aerofit-buss-case

March 16, 2024

Problem Statement: Aerofit is a leading brand in the field of fitness equipment. We Need to identify the characteristics of the target audience for each type of treadmill offered by the company, and provide a better recommendation of the treadmills to new customers.

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('F:\\buss_cass\\data\\aerofit_treadmill.txt')
```

Checking the structure & Characteristics of the dataset

[3]:
```
data.head()
```

[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

[4]:
```
data.shape
```

[4]:
```
(180, 9)
```

[80]:
```
#Checking nulls
data.isna().sum()
```

[80]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
```

```
dtype: int64
```

[6]:
```
#Datatypes of columns
data.dtypes
```

[6]:
```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

[14]:
```
#Describe Data
data.describe()
```

[14]:
```
              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
```

[22]:
```
#Finding Unique values

print(data['Product'].unique())
print(data['Gender'].unique())
print(data['MaritalStatus'].unique())
```

```
['KP281' 'KP481' 'KP781']
['Male' 'Female']
```

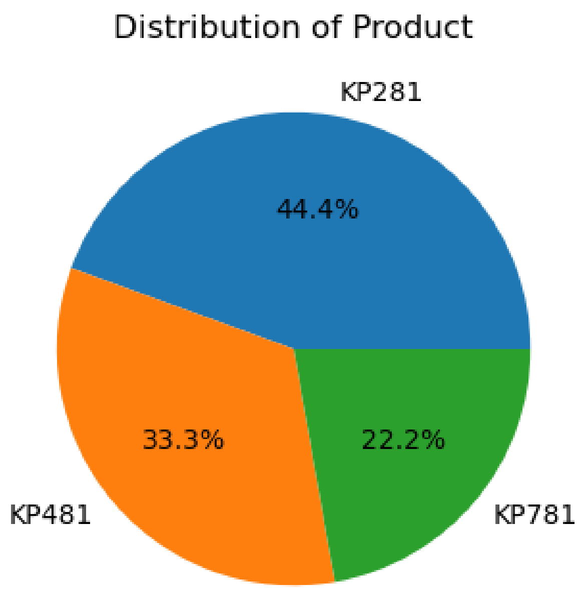
```
['Single' 'Partnered']
```

Finding Distribution of Product, Gender, MaritalStatus

[86]:
```
count = data[data['Product'] == 'KP281'].shape
x = count[0]
count = data[data['Product'] == 'KP481'].shape
y = count[0]
count = data[data['Product'] == 'KP781'].shape
z = count[0]
```

[88]:
```
list_Product = [x,y,z]
```

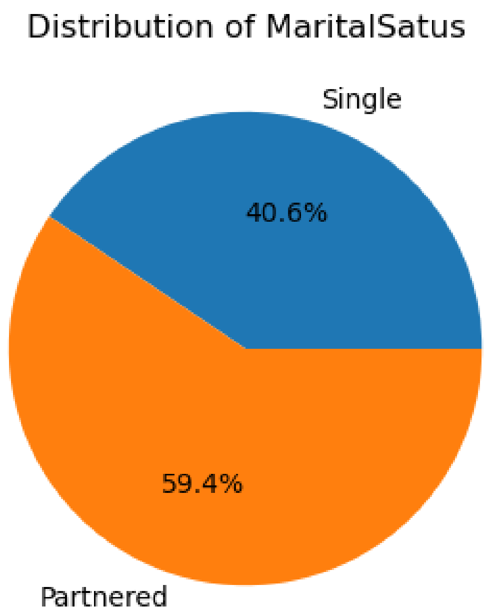
[93]:
```
plt.figure(figsize = (5,4))
plt.pie(list_Product, labels = data['Product'].unique(),autopct = "%2.1f%%")
plt.title("Distribution of Product")
plt.show()
```



[102]:
```
count = data[data['MaritalStatus'] == 'Single'].shape
x = count[0]
count = data[data['MaritalStatus'] == 'Partnered'].shape
y = count[0]
```

[103]:
```
list_MaritalStatus = [x,y]
```

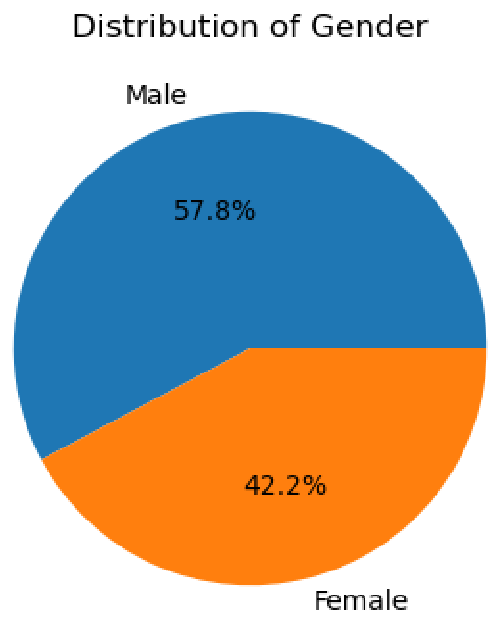
[115]:
```
plt.figure(figsize = (5,4))
plt.pie(list_MaritalStatus, labels = data['MaritalStatus'].unique(),autopct =␣
↪"%2.1f%%")
plt.title("Distribution of MaritalSatus")
plt.show()
```



[ ]:
```
count = data[data['Gender'] == 'Male'].shape
x = count[0]
count = data[data['Gender'] == 'Female'].shape
y = count[0]
```

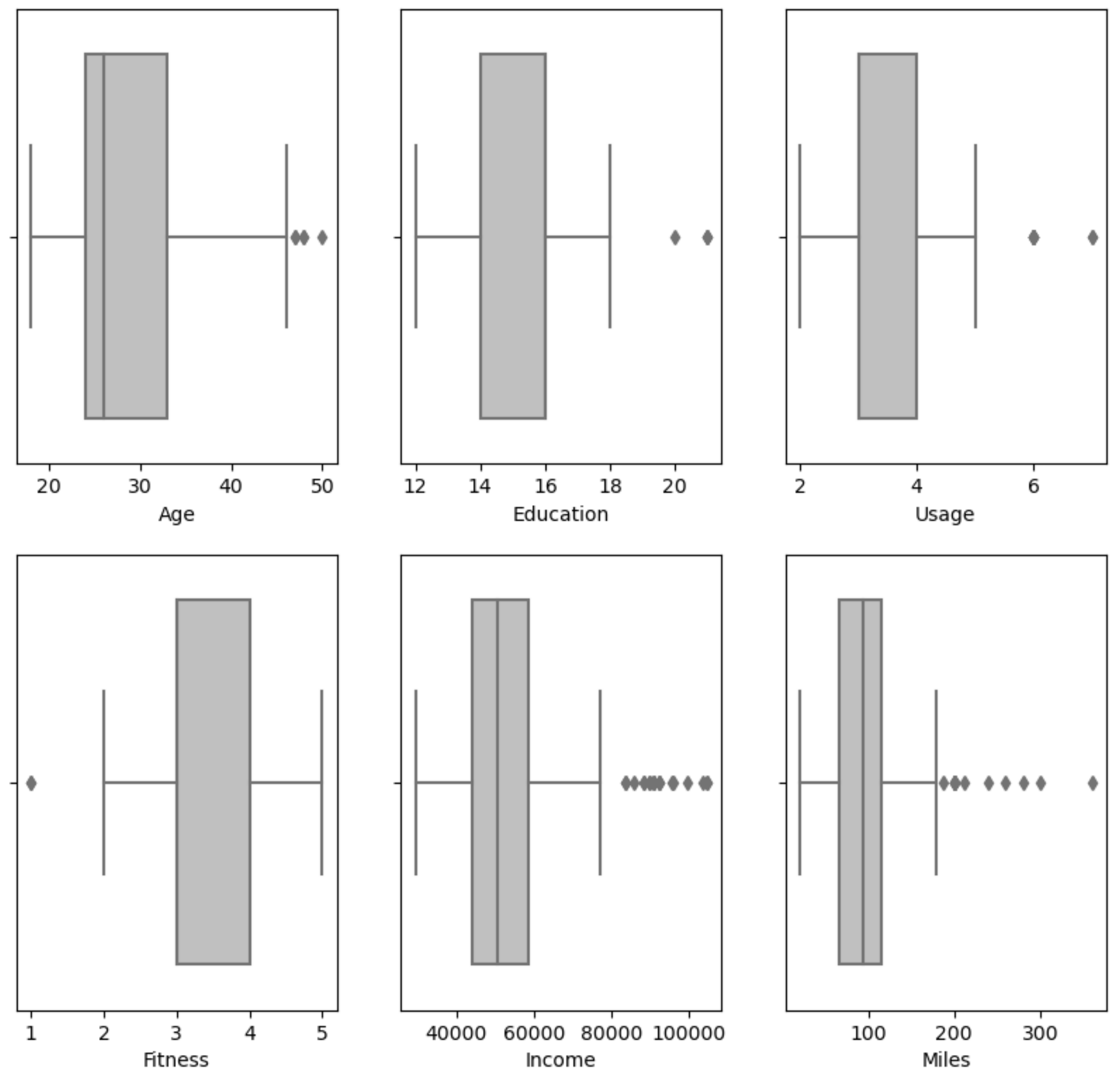
[ ]:
```
list_Gender = [x,y]
```

[116]:
```
plt.figure(figsize = (5,4))
plt.pie(list_Gender, labels = data['Gender'].unique(),autopct = "%2.1f%%")
plt.title("Distribution of Gender")
plt.show()
```



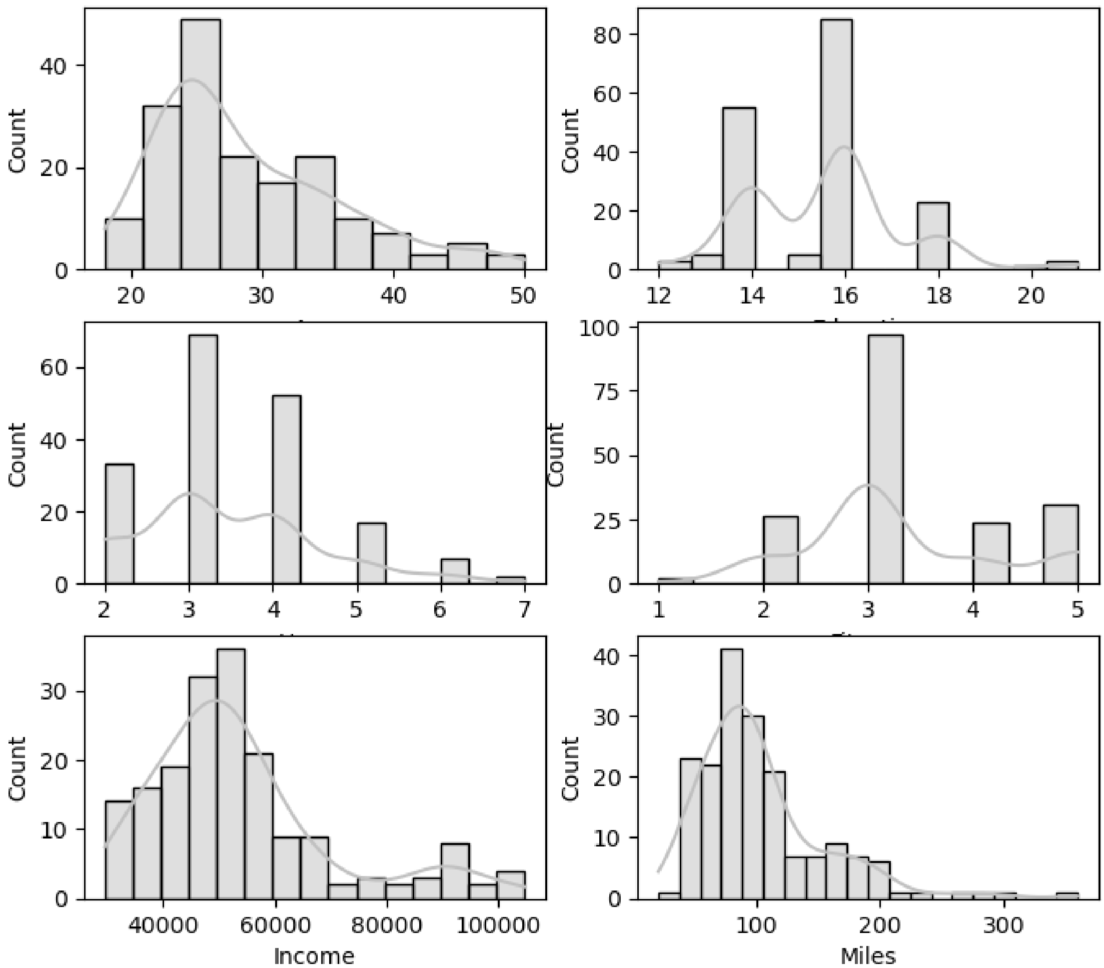

```
[ ]:
```

```
[56]: #Detecting Outliers
fig, axis = plt.subplots(nrows=2, ncols=3, figsize=(10, 8))
fig.subplots_adjust(top=1.0)
sns.boxplot(data=data, x="Age", ax=axis[0,0], color = 'silver')
sns.boxplot(data=data, x="Education", ax=axis[0,1],color = 'silver')
sns.boxplot(data=data, x="Usage",  ax=axis[0,2], color = 'silver')
sns.boxplot(data=data, x="Fitness", ax=axis[1,0], color = 'silver')
sns.boxplot(data=data, x="Income", ax=axis[1,1], color = 'silver')
sns.boxplot(data=data, x="Miles", ax=axis[1,2], color = 'silver')
plt.show()
```



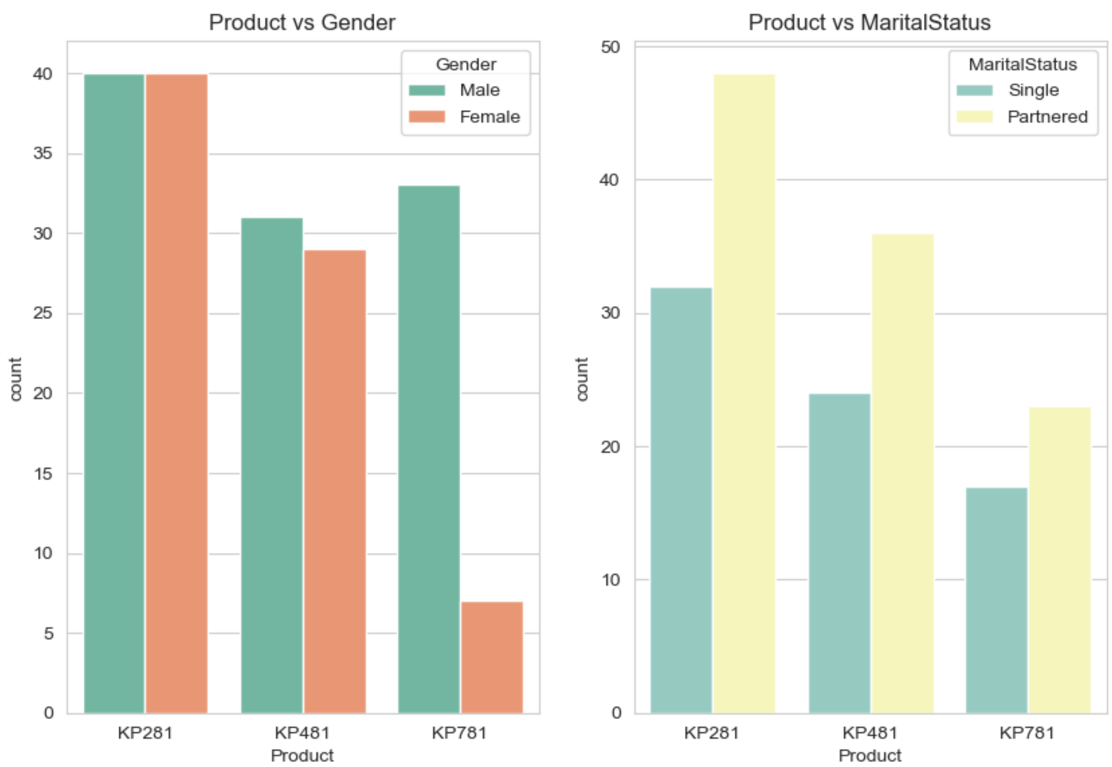
[73]:

```
#Distrubution of Data
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(8, 5))
fig.subplots_adjust(top=1.2)
sns.histplot(data=data, x="Age", kde=True, ax=axis[0,0], color = 'silver')
sns.histplot(data=data, x="Education", kde=True, ax=axis[0,1],color = 'silver')
sns.histplot(data=data, x="Usage", kde=True, ax=axis[1,0],color = 'silver')
sns.histplot(data=data, x="Fitness", kde=True, ax=axis[1,1],color = 'silver')
sns.histplot(data=data, x="Income", kde=True, ax=axis[2,0],color = 'silver')
sns.histplot(data=data, x="Miles", kde=True, ax=axis[2,1],color = 'silver')
plt.show()
```



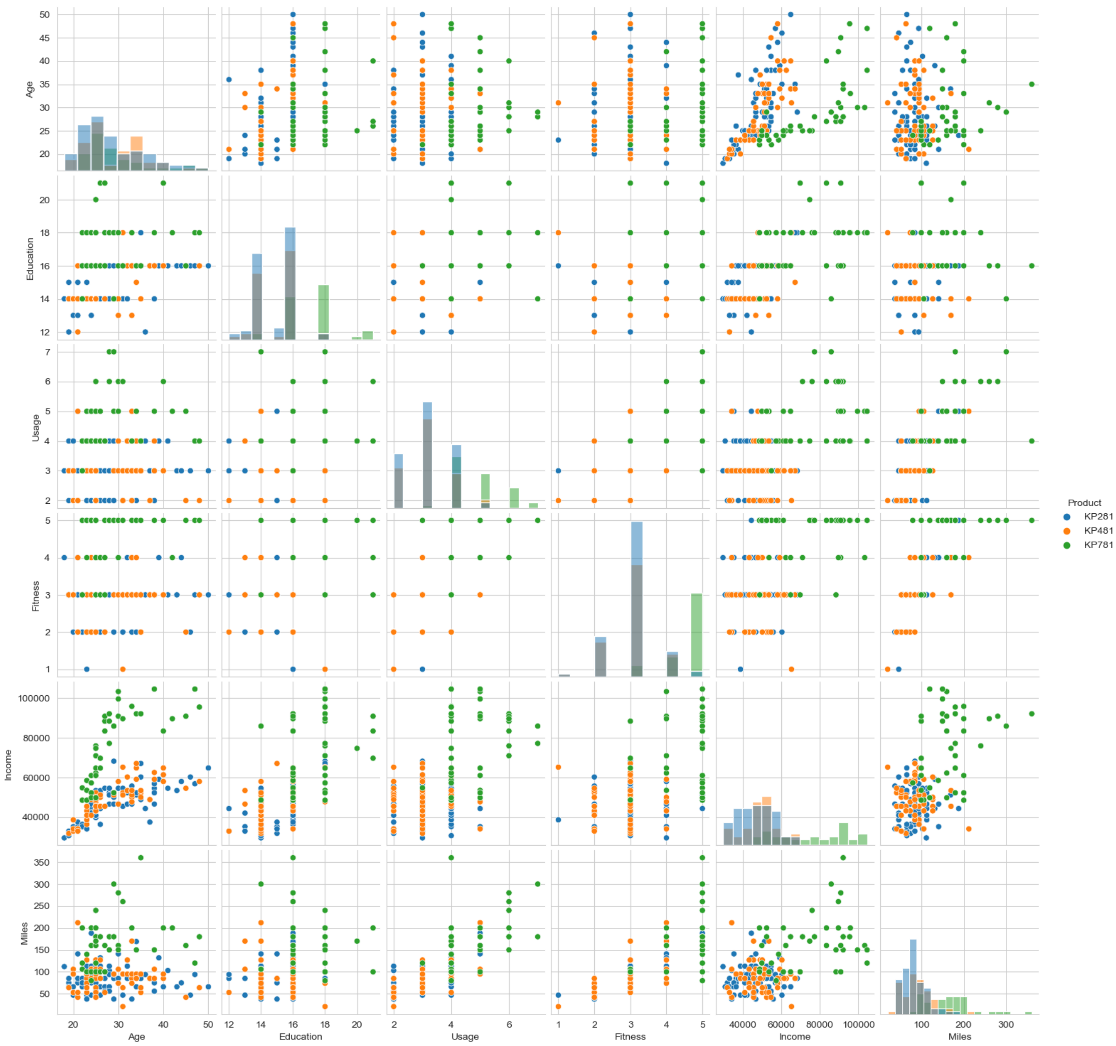
Finding Product Vs Gender and Product vs MaritalStatus

[247]:
```
# Graphical represtion
sns.set_style(style='whitegrid')
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(10, 6.5))
sns.countplot(data=data, x='Product', hue='Gender',
palette='Set2', ax=axs[0])
sns.countplot(data= data, x='Product', hue='MaritalStatus', palette='Set3', ax=axs[1])
axs[0].set_title("Product vs Gender")
axs[1].set_title("Product vs MaritalStatus")
plt.show()
```



[222]:
```
#correlation using Pairplot
sns.pairplot(data, kind='scatter', diag_kind='hist', hue='Product')
plt.show()
```

```
C:\Users\Rajkattari\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118:
UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



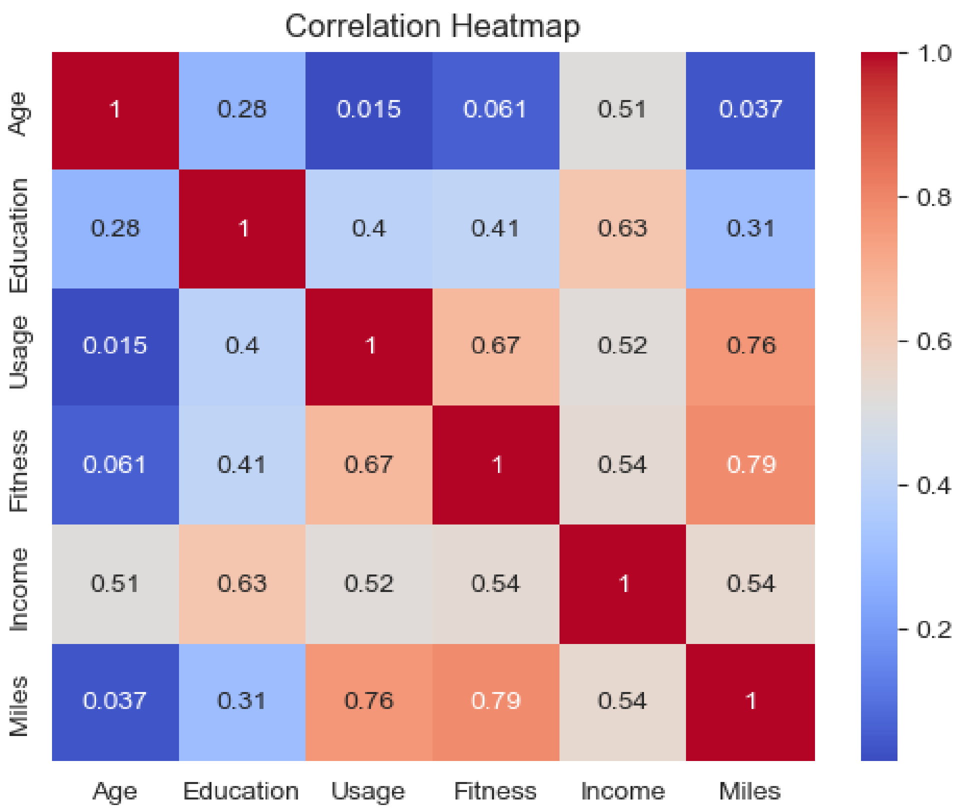
[254]:
```
#creation of corelation martix
df = data[['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']]
df_corr = df.corr()
```

[255]:
```
df_corr
```

[255]:

	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

[256]:
```python
# ploting heatmap
sns.heatmap(df_corr, annot = True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



[153]:
```python
#crosstab Product vs Gender
cross_gen = pd.crosstab(data['Gender'], data['Product'], margins = True)
cross_gen
```

[153]:
```
Product  KP281  KP481  KP781  All
Gender
Female      40     29      7   76
Male        40     31     33  104
All         80     60     40  180
```

[143]:
```python
nor_gen = pd.crosstab(data['Gender'], data['Product'], margins = True,␣
 ↪normalize = True)
nor_gen
```

[143]:
```
Product     KP281     KP481     KP781       All
Gender
```

```
Female   0.222222  0.161111  0.038889  0.422222
Male     0.222222  0.172222  0.183333  0.577778
All      0.444444  0.333333  0.222222  1.000000
```

[ ]:

[161]:
```
#crosstab Product vs MaritalStatus
pd.crosstab(data['MaritalStatus'], data['Product'], margins = True)
```

[161]:
```
Product        KP281  KP481  KP781  All
MaritalStatus
Partnered         48     36     23  107
Single            32     24     17   73
All               80     60     40  180
```

[ ]:

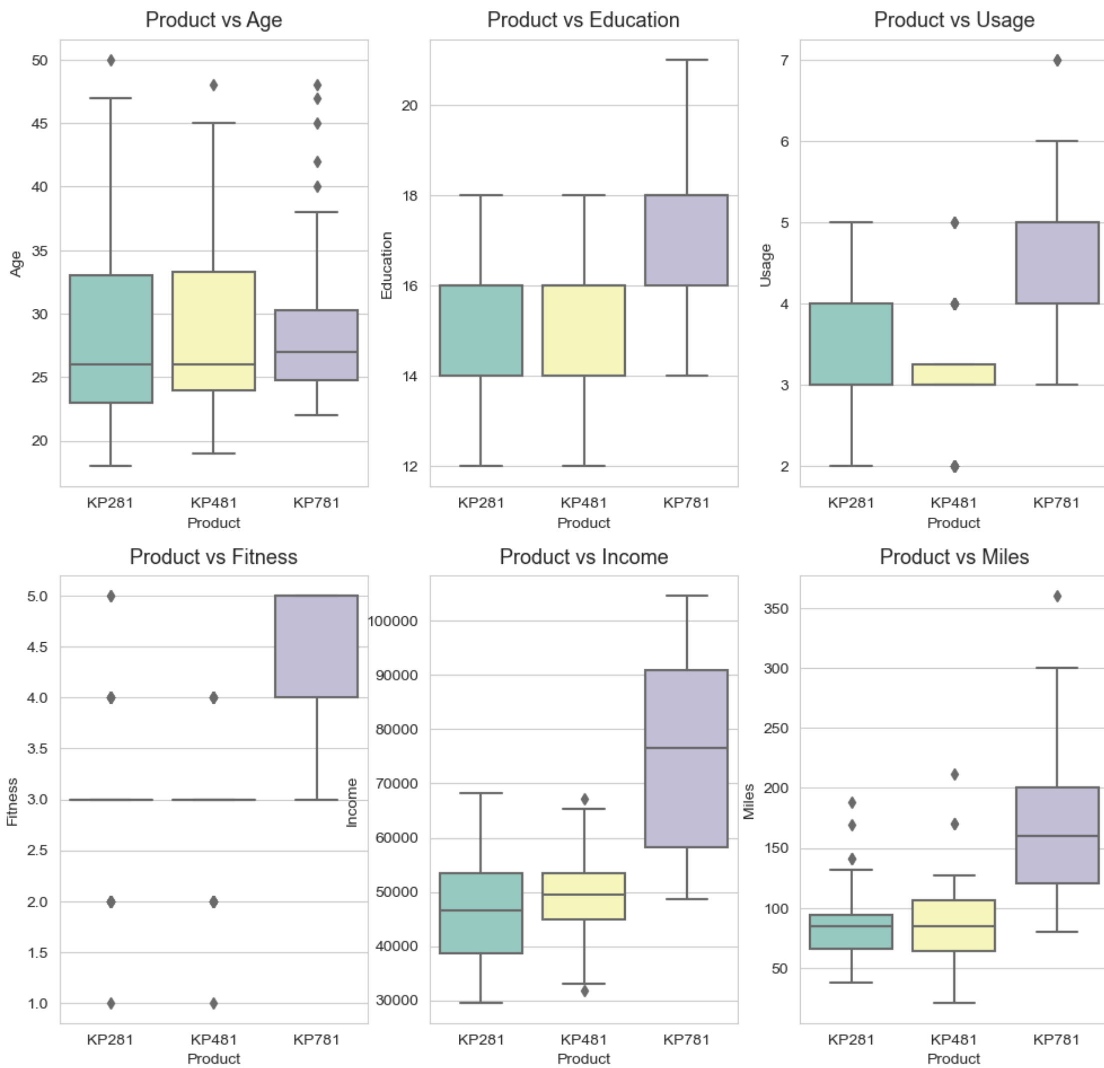
[183]:
```
#Marginal Probability product vs Gender
print('KP281 :' + str(nor_gen['KP281'].iloc[2])[:4])
print('KP481 :' + str(nor_gen['KP481'].iloc[2])[:4])
print('KP781 :' + str(nor_gen['KP781'].iloc[2])[:4])
```

```
KP281 :0.44
KP481 :0.33
KP781 :0.22
```

Finding Relationship between Product vs Age, Education, Usage, Fitness, Income, Miles

[248]:
```
#Barplot repression
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(12, 8))
fig.subplots_adjust(top=1.2)
sns.boxplot(data=data, x='Product', y='Age', ax=axs[0,0], palette='Set3')
axs[0,0].set_title(f"Product vs Age",pad=8, fontsize=13)
sns.boxplot(data=data, x='Product', y='Education', ax=axs[0,1], palette='Set3')
axs[0,1].set_title(f"Product vs Education",pad=8, fontsize=13)
sns.boxplot(data=data, x='Product', y='Usage', ax=axs[0,2], palette='Set3')
axs[0,2].set_title(f"Product vs Usage",pad=8, fontsize=13)
sns.boxplot(data=data, x='Product', y='Fitness', ax=axs[1,0], palette='Set3')
axs[1,0].set_title(f"Product vs Fitness",pad=8, fontsize=13)
sns.boxplot(data=data, x='Product', y='Income', ax=axs[1,1], palette='Set3')
axs[1,1].set_title(f"Product vs Income",pad=8, fontsize=13)
sns.boxplot(data=data, x='Product', y='Miles', ax=axs[1,2], palette='Set3')
axs[1,2].set_title(f"Product vs Miles",pad=8, fontsize=13)
```

[248]: Text(0.5, 1.0, 'Product vs Miles')



[ ]:

[172]:
```
#Crosstab Gender vs MaritalStatus vs Product
p = pd.crosstab(data['MaritalStatus'], [data['Gender'], data['Product']],␣
 ↪margins=True)
p
```

[172]:

Gender	Female			Male			All
Product	KP281	KP481	KP781	KP281	KP481	KP781	
MaritalStatus							
Partnered	27	15	4	21	21	19	107
Single	13	14	3	19	10	14	73
All	40	29	7	40	31	33	180

[176]:
```python
# Assgining values to the variables from the crosstab
f_p_281 = p['Female']['KP281'].iloc[0]
f_s_281 = p['Female']['KP281'].iloc[1]
total_f_281 = p['Female']['KP281'].iloc[2]

f_p_481 = p['Female']['KP481'].iloc[0]
f_s_481 = p['Female']['KP481'].iloc[1]
total_f_481 = p['Female']['KP481'].iloc[2]

f_p_781 = p['Female']['KP781'].iloc[0]
f_s_781 = p['Female']['KP781'].iloc[1]
total_f_781 = p['Female']['KP781'].iloc[2]

M_p_281 = p['Male']['KP281'].iloc[0]
M_s_281 = p['Male']['KP281'].iloc[1]
total_M_281 = p['Male']['KP281'].iloc[2]

M_p_481 = p['Male']['KP481'].iloc[0]
M_s_481 = p['Male']['KP481'].iloc[1]
total_M_481 = p['Male']['KP481'].iloc[2]

M_p_781 = p['Male']['KP781'].iloc[0]
M_s_781 = p['Male']['KP781'].iloc[1]
total_M_781 = p['Male']['KP781'].iloc[2]

total_product_p = p['All'].iloc[0]
total_product_s = p['All'].iloc[1]
total_products = p['All'].iloc[2]
```

Calculate Probability Female who are Partnered

[237]:
```python
print(f"KP281_Female/Partnered : {f_p_281/total_product_p:.2f}")
print(f"KP481_Female/Partnered : {f_p_481/total_product_p:.2f}")
print(f"KP781_Female/Partnered : {f_p_781/total_product_p:.2f}")
```

```
KP281_Female/Partnered : 0.25
KP481_Female/Partnered : 0.14
KP781_Female/Partnered : 0.04
```

Calculate Probability Male who are Partnered

[236]:
```python
print(f"KP281_Male/Partnered : {M_p_281/total_product_p:.2f}")
print(f"KP481_Male/Partnered : {M_p_481/total_product_p :.2f}")
print(f"KP781_Male/Partnered : {M_p_781/total_product_p :.2f}")
```

```
KP281_Male/Partnered : 0.20
KP481_Male/Partnered : 0.20
KP781_Male/Partnered : 0.18
```

Calculate Probability Female who are Single

[234]:
```
print(f"KP281_Female/Single : {f_s_281/total_product_s :.2f}")
print(f"KP481_Female/Single : {f_s_481/total_product_s :.2f}")
print(f"KP781_Female/Single : {f_s_781/total_product_s :.2f}")
```

```
KP281_Female/Single : 0.18
KP481_Female/Single : 0.19
KP781_Female/Single : 0.04
```

Calculate Probability male who are Single

[231]:
```
print(f"KP281_Male/Single : {(M_s_281/total_product_s):.2f}")
print(f"KP481_Male/Single : {(M_s_481/total_product_s):.2f}")
print(f"KP781_Male/Single : {M_s_781/total_product_s :.2f}")
```

```
KP281_Male/Single : 0.26
KP481_Male/Single : 0.14
KP781_Male/Single : 0.19
```

[ ]:
```

```

Insight:

- Age, Education and Usage are having very few outliers. While Income and Miles are having more outliers.
- KP281 is the most frequent product.
- There are more Males in the data than Females.
- More Partnered persons are there in the data.
- From the corelation we can observe there is a postive corelation between KP781 and income, miles

Product

- 44.44% of the customers have purchased KP281 product.
- 33.33% of the customers have purchased KP481 product.
- 22.22% of the customers have purchased KP781 product.

Gender

- 57.78% of the customers are Male.

MaritalStatus

- 59.44% of the customers are Partnered.

Recommendation:

- The Product KP281 & KP481 both Male and Female customer are almost same but for the KP781 Female customer are very low to implement targeted strategies such as offering special promotions and trials exclusively designed for the female customers.
- We can make TV ads to attract more customer who's age is greater than 30 as most of the present customer are in the age 20-30
- It's important to offer the KP281 and KP481 Treadmill at an affordable price point. Additionally, consider EMI payments, this can make more accessible to customers with varying budget

[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```