1. Defining Problem Statement and Analysing basic metrics **(10 Points)**
    1. Observations on shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), statistical summary

Solution-> The market research team at AeroFit wants to identify the characteristics of the target audience for each type of treadmill offered by the company, to provide a better recommendation of the treadmills to the new customers. The team decides to investigate whether there are differences across the product with respect to customer characteristics.

1. Perform descriptive analytics **to create a customer profile** for each AeroFit treadmill product by developing appropriate tables and charts.
2. For each AeroFit treadmill product, construct **two-way contingency tables** and compute all **conditional and marginal probabilities** along with their insights/impact on the business.

```
# Product Purchased:  KP281, KP481, or KP781
```

```
# Age:  In years

# Gender: Male/Female

# Education:  In years

# MaritalStatus:  Single or partnered

# Usage:  The average number of times the customer plans to use the
treadmill each week.

# Income: Annual income (in $)

# Fitness:  Self-rated fitness on a 1-to-5 scale, where 1 is the poor
shape and 5 is the excellent shape.

# Miles:  The average number of miles the customer expects to walk/run
each week
```

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

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

2.Non-Graphical Analysis: Value counts and unique attributes **(10 Points)**

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

```
Product  Age  Gender  Education  MaritalStatus  Usage  Fitness  Income  Miles
KP281    18   Male    14         Single         3      4        29562   112      1
KP481    30   Female  13         Single         4      3        46617   106      1
         31   Female  16         Partnered      2      3        51165   64       1
                      18         Single         2      1        65220   21       1
              Male    16         Partnered      3      3        52302   95       1
                                                                                 ..
KP281    34   Female  16         Single         2      2        52302   66       1
              Male    16         Single         4      5        51165   169      1
         35   Female  16         Partnered      3      3        60261   94       1
                      18         Single         3      3        67083   85       1
KP781    48   Male    18         Partnered      4      5        95508   180      1
Length: 180, dtype: int64
```

3.Visual Analysis - Univariate & Bivariate **(30 Points)**

Observations:

- **There are no missing values in the data.**
- **There are 3 unique products in the dataset.**
- **KP281 is the most frequent product.**
- **Minimum & Maximum age of the person is 18 & 50, mean is 28.79 and 75% of persons have age less than or equal to 33.**
- **Most of the people are having 16 years of education i.e. 75% of persons are having education <= 16 years.**
- **Out of 180 data points, 104's gender is Male and rest are the female.**
- **Standard deviation for Income & Miles is very high. These variables might have the outliers in it.**

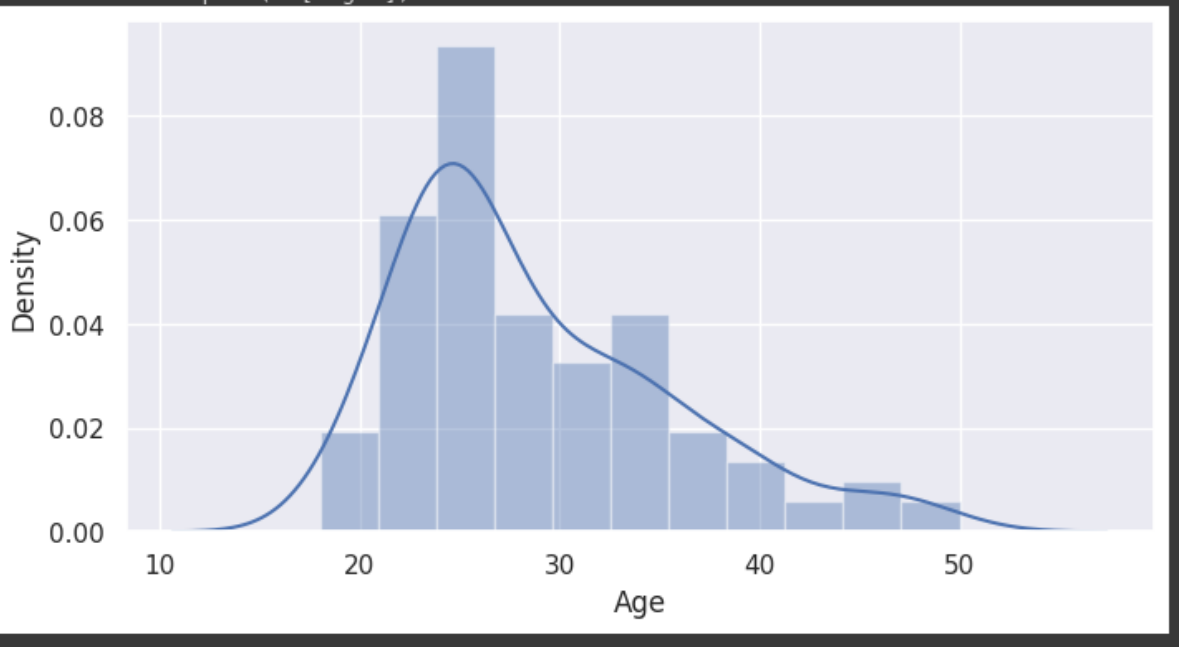
Product wise unique variant

```
df['Product'].unique()
```

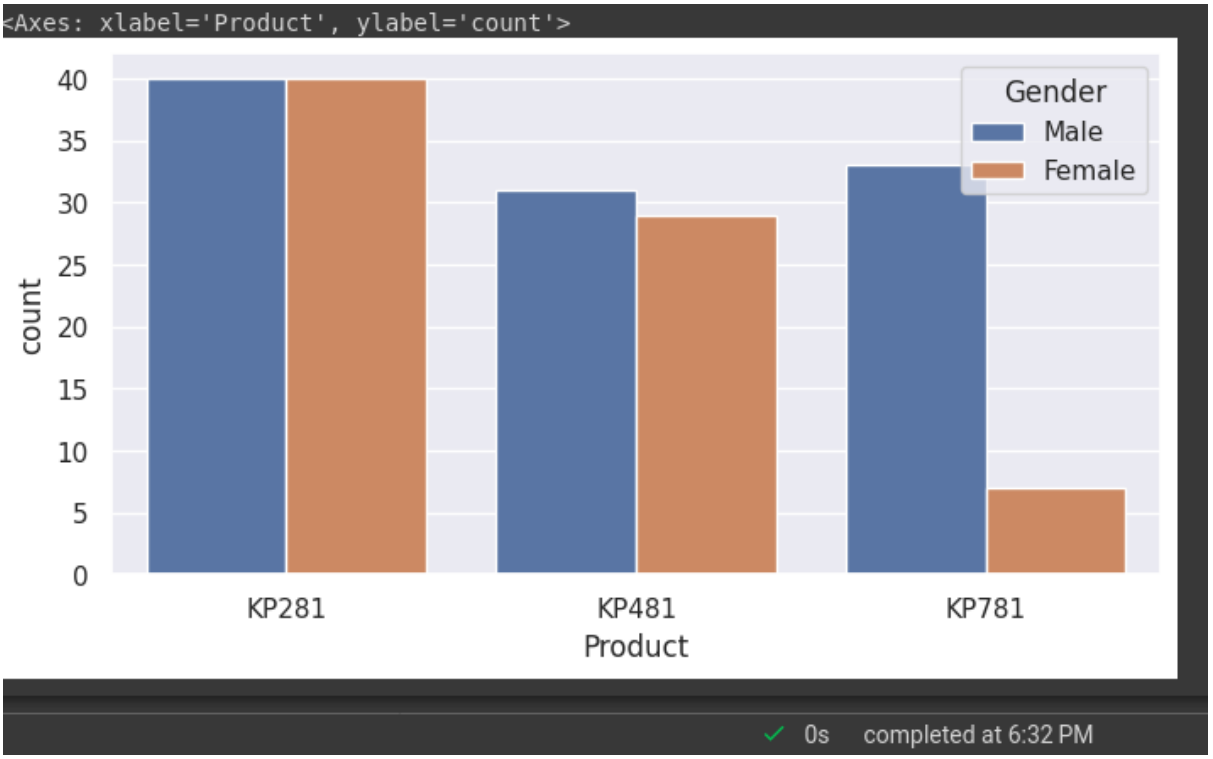
```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

1:For continuous variable(s): Distplot, countplot, histogram for univariate analysis (10 Points)

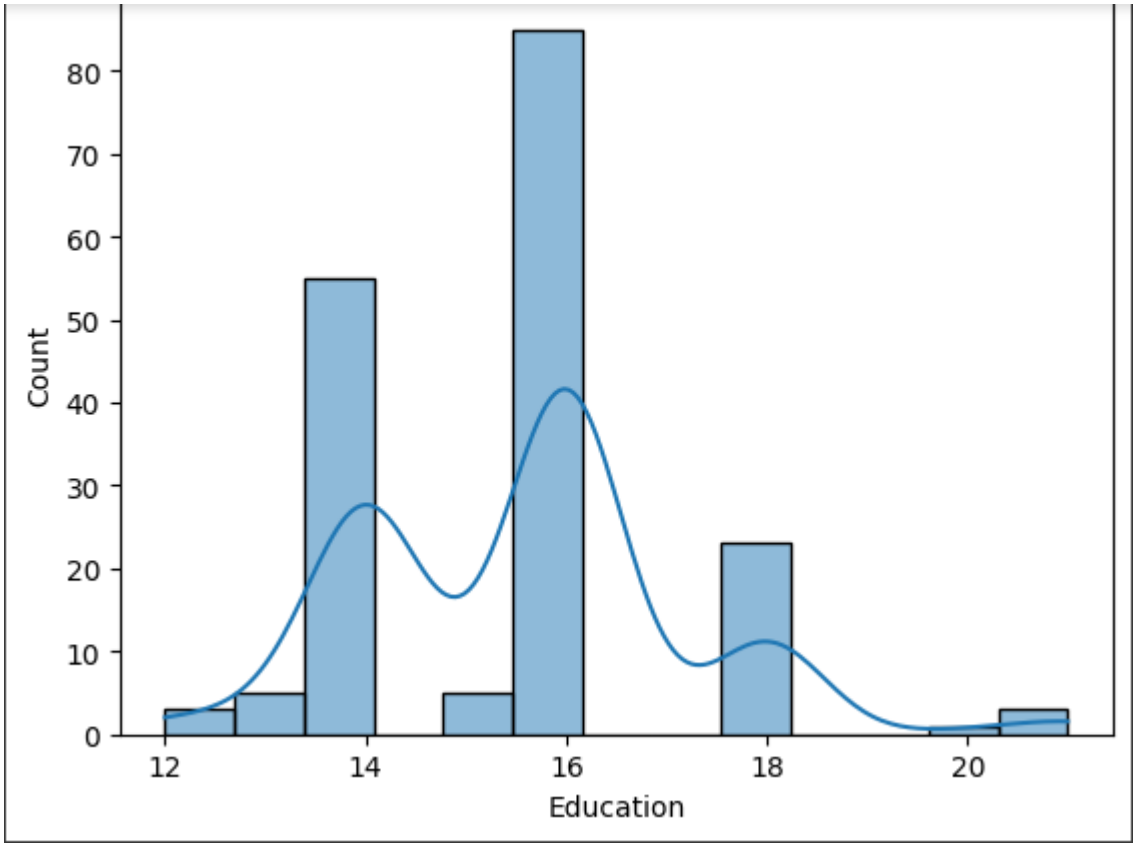
```
ax = sns.distplot(df['Age'])
```

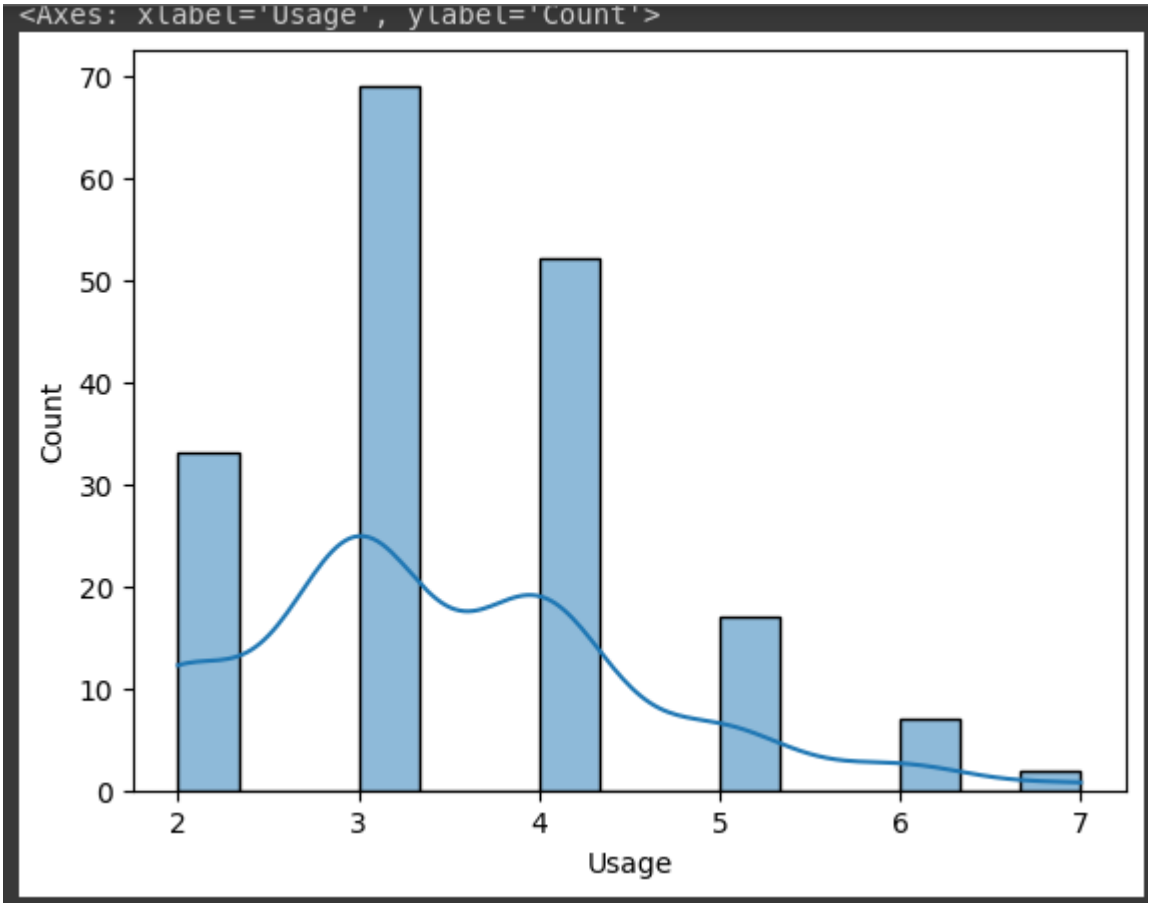


Countplot for products and Gender



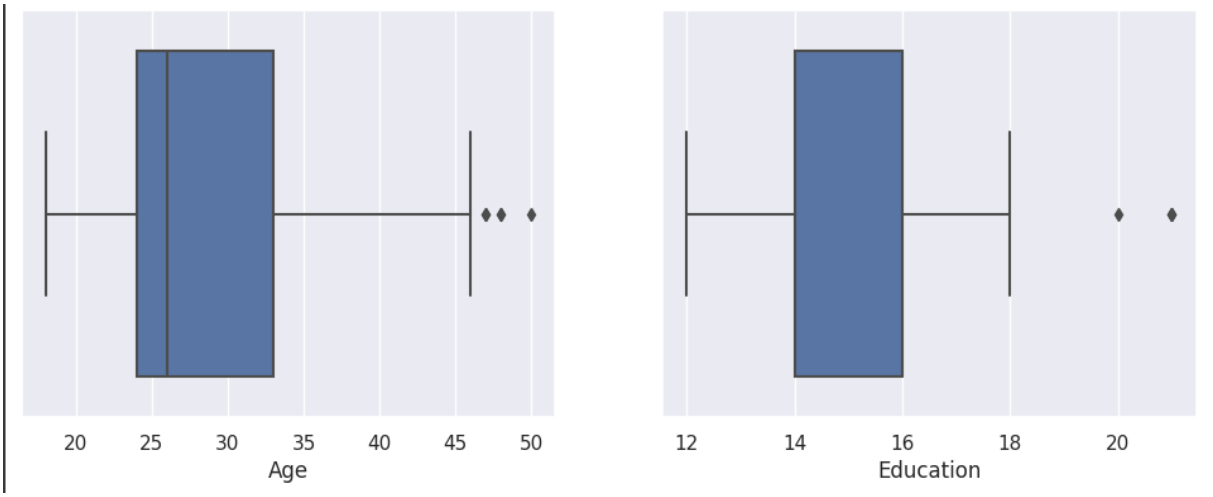
Education hist plot

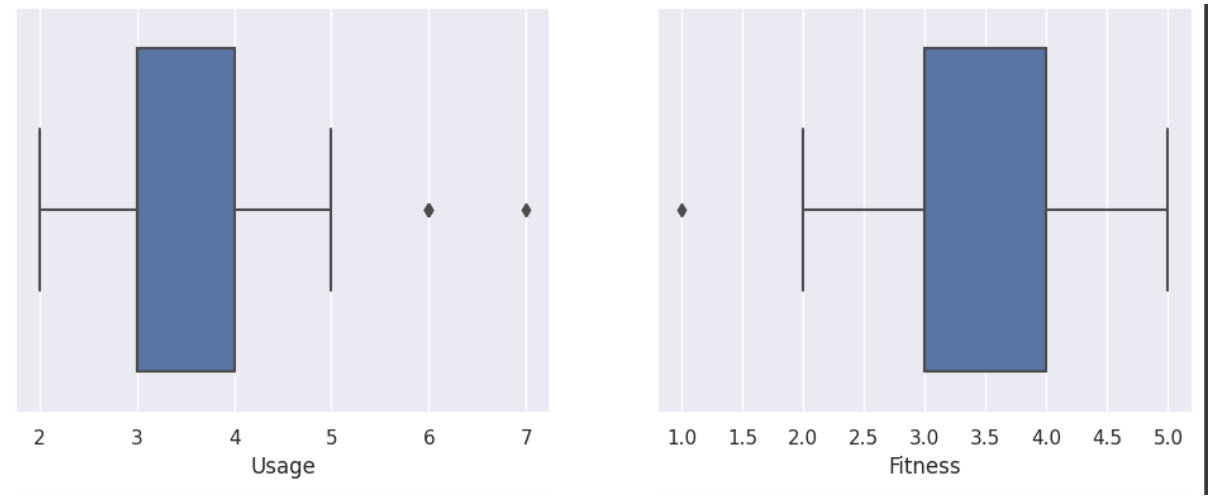


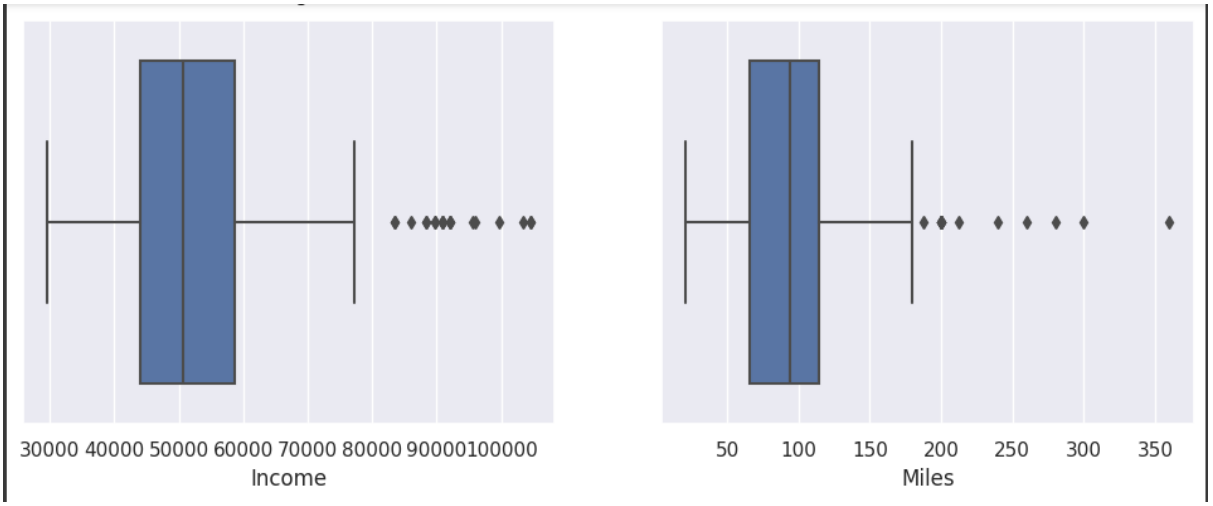


Outliers detection using BoxPlots

```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))

fig.subplots_adjust(top=1.2)



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()
```







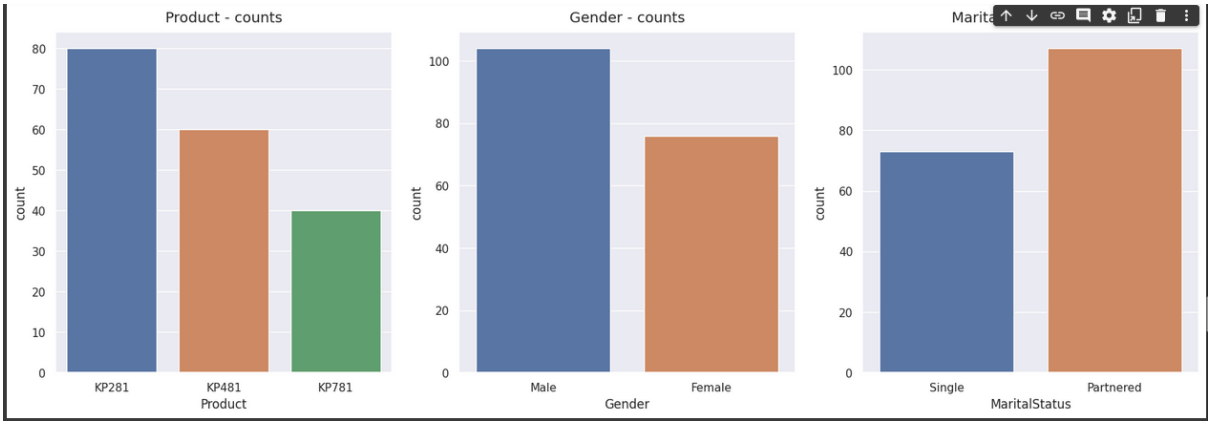
Observation

Even from the boxplots it is quite clear that:

- **Age**, **Education** and **Usage** are having very few outliers.
- While **Income** and **Miles** are having more outliers.

2.For categorical variable(s): Boxplot (10 Points)

```
fig, axs = plt.subplots(nrows=1, ncols=3, figsize=(20, 6))

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()
```



Obervations

- **KP281** is the most frequent product.
- Thare are more Males in the data than Females.
- More **Partnered** persons are there in the data.

```
df1 = df[['Product', 'Gender', 'MaritalStatus']].melt()

df1.groupby(['variable', 'value'])[['value']].count() / len(df)

df1
```

variable	value	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

Observations

- **Product**
  - 44.44% of the customers have purchased **KP2821** 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**.

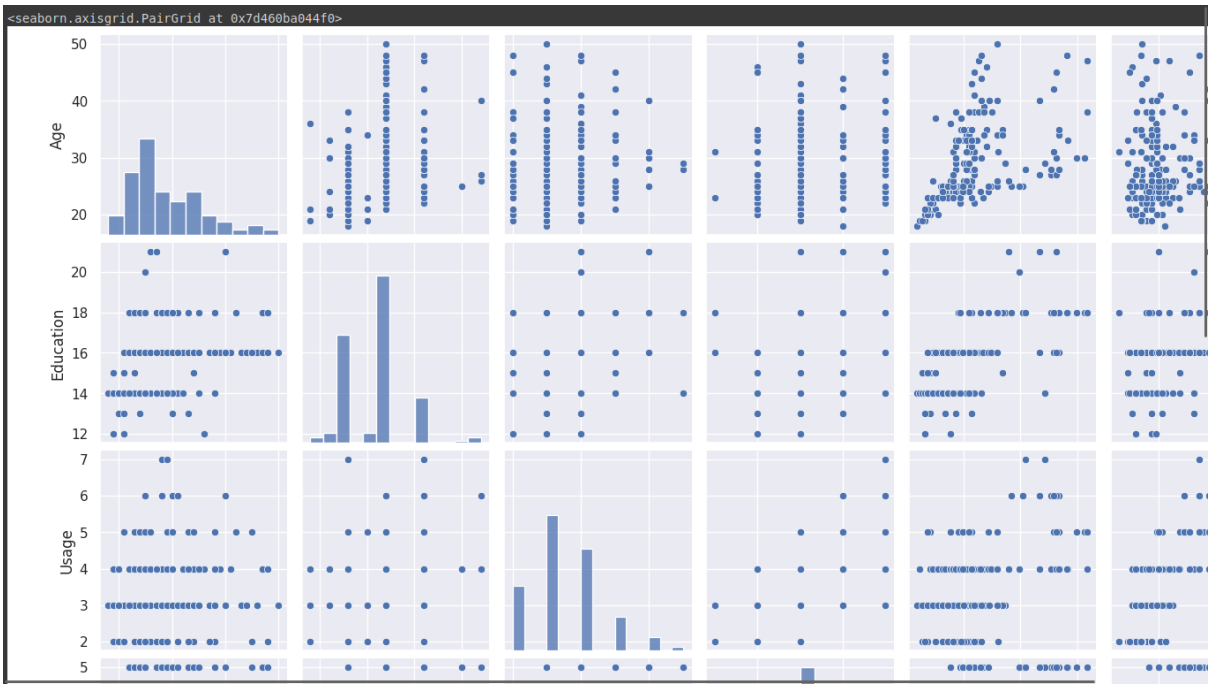
3.For correlation: Heatmaps, Pairplots(10 Points)

```
sns.heatmap(df2)
```

```
<Axes: ylabel='variable-value'>
```

Pairplots :-> `sns.pairplot(df)`

```
<seaborn.axisgrid.PairGrid at 0x7d460ba044f0>
```

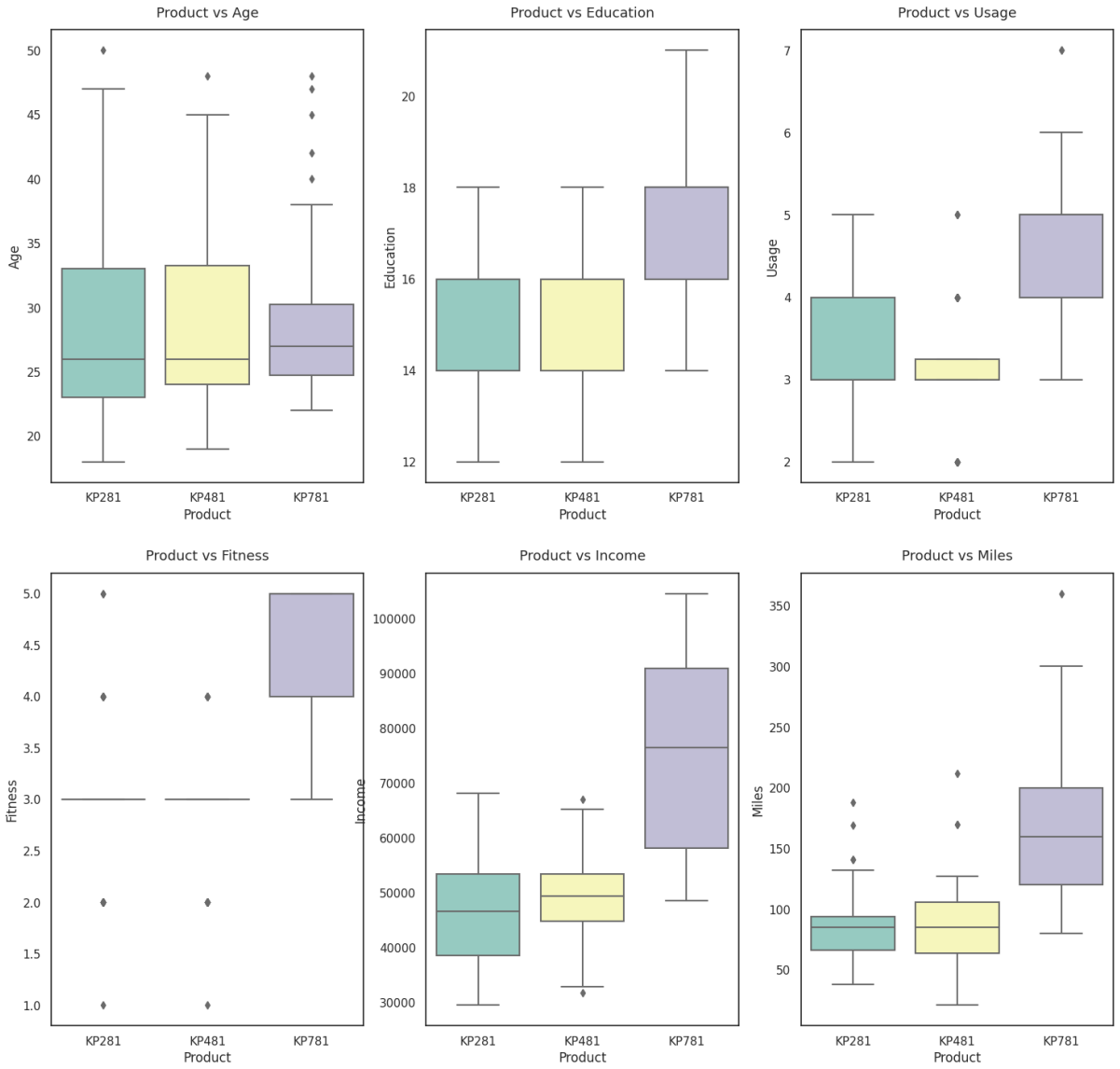


4.Missing Value & Outlier Detection

- **There are no missing values in the data.**
- **There are 3 unique products in the dataset.**
- **KP281 is the most frequent product.**
- **Minimum & Maximum age of the person is 18 & 50, mean is 28.79 and 75% of persons have age less than or equal to 33.**
- **Most of the people are having 16 years of education i.e. 75% of persons are having education <= 16 years.**
- **Out of 180 data points, 104's gender is Male and rest are the female.**
- **Standard deviation for Income & Miles is very high. These variables might have the outliers in it.**

```
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income',
'Miles']

sns.set_style("white")

fig, axs = plt.subplots(nrows=2, ncols=3, figsize=(18, 12))

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=12,
fontsize=13)

        count += 1
```



Observations

1. Product vs Age

- **Customers purchasing products KP281 & KP481 are having same Age median value.**
- **Customers whose age lies between 25-30, are more likely to buy KP781 product**

1. Product vs Education

- **Customers whose `Education` is greater than 16, have more chances to purchase the `KP781` product.**
- **While the customers with `Education` less than 16 have equal chances of purchasing `KP281` or `KP481`.**

1. **`Product vs Usage`**
   - **Customers who are planning to use the treadmill greater than 4 times a week, are more likely to purchase the `KP781` product.**
   - **While the other customers are likely to purchasing `KP281` or `KP481`.**
1. **`Product vs Fitness`**
   - **The more the customer is fit (fitness >= 3), higher the chances of the customer to purchase the `KP781` product.**
1. **`Product vs Income`**
   - **Higher the `Income` of the customer (Income >= 60000), higher the chances of the customer to purchase the `KP781` product.**
1. **`Product vs Miles`**
   - If the customer expects to walk/run greater than 120 Miles per week, it is more likely that the customer will buy `KP781` product.

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

1. **Comments on the range of attributes**

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

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

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

```
def p_prod_given_gender(gender, print_marginal=False):
```

```
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

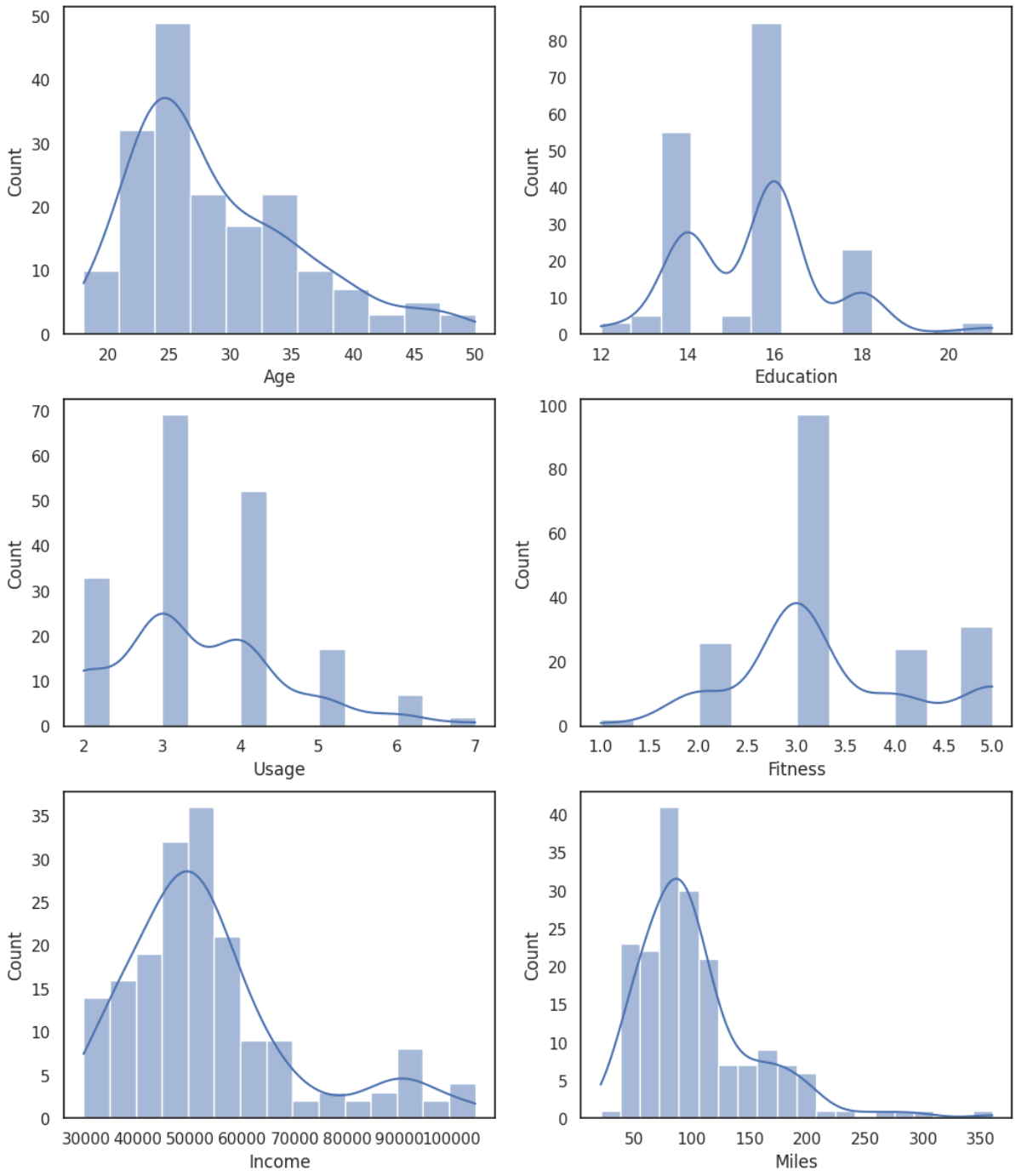
P(KP481/Female): 0.38

P(KP281/Female): 0.53

3. Comments for each univariate and bivariate plot

```
# univariate



fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))

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()
```

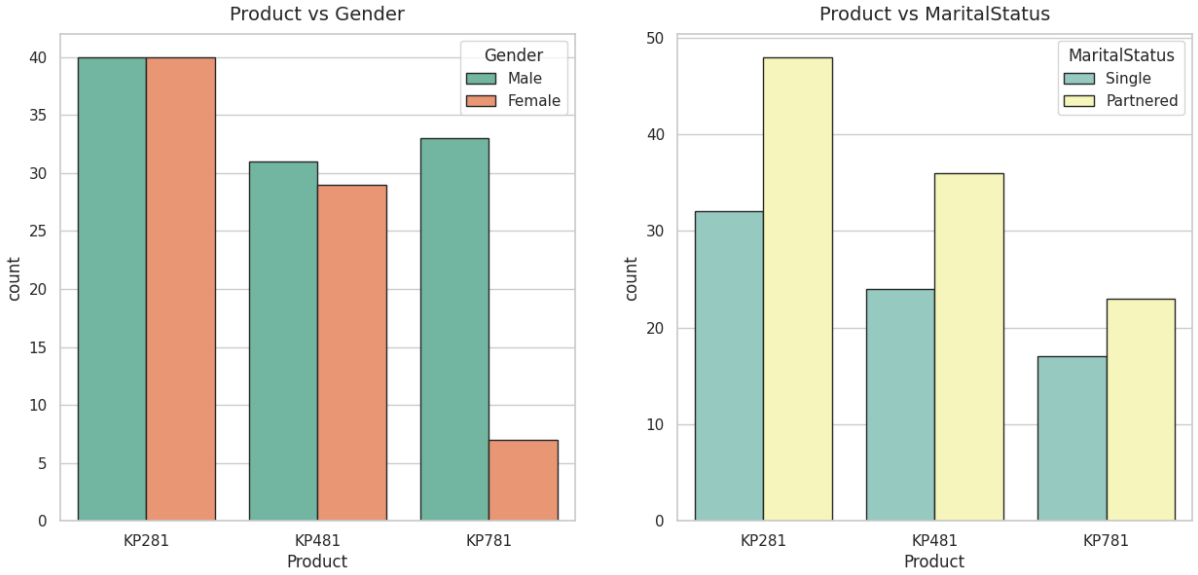


.

Checking if features - `Gender` or `MaritalStatus` have any effect on the product purchased.

```
sns.set_style(style='whitegrid')

fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(15, 6.5))

sns.countplot(data=df, x='Product', hue='Gender', edgecolor="0.15",
palette='Set2', ax=axs[0])
```

```
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()
```



6:Recommendations (10 Points) - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand

1:Learn about Airofit: Take some time to explore and understand what Airofit is and how it can benefit you. Visit their website, watch videos, and read user reviews to get a clear idea.

2:Purchase Airofit: If you find Airofit suitable for your needs, make the purchase either online or through an authorized retailer.

3:Unbox and assemble: Once you receive your Airofit, follow the provided instructions to unbox and assemble the device properly.

4:Breathe correctly: Pay attention to your breathing technique during the exercises. Follow the guidance provided by Airofit to ensure you're using the device effectively.