In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
data = pd.read_csv('https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992
```

In [3]:
```python
data
```

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
```

BASIC INFORMATION

In [4]:
```python
data.head()
```

Out[4]:

	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

In [5]:

```
data.tail()
```

Out[5]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
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

In [6]:

```
data.isna().sum()
#There are no null values
```

Out[6]:

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

In [7]:

```
data.shape
#180 rows and 9 columns
```

Out[7]:

```
(180, 9)
```

In [8]:

```
data.dtypes
```

Out[8]:

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

In [9]:

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

Out[9]:

	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

In [10]:

```
Unique_Pdt = data["Product"].unique()
```

In [11]:

```
Unique_Pdt
```

Out[11]:

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

Outliers detection using Boxplot

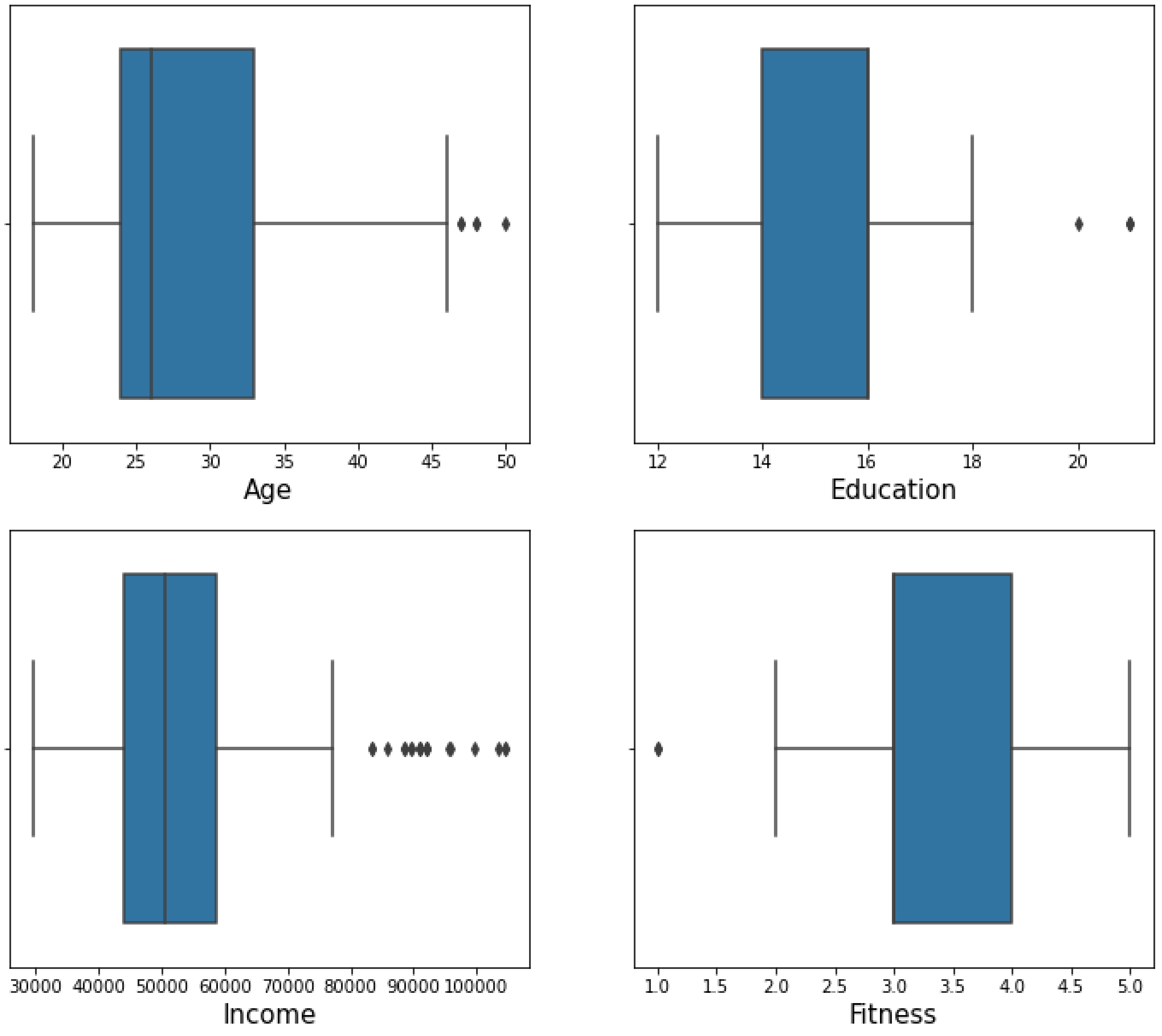
In [12]:

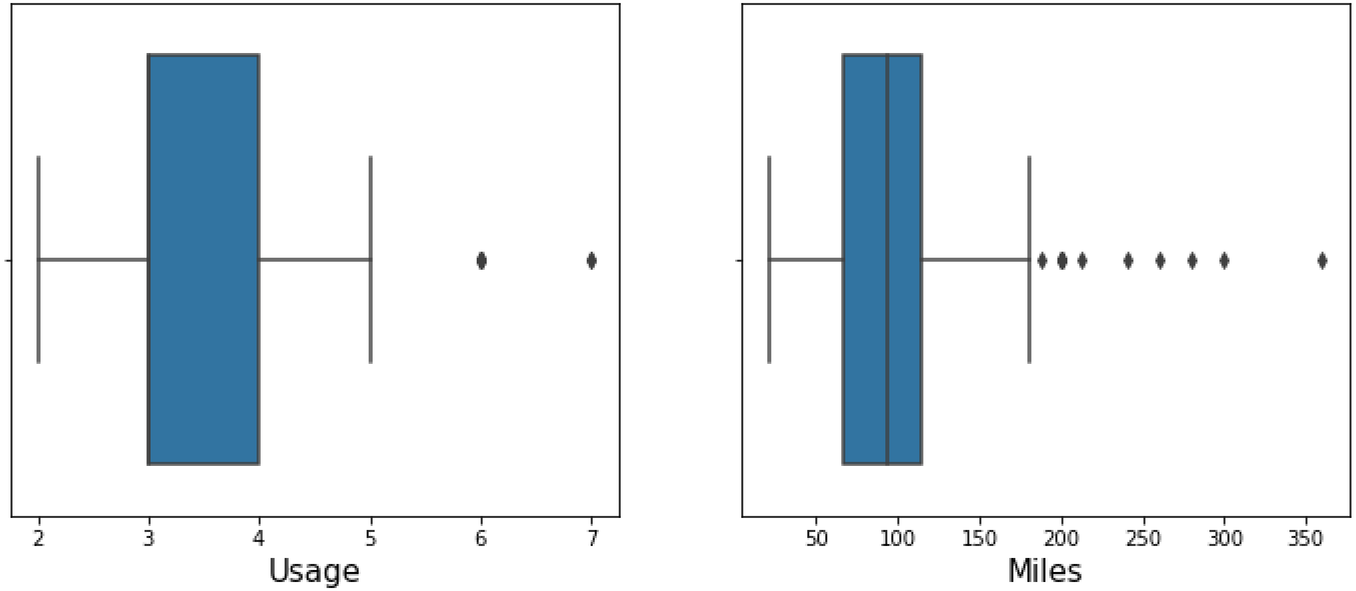
```
plt.figure(figsize = (12, 16))
plt.suptitle('Outliers detection using Boxplot', fontsize = 20)

plt.subplot(3, 2, 1)
plt.xlabel('AGE', fontsize = 15)
sns.boxplot(data = data, x = "Age")

plt.subplot(3, 2, 2)
plt.xlabel('EDUCATION', fontsize = 15)
sns.boxplot(data = data, x = "Education")

plt.subplot(3, 2, 3)
plt.xlabel('INCOME', fontsize = 15)
sns.boxplot(data = data, x = "Income")

plt.subplot(3, 2, 4)
plt.xlabel('FITNESS', fontsize = 15)
sns.boxplot(data = data, x = "Fitness")

plt.subplot(3, 2, 5)
plt.xlabel('USAGE', fontsize = 15)
sns.boxplot(data = data, x = "Usage")

plt.subplot(3, 2, 6)
plt.xlabel('MILES', fontsize = 15)
sns.boxplot(data = data, x = "Miles")
```

Out[12]:

```
<AxesSubplot:xlabel='Miles'>
```

Outliers detection using Boxplot





In [13]:

```
'''
-> Data has 180 rows with 9 columns
-> There are no Null values in the given dataset.
-> Outliers are observed in Income and Miles only.
-> There are 3 unique products i.e., 'KP281', 'KP481', 'KP281' and top one is "KP281"
-> People are in age group of 18 to 50 years. Where 75% of the population has age less than or equal to of 33 years.
-> The product is for Male and Female where Male population is more around 104 compared to female.

'''
```

Out[13]:

```
'\n-> Data has 180 rows with 9 columns\n-> There are no Null values in the given dataset. \n-> Outliers are observed in Income a
nd Miles only.\n-> There are 3 unique products i.e., \'KP281\', \'KP481\', \'KP281\' and top one is "KP281"\n-> People are in ag
e group of 18 to 50 years. Where 75% of the population has age less than or equal to of 33 years.\n-> The product is for Male an
d Female where Male population is more around 104 compared to female.\n\n'
```

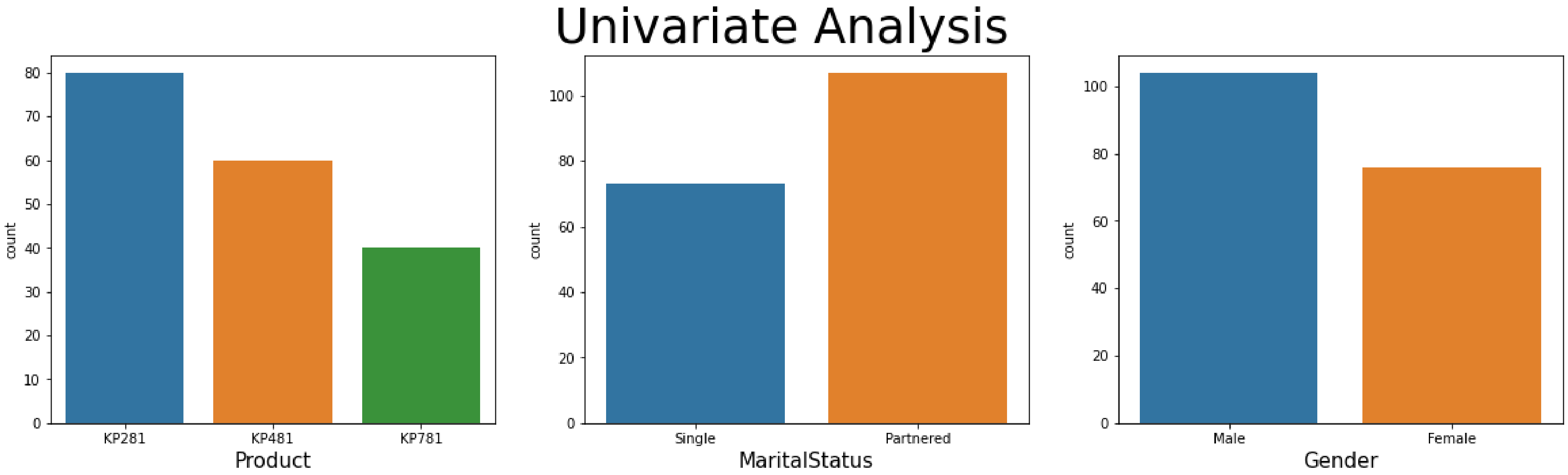
UNIVARIATE ANALYSIS

In [14]:

```
plt.figure(figsize = (20, 5))
plt.suptitle('Univariate Analysis', fontsize = 35)

plt.subplot(1, 3, 1)
plt.xlabel('Product', fontsize = 15)
sns.countplot(x = "Product", data = data)

plt.subplot(1, 3, 2)
plt.xlabel('MaritalStatus', fontsize = 15)
sns.countplot(x = "MaritalStatus", data = data)

plt.subplot(1, 3, 3)
plt.xlabel('Gender', fontsize = 15)
sns.countplot(x = "Gender", data = data)
```

Out[14]:

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



In [15]:

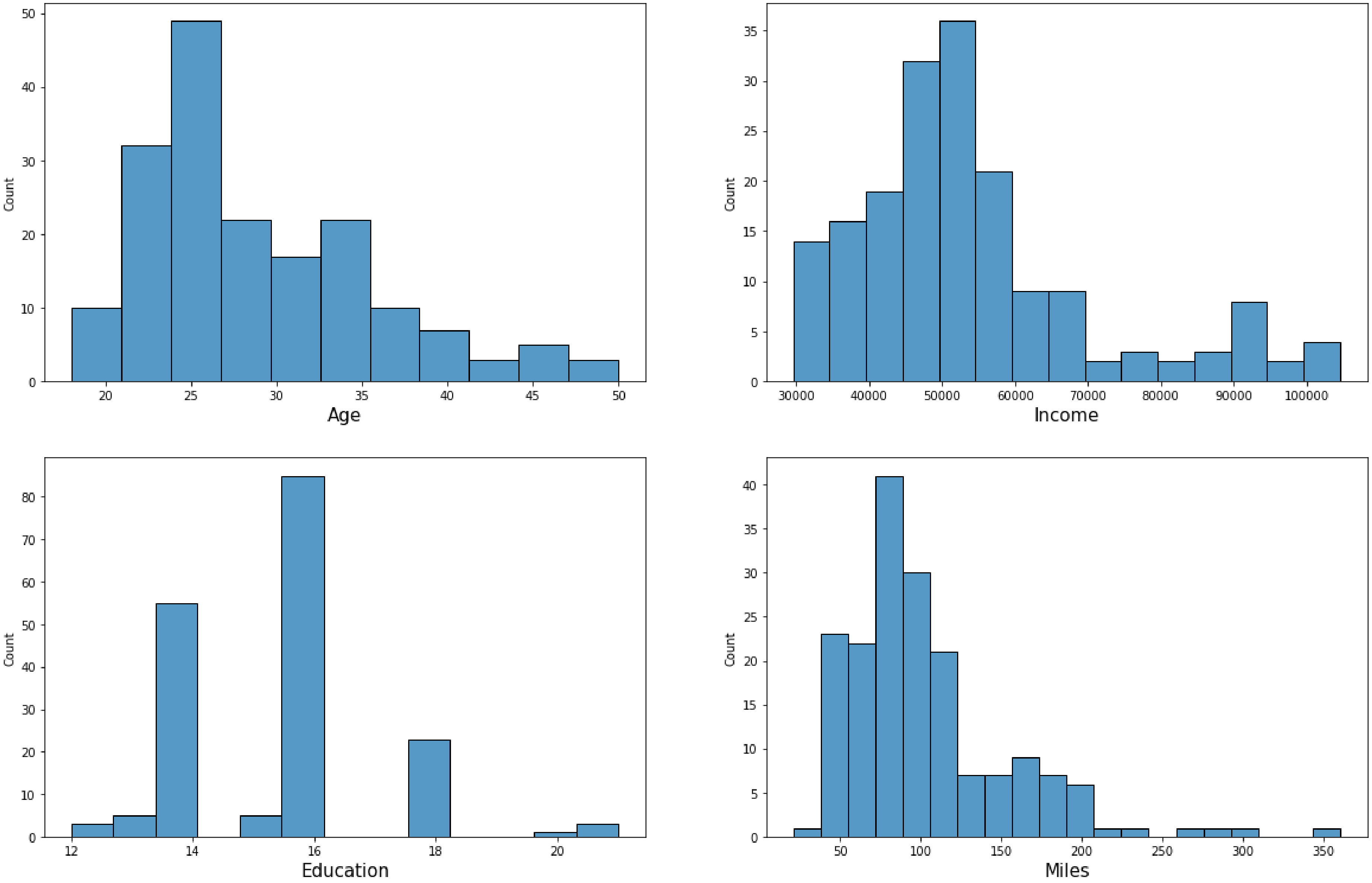
```
plt.figure(figsize = (20, 20))
plt.suptitle('Univariate Analysis', fontsize = 50)

plt.subplot(3, 2, 1)
plt.xlabel('Age', fontsize = 15)
sns.histplot(x = "Age", data = data)

plt.subplot(3, 2, 2)
plt.xlabel('Income', fontsize = 15)
sns.histplot(x = "Income", data = data)

plt.subplot(3, 2, 3)
plt.xlabel('Education', fontsize = 15)
```

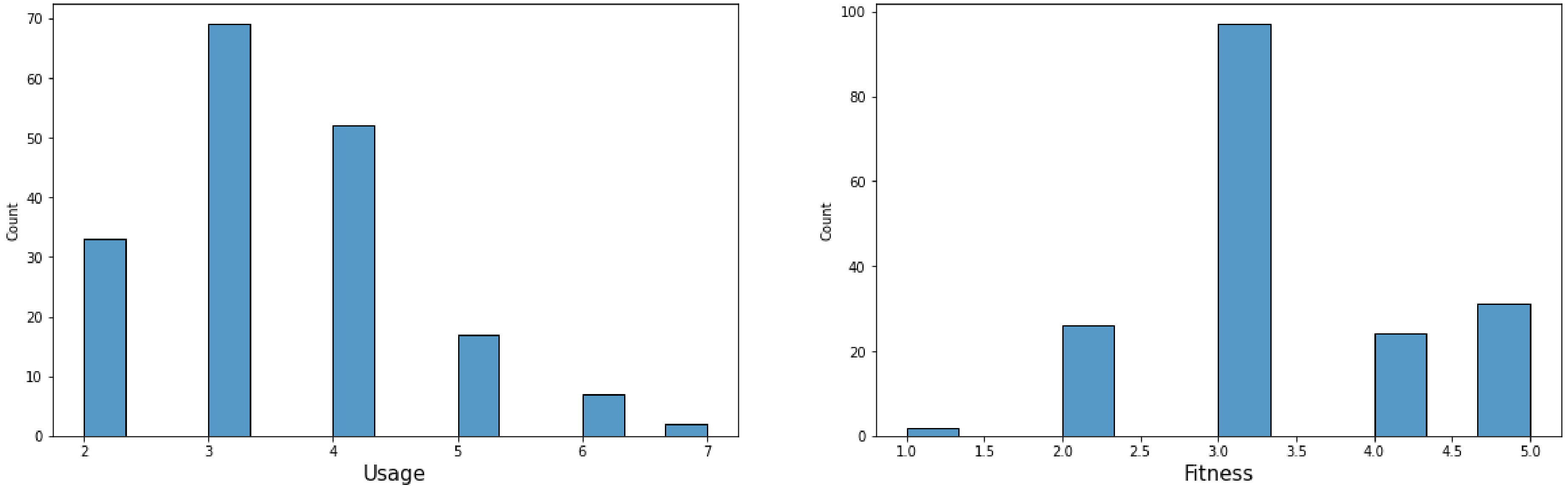
```
sns.histplot(x = "Education", data = data)

plt.subplot(3, 2, 4)
plt.xlabel('Miles', fontsize = 15)
sns.histplot(x = "Miles", data = data)

plt.subplot(3, 2, 5)
plt.xlabel('Usage', fontsize = 15)
sns.histplot(x = "Usage", data = data)

plt.subplot(3, 2, 6)
plt.xlabel('Fitness', fontsize = 15)
sns.histplot(x = "Fitness", data = data)
```

Out[15]:

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

Univariate Analysis





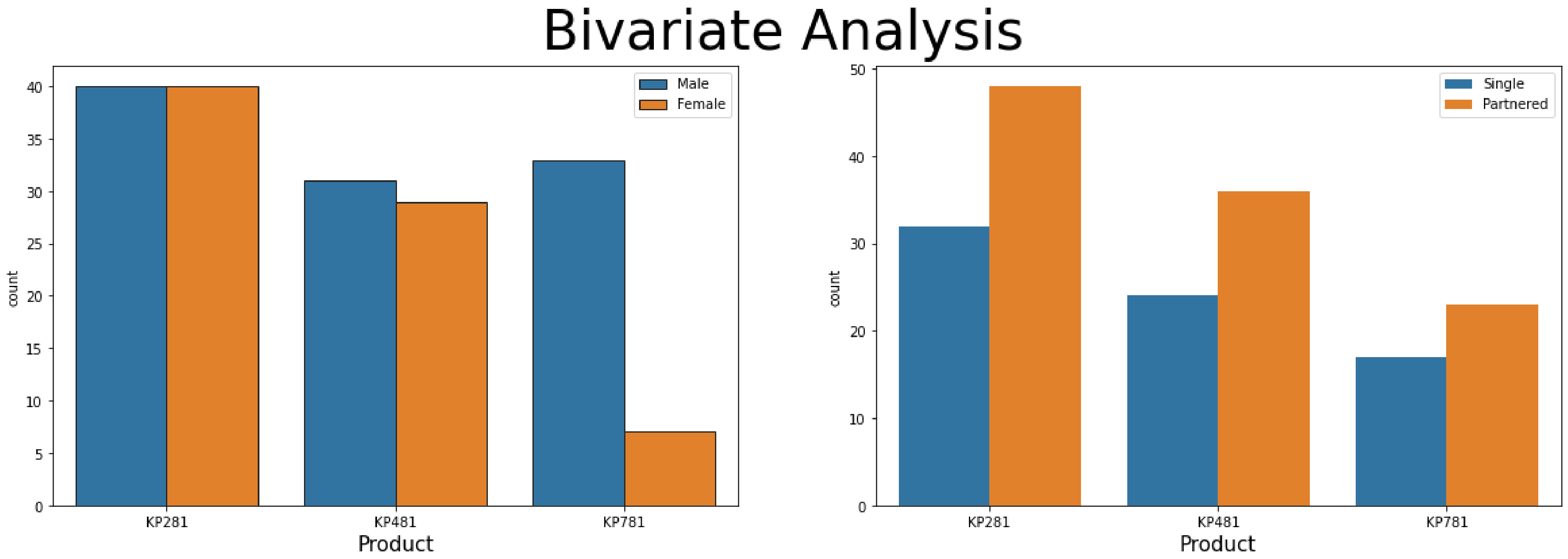
BIVARIATE ANALYSIS

In [17]:

```
plt.figure(figsize = (20, 6))
plt.suptitle('Bivariate Analysis', fontsize = 40)

plt.subplot(1, 2, 1)
plt.xlabel('Product wrt Gender', fontsize = 15)
sns.countplot(data = data, x = "Product", hue = "Gender", edgecolor="0.15")
plt.legend(loc = 'upper right')


plt.subplot(1, 2, 2)
plt.xlabel('Product wrt Mariage', fontsize = 15)
sns.countplot(data = data, x = "Product", hue = "MaritalStatus")
plt.legend(loc = 'upper right')
```

Out[17]:

```
<matplotlib.legend.Legend at 0x2334d0f2e80>
```



In [ ]:

```
'''
-> KP281 is the most purchased product.
-> Equal number of males and females have purchased KP281 product and approximately same for KP481.
-> For Product KP781, it is mostly purchased by Male customers
-> All the 3 products are preferably purchased by "Partnered" population than "Single's"
'''
```

In [20]:

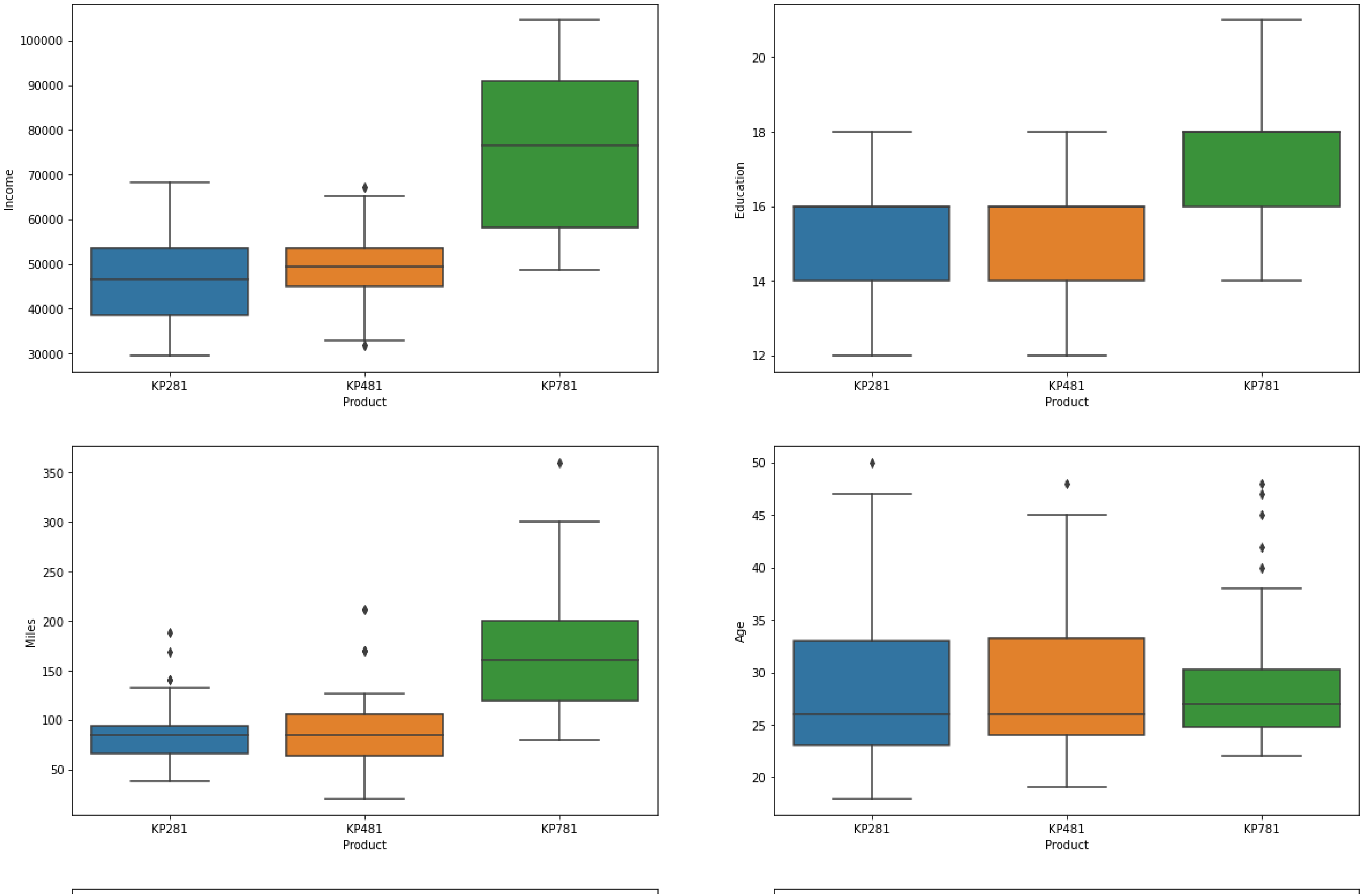
```
plt.figure(figsize = (20, 20))
plt.suptitle('Bivariate Analysis', fontsize = 40)

plt.subplot(3, 2, 1)
sns.boxplot(data = data, x = "Product", y = "Income")

plt.subplot(3, 2, 2)
sns.boxplot(data = data, x = "Product", y = "Education")

plt.subplot(3, 2, 3)
sns.boxplot(data = data, x = "Product", y = "Miles")

plt.subplot(3, 2, 4)
sns.boxplot(data = data, x = "Product", y = "Age")
```

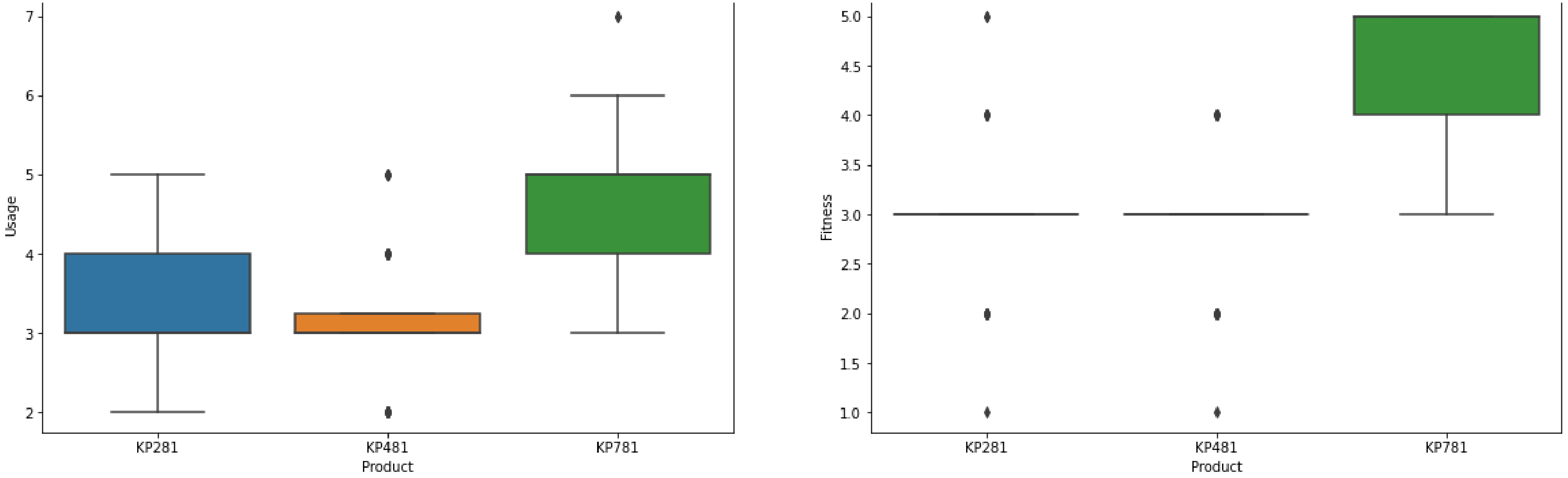
```
plt.subplot(3, 2, 5)
sns.boxplot(data = data, x = "Product", y = "Usage")

plt.subplot(3, 2, 6)
sns.boxplot(data = data, x = "Product", y = "Fitness")
```

Out[20]:

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

Bivariate Analysis





In [ ]:

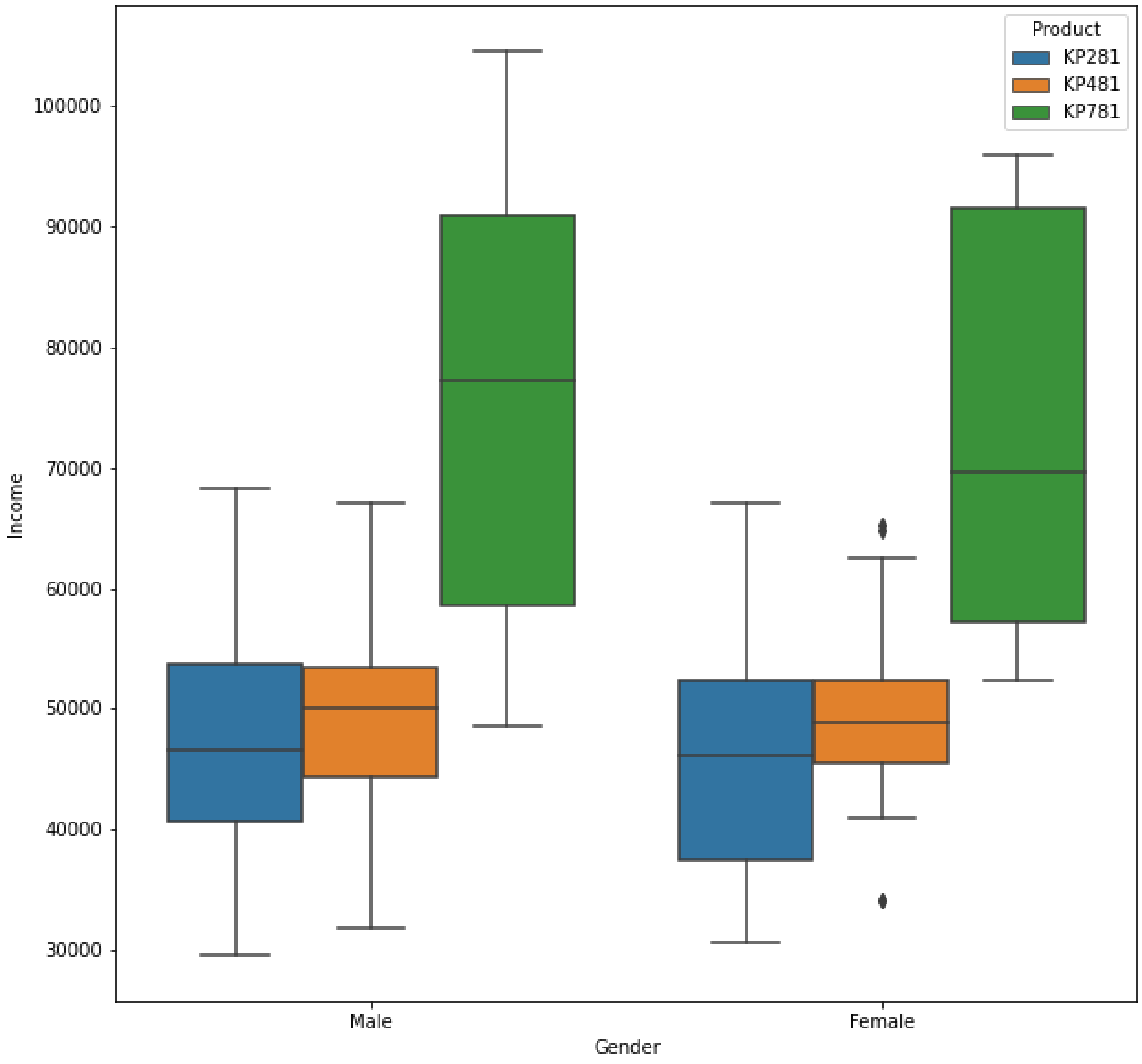
```
'''
From the data we can interpret that,
Product "KP781" is mostly purchased by customers:
->whose income is > 60000, having education >16.
->Customer who is planning to use treadmill more than 4 times a week and expect to walk/run greater than 120 Miles/week
with a fitness level will be more than 3

->Whereas KP281 and KP481 is preferred by customers who does not fall in the above category

Product Capability "KP281 < KP481 < KP781"
'''
```

In [67]:

```
plt.figure(figsize = (10, 10))
sns.boxplot(data = data, x = "Gender", y = "Income", hue = "Product")
```

Out[67]:

```
<AxesSubplot:xlabel='Gender', ylabel='Income'>
```



Probability

In [70]:

```
#Marginal Probability
data["Product"].value_counts()/data["Product"].value_counts().sum()
```

Out[70]:

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

In [71]:

```
data["MaritalStatus"].value_counts()/data["MaritalStatus"].value_counts().sum()
```

Out[71]:

```
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
```

In [72]:

```
data["Gender"].value_counts()/data["Gender"].value_counts().sum()
```

Out[72]:

```
Male      0.577778
Female    0.422222
Name: Gender, dtype: float64
```

In [ ]:

```
'''
44.4% of the customers have purchased "KP281"
33.3% of the customers have purchased "KP481"
22.2% of the customers have purchased "KP781"

59.4% of the customers are Partnered and 40% are Single

57.7% are Male customers wile 42.2 % are Female customers

'''
```

In [ ]:

```
#Conditional Probability for Gender w.r.t Product
```

In [30]:

```
df = pd.crosstab(index = data["Gender"], columns = data["Product"], margins = True)
df
```

Out[30]:

Product	KP281	KP481	KP781	All
Gender				
Female	40	29	7	76
Male	40	31	33	104
All	80	60	40	180

In [43]:

```
Cond1 = pd.crosstab(index = data["Gender"], columns = data["Product"], margins = True, normalize = "columns")*100
Cond1
```

Out[43]:

Product	KP281	KP481	KP781	All
Gender				
Female	50.0	48.333333	17.5	42.222222
Male	50.0	51.666667	82.5	57.777778

In [ ]:

```
'''
Insights Cond1: Conditional Probability

P(Female|KP281) = 0.5
P(Male|KP281) = 0.5

P(Female|KP481) = 0.48
P(Male|KP481) = 0.52

P(Female|KP781) = 0.17
P(Male|KP781) = 0.82
'''
```

In [42]:

```
Cond2 = pd.crosstab(index = data["Gender"], columns = data["Product"], margins = True, normalize = "index")*100
Cond2
```

Out[42]:

Product	KP281	KP481	KP781
Gender			
Female	52.631579	38.157895	9.210526
Male	38.461538	29.807692	31.730769
All	44.444444	33.333333	22.222222

In [ ]:

```
'''
Insights Cond2: Coditional probability

P(KP281|Female) = 0.53
P(KP281|Male) = 0.38
```

```
P(KP481|Female) = 0.38
P(KP481|Male) = 0.30

P(KP781|Female) = 0.1
P(KP781|Male) = 0.31

'''
```

```
In [ ]: #Probability for Marital Status w.r.t Product
```

```
In [44]: MargS = pd.crosstab(index = data["MaritalStatus"], columns = data["Product"], margins = True)
         MargS
```

Out[44]:

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

```
In [45]: Marg1 = pd.crosstab(index = data["MaritalStatus"], columns = data["Product"], margins = True, normalize = "columns")*100
         Marg1
```

Out[45]:

Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	60.0	60.0	57.5	59.444444
Single	40.0	40.0	42.5	40.555556

```
In [ ]: '''
        Insights Marg1: Conditional Probability

        P(Partnered|KP281) = 0.6
        P(Single|KP281) = 0.4

        P(Partnered|KP481) = 0.6
```

```
P(Single|KP481) = 0.4

P(Partnered|KP781) = 0.57
P(Single|KP781) = 0.42
'''
```

In [46]:

```python
Marg2 = pd.crosstab(index = data["MaritalStatus"], columns = data["Product"], margins = True, normalize = "index")*100
Marg2
```

Out[46]:

Product	KP281	KP481	KP781
MaritalStatus			
Partnered	44.859813	33.644860	21.495327
Single	43.835616	32.876712	23.287671
All	44.444444	33.333333	22.222222

In [ ]:

```
'''
Insights Marg2: Coditional probability

P(KP281|Partnered) = 0.45
P(KP281|Single) = 0.44

P(KP481|Partnered) = 0.34
P(KP481|Single)= 0.33

P(KP781|Partnered) = 0.22
P(KP781|Single) = 0.23

'''
```

In [ ]:

In [ ]: