Aerofit - Descriptive Statistics & Probability

In [1]:

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

In [2]:

```
df = pd.read_csv("aerofit_treadmill.csv")
```

1. Defining Problem Statement and Analysing basic metrics

1. Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), statistical summary

In [5]:

```
df.shape   # there are 180 samples in the dataset,along with 9 features.
```

Out[5]:

```
(180, 9)
```

In [7]:

```
df.head()
```

Out[7]:

	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 [6]:

```
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
```

There are no nulls in the dataset, so we dont need to do any imputations to the dataset

Product, Gender, MaritalStatus are categorical variables

Education, Usage, Fitness are discrete numerical variables

Age, Income, Miles are continuous numerical variables

In [22]:

```
def describe(df, stats):
    d = df.describe()
    d = d.append(df.reindex(d.columns, axis = 1).agg(stats))
    return d.apply(round,args=(2,))

describe(df, ['median', 'std'])
```

Out[22]:

	Age	Education	Usage	Fitness	Income	Miles
count	180.00	180.00	180.00	180.00	180.00	180.00
mean	28.79	15.57	3.46	3.31	53719.58	103.19
std	6.94	1.62	1.08	0.96	16506.68	51.86
min	18.00	12.00	2.00	1.00	29562.00	21.00
25%	24.00	14.00	3.00	3.00	44058.75	66.00
50%	26.00	16.00	3.00	3.00	50596.50	94.00
75%	33.00	16.00	4.00	4.00	58668.00	114.75
max	50.00	21.00	7.00	5.00	104581.00	360.00
median	26.00	16.00	3.00	3.00	50596.50	94.00
std	6.94	1.62	1.08	0.96	16506.68	51.86

Age, Education, Usage, Fitness : both mean and median are closer to each other, so there is no much skewness.

Income : mean is around 3k higher than median, higher incomes are dragging mean higher.

Miles : most people have higher targets to run each week, hence the mean is 103 compared median at 94

In [45]:

```
def describe_cat(df):
    df1 = df.describe(include='object')
    df2 = pd.DataFrame(index=['percent'],data=dict(df1.loc['freq']/df1.loc['count']
    return pd.concat([df1,df2],axis=0)
describe_cat(df)
```

Out[45]:

	Product	Gender	MaritalStatus
count	180	180	180
unique	3	2	2
top	KP281	Male	Partnered
freq	80	104	107
percent	0.444444	0.577778	0.594444

Product : KP281 product has 44% of values out of 3 product

Gender : Male holds 58% of the purchases

MaritalStatus : 60% of the people who purchase the product are Married

In [ ]:

In [ ]:

2. Non-Graphical Analysis: Value counts and unique attributes

In [46]:

```
df.head()
```

Out[46]:

	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 [57]:

```
df['Product'].nunique()  # there are total 3 categories in Product column
```

Out[57]:

```
3
```

In [59]:

```
df['Product'].unique()   # 'KP281', 'KP481', 'KP781' are the 3 categories in Produc
```

Out[59]:

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

In [50]:

```
df['Product'].value_counts()    # count of each product category
```

Out[50]:

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

In [ ]:

```

```

In [51]:

```
df['Gender'].nunique()   # there are 2 genders in Gender column
```

Out[51]:

```
2
```

In [52]:

```
df['Gender'].unique()   # 'Male', 'Female' are the gender representaion in the Gend
```

Out[52]:

```
array(['Male', 'Female'], dtype=object)
```

In [53]:

```
df['Gender'].value_counts()   # count of each gender
```

Out[53]:

```
Male      104
Female     76
Name: Gender, dtype: int64
```

In [ ]:

```

```

In [54]:

```
df['MaritalStatus'].nunique()   # there are 2 categories in the MaritaStatus
```

Out[54]:

```
2
```

In [55]:

```
df['MaritalStatus'].unique()   # 'Single', 'Partnered' are the 2 categories
```

Out[55]:

```
array(['Single', 'Partnered'], dtype=object)
```

In [60]:

```
df['MaritalStatus'].value_counts()   # counts of each category of MaritalStatus
```

Out[60]:

```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

In [ ]:

```

```

4. Missing Value & Outlier Detection

In [223]:

```
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
```

There are no missing values to impute, but we have extra information regarding the product pricing given in the documentation, so lets append that to our data

The KP281 is an entry-level treadmill that sells for 1,500.

The KP481 is for mid-level runners that sell for 1,750.

The KP781 treadmill is having advanced features that sell for 2,500.

In [227]:

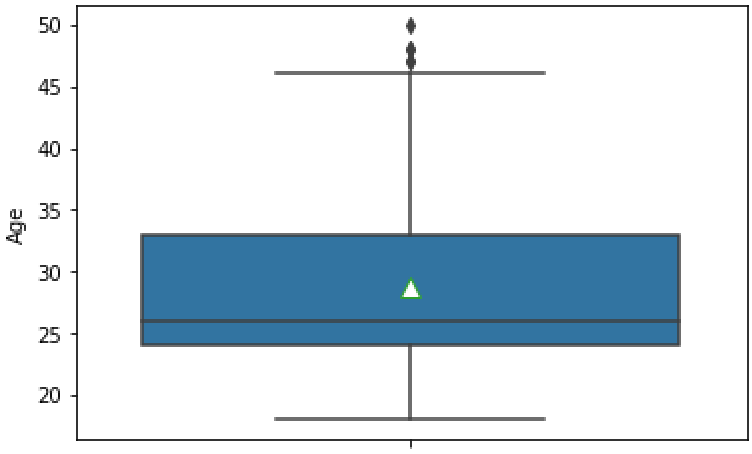
```
df['ProductPrice'] = df['Product'].replace({
    'KP281':1500,
    'KP481':1750,
    'KP781':2500
})
```

In [243]:

```
sns.boxplot(y='Age',data=df,showmeans=True,meanprops={"markerfacecolor":"white","ma
# Age has few outliers which are above 47
```

Out[243]:

<AxesSubplot:ylabel='Age'>

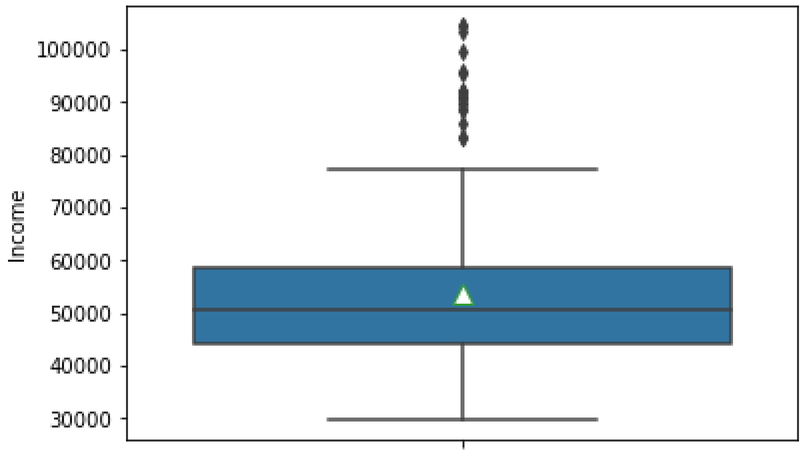


In [244]:

```
sns.boxplot(y='Income',data=df,showmeans=True,meanprops={"markerfacecolor":"white",
# Income has outliers which are above 78000
```

Out[244]:

<AxesSubplot:ylabel='Income'>

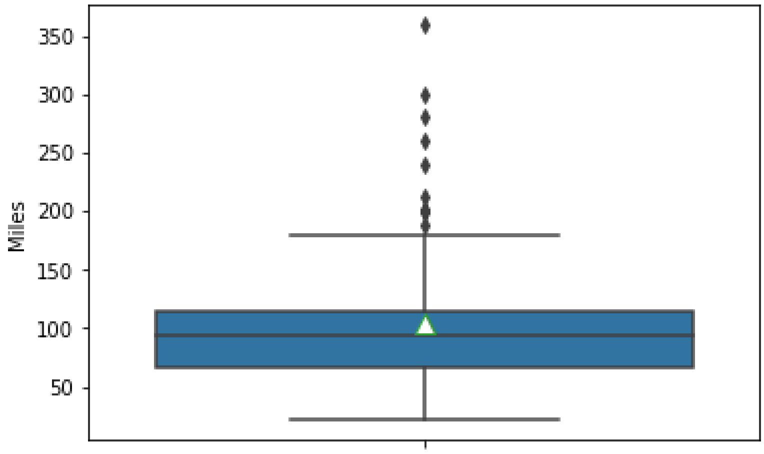


In [245]:

```
sns.boxplot(y='Miles',data=df,showmeans=True,meanprops={"markerfacecolor":"white","
# Miles have outliers which are above 180
```

Out[245]:

```
<AxesSubplot:ylabel='Miles'>
```



We didnt find any outliers which are below the lower whisker, all the outliers for the 3 continuous variables are above the higher whisker, so we can assume that the data is right skewed

We dont need to remove outliers because those outliers are due to certain segment of people, which we will come to know in the next parts of the analysis

In [229]:

```
df.head()
```

Out[229]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	ProductPri
0	KP281	18	Male	14	Single	3	4	29562	112	15
1	KP281	19	Male	15	Single	2	3	31836	75	15
2	KP281	19	Female	14	Partnered	4	3	30699	66	15
3	KP281	19	Male	12	Single	3	3	32973	85	15
4	KP281	20	Male	13	Partnered	4	2	35247	47	15

In [ ]:

In [ ]:

3. Visual Analysis - Univariate & Bivariate

1. For continuous variable(s): Distplot, countplot, histogram for univariate analysis

2. For categorical variable(s): Boxplot

3. For correlation: Heatmaps, Pairplots

5. Business Insights based on Non-Graphical and Visual Analysis (10 Points)

1. Comments on the range of attributes

2. Comments on the distribution of the variables and relationship between them

3. Comments for each univariate and bivariate plot

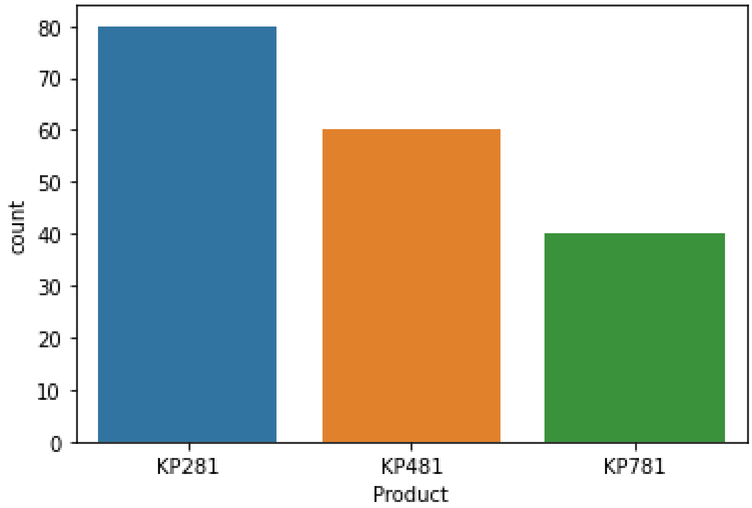
Will work with both 3rd and 5th questions combined, as both are related to each other

In [114]:

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

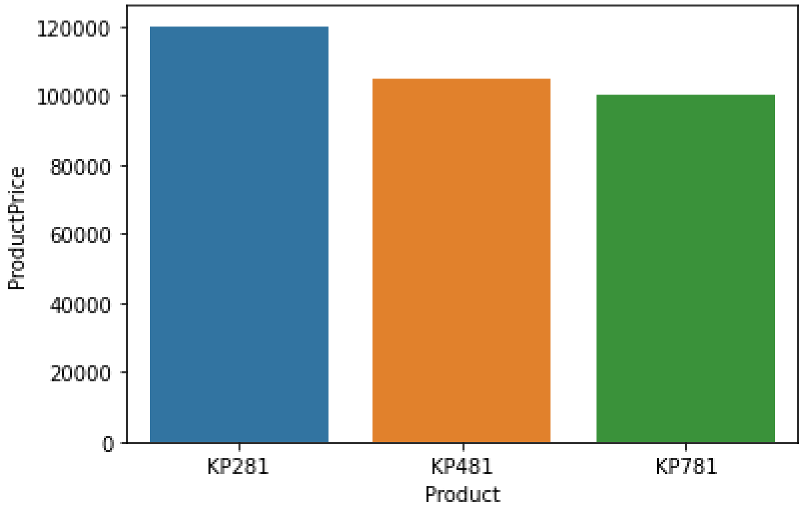
Out[114]:

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



In [239]:

```
sns.barplot(x='Product',y='ProductPrice',data=df,estimator=np.sum)
plt.show()
```



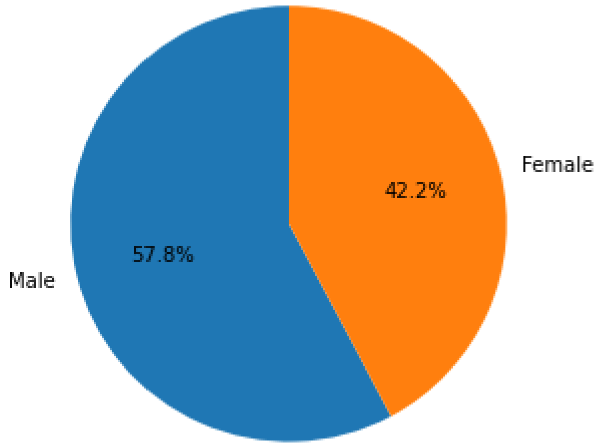
Though the sales volume of KP781 is much lower compared to others, the sum of amount made from the product is almost nearer to other 2 products

In [ ]:

```

```

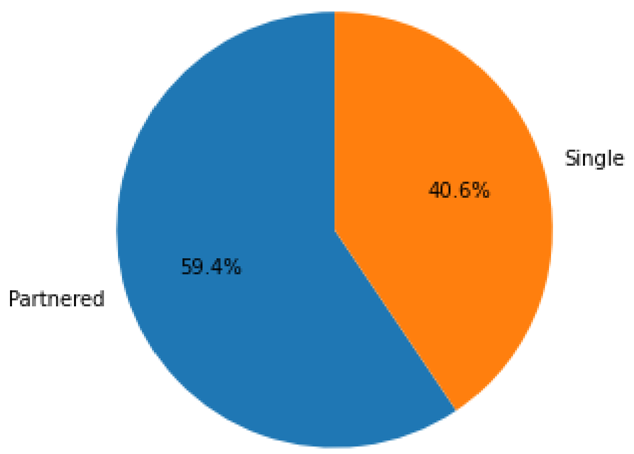
In [246]:

```
# 58% of the purchases made are from Male and 42 percent of purchases are made from
plt.figure(figsize=(10,5))
plt.pie(df['Gender'].value_counts().values, labels = df['Gender'].value_counts().in
plt.show()
```



In [247]:

```
plt.figure(figsize=(10,5))
plt.pie(df['MaritalStatus'].value_counts().values, labels = df['MaritalStatus'].val
plt.show()
```



60% of the purchases are from Partnered and only 40% of the sales are from Single, this tells us that married people tend to buy more compared to single

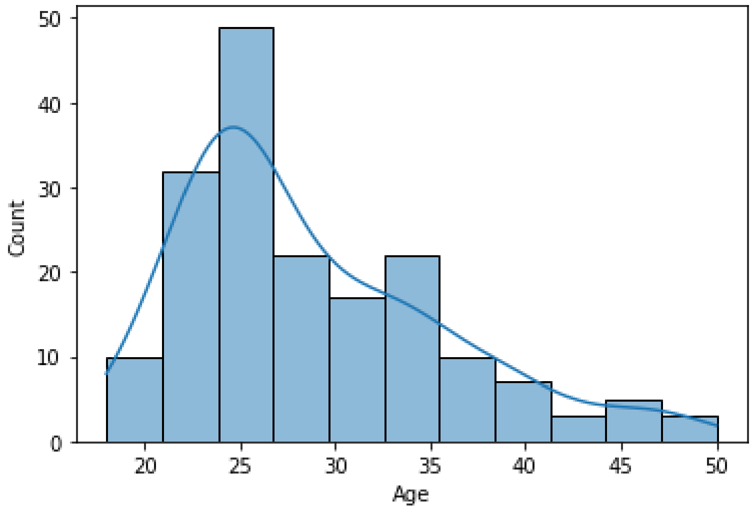
In [ ]:

In [248]:

```
# Age of people who buy the products are mostly between 22 and 29
sns.histplot(x="Age",data=df,kde=True)
```

Out[248]:

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

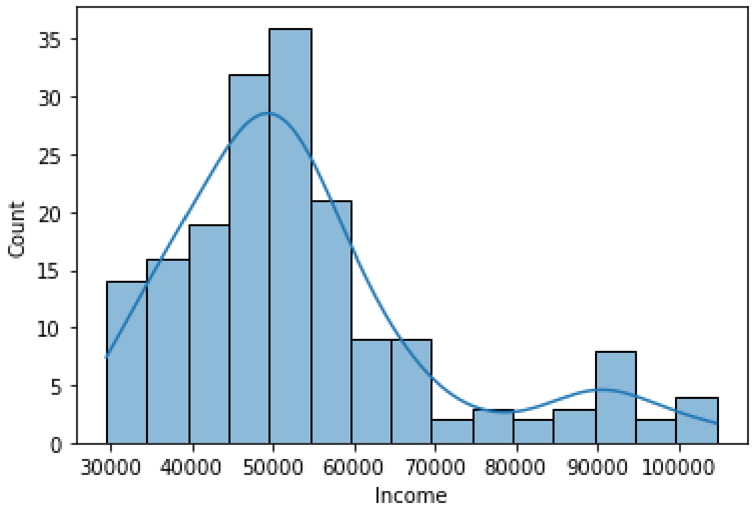


In [263]:

```
# Income levels of most of the people who buy are less than 69k
sns.histplot(x="Income",data=df,kde=True)
```

Out[263]:

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



In [270]:

```
df[df['Income']<69000][['Product']].value_counts()
```

Out[270]:

```
Product
KP281      80
KP481      60
KP781      16
dtype: int64
```

In [262]:

```
df[df['Income']>69000][['Product']].value_counts()
```

Out[262]:

```
Product
KP781      24
dtype: int64
```

From the above filter we can see that, all the people who have income above 69k buy the advanced featured treadmill i.e KP781

In [ ]:

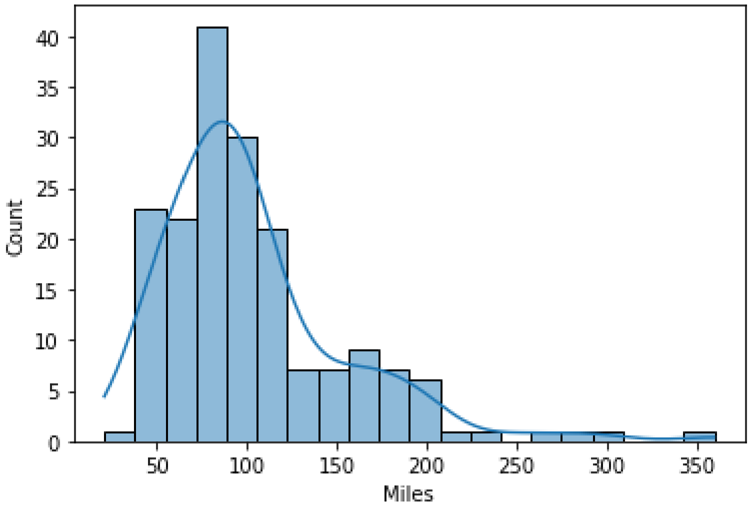
```

```

In [80]:

```
sns.histplot(x="Miles",data=df,kde=True)
```

Out[80]:

<AxesSubplot:xlabel='Miles', ylabel='Count'>

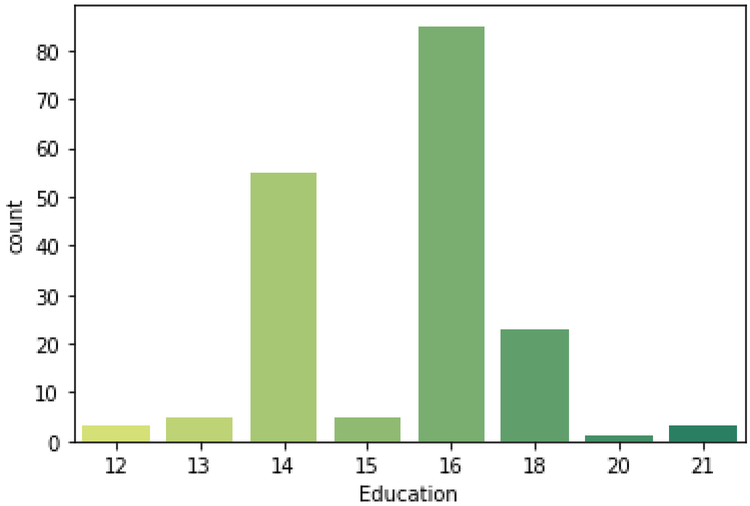


In [78]:

```
sns.countplot(x="Education",data=df,palette='summer_r')
```

Out[78]:

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

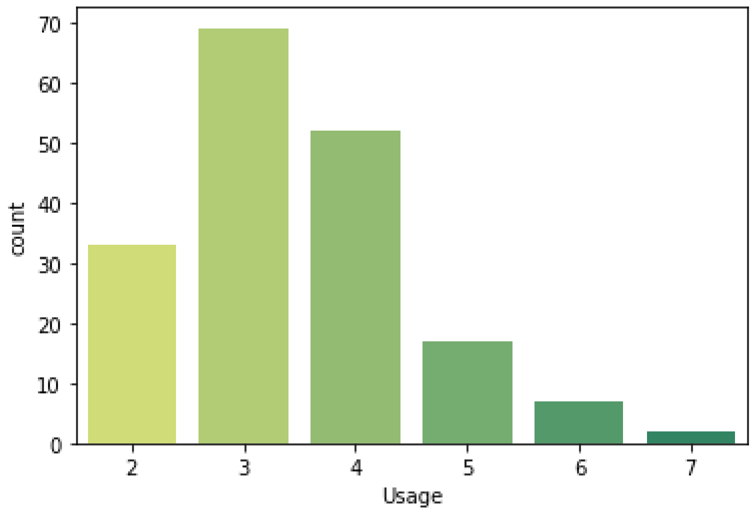


In [265]:

```
# Most of the people who want to use the treadmill wants to use it 2-4 times a week
sns.countplot(x="Usage",data=df,palette='summer_r')
```

Out[265]:

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

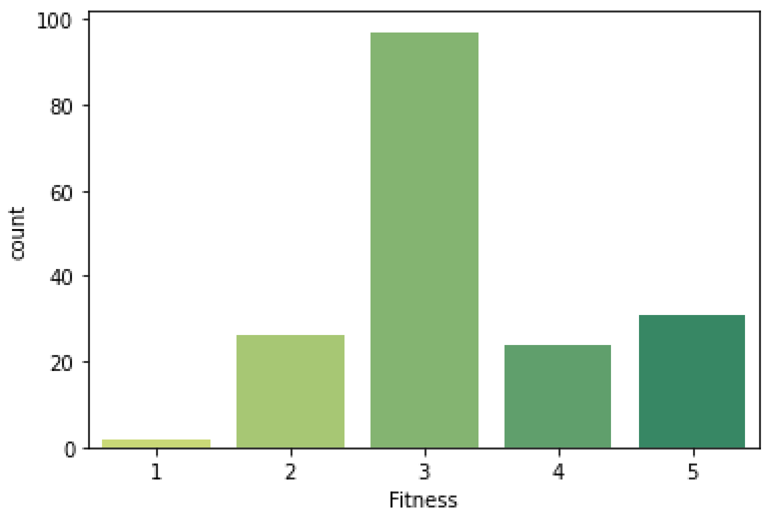


In [266]:

```
# most of the people have fitness rating of 3, which is average and also people who
# as we can see from the below plotted graph
sns.countplot(x="Fitness",data=df,palette='summer_r')
```

Out[266]:

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



In [ ]:

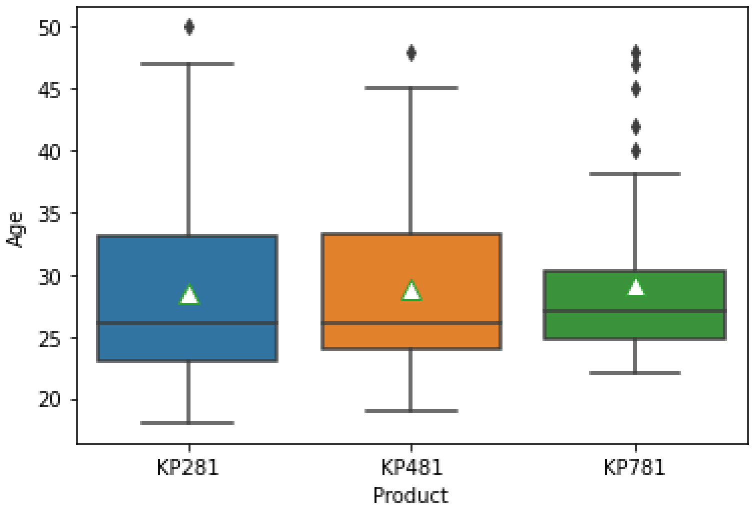
```

```

In [104]:

```
sns.boxplot(x='Product',y='Age',data=df,showmeans=True,meanprops={"markerfacecolor"
```

Out[104]:

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



In [ ]:

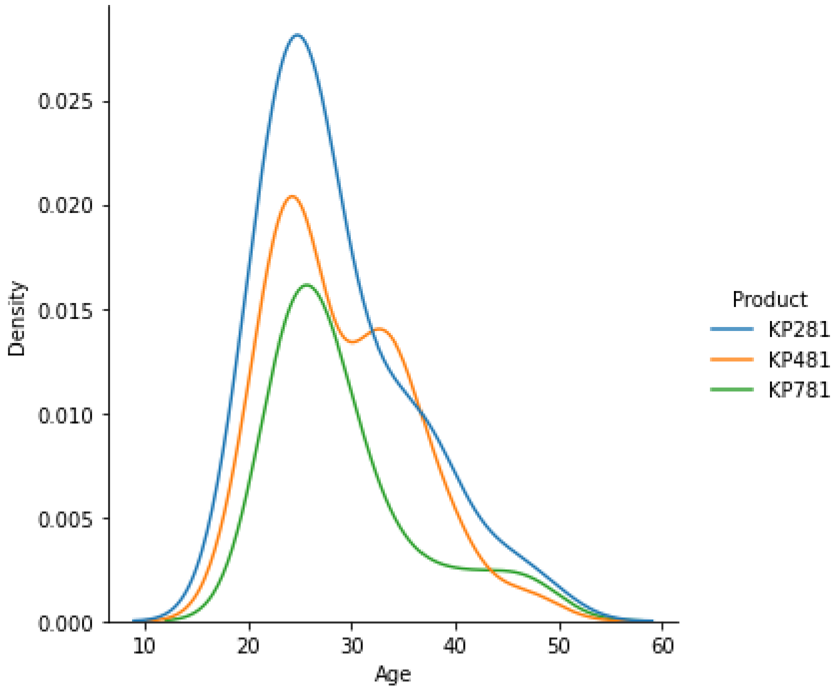
```

```

In [125]:

```
sns.displot(x="Age",hue='Product',data=df, kind="kde")
```

Out[125]:

```
<seaborn.axisgrid.FacetGrid at 0x7ff17a3da3a0>
```



Most people who buy KP781 treadmill are younger age group, compared with other products

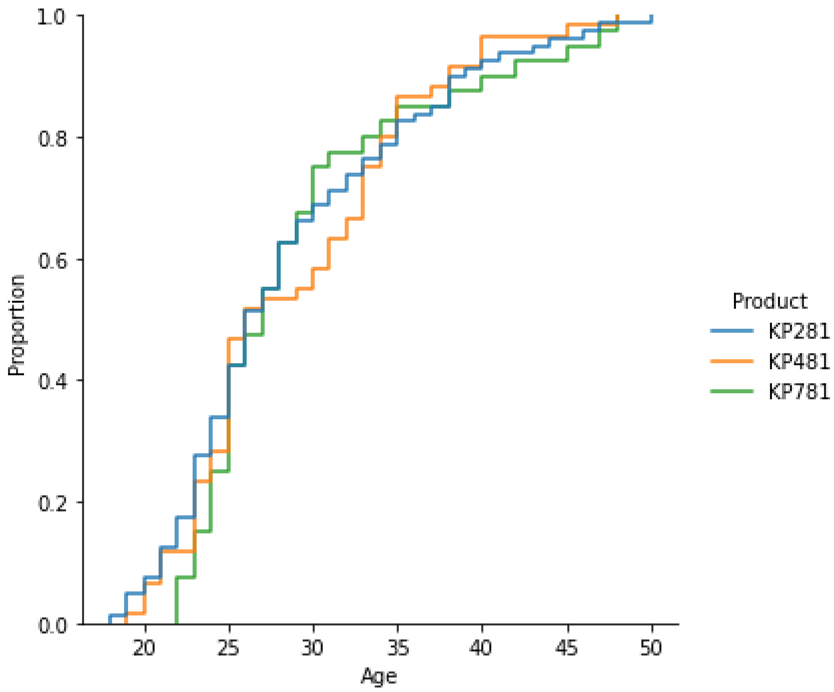
In [ ]:

In [124]:

```
sns.displot(x="Age",hue='Product',data=df, kind="ecdf")
```

Out[124]:

<seaborn.axisgrid.FacetGrid at 0x7ff17a6bed00>

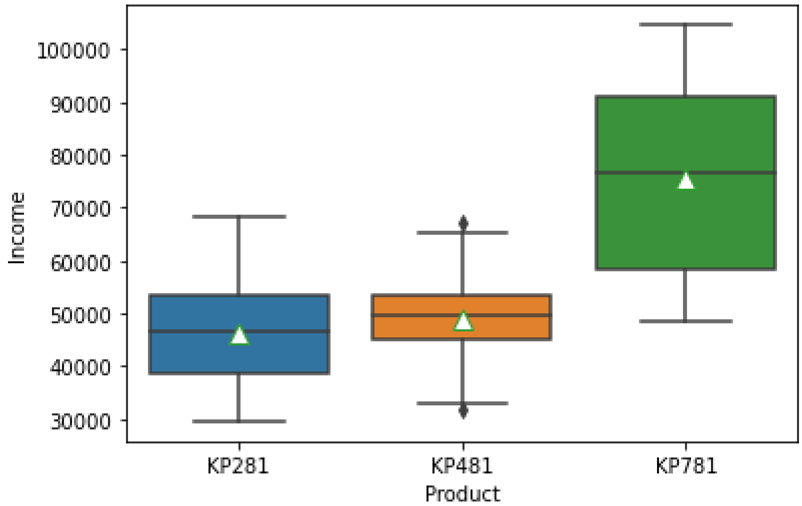


In [100]:

```
sns.boxplot(x='Product',y='Income',data=df,showmeans=True,meanprops={"markerfacecol
```

Out[100]:

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

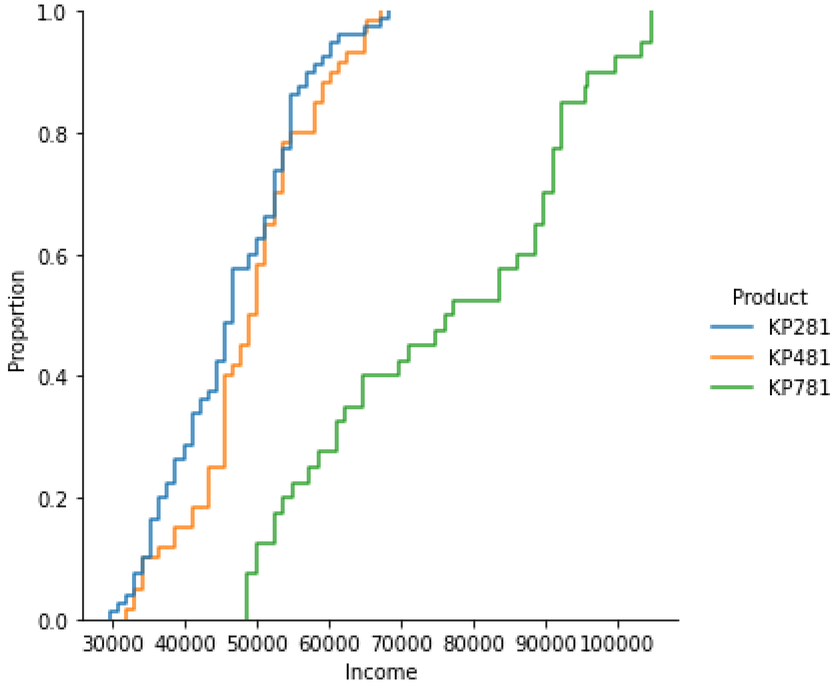


In [122]:

```
sns.displot(x="Income",hue='Product',data=df, kind="ecdf")
```

Out[122]:

<seaborn.axisgrid.FacetGrid at 0x7ff17a56a700>



Most of the people who buy KP781 are higher income people

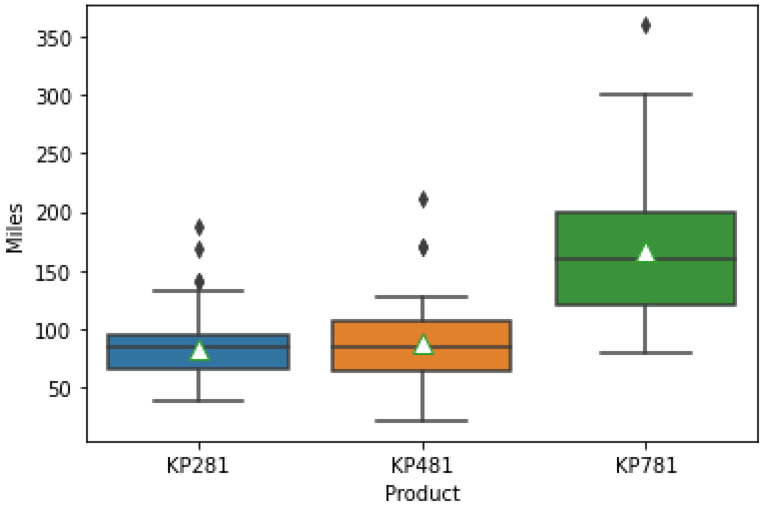
In [ ]:

In [102]:

```
sns.boxplot(x='Product',y='Miles',data=df,showmeans=True,meanprops={"markerfacecolo
```

Out[102]:

<AxesSubplot:xlabel='Product', ylabel='Miles'>

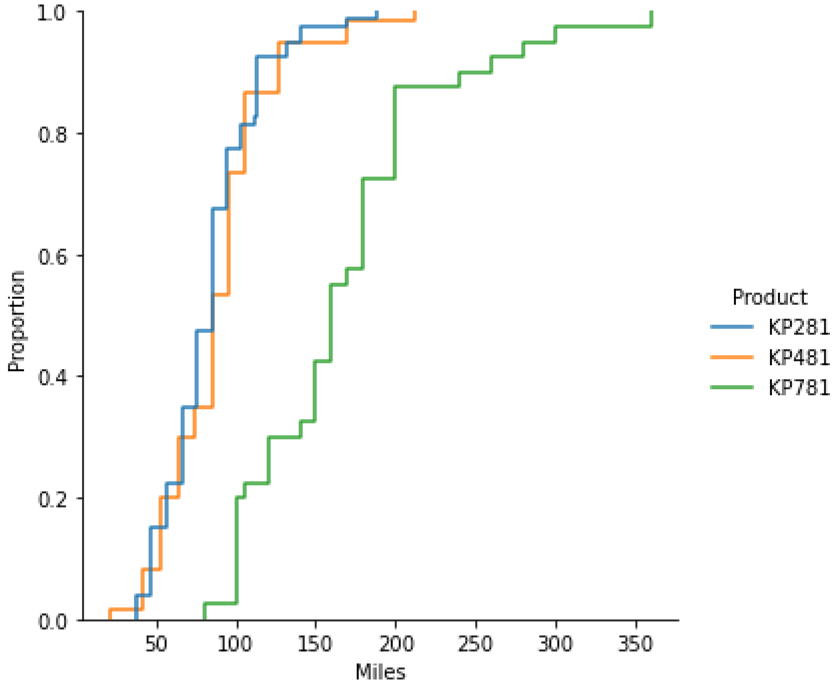


In [121]:

```
sns.displot(x="Miles",hue='Product',data=df, kind="ecdf")
```

Out[121]:

```
<seaborn.axisgrid.FacetGrid at 0x7ff17a6ab940>
```



People who buy KP781 treadmill wanted to run more miles compared to other products, we can illustrate from the above graphs

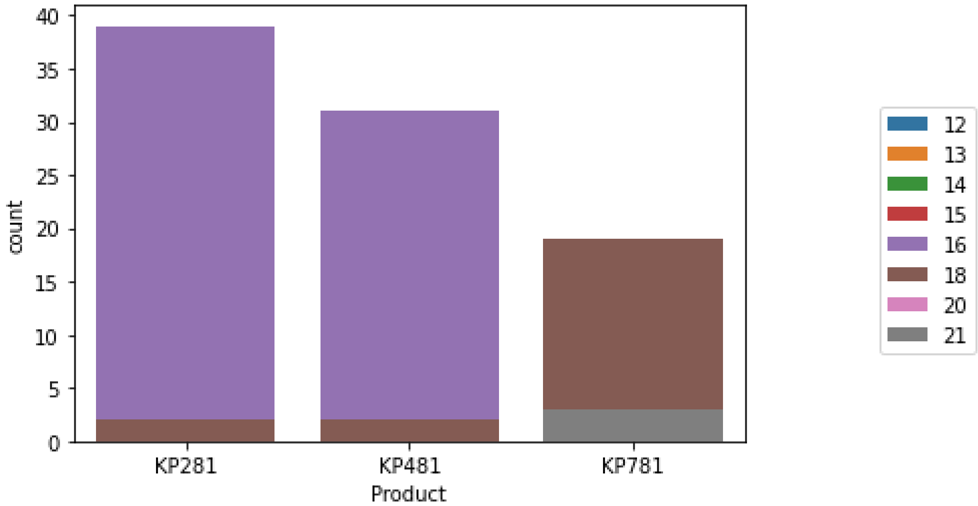
In [ ]:

In [166]:

```
sns.countplot(x='Product',hue='Education',data=df,dodge=False)
plt.legend( loc=(1.2,0.2))
```

Out[166]:

```
<matplotlib.legend.Legend at 0x7ff1790fca30>
```

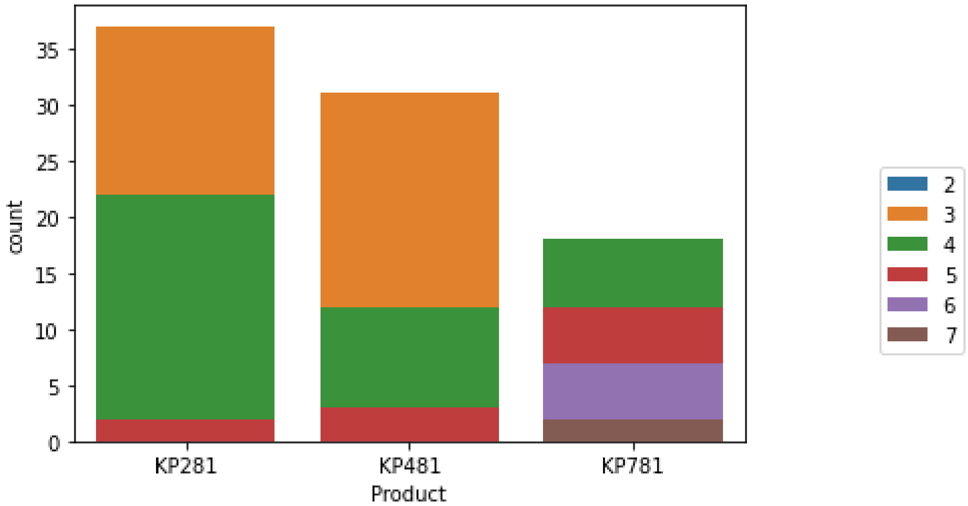


In [167]:

```
sns.countplot(x='Product',hue='Usage',data=df,dodge=False)
plt.legend( loc=(1.2,0.2))
```

Out[167]:

```
<matplotlib.legend.Legend at 0x7ff179037280>
```



People who buy KP781 product wanted to use it 4-7 times a week, while other 2 products is only 3-5 times a week

KP781 product purchased people wanted to run more times a week compared to lower cost product purchased people

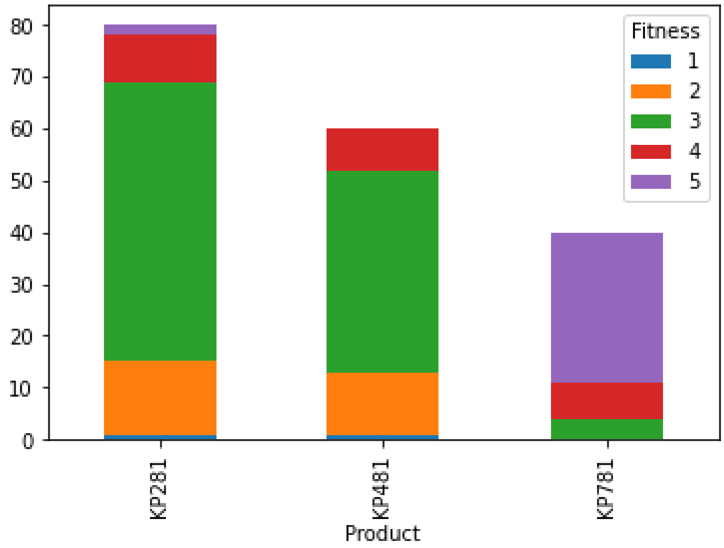
In [ ]:

In [273]:

```
df_plot = df.groupby(['Product', 'Fitness']).size().reset_index().pivot(columns='Fi
df_plot.plot(kind='bar', stacked=True)
```

Out[273]:

```
<AxesSubplot:xlabel='Product'>
```



People who brought the premium product i.e KP781 have fitness levels of 5 out of 5 for almost all people and only a few are with 3 and 4 rating and most people from other purchases are only fit with 3 rating

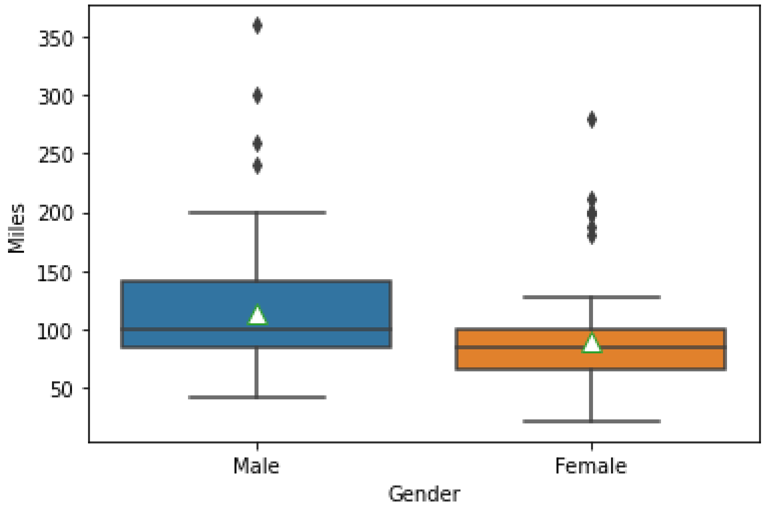
In [ ]:

In [131]:

```
sns.boxplot(x='Gender',y='Miles',data=df,showmeans=True,meanprops={"markerfacecolor
```

Out[131]:

<AxesSubplot:xlabel='Gender', ylabel='Miles'>

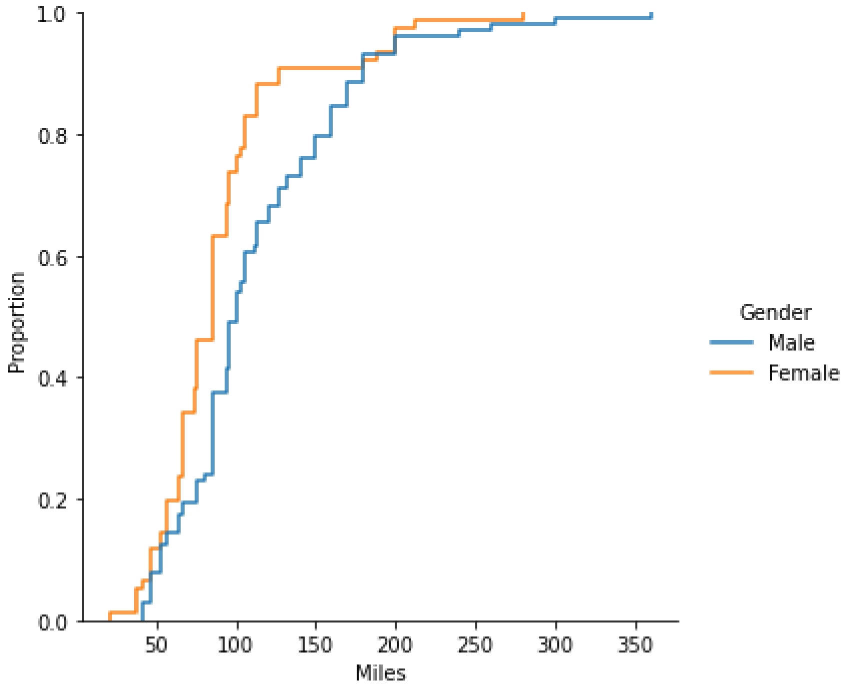


In [132]:

```
sns.displot(x="Miles",hue='Gender',data=df, kind="ecdf")
```

Out[132]:

<seaborn.axisgrid.FacetGrid at 0x7ff17a8be040>



In [ ]:

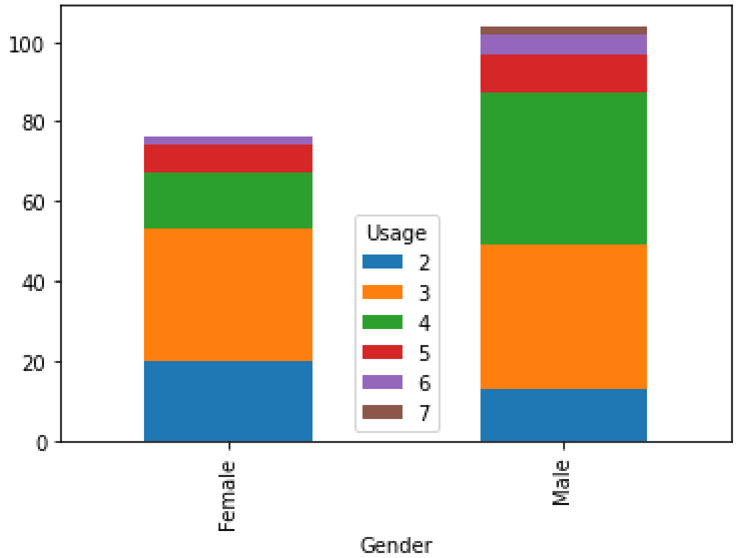
```

```

In [275]:

```
df_plot = df.groupby(['Gender', 'Usage']).size().reset_index().pivot(columns='Usage
df_plot.plot(kind='bar', stacked=True)
```

Out[275]:

<AxesSubplot:xlabel='Gender'>

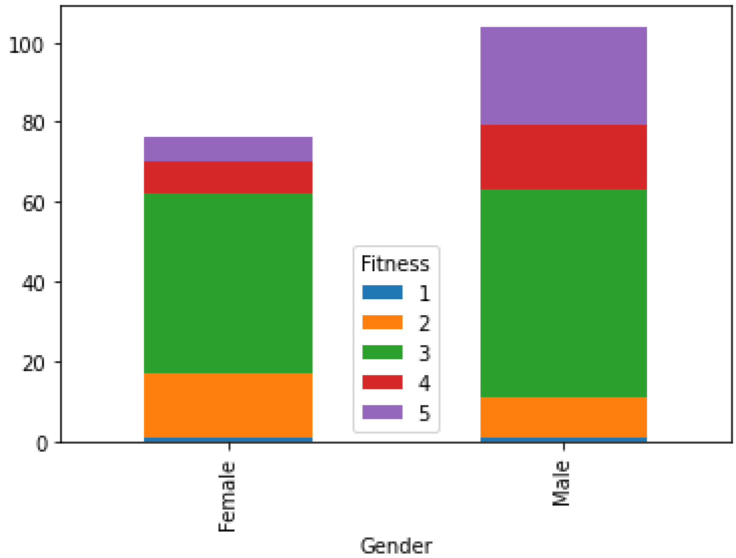


In [277]:

```
df_plot = df.groupby(['Gender', 'Fitness']).size().reset_index().pivot(columns='Fit
df_plot.plot(kind='bar', stacked=True)
```

Out[277]:

<AxesSubplot:xlabel='Gender'>



Male people are more fit compared to Female

In [ ]:

```

```

In [133]:

```
df.head()
```

Out[133]:

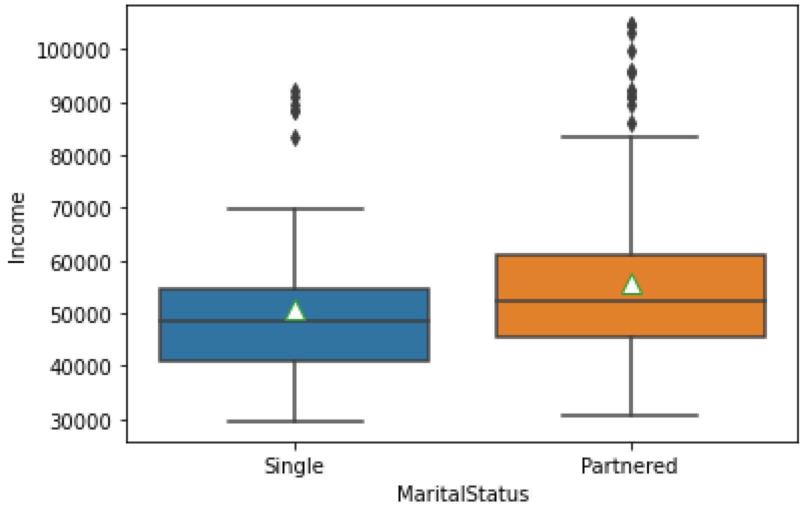
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

In [156]:

```
sns.boxplot(x='MaritalStatus',y='Income',data=df,showmeans=True,meanprops={"markerf
```

Out[156]:

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

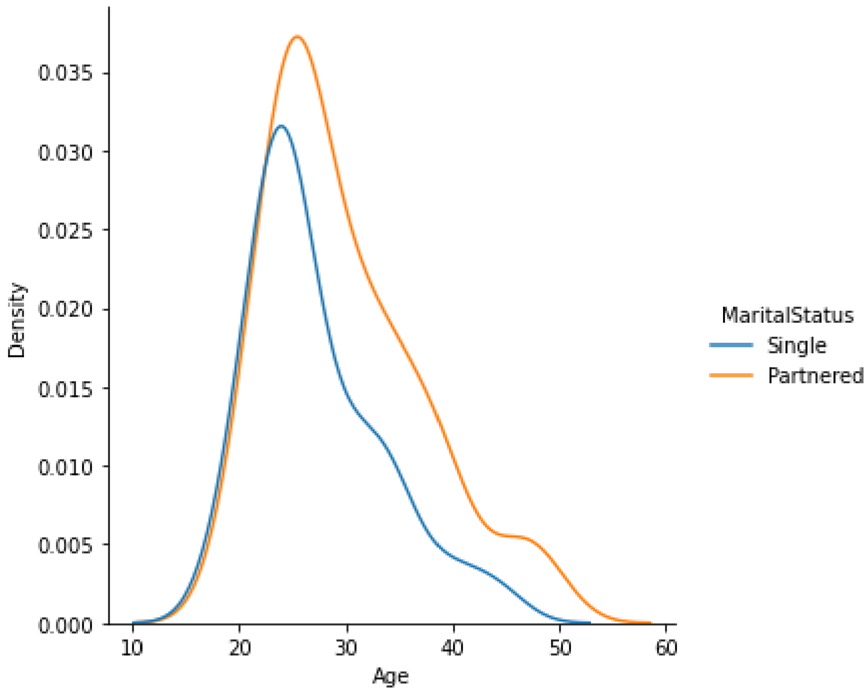


In [152]:

```
sns.displot(x="Age",hue='MaritalStatus',data=df, kind="kde")
```

Out[152]:

<seaborn.axisgrid.FacetGrid at 0x7ff1799bf3a0>



In [ ]:

In [ ]:

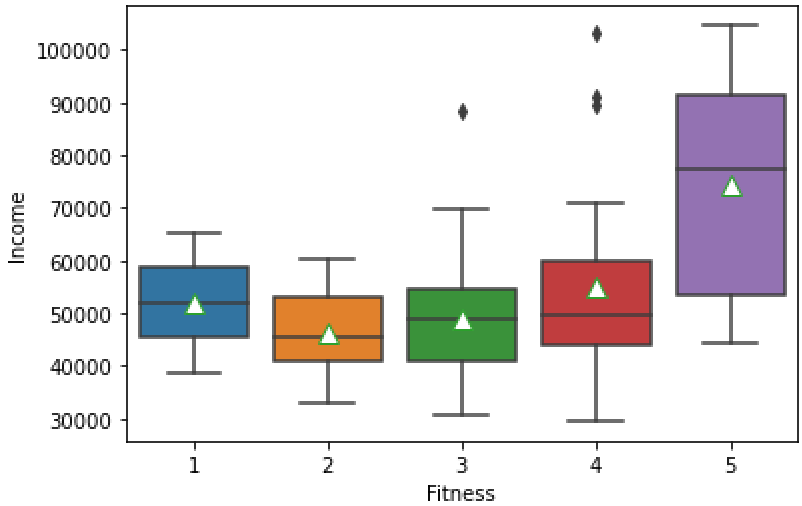
In [ ]:

In [170]:

```
sns.boxplot(x='Fitness',y='Income',data=df,showmeans=True,meanprops={"markerfacecol
```

Out[170]:

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

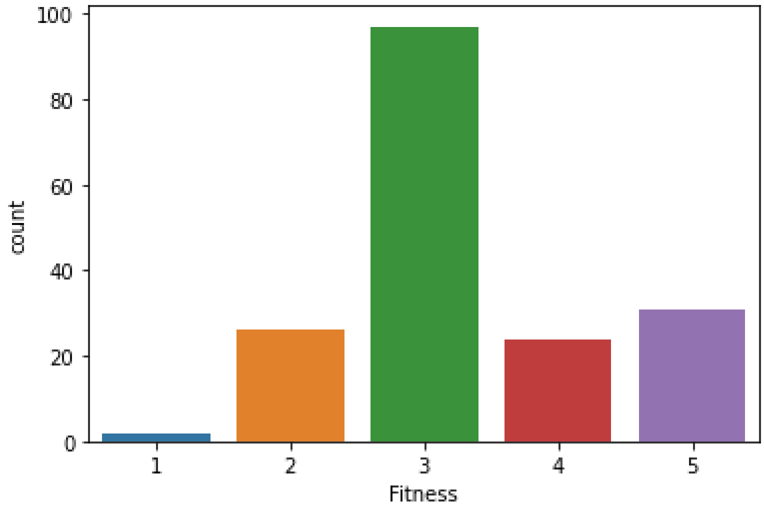


In [171]:

```
sns.countplot(x='Fitness',data=df)
```

Out[171]:

<AxesSubplot:xlabel='Fitness', ylabel='count'>



Most of the people who use threadmill are with fitness rating 3 and people with fitness rating 1 are not buying much

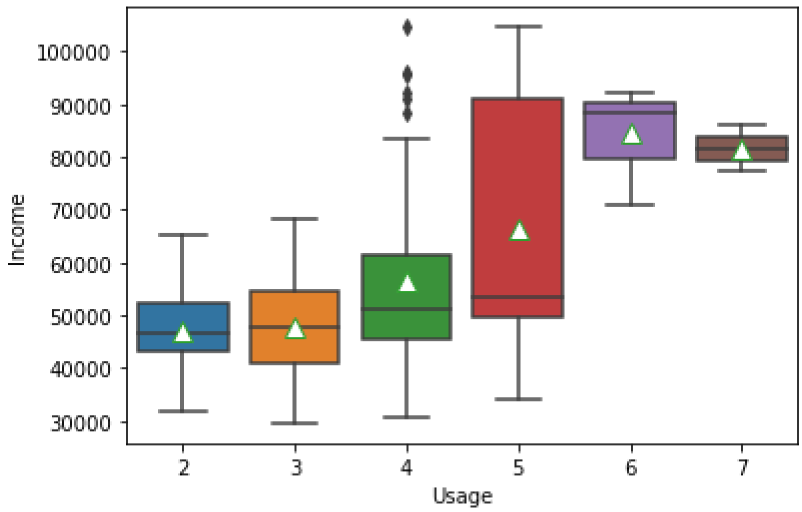
In [ ]:

In [182]:

```
sns.boxplot(x='Usage',y='Income',data=df,showmeans=True,meanprops={"markerfacecolor
```

Out[182]:

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

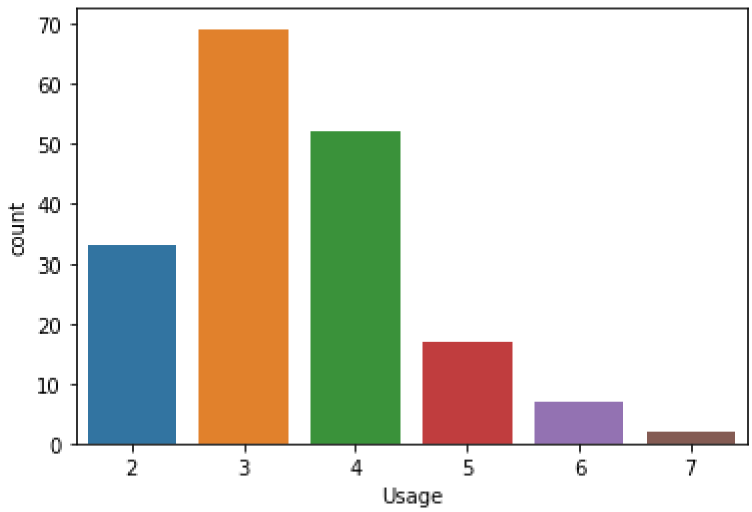


In [183]:

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

Out[183]:

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



Most of the peope who purchase the threadmill wanted to use 2-4 times a day

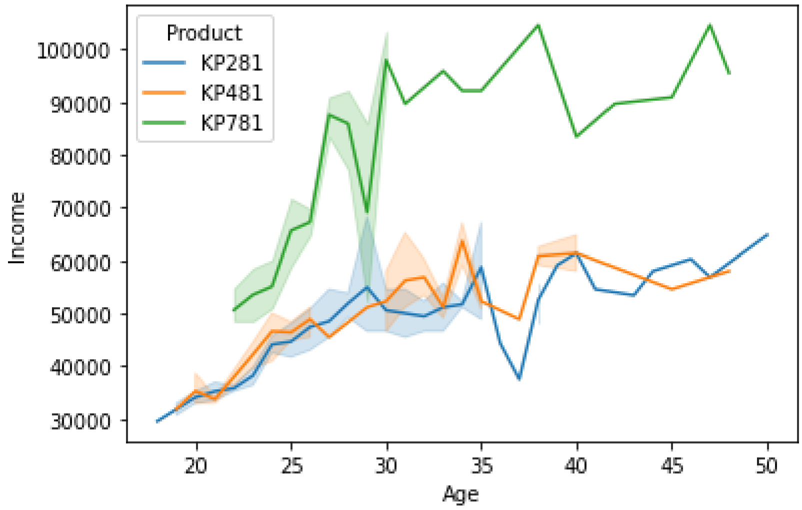
In [ ]:

In [15]:

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

Out[15]:

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

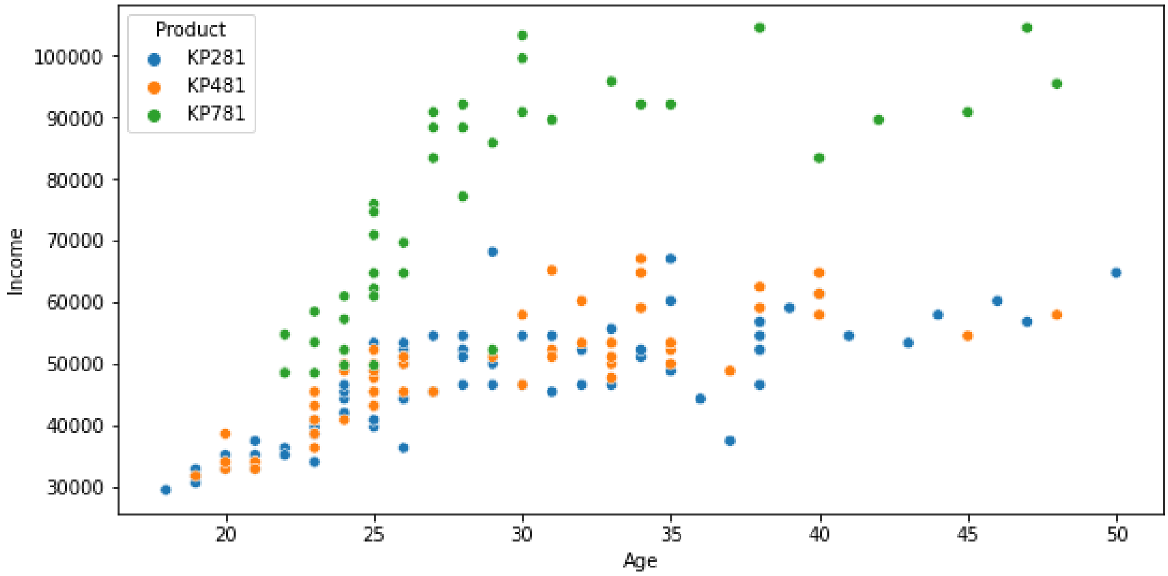


In [14]:

```
plt.figure(figsize=(10, 5))
sns.scatterplot(x="Age", y="Income", data=df, hue='Product')
```

Out[14]:

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

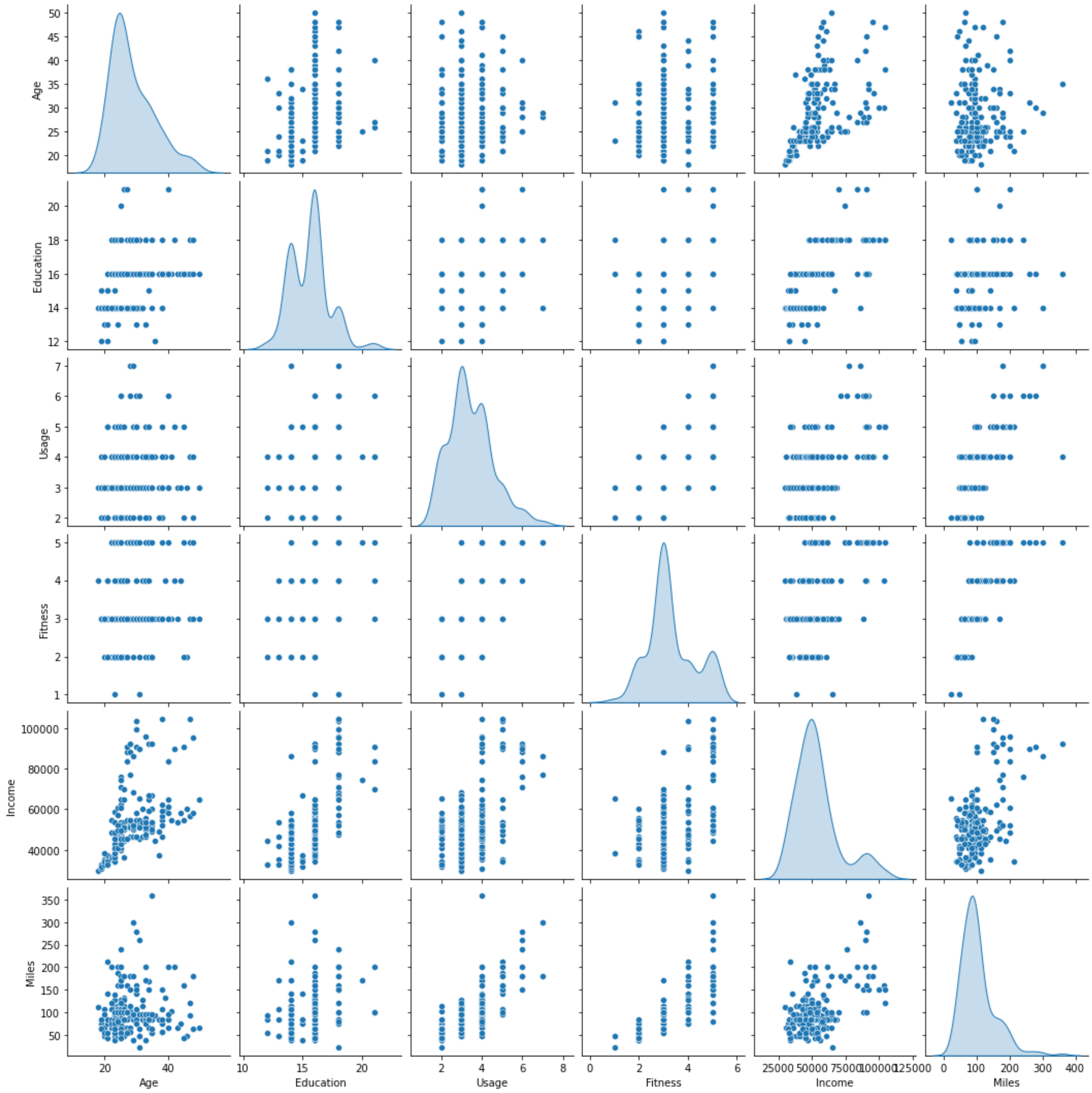


In [185]:

```
sns.pairplot(data=df,diag_kind='kde')
```

Out[185]:

<seaborn.axisgrid.PairGrid at 0x7ff178933d60>

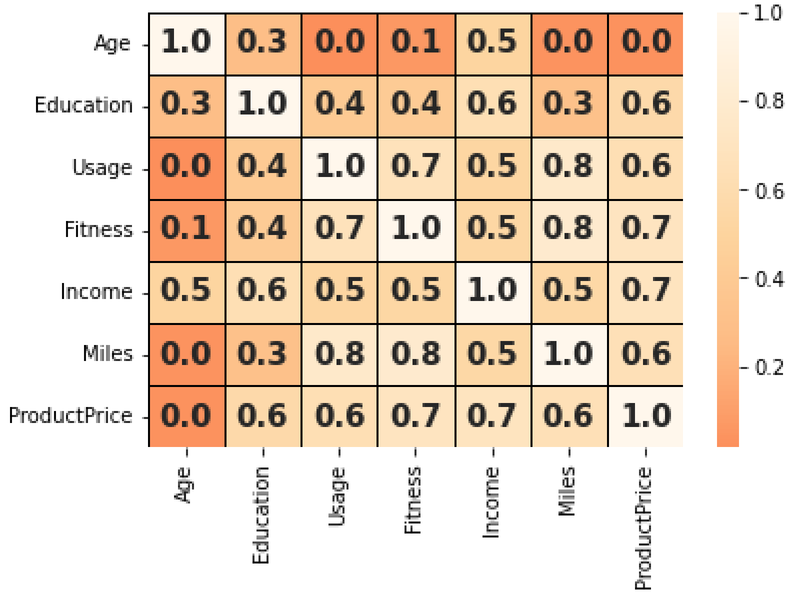


In [279]:

```
sns.heatmap(df.corr()
#           ,vmin=-1
            ,vmax=1
            ,center=0
            ,cmap='OrRd_r'
            ,annot=True
            ,fmt='.1f'
            ,annot_kws=dict(size=15,weight='bold')
            ,linecolor='black'
            ,linewidths=0.5)
```

Out[279]:

```
<AxesSubplot:>
```



From the above plots we can observe that Usage, Fitness, Miles are correlated with each other

In [ ]:

In [ ]:

In [195]:

```
df['Product'].value_counts(normalize=True).apply(round,args=(2,))
```

Out[195]:

```
KP281    0.44
KP481    0.33
KP781    0.22
Name: Product, dtype: float64
```

Probability of people who buy KP281 product is 0.44, KP481 is 0.33, KP781 is 0.22

In [221]:

```
df['Gender'].value_counts(normalize=True).apply(round,args=(2,))
```

Out[221]:

```
Male      0.58
Female    0.42
Name: Gender, dtype: float64
```

Probability of Female who buy product is 0.42, Male is 0.58

In [220]:

```
df['MaritalStatus'].value_counts(normalize=True).apply(round,args=(2,))
```

Out[220]:

```
Partnered    0.59
Single       0.41
Name: MaritalStatus, dtype: float64
```

Probability of Single who buy product is 0.41, Male is 0.59

In [188]:

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

Out[188]:

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

In [281]:

```
# P(Male / KP781)  # 82% of the people who buy KP781 are males
round(33/40,2)
```

Out[281]:

0.82

P(KP281 / Female) - the probability of Female who buy KP281 product is 0.53, so most women tend to buy the cheper product

male more or less have same distribution across products

For KP281, KP481 the product probability distribution is approx .5 for both male and female, only for KP781 its highly uneven.

In [192]:

```
round(40/76,2)
```

Out[192]:

0.53

In [196]:

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

Out[196]:

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

In [217]:

```
round(46/107,2) # P(Female/Partnered)
```

Out[217]:

0.43

In [210]:

```
round(30/73 ,2) # P(Female/Single)
```

Out[210]:

0.41

In [211]:

```
round(61/107 ,2) # P(Male/Partnered)
```

Out[211]:

```
0.57
```

In [212]:

```
round(43/73 ,2) # P(Male/Single)
```

Out[212]:

```
0.59
```

In [213]:

```
round(46/76 ,2) # P(Partnered/Female)
```

Out[213]:

```
0.61
```

In [214]:

```
round(30/76 ,2) # P(Single/Female)
```

Out[214]:

```
0.39
```

In [215]:

```
round( 61/104 ,2) # P(Partnered/Male)
```

Out[215]:

```
0.59
```

In [216]:

```
round(43/104 ,2) # P(Single/Male)
```

Out[216]:

```
0.41
```

Looks like all the Conditional probabilities are aligned approx equal to marginal probability, so we dont find much information from relation between MaritalStatus and Gender

In [ ]:

```

```

In [218]:

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

Out[218]:

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

Looks like all the Conditional probabilities are aligned approx equal to marginal probability, so we dont find much information from relation between MaritalStatus and Product

Partnered tend to buy 50% of products higher compared to Single across all products

In [ ]:

```

```

6. Recommendations - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand

1. KP281-entry-level and low price product has 44% of sales out of all 3 product - more people tend to buy cheaper products, so we can increase the distribution channel and optimize the supply chain with this product as we can increase the sales by catering the demand

2. 60% of the people who purchased the product are Married, so married people are much interested in buying this product, we can target married people for sales

3. People between 22-29 age groups have most sales, we can target this group of people for sales

4. All the people with income above 69k brought the KP781 high cost product, so we can target people with higher income than 69k for the sales of our premium product

5. As higher income has high correlation with education, we can target educated people as well to sell the premium product

6. Higher fitness rating people tend to buy costiler product, so people who are most fit tends to be more fitness freak and hence purchases the advanced threadmill, so we can target this people to purchase advanced product

7. 82% of the people who buy KP781 are males, so we can target males to buy our premium product

8. As the consumer metrics of both KP281 and KP481 are more similar to each other, we can push to sell the higher price product i.e KP481 to the customer, so our sale value increases

9. Customers who purchase KP481 are slightly higher aged people, so people with low - mid level income with age greater than 35 may choose this product due to decent number of features, we can market to this product to these type of customers

In [ ]:

```

```