In [2]:

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


Aerofit = pd.read_csv("C:\\Users\\bolla\\OneDrive\\Desktop\\Notes\\aerofit_treadmill.txt")
```

In [3]:

```
Aerofit
```

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
...	...	...	...	...	...	...	...	...	...
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 [4]:

```
print("Size of the data:", Aerofit.size, "elements")
```

```
Size of the data: 1620 elements
```

In [5]:

```
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: 12.8+ KB
```

In [6]:

```
Aerofit.shape
```

Out[6]:

```
(180, 9)
```

In [75]:

```
# % of missing values in each column
missing = Aerofit.isna().sum()/len(Aerofit)*100
missing
# There are no missing vlues n dataset
```

Out[75]:

```
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
```

Dataset has 1620 elements with 180 rows × 9 columns , no missing values

In [7]:

```
Aerofit.describe()
```

Out[7]:

	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

Age min is 18 & max is 50 , 75% of the population has 16 years of education Among 180 people 104 are male and 76 are female , Standard deviation is high in both Income & Miles when compared to others

In [9]:

```
Aerofit[["Product"]].value_counts()
```

Out[9]:

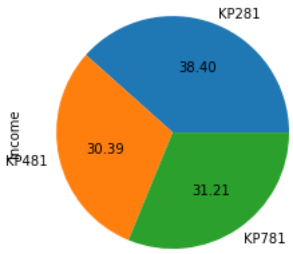
```
Product
KP281      80
KP481      60
KP781      40
dtype: int64
```

In [10]:

```
# Annual income % based on the trademill model

Aerofit.groupby("Product")["Income"].sum().plot(kind='pie',autopct="%.2f")
```

Out[10]:

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



The KP281 is an entry-level treadmill that sells for $1,500 has more Income and famous compared to other two types KP481 , KP781

In [11]:

```
Aerofit.nunique()
```

Out[11]:

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

In [12]:

```
Aerofit.groupby("Gender")["Product"].count()
```

Out[12]:

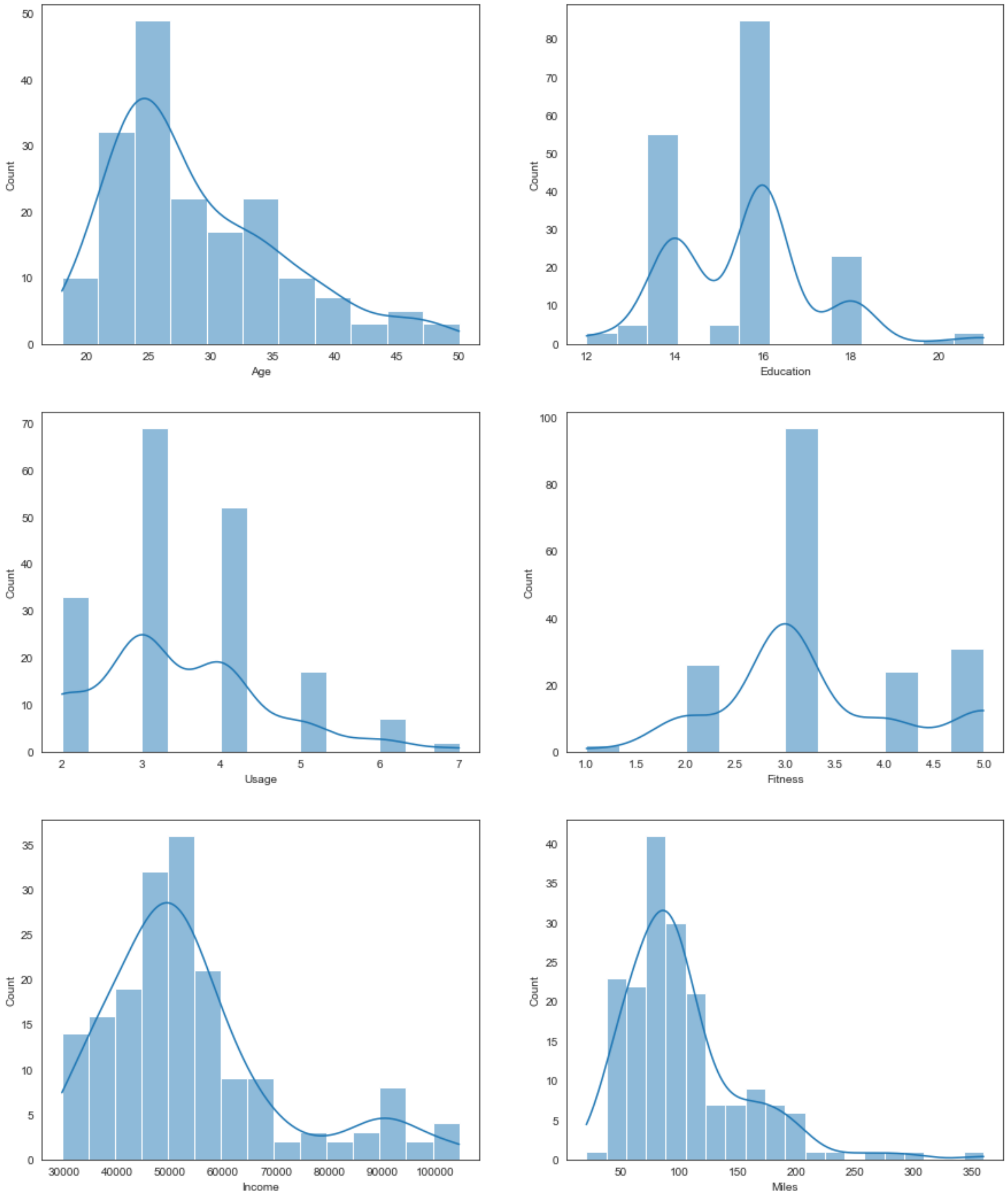
```
Gender
Female     76
Male      104
Name: Product, dtype: int64
```

In [13]:

```
### Quantative attributes in data are

### Age     Education   Usage   Fitness Income  Miles
```

In [76]:

```
fig, axis = plt.subplots(nrows = 3, ncols = 2, figsize =(15,13))
fig.subplots_adjust(top=1.2)

sns.histplot(data =Aerofit , x ='Age', kde = True , ax =axis[0,0] )
sns.histplot(data =Aerofit , x ='Education', kde = True , ax =axis[0,1] )
sns.histplot(data =Aerofit , x ='Usage', kde = True , ax =axis[1,0] )
sns.histplot(data =Aerofit , x ='Fitness', kde = True , ax =axis[1,1] )
sns.histplot(data =Aerofit , x ='Income', kde = True , ax =axis[2,0] )
sns.histplot(data =Aerofit , x ='Miles', kde = True , ax =axis[2,1] )

plt.show()
```



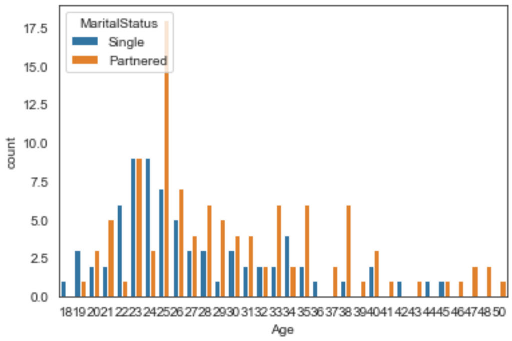
Highest number of people using the products are young, with age 24. People with education 15 and 16 are more, People with Fitness recording 3 are most common, Income is high in 50000 recording compared to others , Average miles covered is 85-90

In [77]:

```
sns.countplot(data =Aerofit , x ='Age',  hue = "MaritalStatus")
```

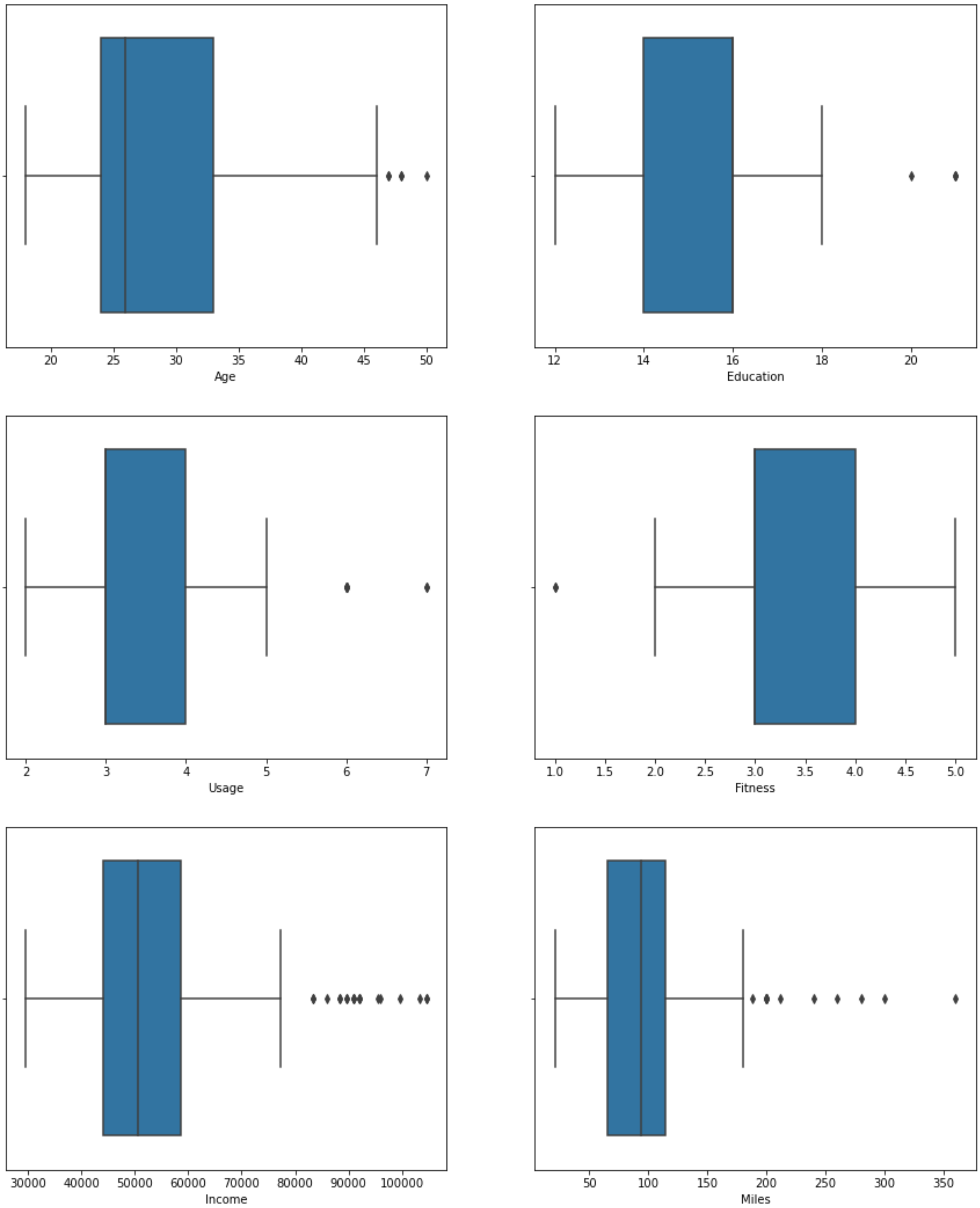
Out[77]:

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



In [15]:

```
fig, axis = plt.subplots(nrows = 3, ncols = 2, figsize =(15,13))
fig.subplots_adjust(top=1.2)

sns.boxplot(data =Aerofit , x ='Age',  ax =axis[0,0])
sns.boxplot(data =Aerofit , x ='Education',  ax =axis[0,1])
sns.boxplot(data =Aerofit , x ='Usage',  ax =axis[1,0])
sns.boxplot(data =Aerofit , x ='Fitness',  ax =axis[1,1])
sns.boxplot(data =Aerofit , x ='Income',  ax =axis[2,0])
sns.boxplot(data =Aerofit , x ='Miles',  ax =axis[2,1])

plt.show()
```



Income and Miles columns have more outliers compared to others , this states the higher the Standard deviation the higher the outliers, we can even see a high difference in mean and median in both Income and Miles columns compared to others

In [16]:

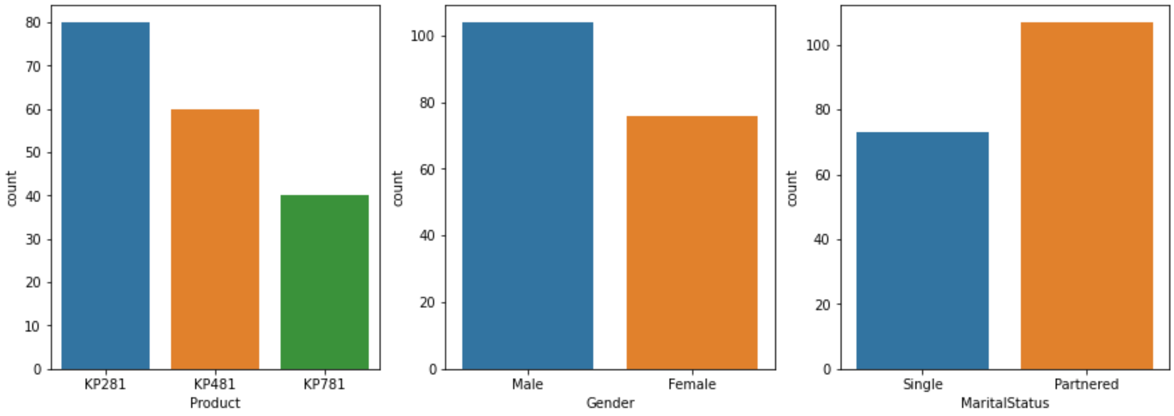
```
### Qualitative attributes in data are

### Product     Gender  MaritalStatus
```

In [17]:

```
fig , axis = plt.subplots(nrows = 1 , ncols = 3 , figsize = (15,5))

sns.countplot(data =Aerofit , x ='Product',  ax =axis[0])
sns.countplot(data =Aerofit , x ='Gender',  ax =axis[1])
sns.countplot(data =Aerofit , x ='MaritalStatus',  ax =axis[2])
```

Out[17]:

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



KP281 is the most purchased product as per data , Males tend to purchase this product more than females, Customers with MaritalStatus as Partnered have more purchases than Single ,

In [18]:

```
pb_df = Aerofit[["Product","Gender","MaritalStatus"]].melt()
```

In [19]:

```
pb_df.groupby(["variable","value"])["value"].count()/len(Aerofit)
```

Out[19]:

```
variable       value
Gender         Female       0.422222
               Male         0.577778
MaritalStatus  Partnered    0.594444
               Single       0.405556
Product        KP281        0.444444
               KP481        0.333333
               KP781        0.222222
Name: value, dtype: float64
```

Marginal probability -
In Gender 42% probability of females and 57.7% probability of Males purchasing the product. In Marital Status 59.4% probability of Partnered and 40.5% probability of single's purchasing the product. Product type overall KP281 has more sales with 44.4% probability , KP481 stands second with 33.3% probability , KP781 has least sales with 22.2% probability

In [20]:

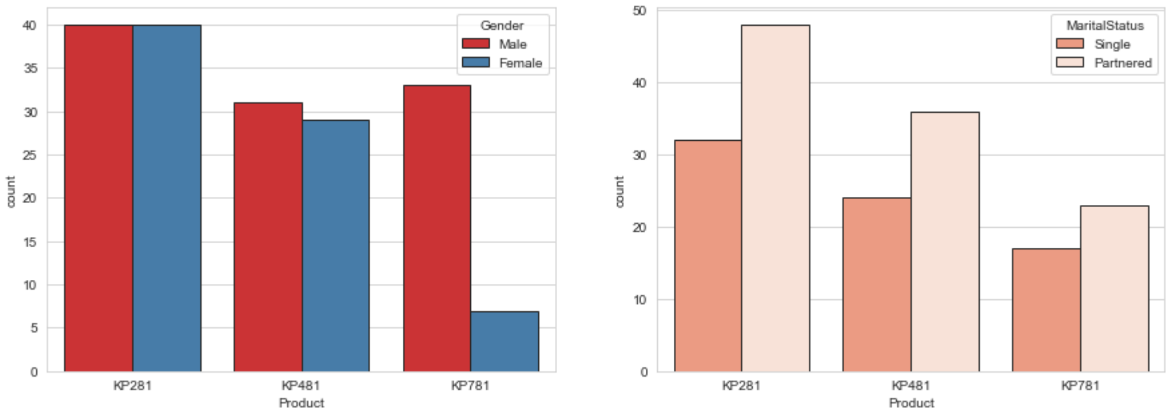
```
# Bivariant analysis , influence of Gender,MaritalStatus on type of product purchase.
```

In [21]:

```
sns.set_style("whitegrid")

fig , axis = plt.subplots(nrows = 1 , ncols = 2 , figsize = (15,5))
sns.countplot(data =Aerofit , x ='Product',hue = "Gender" ,edgecolor="0.15", palette='Set1', ax =axis[0])
sns.countplot(data =Aerofit , x ='Product', hue = "MaritalStatus",edgecolor="0.15", palette=["#fc9272","#fee0d2"] , ax =axis[1])
```

Out[21]:

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



Equal number of Males and females purchased KP281 and almost same for purchased KP481 , KP781 was mostly purchased by males, Partnered people have more purchases when compared to single people.

In [22]:

```
plt.figure(figsize=(10,8))
sns.heatmap(Aerofit.corr(),cmap="crest",annot= True)
```

Out[22]:

```
<AxesSubplot:>
```

In [23]:

```
sns.pairplot(Aerofit, hue = "Product",height=1.5)
```

Out[23]:

```
<seaborn.axisgrid.PairGrid at 0x2b65d520c10>
```

In [ ]:

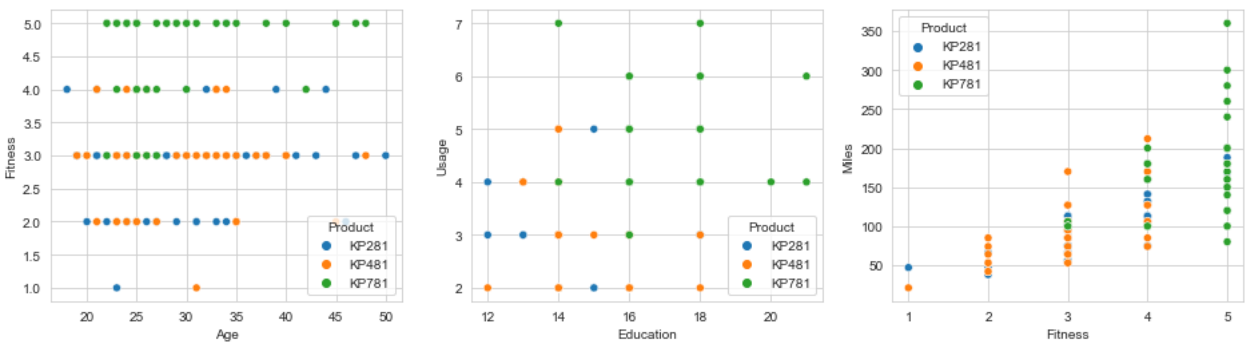
```
Product Age Gender  Education   MaritalStatus   Usage   Fitness Income  s
```

In [34]:

```
fig, axis = plt.subplots(nrows = 1, ncols = 3, figsize =(16,4))

sns.scatterplot(data =Aerofit , x ='Age',y ="Fitness", hue ="Product"  , ax =axis[0] )
sns.scatterplot(data =Aerofit , x ='Education',y ="Usage", hue ="Product"  , ax =axis[1] )
sns.scatterplot(data =Aerofit , x ='Fitness',y ="Miles", hue ="Product"  , ax =axis[2] )
```

Out[34]:

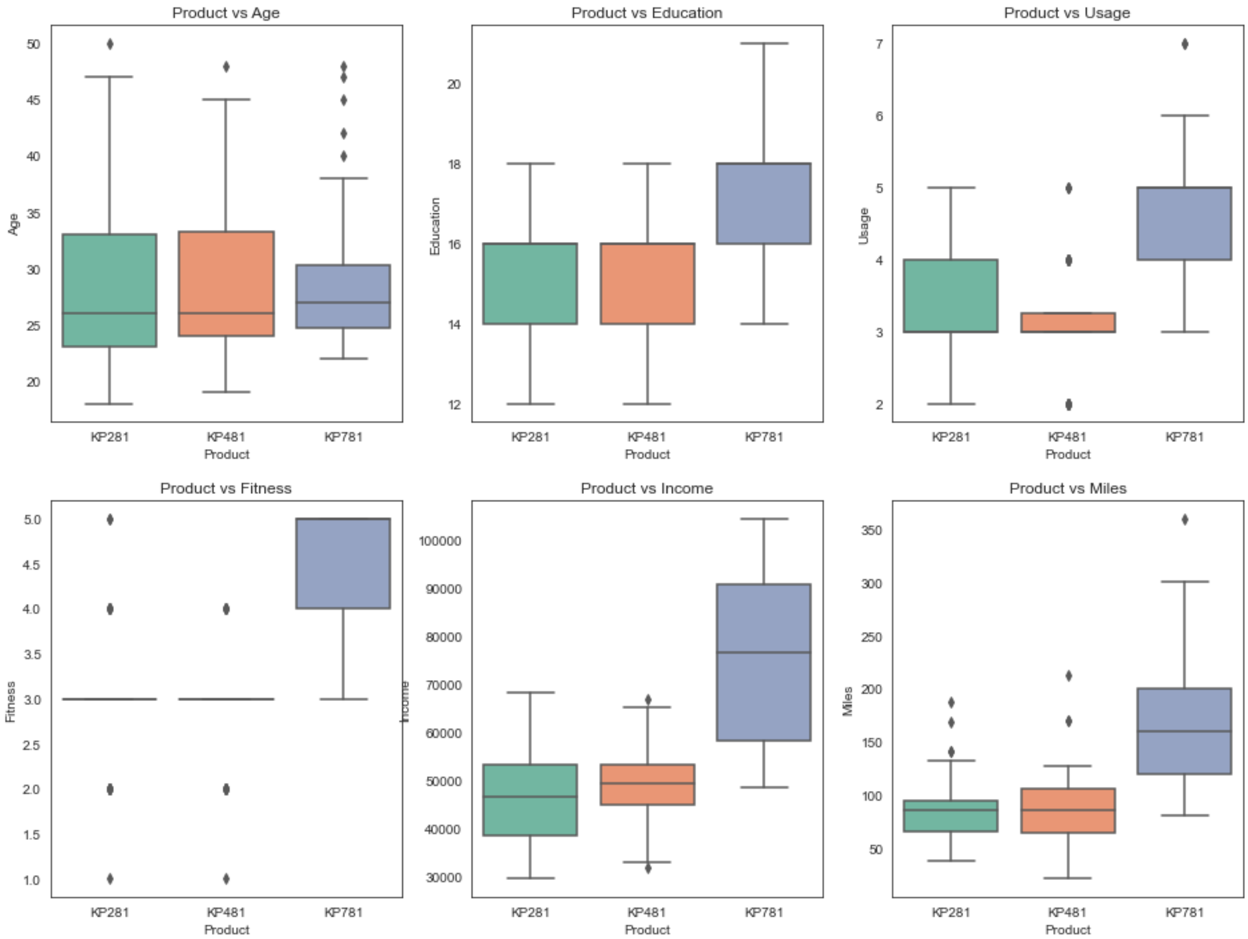
```
<AxesSubplot:xlabel='Fitness', ylabel='Miles'>
```



By analyzing relationship between “Age,Education,Usage,Fitness,Miles” with Product type , we can understand that people with High fitness , Usage and more number of miles have preferred KP781 which has advanced features . Customers with education more than 16 have higher chance of buying KP781 , KP481 has a an average affect on Fitness and and miles used

In [40]:

```
fig, axis = plt.subplots(nrows = 2, ncols = 3, figsize =(16,12))
sns.set_style("white")

count = 0
vr=["Age","Education","Usage","Fitness","Income","Miles"]

for i in range(2):
    for j in range(3):
        sns.boxplot(data= Aerofit,x="Product",y=vr[count],ax=axis[i,j],palette='Set2')

        axis[i,j].set_title(f"Product vs {vr[count]}")

        count += 1
```



Median Age of KP281 and KP481 are same , KP781 has slightly higher median in age, also more outliers than remaining two types. KP781 purchasers seem to be more educated than KP281,KP481 purchasers . Product usage also seems high in KP781 compared to other types but KP481 has outliers in usage . Even in fitness KP781 is higher than remaining and doesn't have any outliers like remaining two types. Income range is high for KP781 even though KP281 has more sum of income. In miles KP781 has a stable and high recordings when compared to other two with outliers.

In [44]:

```
# Probability of Individual Product type with respect to Gender for overall data (Marginal Probability)
# example - Probability of males purchasing KP281

pd.crosstab(Aerofit["Product"],Aerofit["Gender"], normalize=True, margins=True, margins_name = "Total").round(2)
```

Out[44]:

Gender	Female	Male	Total
Product			
KP281	0.22	0.22	0.44
KP481	0.16	0.17	0.33
KP781	0.04	0.18	0.22
Total	0.42	0.58	1.00

In [45]:

```
# Probability of Individual Product type with respect to MaritalStatus for overall data (Marginal Probability)
# example - Probability of Singles purchasing KP281

pd.crosstab(Aerofit["Product"],Aerofit["MaritalStatus"], normalize=True, margins=True, margins_name = "Total").round(2)
```

Out[45]:

MaritalStatus Product	Partnered	Single	Total
KP281	0.27	0.18	0.44
KP481	0.20	0.13	0.33
KP781	0.13	0.09	0.22
Total	0.59	0.41	1.00

In [74]:

```
# Probability of Individual Product type with respect to Individual Gender  (Conditional Probability)
# example - Probability of males purchasing KP281 among all the males

pb_kp281_males = round(len(Aerofit[(Aerofit["Product"]== "KP281") & (Aerofit["Gender"]==
                                                                "Male")])/len(Aerofit[(Aerofit["Gender"]== "Male")]),2)
pb_kp481_males = round(len(Aerofit[(Aerofit["Product"]== "KP481") & (Aerofit["Gender"]==
                                                                "Male")])/len(Aerofit[(Aerofit["Gender"]== "Male")]),2)
pb_kp781_males = round(len(Aerofit[(Aerofit["Product"]== "KP781") & (Aerofit["Gender"]==
                                                                "Male")])/len(Aerofit[(Aerofit["Gender"]== "Male")]),2)

pb_kp281_females = round(len(Aerofit[(Aerofit["Product"]== "KP281") & (Aerofit["Gender"]==
                                                                "Female")])/len(Aerofit[(Aerofit["Gender"]== "Female")]),2)
pb_kp481_females = round(len(Aerofit[(Aerofit["Product"]== "KP481") & (Aerofit["Gender"]==
                                                                "Female")])/len(Aerofit[(Aerofit["Gender"]== "Female")]),2)
pb_kp781_females = round(len(Aerofit[(Aerofit["Product"]== "KP781") & (Aerofit["Gender"]==
                                                                "Female")])/len(Aerofit[(Aerofit["Gender"]== "Female")]),2)


print("P(KP281|Male) = ",pb_kp281_males)
print("P(KP481|Male) = ",pb_kp481_males)
print("P(KP781|Male) = ",pb_kp781_males)
print("  ")
print("P(KP281|Female) = ",pb_kp281_females)
print("P(KP481|Female) = ",pb_kp481_females)
print("P(KP781|Female) = ",pb_kp781_females)
```

```
P(KP281|Male) =  0.38
P(KP481|Male) =  0.3
P(KP781|Male) =  0.32

P(KP281|Female) =  0.53
P(KP481|Female) =  0.38
P(KP781|Female) =  0.09
```

In [71]:

```
# Probability of Individual Product type with respect to Individual MaritalStatus  (Conditional Probability)
# example - Probability of Singles purchasing KP281 among all the Singles

pb_kp281_Single = round(len(Aerofit[(Aerofit["Product"]== "KP281") & (Aerofit["MaritalStatus"]==
                                                            "Single")])/len(Aerofit[(Aerofit["MaritalStatus"]=="Single")]),2)
pb_kp481_Single = round(len(Aerofit[(Aerofit["Product"]== "KP481") & (Aerofit["MaritalStatus"]==
                                                            "Single")])/len(Aerofit[(Aerofit["MaritalStatus"]=="Single")]),2)
pb_kp781_Single = round(len(Aerofit[(Aerofit["Product"]== "KP781") & (Aerofit["MaritalStatus"]==
                                                            "Single")])/len(Aerofit[(Aerofit["MaritalStatus"]=="Single")]),2)

pb_kp281_Partnered = round(len(Aerofit[(Aerofit["Product"]=="KP281") & (Aerofit["MaritalStatus"]==
                                                            "Partnered")])/len(Aerofit[(Aerofit["MaritalStatus"]=="Partnered")]),2)
pb_kp481_Partnered = round(len(Aerofit[(Aerofit["Product"]=="KP481") & (Aerofit["MaritalStatus"]==
                                                            "Partnered")])/len(Aerofit[(Aerofit["MaritalStatus"]=="Partnered")]),2)
pb_kp781_Partnered = round(len(Aerofit[(Aerofit["Product"]=="KP781") & (Aerofit["MaritalStatus"]==
                                                            "Partnered")])/len(Aerofit[(Aerofit["MaritalStatus"]=="Partnered")]),2)


print("P(KP281|Single) = ",pb_kp281_Single)
print("P(KP481|Single) = ",pb_kp481_Single)
print("P(KP781|Single) = ",pb_kp781_Single)
print("  ")
print("P(KP281|Partnered) = ",pb_kp281_Partnered)
print("P(KP481|Partnered) = ",pb_kp481_Partnered)
print("P(KP781|Partnered) = ",pb_kp781_Partnered)
```

```
P(KP281|Single) =  0.44
P(KP481|Single) =  0.33
P(KP781|Single) =  0.23

P(KP281|Partnered) =  0.45
P(KP481|Partnered) =  0.34
P(KP781|Partnered) =  0.21
```

Both in males and females KP281 has highest probability for purchasing, KP781 has a good chance of purchase among males than in females , KP281 has high probability among both single and partnered customers.

In [73]:

```
Aerofit.head(10)
```

Out[73]:

	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

Recommendations

Customers to Target - Age 18 to 35 . Education 13 and above. Males customers for KP781 as males have high probability of buying advanced models. Partnered status people tend to buy Treadmill more as it could be used by multiple people and also for fitness. People with High usage tend to buy KP781 and people with minimal usage tend to buy KP281. People with Income 60000 and more buy KP781 and with Income 55000 to 45000 buy KP481 , people with Income 55000 to 38000 buy KP281 , hence depending on income customer can be targeted. People with High fitness and more number of miles covered buy KP781 having advanced features.

Regions that need offers and promotion for increase in sales - Creating more awareness using promotions and attracting with offers specified to Female customers can increase sales in Female customers. As singles think it would not be viable to purchase a treadmill for their own , it would be great to provide options like Try & Buy , selling refurbished at less cost . People with low Income can be attracted with different finance options and offers. Customers that have low education can be attracted using free 1 month online recorded lessons based on the type of product. Costumes with low Usage and less miles covered can be attracted using Exchange offer between KP281 and KP781 or KP481 , as then tend to buy basic versions for trail and minimal usage which later can be persuaded to pick a advanced one like KP781.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: