In [1]:

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

In [5]:

```
df = pd.read_csv(r"C:\Users\akash.bana\Desktop\Akash_backup\Akash\Scaler\Prob & Stats\Pr
df.head()
```

Out[5]:

	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 [4]:

```
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 [5]:

```
df.describe()
```

Out[5]:

	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

In [8]:

```
df.shape
```

Out[8]:

```
(180, 9)
```

In [35]:

```
df.isna().sum()
```

Out[35]:

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

In [13]:

```
df.nunique()
```

Out[13]:

```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

In [15]:

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

Out[15]:

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

In [16]:

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

Out[16]:

```
array(['Male', 'Female'], dtype=object)
```

In [17]:

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

Out[17]:

```
array(['Single', 'Partnered'], dtype=object)
```

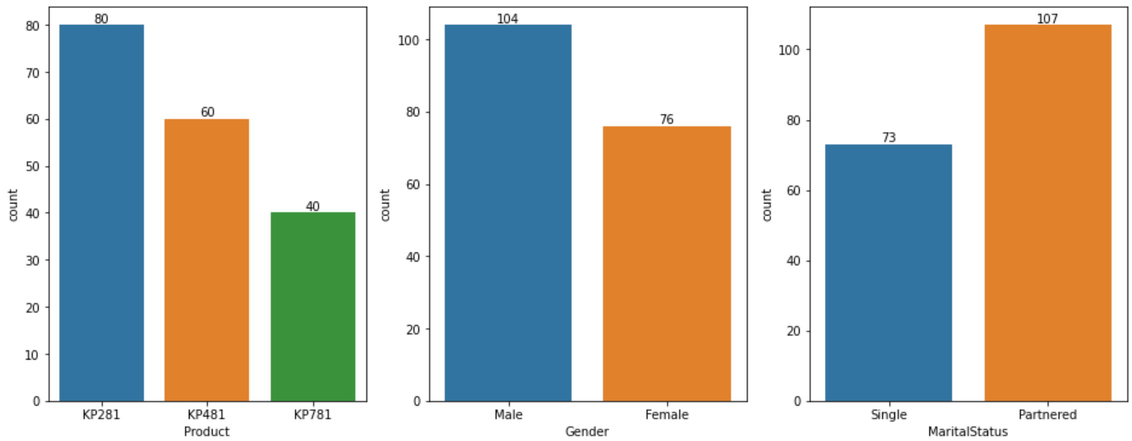
Insights:

1. There are no missing values in the dataset
2. There are 3 products - KP281, KP481, KP781
3. Standard deviation for 'Income' and 'Miles' are high. Outliers are possible in those columns
4. Data types for all the columns are in desired form
5. Age of people varies between 18 and 50

Univariate analysis

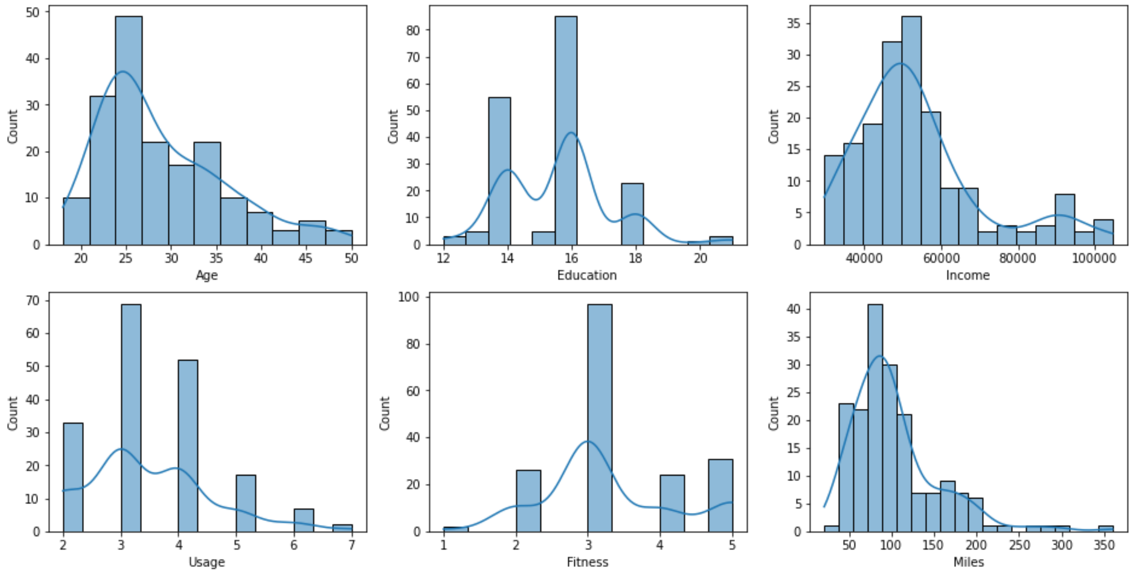
In [32]:

```
plt.figure(figsize=(16,6))
plt.subplot(1,3,1)
x = sns.countplot(data=df,x='Product')
for i in x.containers:
    x.bar_label(i,)
plt.subplot(1,3,2)
y = sns.countplot(data=df,x='Gender')
for i in y.containers:
    y.bar_label(i,)
plt.subplot(1,3,3)
z = sns.countplot(data=df,x='MaritalStatus')
for i in z.containers:
    z.bar_label(i,)
plt.show()
```



In [34]:

```
plt.figure(figsize=(16,8))
plt.subplot(2,3,1)
sns.histplot(data=df,x='Age',kde=True)
plt.subplot(2,3,2)
sns.histplot(data=df,x='Education',kde=True)
plt.subplot(2,3,3)
sns.histplot(data=df,x='Income',kde=True)
plt.subplot(2,3,4)
sns.histplot(data=df,x='Usage',kde=True)
plt.subplot(2,3,5)
sns.histplot(data=df,x='Fitness',kde=True)
plt.subplot(2,3,6)
sns.histplot(data=df,x='Miles',kde=True)
plt.show()
```



Insights:

1. Number of units sold: KP281 > KP481 > KP781
2. Men have purchased more number of units than women, while partnered have purchased more than singles
3. Majority of buyers are aged between 20 & 35
4. Majority of buyers have income between 40,000 & 60,000
5. Majority of the buyers rated themselves 3, these could be buyers with decent phisique wanting to get better

Detecting outliers

In [139]:

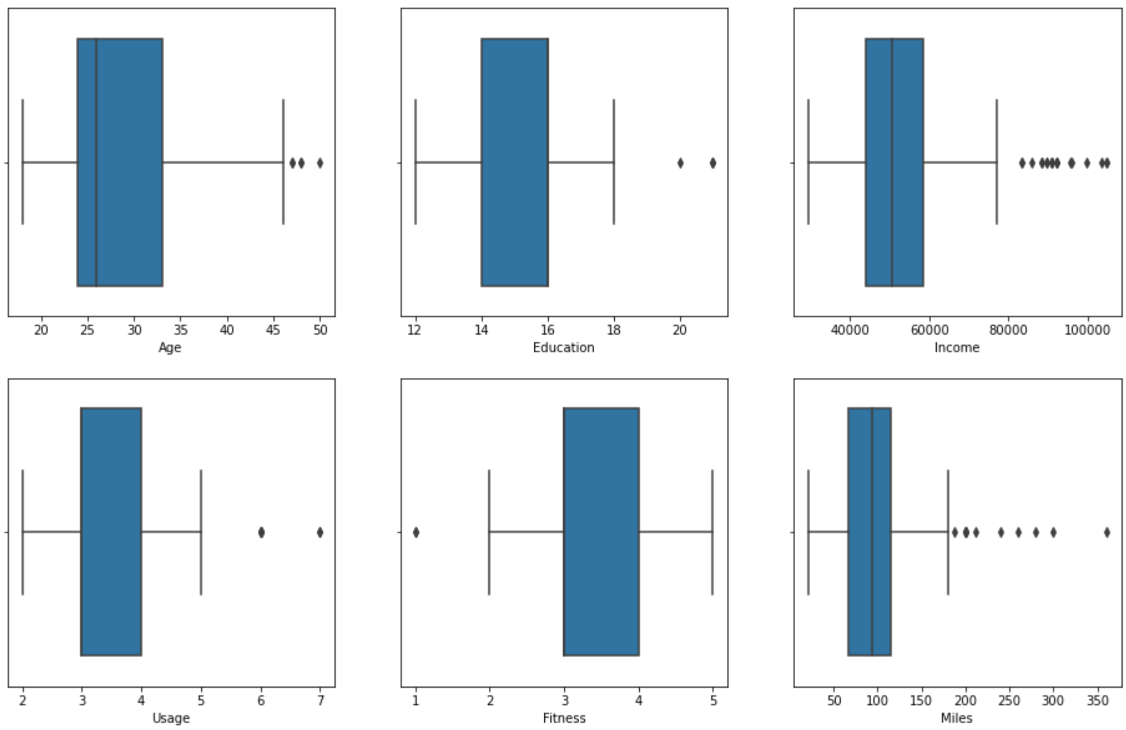
```
outliers = df.describe().loc['mean':'std']
outliers.loc['% deviation'] = outliers.loc['std']/outliers.loc['mean']*100
outliers
```

Out[139]:

	Age	Education	Usage	Fitness	Income	Miles
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
% deviation	24.118674	10.384227	31.392840	28.959118	30.727502	50.258136

In [45]:

```
plt.figure(figsize=(16,10))
plt.subplot(2,3,1)
sns.boxplot(data=df,x='Age')
plt.subplot(2,3,2)
sns.boxplot(data=df,x='Education')
plt.subplot(2,3,3)
sns.boxplot(data=df,x='Income')
plt.subplot(2,3,4)
sns.boxplot(data=df,x='Usage')
plt.subplot(2,3,5)
sns.boxplot(data=df,x='Fitness')
plt.subplot(2,3,6)
sns.boxplot(data=df,x='Miles')
plt.show()
```



Insights:

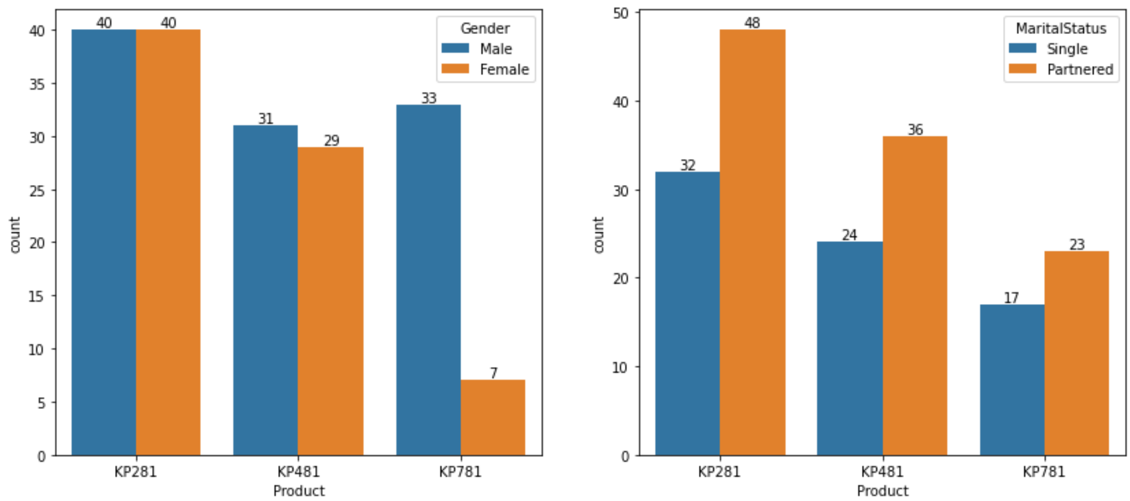
1. 'Income' and 'Miles' have higher number of outliers compared to other variables

Bi-variate analysis

Impact of 'Gender' & 'MaritalStatus' on product purchase

In [53]:

```
plt.figure(figsize=(14,6))
plt.subplot(1,2,1)
x = sns.countplot(data=df,x='Product',hue='Gender')
for i in x.containers:
    x.bar_label(i,)
plt.subplot(1,2,2)
y = sns.countplot(data=df,x='Product',hue='MaritalStatus')
for i in y.containers:
    y.bar_label(i,)
plt.show()
```



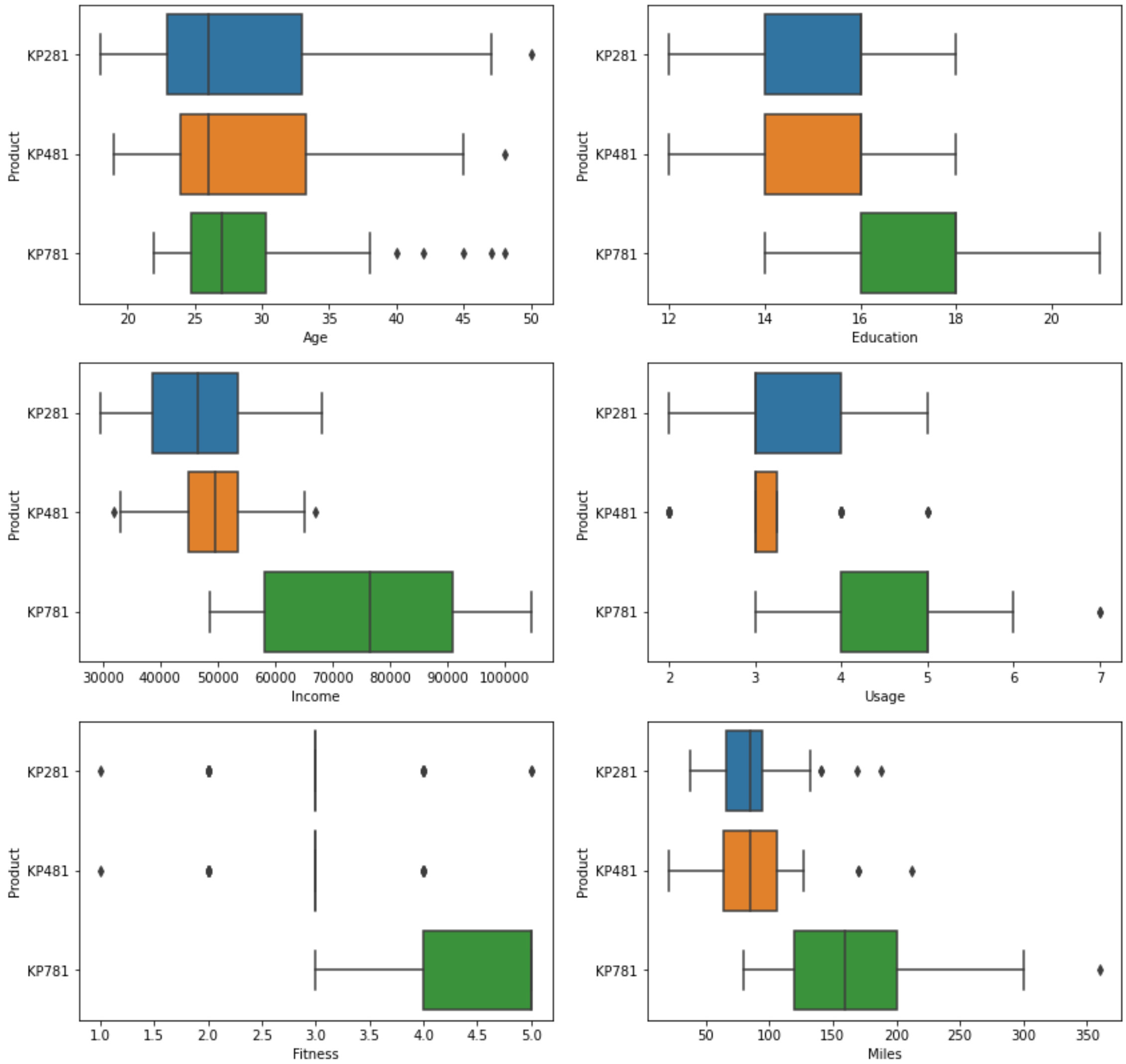
Insights:

1. For products KP281 & KP481, men & women customers could equally purchase the product (i.e. 50% each)
2. For KP781, majority of the buyers are men, around 82.5% of total buyers are men
3. Partnered customers are more likely to purchase the products than singles

Impact of age, education, income, usage, fitness, miles on product purchase

In [68]:

```
plt.figure(figsize=(14,14))
plt.subplot(3,2,1)
sns.boxplot(data=df,y='Product',x='Age')
plt.subplot(3,2,2)
sns.boxplot(data=df,y='Product',x='Education')
plt.subplot(3,2,3)
sns.boxplot(data=df,y='Product',x='Income')
plt.subplot(3,2,4)
sns.boxplot(data=df,y='Product',x='Usage')
plt.subplot(3,2,5)
sns.boxplot(data=df,y='Product',x='Fitness')
plt.subplot(3,2,6)
sns.boxplot(data=df,y='Product',x='Miles')
plt.show()
```



Insights:

1. Mean age of buyers of different products are almost same, lies between the age of 25 & 28
2. Older people (age>40) tend to prefer the product KP781 over other products

3. Customers with education < 16years could prefer KP281 & KP481 while customers with education > 16years could prefer KP781
4. Customers with income lesser than 50,000 USD could prefer comparatively cheaper products, KP281 & KP481 while customers with income greater than 50,000 USD could prefer KP781
5. Customers who want to use the treadmill more than 4 times a week are more likely to prefer KP781
6. Customer with fit body could prefer KP781
7. Customers who are planning to run for more than 100 miles / week could prefer KP781

Marginal probability

P[product]

In [70]:

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

Out[70]:

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

P[gender]

In [72]:

```
df['Gender'].value_counts(normalize=True)
```

Out[72]:

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

P[Marital_status]

In [74]:

```
df['MaritalStatus'].value_counts(normalize=True)
```

Out[74]:

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

Conditional probability

P [product / gender]

In [8]:

```
pd.crosstab(df['Product'],df['Gender'],normalize='columns')
```

Out[8]:

Gender Product	Female	Male
KP281	0.526316	0.384615
KP481	0.381579	0.298077
KP781	0.092105	0.317308

P [product / marital_status]

In [79]:

```
pd.crosstab(df['Product'],df['MaritalStatus'],normalize='columns')
```

Out[79]:

MaritalStatus Product	Partnered	Single
KP281	0.448598	0.438356
KP481	0.336449	0.328767
KP781	0.214953	0.232877

P [product / fitness]

In [81]:

```
pd.crosstab(df['Product'],df['Fitness'],normalize='columns')
```

Out[81]:

Fitness Product	1	2	3	4	5
KP281	0.5	0.538462	0.556701	0.375000	0.064516
KP481	0.5	0.461538	0.402062	0.333333	0.000000
KP781	0.0	0.000000	0.041237	0.291667	0.935484

P [product / usage]

In [82]:

```
pd.crosstab(df['Product'],df['Usage'],normalize='columns')
```

Out[82]:

Usage Product	2	3	4	5	6	7
KP281	0.575758	0.536232	0.423077	0.117647	0.0	0.0
KP481	0.424242	0.449275	0.230769	0.176471	0.0	0.0
KP781	0.000000	0.014493	0.346154	0.705882	1.0	1.0

Recommendation system:

KP281:

1. Majority of the customers could prefer KP281 with men & women equally purchasing the product
2. Mean age - 28.5 years
3. Average Education = 15 years
4. Mean annual salary - 46,000 USD
5. Expects to use - 5 times / week or less
6. Self rating on fitness - 4 / 5 or less
7. Expects to walk - 82 miles / week

KP481:

1. Product is expected to be equally bought by both men & women
2. Mean age - 28.9 years
3. Average Education = 15 years
4. Mean annual salary - 49,000 USD
5. Expects to use - 5 times / week or less
6. Self rating on fitness - 4 / 5 or less
7. Expects to walk - 87 miles / week

KP781:

1. Product is mostly preferred by men
2. Mean age - 29.1 years
3. Average Education = 17 years
4. Mean annual salary - 75,000 USD
5. Expects to use - 4 times / week or more
6. Self rating on fitness - 4 / 5 or more
7. Expects to walk - 166 miles / week

In [ ]: