In [4]:

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

In [6]:

```
#data_aero="aerofit.treademill.csv"
df=pd.read_csv('aerofit_treademill.csv')
df
```

Out[6]:

| | 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 [13]:

```
print(f"Number of rows: {df.shape[0]} \nNumber of columns: {df.shape[1]}")
```

```
Number of rows: 180
Number of columns: 9
```

In [14]:

```
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
```

In [15]:

```
df.describe(include="all")
```

Out[15]:

|  | Product | Age | Gender | Education | MaritalStatus | Usage | Fitness | In |
|---|---|---|---|---|---|---|---|---|
| **count** | 180 | 180.000000 | 180 | 180.000000 | 180 | 180.000000 | 180.000000 | 180.00 |
| **unique** | 3 | NaN | 2 | NaN | 2 | NaN | NaN |  |
| **top** | KP281 | NaN | Male | NaN | Partnered | NaN | NaN |  |
| **freq** | 80 | NaN | 104 | NaN | 107 | NaN | NaN |  |
| **mean** | NaN | 28.788889 | NaN | 15.572222 | NaN | 3.455556 | 3.311111 | 53719.57 |
| **std** | NaN | 6.943498 | NaN | 1.617055 | NaN | 1.084797 | 0.958869 | 16506.68 |
| **min** | NaN | 18.000000 | NaN | 12.000000 | NaN | 2.000000 | 1.000000 | 29562.00 |
| **25%** | NaN | 24.000000 | NaN | 14.000000 | NaN | 3.000000 | 3.000000 | 44058.75 |
| **50%** | NaN | 26.000000 | NaN | 16.000000 | NaN | 3.000000 | 3.000000 | 50596.50 |
| **75%** | NaN | 33.000000 | NaN | 16.000000 | NaN | 4.000000 | 4.000000 | 58668.00 |
| **max** | NaN | 50.000000 | NaN | 21.000000 | NaN | 7.000000 | 5.000000 | 104581.00 |

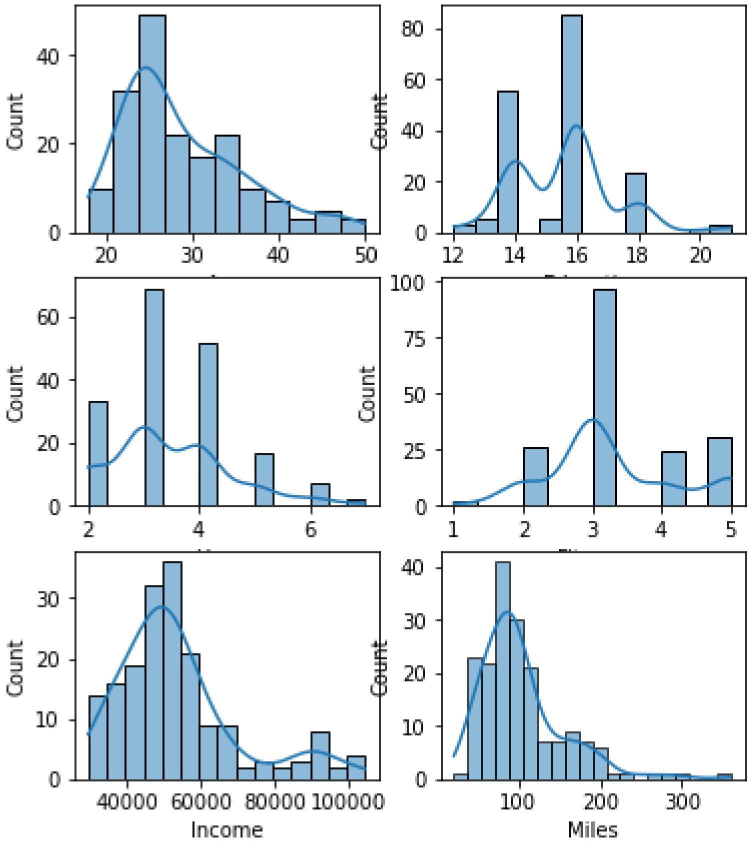
In [17]:

```
print("columns with missing value")
print(df.isnull().any())
```

```
columns with missing value
Product          False
Age              False
Gender           False
Education        False
MaritalStatus    False
Usage            False
Fitness          False
Income           False
Miles            False
dtype: bool
```

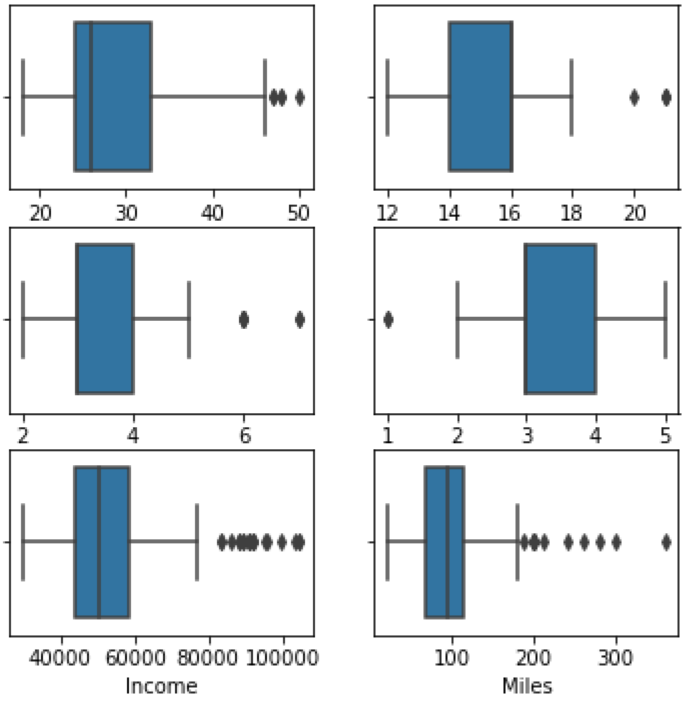
In [23]:

```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(6, 5))
fig.subplots_adjust(top=1.2)
sns.histplot(data=df, x="Age", kde=True, ax=axis[0, 0])
sns.histplot(data=df, x="Education", kde=True, ax=axis[0, 1])
sns.histplot(data=df, x="Usage", kde=True, ax=axis[1, 0])
sns.histplot(data=df, x="Fitness", kde=True, ax=axis[1, 1])
sns.histplot(data=df, x="Income", kde=True, ax=axis[2, 0])
sns.histplot(data=df, x="Miles", kde=True, ax=axis[2, 1])
plt.show()
```



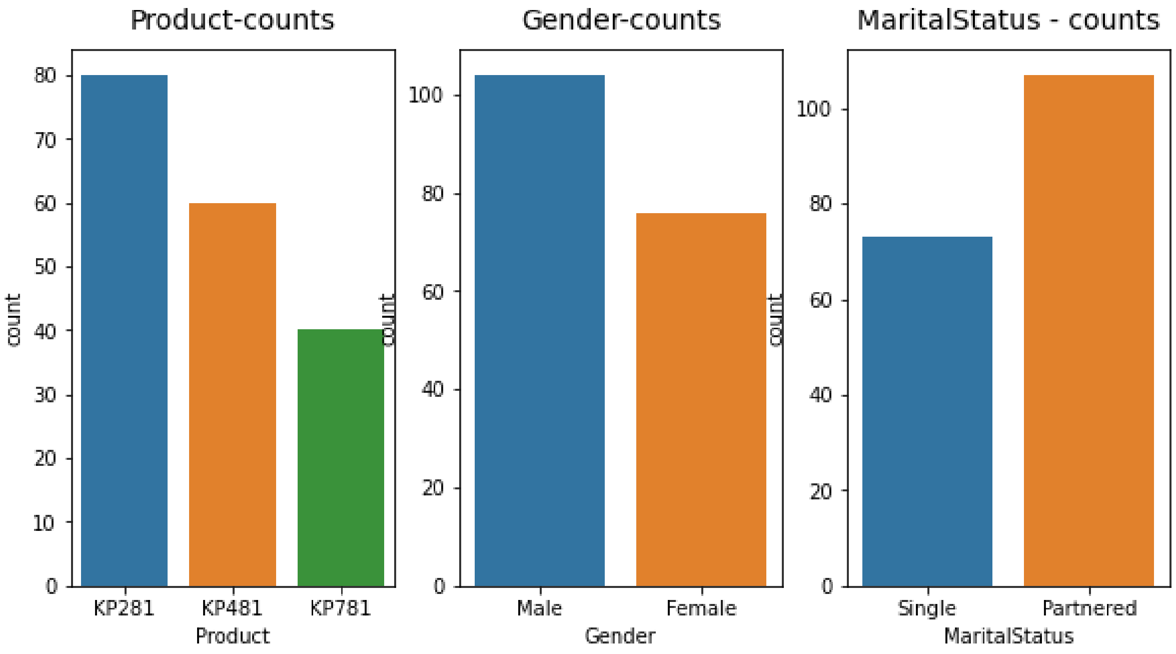
In [26]:

```
fig, axis = plt.subplots (nrows=3, ncols=2, figsize=(6, 5))
fig.subplots_adjust(top=1.0)
sns.boxplot(data=df, x="Age", orient='h', ax=axis[0,0])
sns.boxplot(data=df, x="Education", orient='h', ax=axis [0,1])
sns.boxplot(data=df, x="Usage", orient='h', ax=axis[1,0])
sns.boxplot (data=df, x="Fitness", orient='h', ax=axis[1,1])
sns.boxplot(data=df, x="Income", orient='h', ax=axis [2,0])
sns.boxplot (data=df, x="Miles", orient='h', ax=axis [2,1])
plt.show()
```



In [27]:

```
fig, axs = plt.subplots (nrows=1, ncols=3, figsize=(10,5))
sns.countplot(data=df, x='Product', ax=axs [0])
sns.countplot(data=df, x='Gender', ax=axs[1])
sns.countplot(data=df, x='MaritalStatus', ax=axs [2])
axs[0].set_title("Product-counts", pad=10, fontsize=14)
axs[1].set_title("Gender-counts", pad=10, fontsize=14)
axs[2].set_title("MaritalStatus - counts", pad=10, fontsize=14)
plt.show()
```



In [28]:

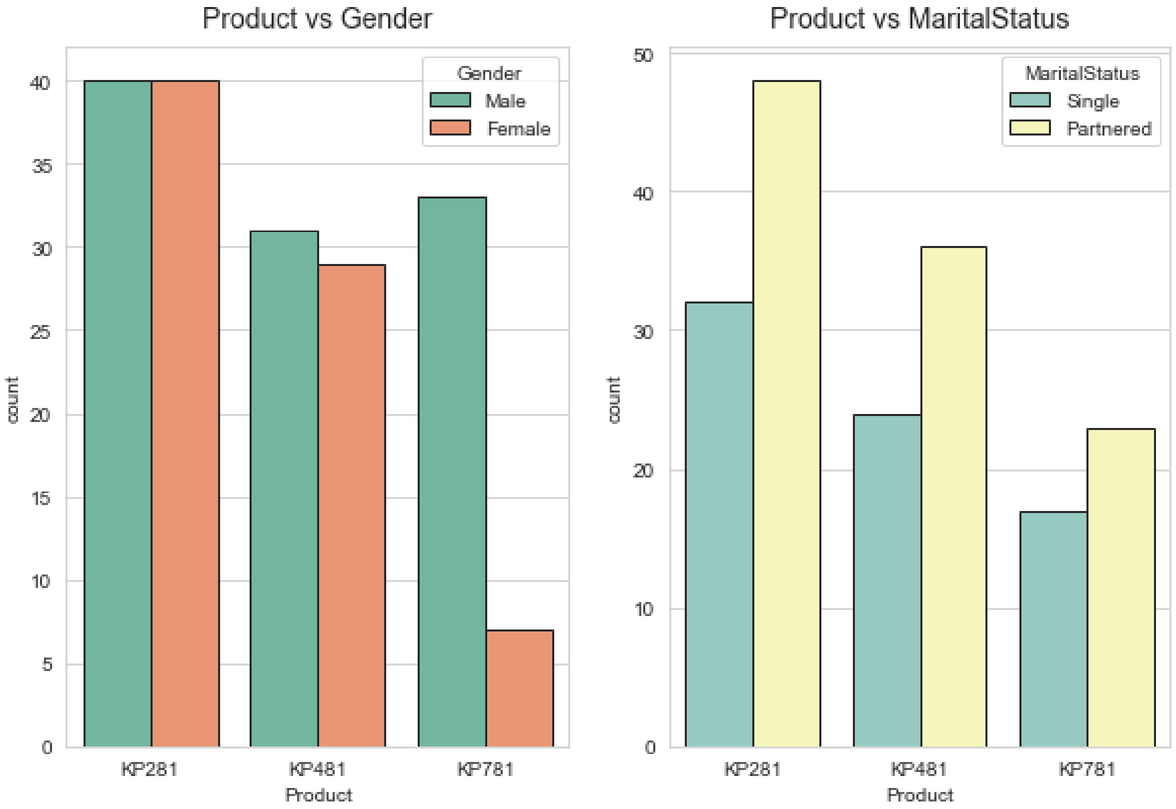
```
dfl = df [['Product', 'Gender', 'MaritalStatus']].melt()
dfl.groupby(['variable', 'value']) [['value']].count() / len (df)
```

Out[28]:

| | | value |
|---|---|---|
| **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 |

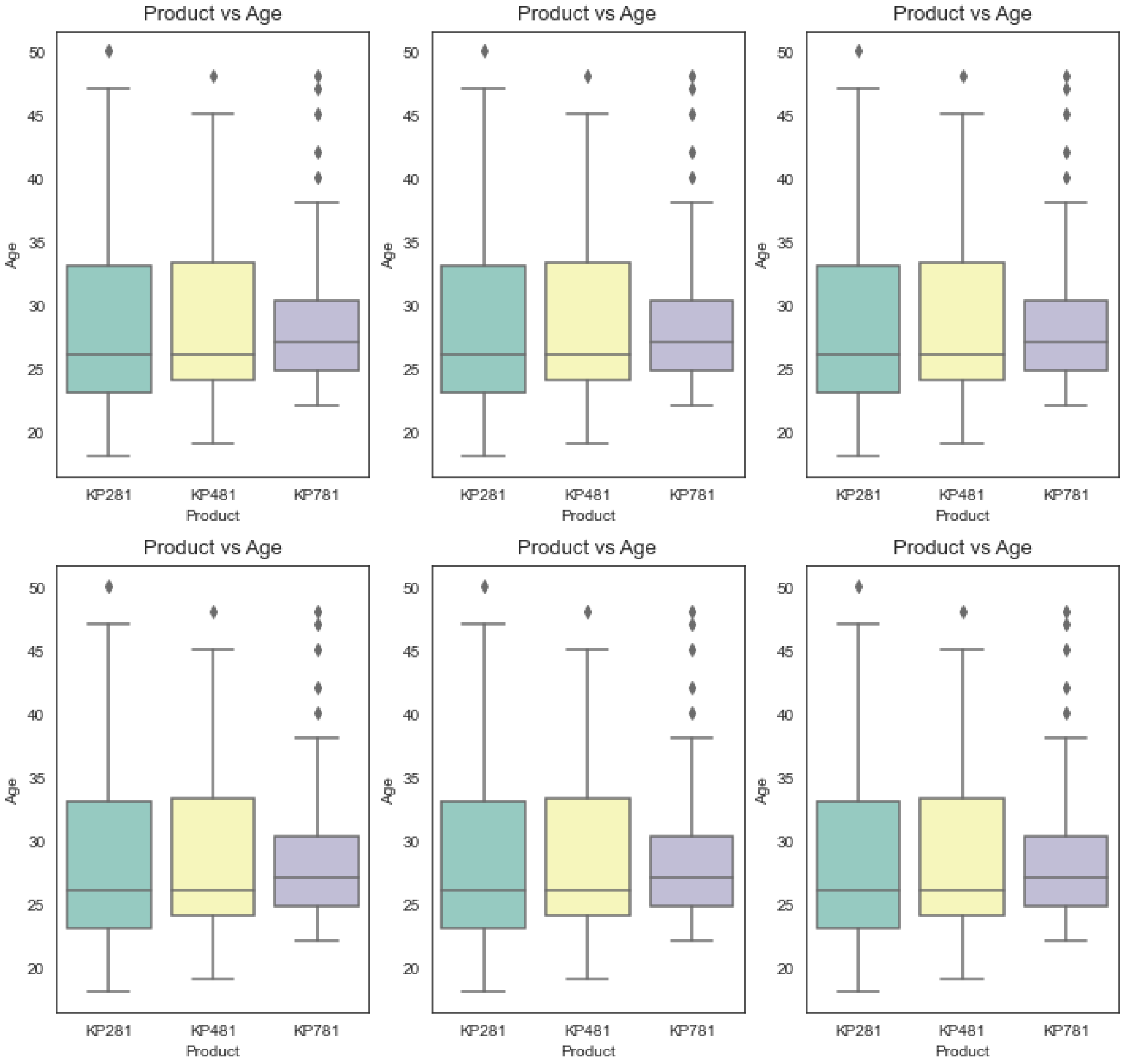
In [30]:

```
sns.set_style(style='whitegrid')
fig, axs = plt.subplots (nrows=1, ncols=2, figsize=(10, 6.5))
sns.countplot(data=df, x='Product', hue='Gender', edgecolor="0.15", palette='Set2'
sns.countplot(data=df, x='Product', hue='MaritalStatus',
edgecolor="0.15", palette='Set3', ax=axs[1])
axs[0].set_title("Product vs Gender", pad=10, fontsize=14)
axs[1].set_title("Product vs MaritalStatus", pad=10, fontsize=14)
plt.show()
```

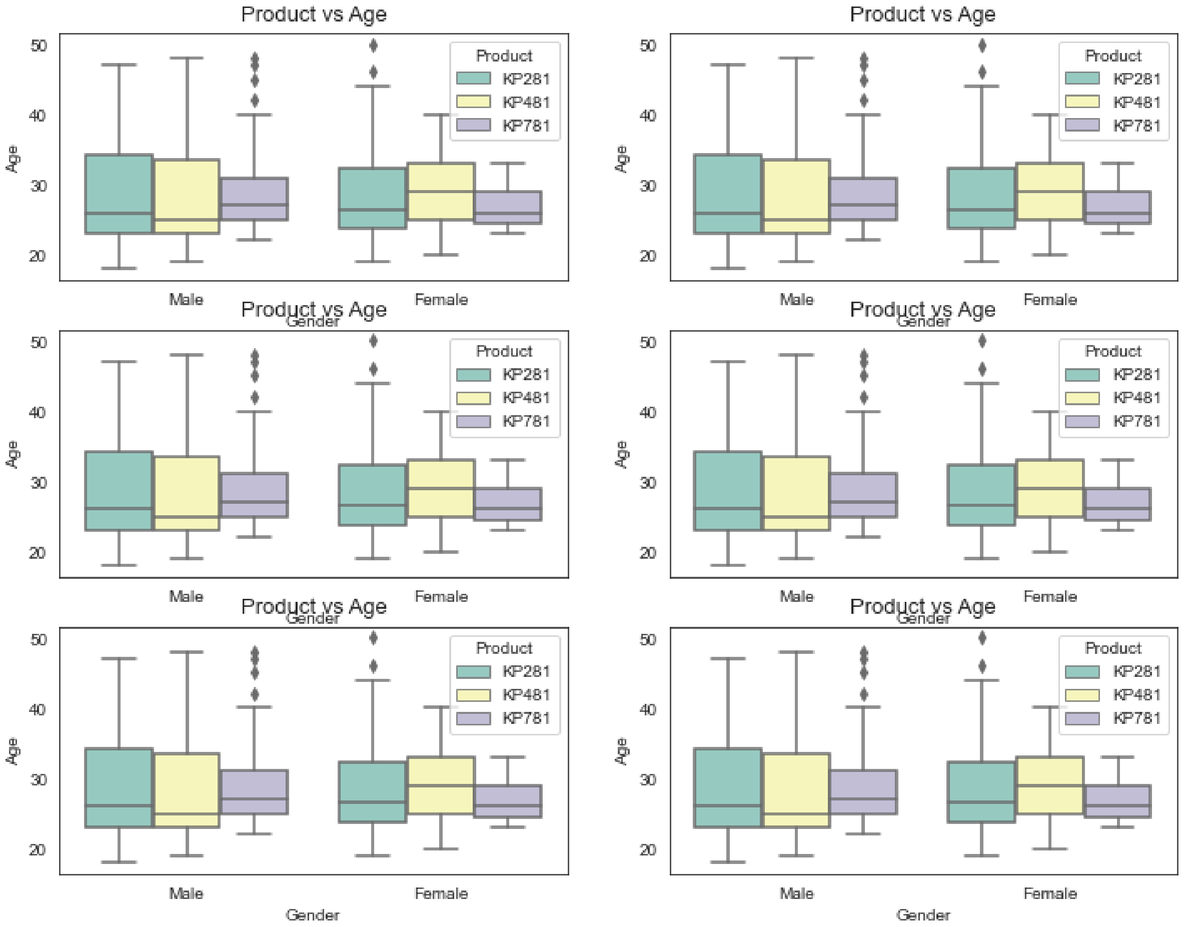


In [34]:
```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(12, 8))
fig.subplots_adjust(top=1.2)
count = 0

for i in range (2):
    for j in range (3):
        sns.boxplot(data=df, x='Product', y=attrs[count],
ax=axs[i,j], palette='Set3')
        axs[i, j].set_title(f"Product vs {attrs[count]}",
pad=8, fontsize=13)
count += 1
```



In [37]:
```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income' 'Miles']
sns.set_style ("white")
fig, axs = plt.subplots (nrows=3, ncols=2, figsize=(12, 8))
fig.subplots_adjust(top=1)
count = 0
for i in range (3):
    for j in range (2):
        sns.boxplot(data=df, x='Gender', y=attrs[count], hue='Product',
ax=axs[i, j], palette='Set3')
        axs[i, j].set_title(f"Product vs {attrs[count]}", pad=8,
fontsize=13)
count += 1
```



```
In [38]: df['Product'].value_counts(normalize=True)
```

```
Out[38]: KP281    0.444444
         KP481    0.333333
         KP781    0.222222
         Name: Product, dtype: float64
```

```
In [42]: # Define df, df1, and dfl DataFrames if they are not defined elsewhere in your code

         def p_prod_given_gender(gender, print_marginal=False):
             if gender not in ["Female", "Male"]:
                 return "Invalid gender value."

             df1 = pd.crosstab(index=df['Gender'], columns=[df['Product']])
             p_781 = df1['KP781'][gender] / df1.loc[gender].sum()
             p_481 = df1['KP481'][gender] / df1.loc[gender].sum()
             p_281 = df1['KP281'][gender] / df1.loc[gender].sum()

             if print_marginal:
                 print(f"P (Male): {df1.loc['Male'].sum() / len(df):.2f}")
                 print(f"P (Female): {df1.loc['Female'].sum() / len(df):.2f}\n")

             print(f"P (KP781/{gender}): {p_781:.2f}")
             print(f"P (KP481/{gender}): {p_481:.2f}")
             print(f"P (KP281/{gender}): {p_281:.2f}\n")

         p_prod_given_gender('Male', True)
         p_prod_given_gender('Female')
```

```
P (Male): 0.58
P (Female): 0.42

P (KP781/Male): 0.32
P (KP481/Male): 0.30
P (KP281/Male): 0.38

P (KP781/Female): 0.09
P (KP481/Female): 0.38
P (KP281/Female): 0.53
```

In [ ]:

In [ ]: