```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statistics as stat
import warnings
warnings.filterwarnings('ignore')
```

```
aero_data = pd.read_csv('/content/aerofit_treadmill.csv')
aero_data.head(10)
```

	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

Next steps: Generate code with `aero_data` View recommended plots

```
aero_data.sample(10)
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
62	KP281	34	Female	16	Single	2	2	52302	66
176	KP781	42	Male	18	Single	5	4	89641	200
117	KP481	31	Female	18	Single	2	1	65220	21
23	KP281	24	Female	16	Partnered	5	5	44343	188
69	KP281	38	Female	14	Partnered	2	3	54576	56
64	KP281	35	Female	16	Partnered	3	3	60261	94
141	KP781	22	Male	16	Single	3	5	54781	120
100	KP481	25	Female	14	Partnered	5	3	47754	106
143	KP781	23	Male	16	Single	4	5	58516	140
155	KP781	25	Male	18	Partnered	6	5	75946	240

```
aero_data.shape
```

```
(180, 9)
```

Observations

1. Given Dataset, has 180 rows and 9 columns

```
aero_data.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
```

```
aero_data.columns
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
```

```
          dtype='object')
```

```
# Lets Check the Null entries in our dataset

aero_data.isnull().sum()
# aero_data.isnull().any()
```

```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

There are no null values in our data

```
# Check for duplicated entries in our data

aero_data.duplicated().sum()
```

```
0
```

There are no duplicate entries in our data.

```
aero_data.columns
```

```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus', 'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

In Given Data, we have ['Age', 'Education', 'Usage','Fitness', 'Income', 'Miles'] as **numerical columns**. While ['Product','Gender','MaritalStatus'] as **categorical columns**.

```
# Lets describe the numerical columns

aero_data.describe()
```

	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

Above, is the statastical metrics for the given continuous column.

```
# Lets describe the categorical columns

aero_data.describe(include='object')
```

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

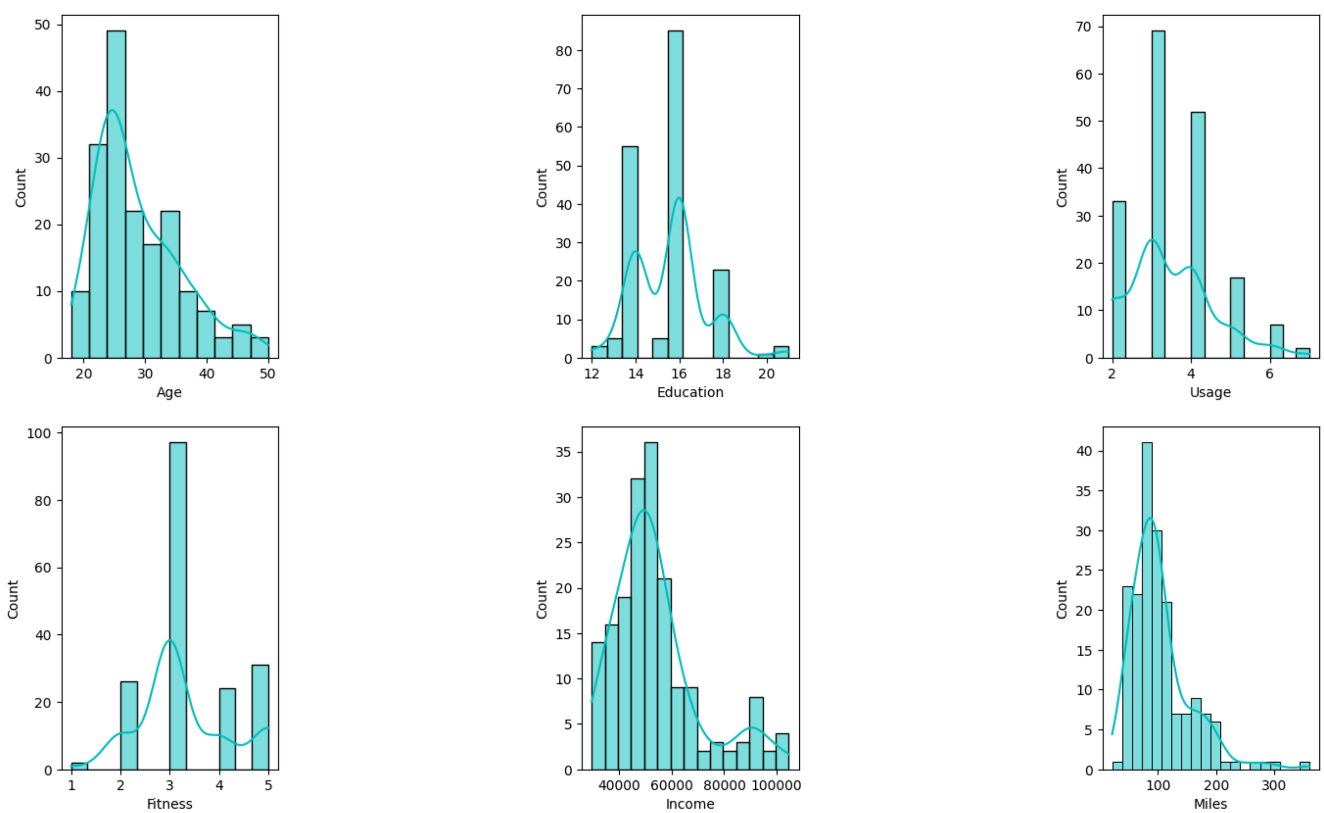
Above, is the describtion for our categorical columns. where "**Product**" has 3 unique values, "**Gender**" has 2 unique values and "**MaritalStatus**" has 2 unique values.

```
col = aero_data.select_dtypes(include='int64').columns
col
```

```
Index(['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles'], dtype='object')
```

Univariate Analysis of Quantitative/Continuous Data

```
# ploting histogram to check the distributions of continuous columns


plt.figure(figsize = (20,10))
plt.subplot(2,6,1)
sns.histplot(data = aero_data.Age, kde = True, color = 'c')

plt.subplot(2,6,3)
sns.histplot(data = aero_data.Education, kde = True, color = 'c')

plt.subplot(2,6,5)
sns.histplot(data = aero_data.Usage, kde = True, color = 'c')

plt.subplot(2,6,7)
sns.histplot(data = aero_data.Fitness, kde = True, color = 'c')

plt.subplot(2,6,9)
sns.histplot(data = aero_data.Income, kde = True, color = 'c')

plt.subplot(2,6,11)
sns.histplot(data = aero_data.Miles, kde = True, color = 'c')

plt.show()
```



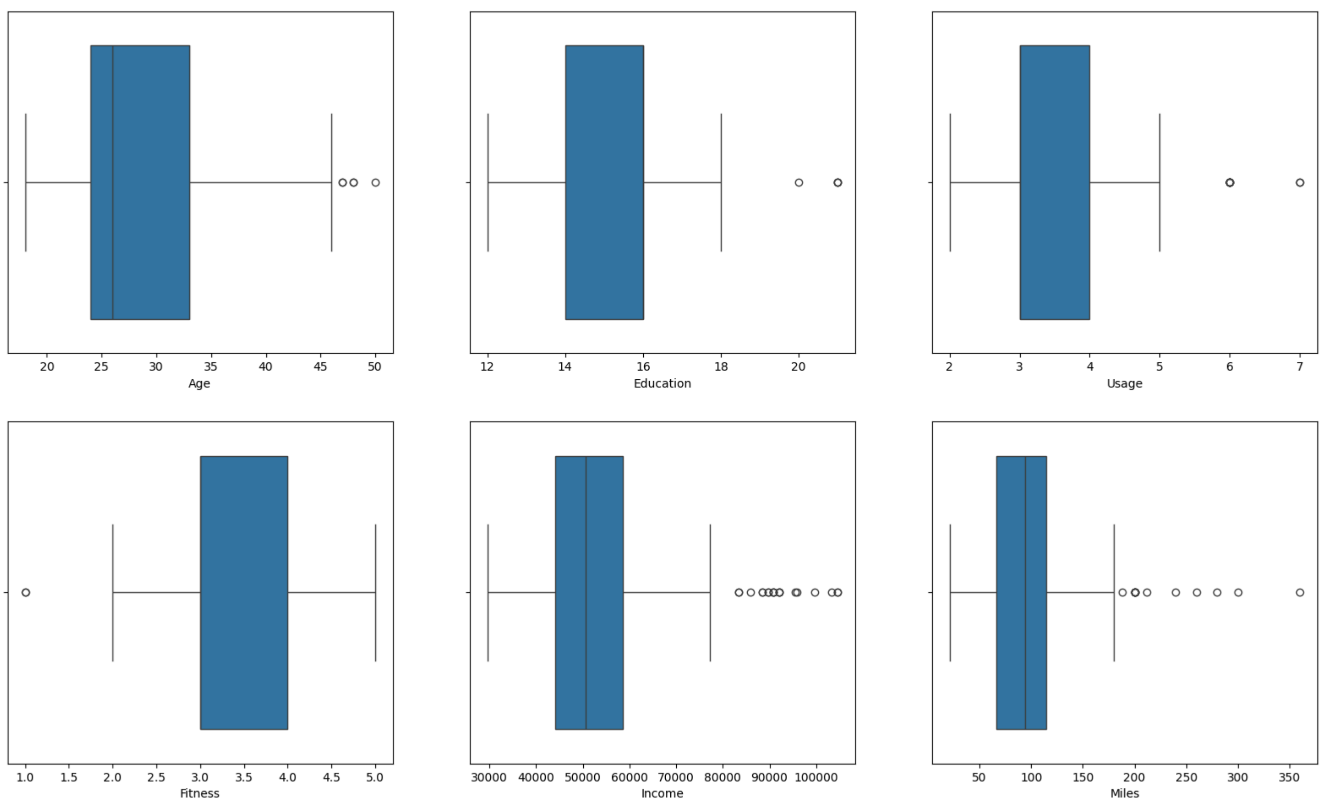
Above is the distribution of the data for the quanatative attributes: **Observations**

1. 'Age', 'Income' and 'Miles' are Right skewed Histogram.
2. Majority 'Age' of the individual lies in between 20 to 30.
3. Most of the 'Employed' individual has an Income in 40,000 to 60,000 range.
4. Most 'Miles' ranges between 60 miles to 100 miles.

```
aero_data.select_dtypes(include = 'object').columns
```

```
Index(['Product', 'Gender', 'MaritalStatus'], dtype='object')
```

```
# Lets try to check the outliers by using boxplots.

fig, ax = plt.subplots(nrows = 2, ncols = 3, figsize = (20,10))
fig.subplots_adjust(top = 1.0)
sns.boxplot(data = aero_data, x = 'Age', ax = ax[0,0])
sns.boxplot(data = aero_data, x = 'Education', ax = ax[0,1])
sns.boxplot(data = aero_data, x = 'Usage', ax = ax[0,2])
sns.boxplot(data = aero_data, x = 'Fitness', ax = ax[1,0])
sns.boxplot(data = aero_data, x = 'Income', ax = ax[1,1])
sns.boxplot(data = aero_data, x = 'Miles', ax = ax[1,2])
plt.show()
```

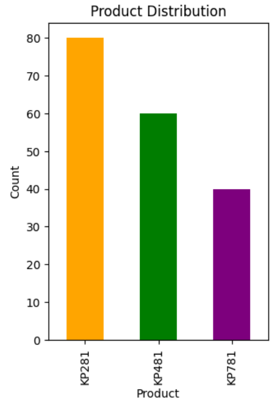


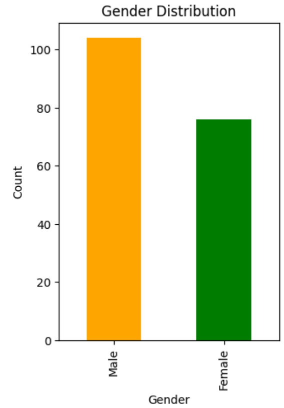
Observations

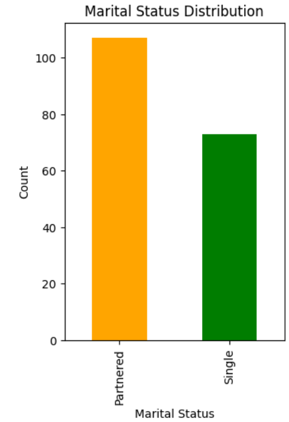
1. 'Income' and 'Miles' have more outliers than any other attributes.

Univariate Analysis of Qualitative/Categorical Data

```
# Create the figure with enough width to display all subplots
plt.figure(figsize=(20, 5))

# Define the colors for each category
colors_product = ['orange', 'green', 'purple', 'cyan', 'blue']  # Adjust the number of colors
colors_gender = ['orange', 'green']  # Adjust the number of colors as needed
colors_marital_status = ['orange', 'green', 'purple', 'cyan', 'blue', 'red']  # Adjust the nu

# Create the first subplot for Product Distribution
plt.subplot(1, 5, 1)
product_counts = aero_data.Product.value_counts()
product_counts.plot(kind = 'bar', color = colors_product[:len(product_counts)])
plt.title('Product Distribution')
plt.xlabel('Product')
plt.ylabel('Count')

# Create the second subplot for Gender Distribution
plt.subplot(1, 5, 3)
gender_counts = aero_data.Gender.value_counts()
gender_counts.plot(kind='bar', color=colors_gender[:len(gender_counts)])
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')

# Create the third subplot for Marital Status Distribution
plt.subplot(1, 5, 5)
marital_status_counts = aero_data.MaritalStatus.value_counts()
marital_status_counts.plot(kind='bar', color=colors_marital_status[:len(marital_status_counts
plt.title('Marital Status Distribution')
plt.xlabel('Marital Status')
plt.ylabel('Count')

# 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=12)
# axs[1].set_title("Gender - counts", pad=10, fontsize=12)
# axs[2].set_title("MaritalStatus - counts", pad=10, fontsize=12)
# plt.show()

# Show the plot
plt.show()
```







Observations Above Distribution Shows following Observations.

1. In Product Distribution, we have three unique products. 'KP281' is the most purchased product, 'KP481' is second least purchased product and 'KP781' is least purchased product.
2. In Gender Distribution, we can see Males are 20% more than Womens.
3. In Marital Status Distribution, there are more people with marital status as Partnered than single.

```
aero_data2 = aero_data[['Product', 'Gender', 'MaritalStatus']].melt() #melting down the data
aero_data2.groupby(['variable', 'value'])[['value']].count() / len(aero_data)
```

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

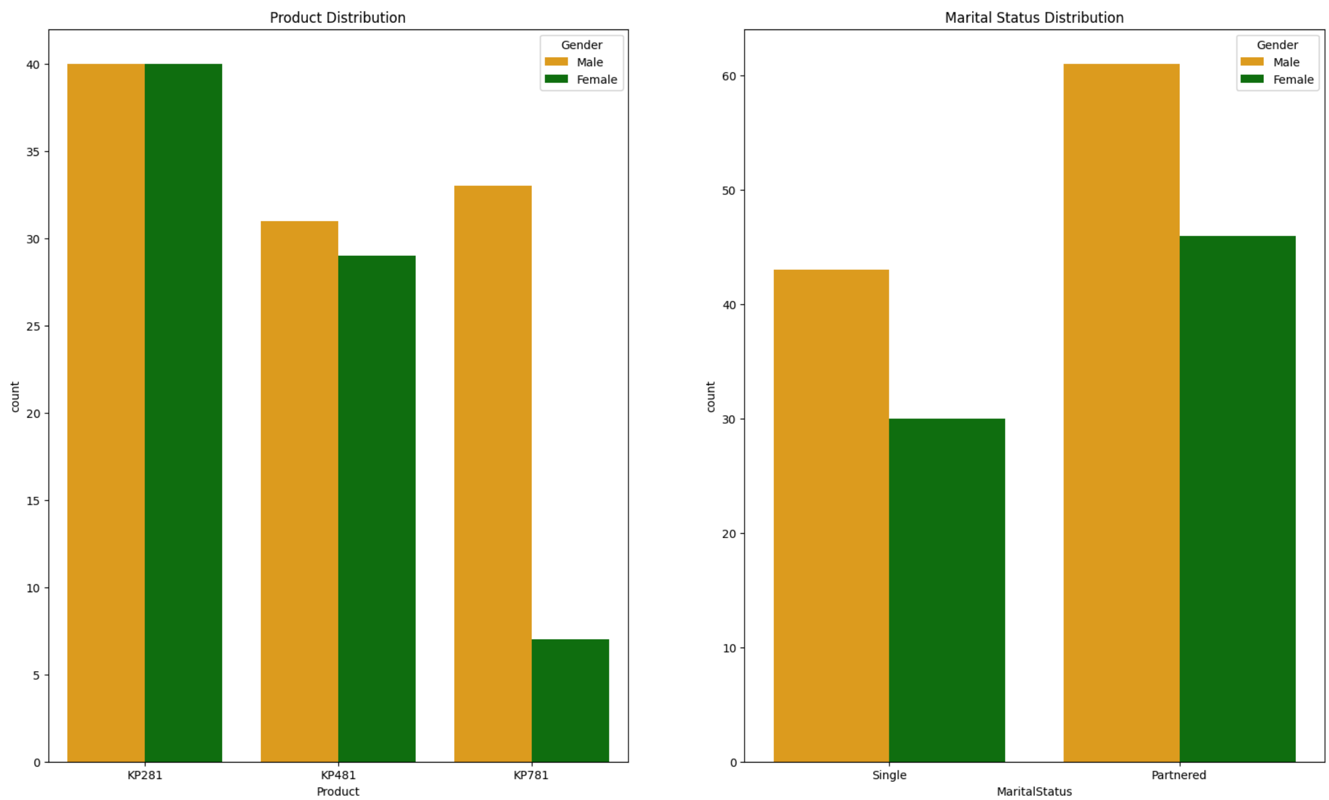
Observations

1. As we can see that 57.78% of the customers are Male.
2. 59.44% of the customers are Partnered.
3. 44.44% of the customers have purchased KP2821 product.
4. 33.33% of the customers have purchased KP481 product.

5. 22.22% of the customers have purchased KP781 product

Bivariate Analysis of Qualitative/Categorical Data

```
color = ['Orange', 'Green']
fig, ax = plt.subplots(nrows = 1, ncols = 2, figsize = (20,10))
fig.subplots_adjust(top = 1.0)
sns.countplot(data = aero_data, x = 'Product',hue = 'Gender', palette = color,  ax = ax[0])
sns.countplot(data = aero_data, x = 'MaritalStatus',hue = 'Gender', palette = color, ax = ax[
ax[0].set_title('Product Distribution')
ax[1].set_title('Marital Status Distribution')
plt.show()
```

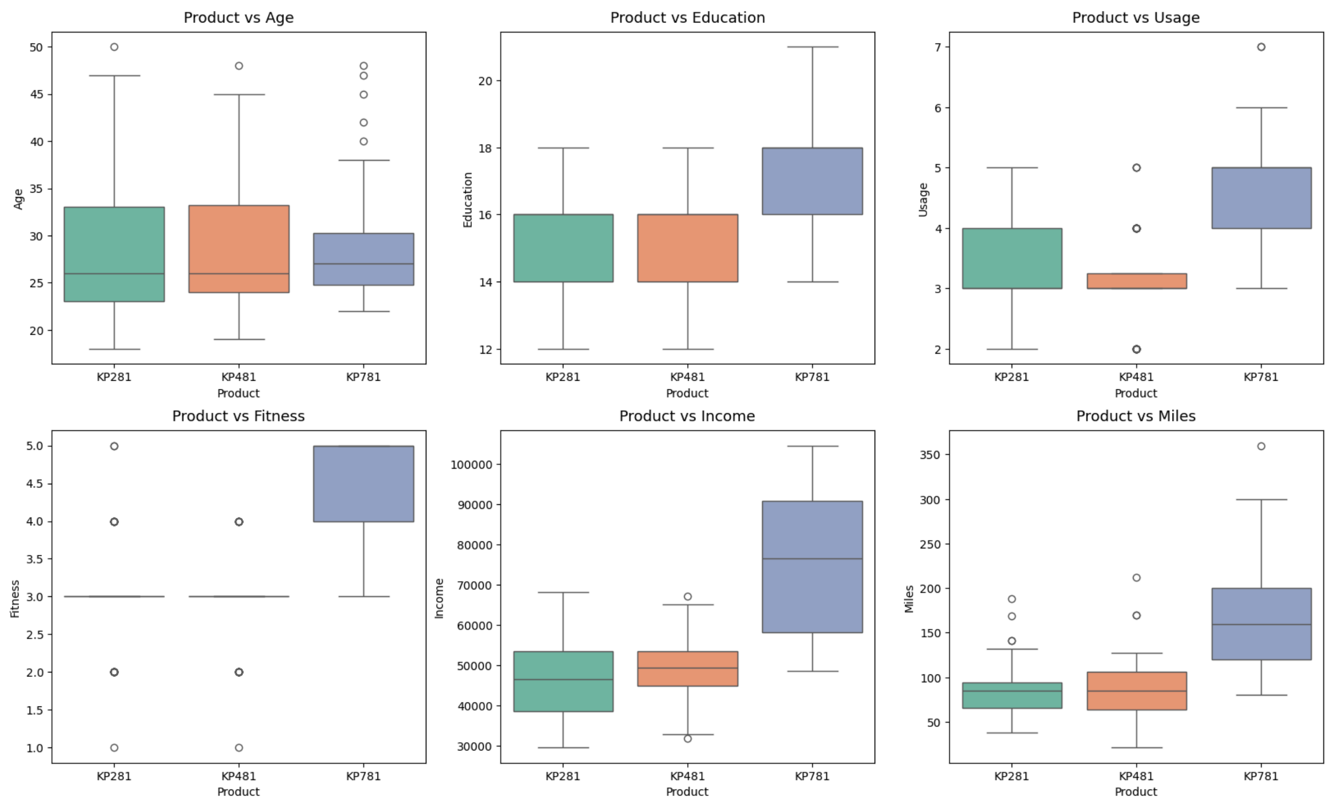


Observations

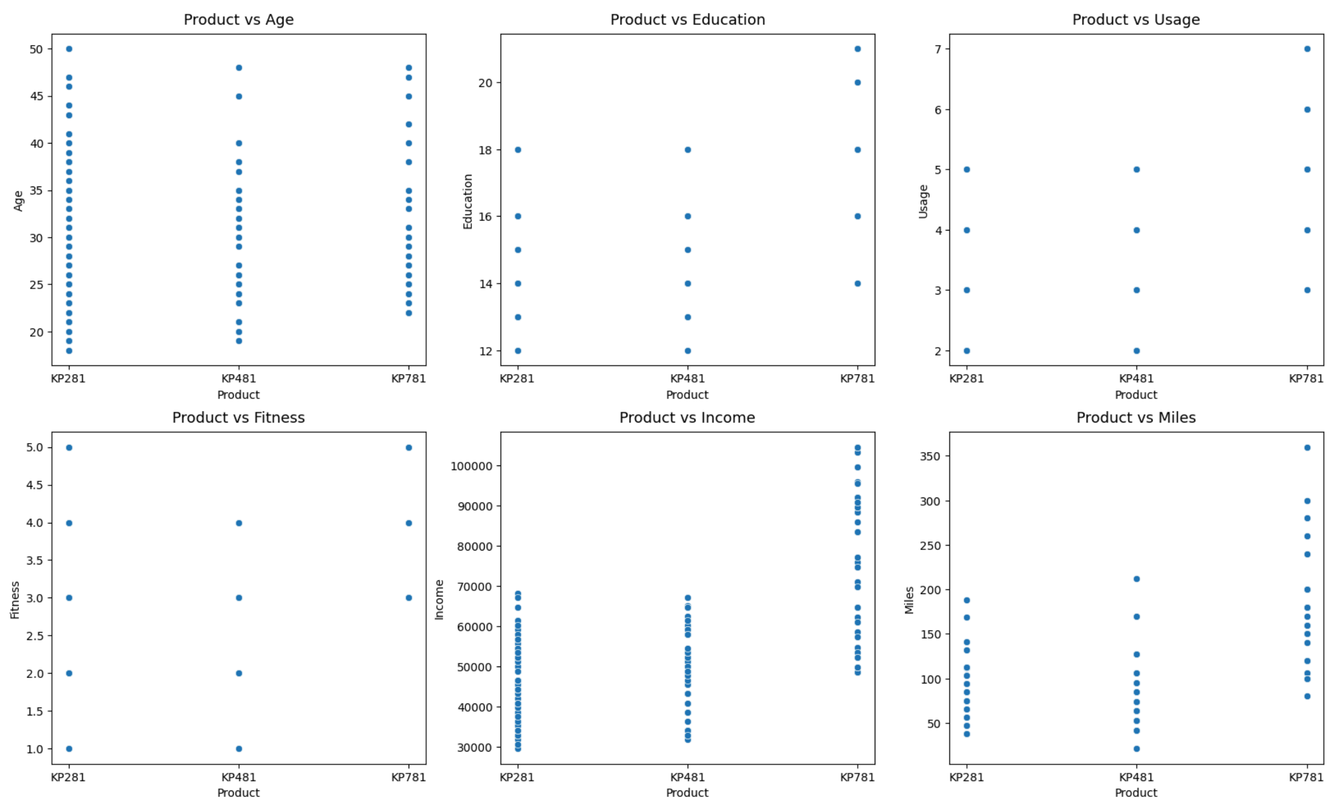
1. Equal number of males and females have purchased KP281 product and Almost same for the product KP481.
2. Most of the Male customers have purchased the KP781 product.
3. Customer who is Partnered, is more likely to purchase the product

Lets see if quantative atributes have any affect on product

```
attr = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
fig, ax = plt.subplots(nrows = 2, ncols = 3, figsize = (20,10))
fig.subplots_adjust(top = 1.0)
count = 0

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



```
attr = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
fig, ax = plt.subplots(nrows = 2, ncols = 3, figsize = (20,10))
fig.subplots_adjust(top = 1.0)
count = 0

for i in range(2):
  for j in range(3):
    sns.scatterplot(data = aero_data, x = 'Product', y = attr[count], ax = ax[i,j], palette =
    ax[i,j].set_title(f"Product vs {attr[count]}", pad = 8, fontsize = 13)
    count += 1
plt.show()
```



Observations

Product vs Age

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

Product vs Education

3. Customers whose Education is greater than 16, have more chances to purchase the KP781 product.

4. While the customers with Education less than 16 have equal chances of purchasing KP281 or KP481.

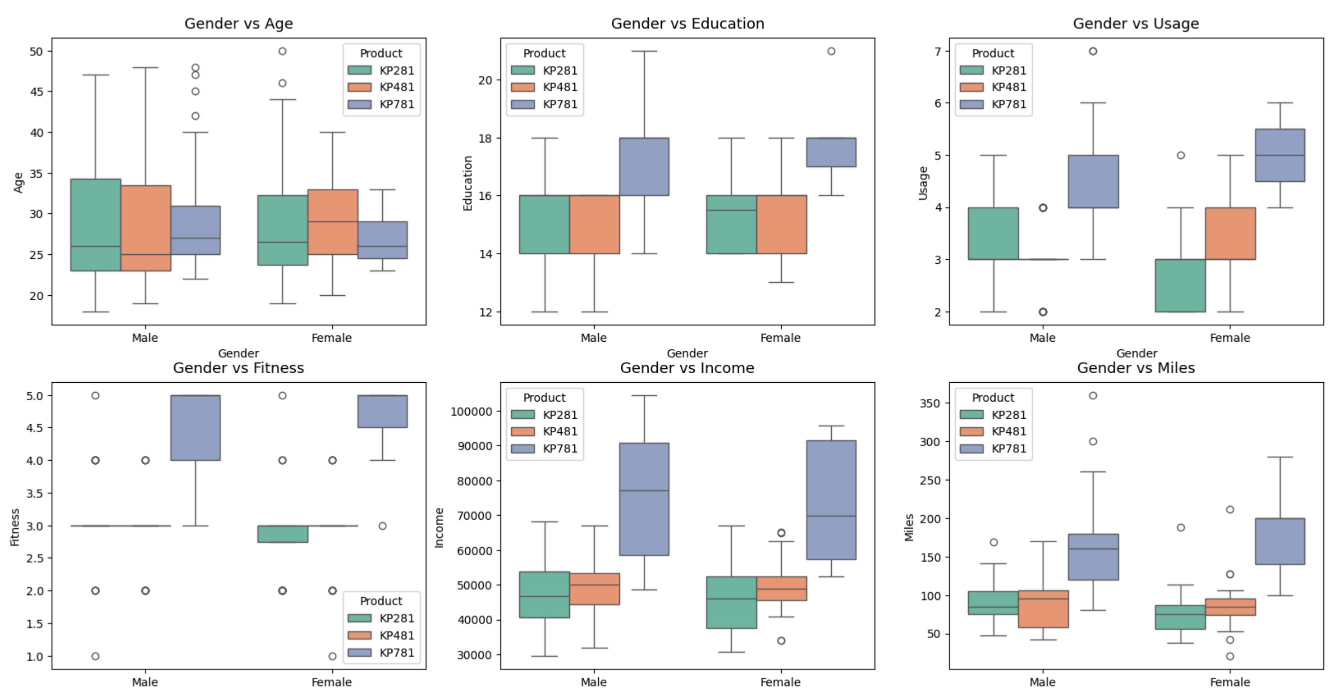
Product vs Usage

5. Customers who are planning to use the treadmill greater than 4 times a week, are more likely to purchase the KP781 product.
6. While the other customers are likely to purchasing KP281 or KP481.

Product vs Fitness

7. The more the customer is fit (fitness >= 3), higher the chances of the customer to purchase the KP781 product

```
attr = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
fiog, ax = plt.subplots(nrows = 2, ncols = 3, figsize = (20,10))
fig.subplots_adjust(top = 1.0)
count = 0

for i in range(2):
  for j in range(3):
    sns.boxplot(data = aero_data, x = 'Gender', y = attr[count],hue = 'Product', ax = ax[i,j]
    ax[i,j].set_title(f"Gender vs {attr[count]}", pad = 8, fontsize = 13)
    count += 1
plt.show()
```



Observation

Gender vs Usage

1. Female usage distribution of all three products are in near to equal proportion as compared to males who are having KP481 usage is very low compared to other two.

Gender vs Income

1. Individual having salary above 60,000 are more tends to buy KP781 Product

```
aero_data['Product'].value_counts(normalize=True)
```

```
Product
KP281    0.444444
KP481    0.333333
KP781    0.222222
Name: proportion, dtype: float64
```

Now Lets Calculate Marginal & Conditional Probabilities:

```
def p_prod_given_gender(gender, print_marginal=False):
  if gender != "Female" and gender != "Male":
    return "Invalid gender value."
  aero_data_01 = pd.crosstab(index=aero_data['Gender'], columns=[aero_data['Product']])
  p_781 = aero_data_01['KP781'][gender] / aero_data_01.loc[gender].sum()
  p_481 = aero_data_01['KP481'][gender] / aero_data_01.loc[gender].sum()
  p_281 = aero_data_01['KP281'][gender] / aero_data_01.loc[gender].sum()
  if print_marginal:
    print(f"P(Male): {aero_data_01.loc['Male'].sum()/len(aero_data):.2f}")
    print(f"P(Female): {aero_data_01.loc['Female'].sum()/len(aero_data):.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
```