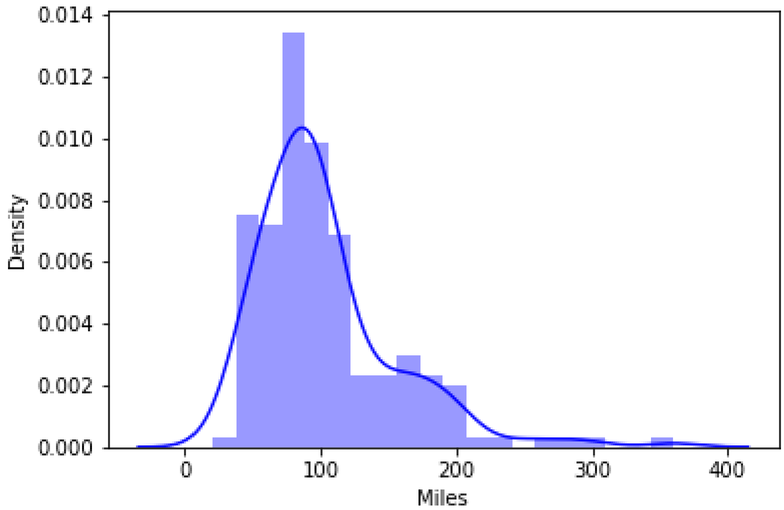
In [12]:

```
sns.distplot(df['Miles'],color='blue')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: Fu
tureWarning: `distplot` is a deprecated function and will be removed in a fu
ture version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

Out[12]:

```
<AxesSubplot:xlabel='Miles', ylabel='Density'>
```



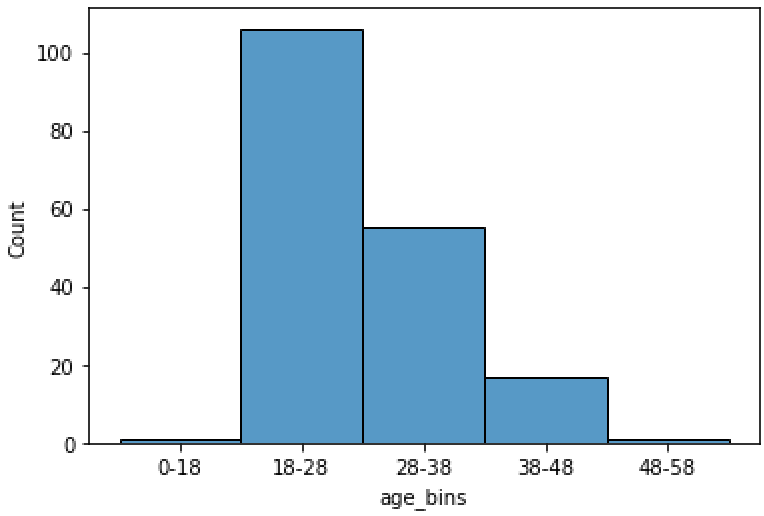
In [ ]:

In [13]:

```
sns.histplot(x='age_bins',data=df)
```

Out[13]:

```
<AxesSubplot:xlabel='age_bins', ylabel='Count'>
```



In [14]:

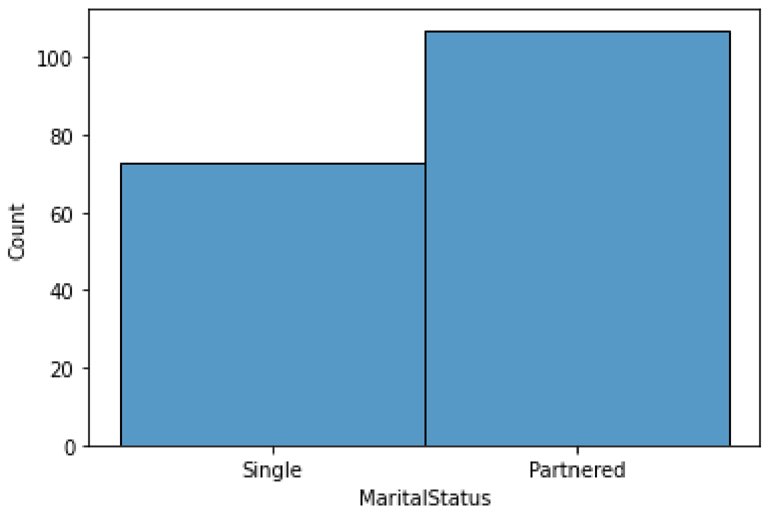
```
# Can notice that maximum users are the 18-28 age group
```

In [15]:

```
sns.histplot(x='MaritalStatus',data=df)
```

Out[15]:

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



In [ ]:

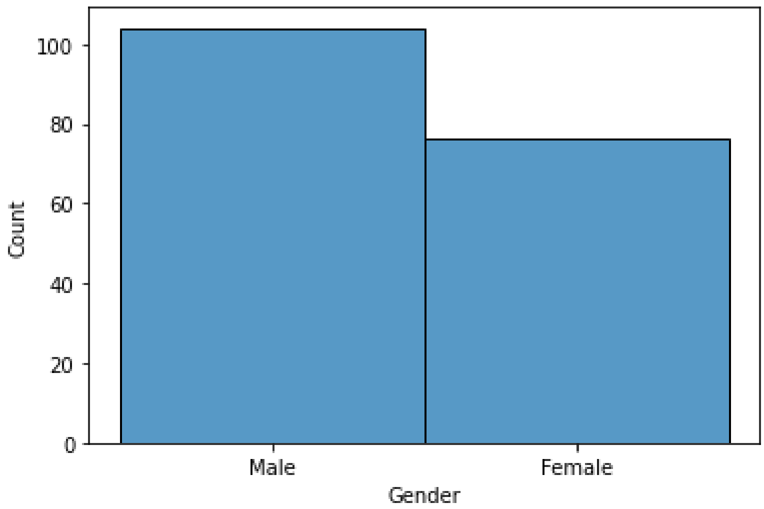
```

```

In [16]:

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

Out[16]:

```
<AxesSubplot:xlabel='Gender', ylabel='Count'>
```



In [17]:

```
# Male are More
```

In [ ]:

```
```

In [18]:

```
# Univarient
```

In [ ]:

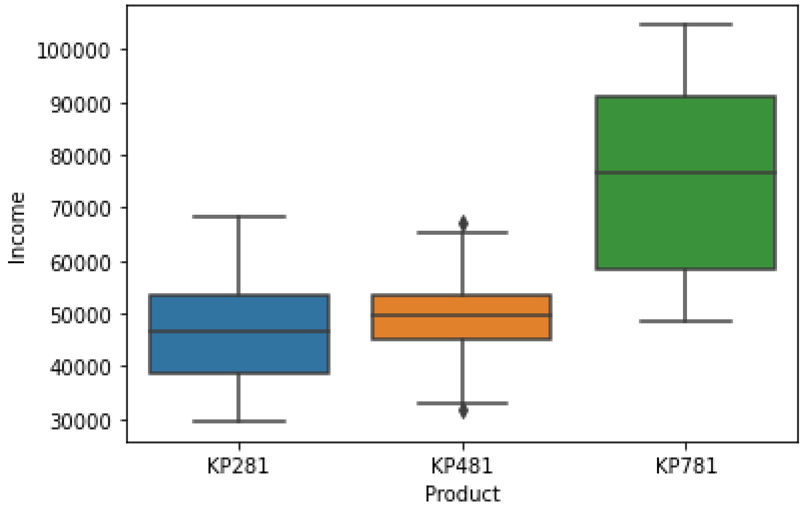
```
```

In [19]:

```
sns.boxplot(x='Product',y='Income',data=df)
```

Out[19]:

```
<AxesSubplot:xlabel='Product', ylabel='Income'>
```



In [20]:

```
# Here we can see that the Highest income clients are using the costliest Product
# We can also see here both Basic and Advanced Products are Used by more number of clients
# Advertising comparison of 3 models can make basic user push for buying at least  Mid vers
```

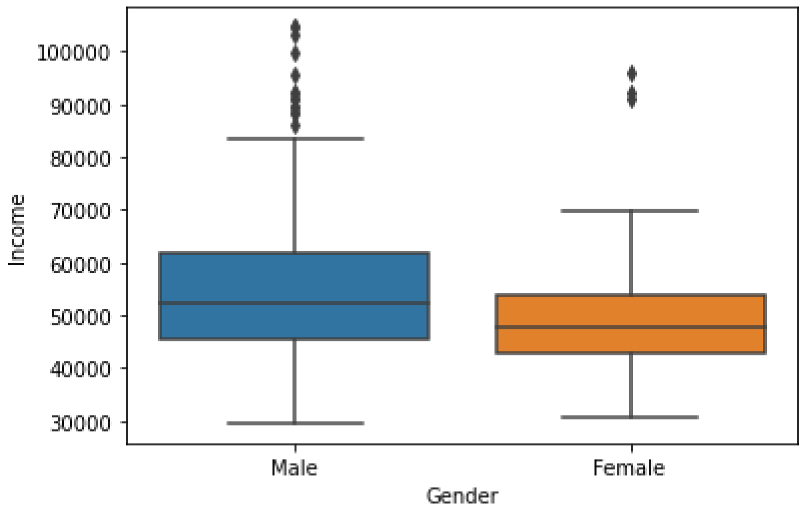
In [ ]:

In [21]:

```
sns.boxplot(x='Gender',y='Income',data=df)
```

Out[21]:

```
<AxesSubplot:xlabel='Gender', ylabel='Income'>
```



In [22]:

```
# observed that lot of outliers in Male compared to female
```

In [23]:

```
df.groupby('Gender')['Income'].mean()
```

Out[23]:

```
Gender
Female    49828.907895
Male      56562.759615
Name: Income, dtype: float64
```

In [24]:

```
# Removing Outliers

q1=df['Income'].quantile(0.25)
q3=df['Income'].quantile(0.75)
iqr=q3-q1

df=df[(df['Income']>q1-1.5*iqr)&(df['Income']<q3+1.5*iqr)]
```

In [25]:

```
df.groupby('Gender')['Income'].mean()
```

Out[25]:

```
Gender
Female    48056.356164
Male      50000.840909
Name: Income, dtype: float64
```

In [26]:

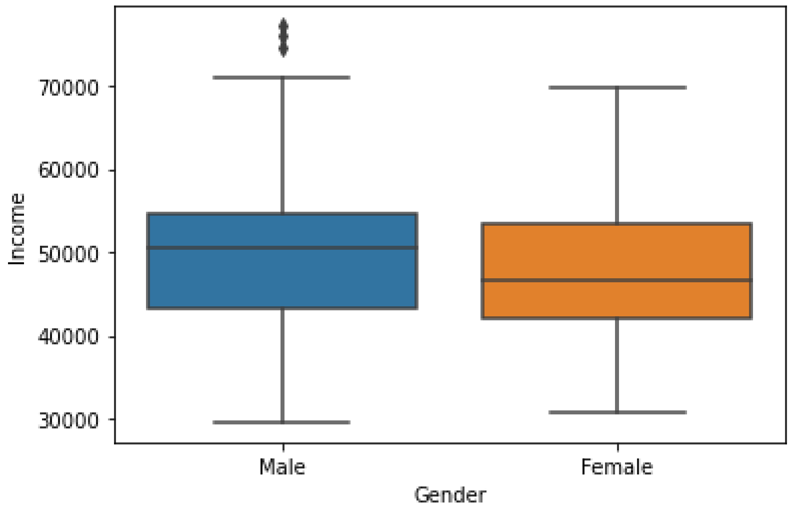
```
# Outliers are removed
```

In [27]:

```
sns.boxplot(x='Gender',y='Income',data=df)
```

Out[27]:

```
<AxesSubplot:xlabel='Gender', ylabel='Income'>
```



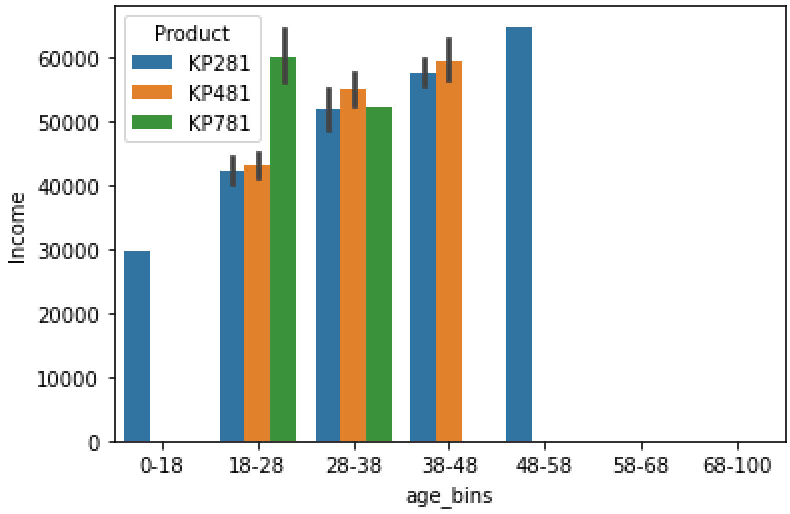
In [ ]:

In [28]:

```
sns.barplot(x='age_bins',y='Income',hue='Product',data=df)
```

Out[28]:

```
<AxesSubplot:xlabel='age_bins', ylabel='Income'>
```



In [ ]:

In [29]:

```
# Its noticable that as age increases Income also increases so must focus on aged people
```

In [ ]:

In [30]:

```
# To check which gender is using the products more
pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True)
```

Out[30]:

Product	KP281	KP481	KP781	All
Gender				
Female	40	29	4	73
Male	40	31	17	88
All	80	60	21	161

In [31]:

```
pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True,normalize=True)*100
```

Out[31]:

Product Gender	KP281	KP481	KP781	All
Female	24.844720	18.012422	2.484472	45.341615
Male	24.844720	19.254658	10.559006	54.658385
All	49.689441	37.267081	13.043478	100.000000

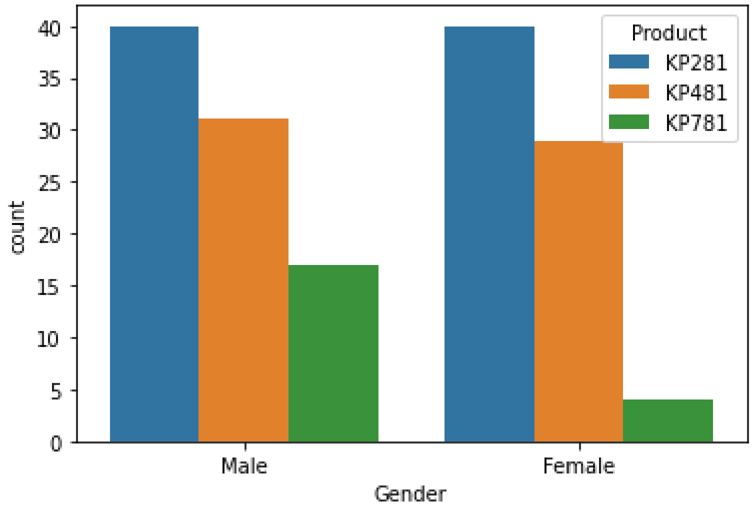
In [ ]:

In [32]:

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

Out[32]:

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



In [33]:

```
# Its notable that advanced product is mostly used by Male
```

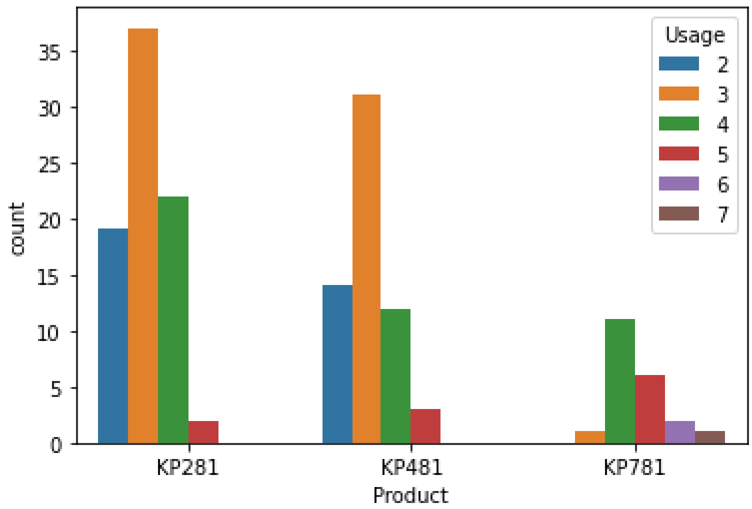
In [ ]:

In [34]:

```
sns.countplot(x='Product',hue='Usage',data=df)
```

Out[34]:

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



In [35]:

```
# Advanced Product is highly used among people who are buying it
```

In [36]:

```
df['Usage'].value_counts()
```

Out[36]:

```
3    69
4    45
2    33
5    11
6     2
7     1
Name: Usage, dtype: int64
```

In [ ]:

```

```

In [37]:

```
# Checking Unique values
df.nunique()
```

Out[37]:

```
Product           3
Age              31
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           51
Miles            32
age_bins          5
dtype: int64
```

In [ ]:

```

```

In [38]:

```
df.corr()
```

Out[38]:

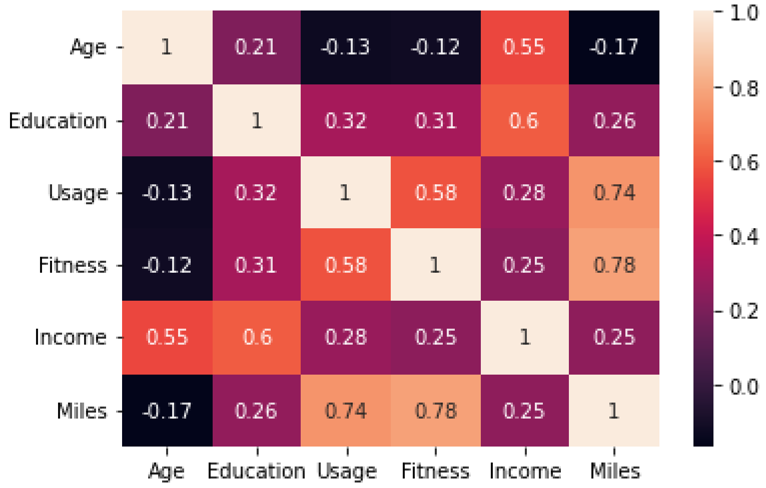
	Age	Education	Usage	Fitness	Income	Miles
Age	1.000000	0.208992	-0.125330	-0.118570	0.551113	-0.165710
Education	0.208992	1.000000	0.315696	0.313260	0.600964	0.260524
Usage	-0.125330	0.315696	1.000000	0.578850	0.279502	0.744355
Fitness	-0.118570	0.313260	0.578850	1.000000	0.246177	0.780566
Income	0.551113	0.600964	0.279502	0.246177	1.000000	0.252686
Miles	-0.165710	0.260524	0.744355	0.780566	0.252686	1.000000

In [39]:

```
sns.heatmap(df.corr(),annot=True)
```

Out[39]:

<AxesSubplot:>



In [40]:

```
# Usage increases fitness also increases
# Miles increase fitness increases
# Age increases Income also increases
```

In [ ]:

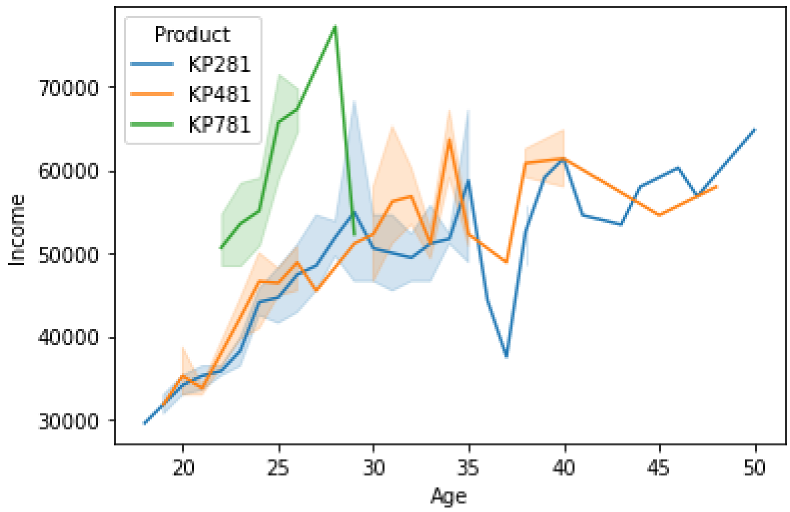
```

```

In [41]:

```
sns.lineplot(x='Age',y='Income',data=df, hue='Product')
```

Out[41]:

<AxesSubplot:xlabel='Age', ylabel='Income'>



In [42]:

```
# Highest Income customers are buying advanced products
```

In [ ]:

In [43]:

```
df.groupby(['Gender','MaritalStatus','Product']).sum()['Miles'].unstack()
```

Out[43]:

	Product	KP281	KP481	KP781
Gender	**MaritalStatus**			
Female	**Partnered**	2023	1410	200
	Single	1025	1123	400
Male	**Partnered**	1684	1832	1410
	Single	1891	911	1106

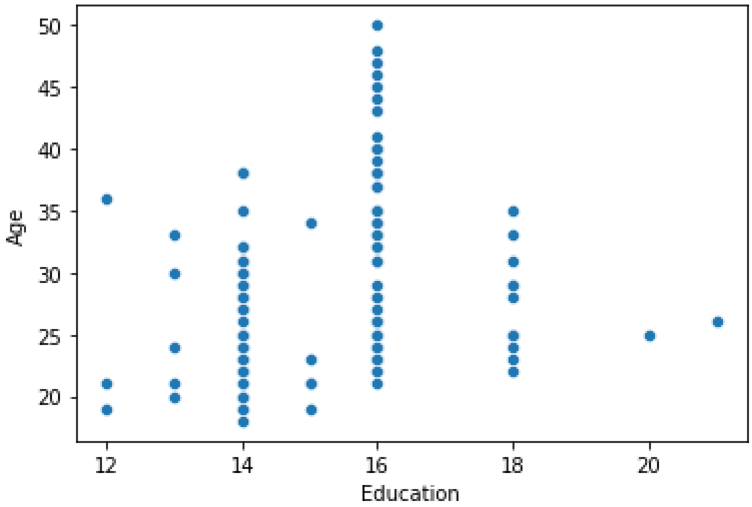
In [ ]:

In [44]:

```
sns.scatterplot(x='Education',y='Age',data=df)
```

Out[44]:

<AxesSubplot:xlabel='Education', ylabel='Age'>



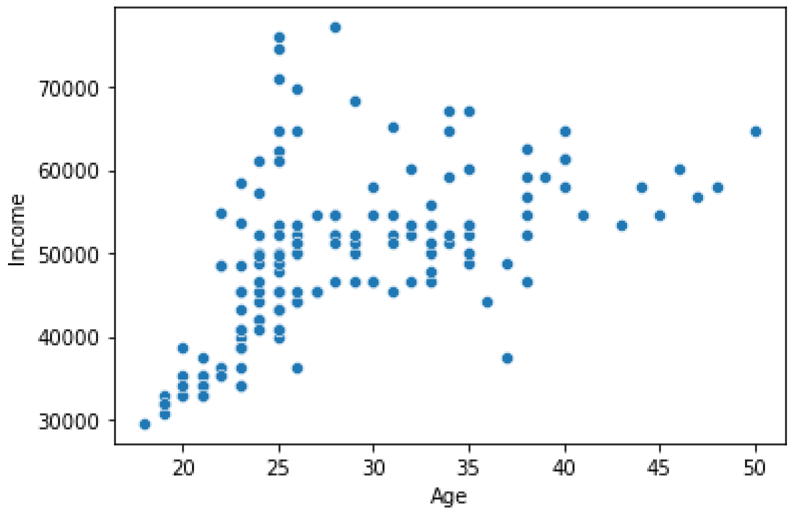
In [ ]:

In [45]:

```
sns.scatterplot(x='Age',y='Income',data=df)
```

Out[45]:

```
<AxesSubplot:xlabel='Age', ylabel='Income'>
```

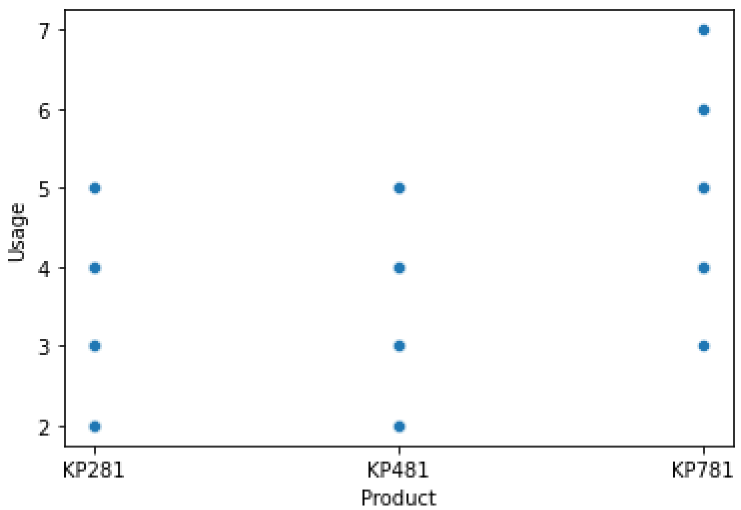


In [46]:

```
sns.scatterplot(x='Product',y='Usage',data=df)
```

Out[46]:

```
<AxesSubplot:xlabel='Product', ylabel='Usage'>
```



In [47]:

```
# Usage of Advanced Product is High
```

In [ ]:

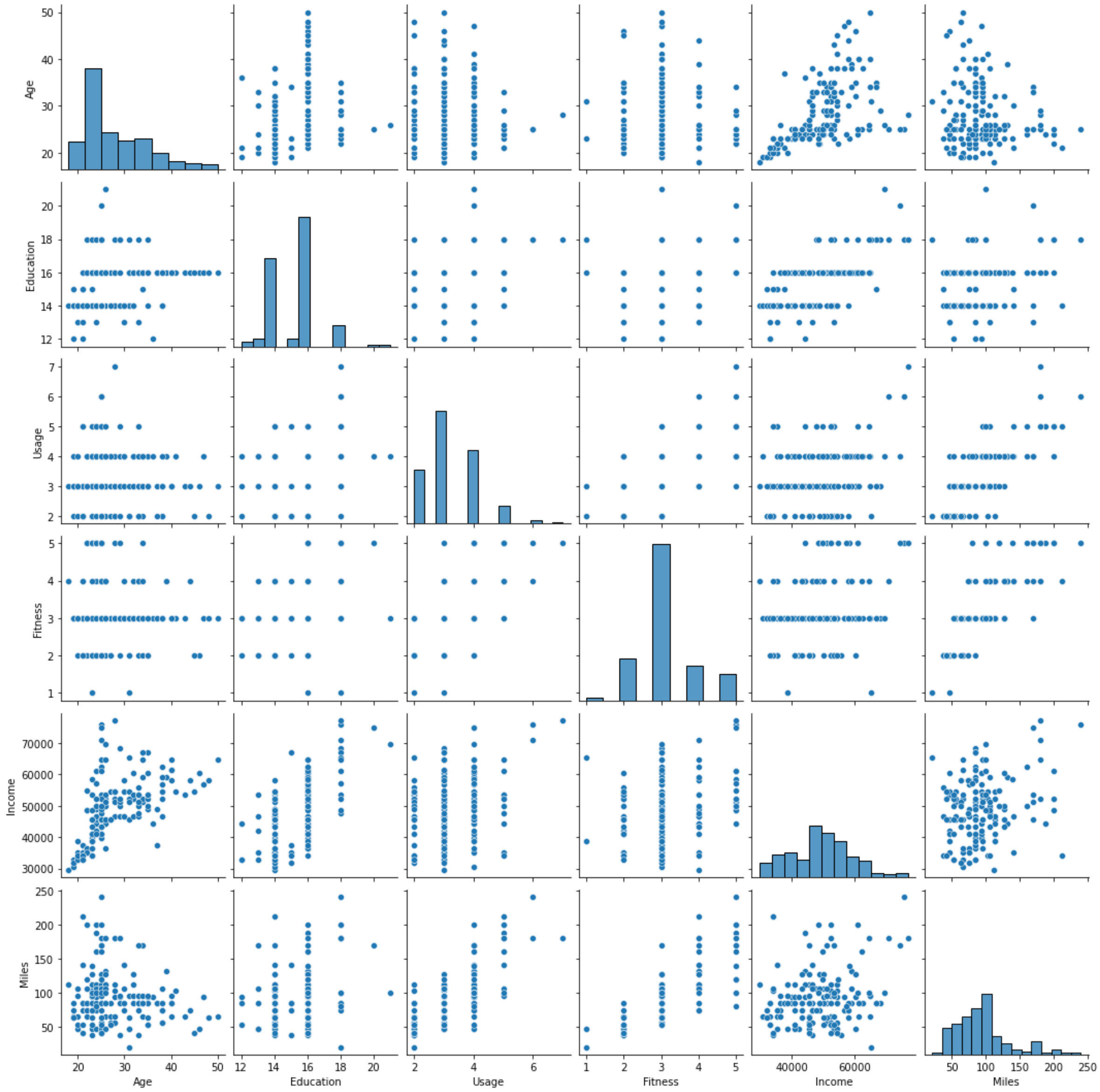
In [ ]:

In [48]:

```
sns.pairplot(df)
```

Out[48]:

```
<seaborn.axisgrid.PairGrid at 0x25da148f5e0>
```



In [ ]: