In [2]:

```
#importing libraries for our purpose
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv('aerofit_treadmill.csv')
df
```

Out[2]:

	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 [5]:

```
1 df.head(30)
```

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
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
10	KP281	22	Male	14	Single	3	3	36384	85
11	KP281	22	Female	14	Partnered	3	2	35247	66
12	KP281	22	Female	16	Single	4	3	36384	75
13	KP281	22	Female	14	Single	3	3	35247	75
14	KP281	23	Male	16	Partnered	3	1	38658	47
15	KP281	23	Male	16	Partnered	3	3	40932	75
16	KP281	23	Female	14	Single	2	3	34110	103
17	KP281	23	Male	16	Partnered	4	3	39795	94
18	KP281	23	Female	16	Single	4	3	38658	113
19	KP281	23	Female	15	Partnered	2	2	34110	38
20	KP281	23	Male	14	Single	4	3	38658	113
21	KP281	23	Male	16	Single	4	3	40932	94
22	KP281	24	Female	16	Single	4	3	42069	94
23	KP281	24	Female	16	Partnered	5	5	44343	188
24	KP281	24	Male	14	Single	2	3	45480	113
25	KP281	24	Male	13	Partnered	3	2	42069	47
26	KP281	24	Female	16	Single	4	3	46617	75
27	KP281	25	Female	14	Partnered	3	3	48891	75
28	KP281	25	Male	14	Partnered	2	3	45480	56
29	KP281	25	Female	14	Partnered	2	2	53439	47

In [3]:

```
#Length of Data
len(df)
```

Out[3]:

```
180
```

In [4]:

```
df.shape # rows and columns
```

Out[4]:

```
(180, 9)
```

In [5]:

```
#What are the different column name?
df.columns
```

Out[5]:

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

In [6]:

```
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
```

Observation:

1) Product,Gender and Martial status are string datatype . 2) While Age, Education,Usage,Fitness,Income,Miles are integer datatype.

In [7]:

```
#Checking Datatypes
df.dtypes
```

Out[7]:

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

Data Preprocessing

In [8]:

```
# changing it to  object dtype to category  to save memory
df.Product=df["Product"].astype("category")
df.Gender=df["Gender"].astype("category")
df.MaritalStatus=df["MaritalStatus"].astype("category")
```

In [9]:

```
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    category
 1   Age            180 non-null    int64   
 2   Gender         180 non-null    category
 3   Education      180 non-null    int64   
 4   MaritalStatus  180 non-null    category
 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: category(3), int64(6)
memory usage: 9.5 KB
```

In [10]:

```
#Are there any duplicate values?
df.duplicated().sum()
```

Out[10]:

```
0
```

In [11]:

```
#No. of unique values in our Data
for  i in df.columns:
    print(i,":",df[i].nunique())
```

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

In [14]:

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

Out[14]:

```
['KP281', 'KP481', 'KP781']
Categories (3, object): ['KP281', 'KP481', 'KP781']
```

In [15]:

```
df["Age"].unique()
```

Out[15]:

```
array([18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 43, 44, 46, 47, 50, 45, 48, 42],
      dtype=int64)
```

In [16]:

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

Out[16]:

```
['Male', 'Female']
Categories (2, object): ['Female', 'Male']
```

In [17]:

```
df["Education"].unique()
```

Out[17]:

```
array([14, 15, 12, 13, 16, 18, 20, 21], dtype=int64)
```

In [18]:

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

Out[18]:

```
['Single', 'Partnered']
Categories (2, object): ['Partnered', 'Single']
```

In [19]:

```
df["Usage"].unique()
```

Out[19]:

```
array([3, 2, 4, 5, 6, 7], dtype=int64)
```

In [20]:

```
df["Fitness"].unique()
```

Out[20]:

```
array([4, 3, 2, 1, 5], dtype=int64)
```

In [21]:

```
df["Income"].unique()
```

Out[21]:

```
array([ 29562,  31836,  30699,  32973,  35247,  37521,  36384,  38658,
        40932,  34110,  39795,  42069,  44343,  45480,  46617,  48891,
        53439,  43206,  52302,  51165,  50028,  54576,  68220,  55713,
        60261,  67083,  56850,  59124,  61398,  57987,  64809,  47754,
        65220,  62535,  48658,  54781,  48556,  58516,  53536,  61006,
        57271,  52291,  49801,  62251,  64741,  70966,  75946,  74701,
        69721,  83416,  88396,  90886,  92131,  77191,  52290,  85906,
       103336,  99601,  89641,  95866, 104581,  95508], dtype=int64)
```

In [22]:

```
df["Miles"].unique()
```

Out[22]:

```
array([112,  75,  66,  85,  47, 141, 103,  94, 113,  38, 188,  56, 132,
       169,  64,  53, 106,  95, 212,  42, 127,  74, 170,  21, 120, 200,
       140, 100,  80, 160, 180, 240, 150, 300, 280, 260, 360], dtype=int64)
```

Observation

No abnormalities were found in data

In [15]:

```
# Checking null values in every column of our Data
df.isnull().sum()
```

Out[15]:

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

Dervied Columns¶

Added 2 new feature from Age

"AgeCategory" - Teens, 20s, 30s and Above 40s
"AgeGroup" - 14-20 , 20-30, 30-40 & 40-60

Added 1 new categorial feature based on the income

"IncomeSlab" - Low Income, Lower-middle income,Upper-Middle income and High income

In [69]:

```
bins = [14,20,30,40,60]
labels =["Teens","20s","30s","Above 40s"]
df['AgeGroup'] = pd.cut(df['Age'], bins)
df['AgeCategory'] = pd.cut(df['Age'], bins,labels=labels)
```

In [70]:

```
df.head()
```

Out[70]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeGroup
0	KP281	18	Male	14	Single	3	4	29562	112	(14, 20]
1	KP281	19	Male	15	Single	2	3	31836	75	(14, 20]
2	KP281	19	Female	14	Partnered	4	3	30699	66	(14, 20]
3	KP281	19	Male	12	Single	3	3	32973	85	(14, 20]
4	KP281	20	Male	13	Partnered	4	2	35247	47	(14, 20]

In [71]:

```
bins_income = [29000, 35000, 60000, 85000,105000]
labels_income = ['Low Income','Lower-middle income','Upper-Middle income', 'High income
df['IncomeSlab'] = pd.cut(df['Income'],bins_income,labels = labels_income)

```

In [72]:

```
df.head()
```

Out[72]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeGroup
0	KP281	18	Male	14	Single	3	4	29562	112	(14, 20]
1	KP281	19	Male	15	Single	2	3	31836	75	(14, 20]
2	KP281	19	Female	14	Partnered	4	3	30699	66	(14, 20]
3	KP281	19	Male	12	Single	3	3	32973	85	(14, 20]
4	KP281	20	Male	13	Partnered	4	2	35247	47	(14, 20]

Function for Outlier detection

Box plot - for checking range of outliers

distplot - For checking skewness

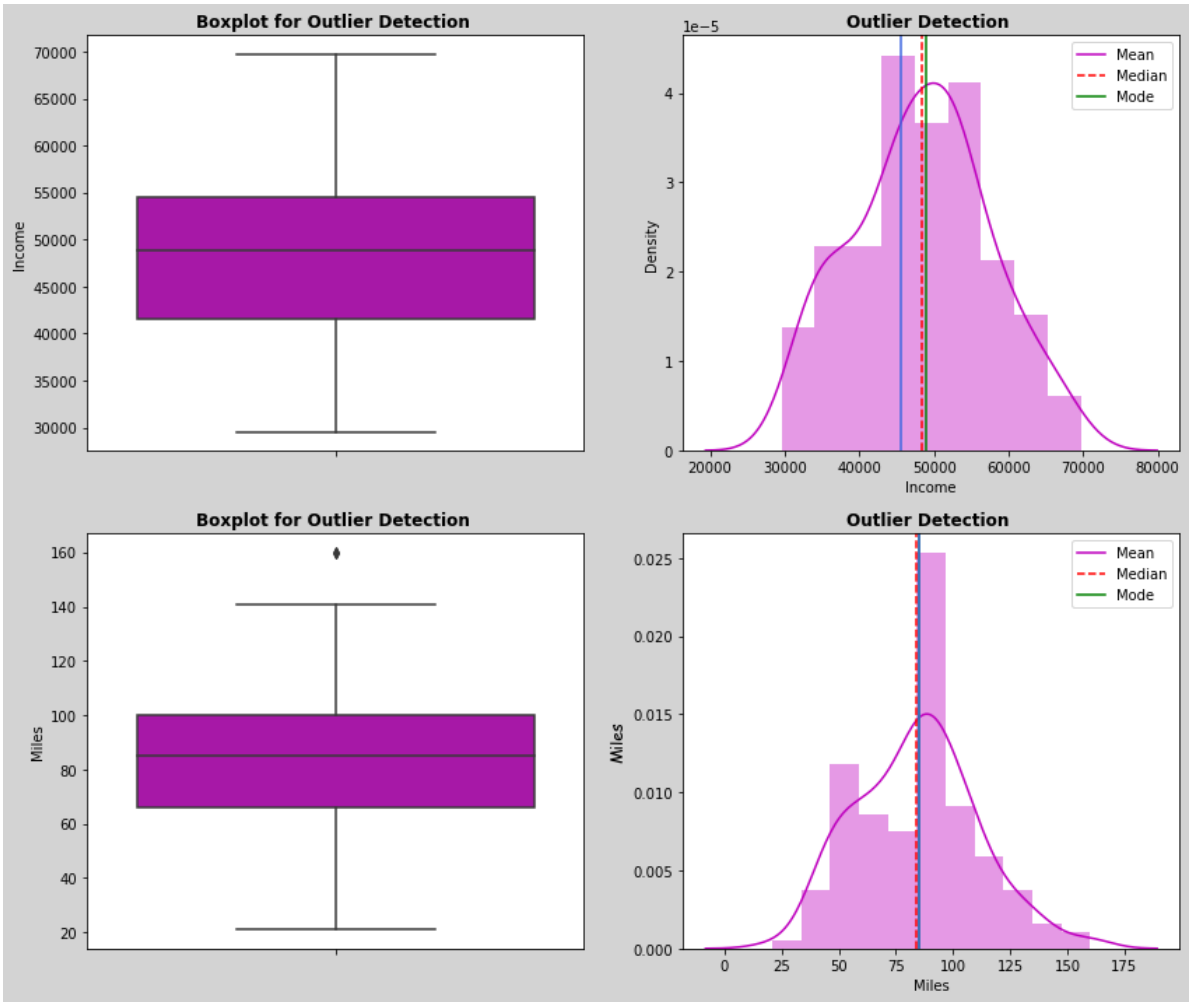
In [31]:

```
def outlier_detect(df,colname,nrows=2,mcols=2,width=20,height=15):
    fig , ax = plt.subplots(nrows,mcols,figsize=(width,height))
    fig.set_facecolor("lightgrey")
    rows = 0
    for var in colname:
        ax[rows][0].set_title("Boxplot for Outlier Detection ", fontweight="bold")
        plt.ylabel(var, fontsize=12,family = "Comic Sans MS")
        sns.boxplot(y = df[var],color='m',ax=ax[rows][0])

        # plt.subplot(nrows,mcols,pltcounter+1)
        sns.distplot(df[var],color='m',ax=ax[rows][1])
        ax[rows][1].axvline(df[var].mean(), color='r', linestyle='--', label="Mean")
        ax[rows][1].axvline(df[var].median(), color='g', linestyle='-', label="Median")
        ax[rows][1].axvline(df[var].mode()[0], color='royalblue', linestyle='-', label=
        ax[rows][1].set_title("Outlier Detection ", fontweight="bold")
        ax[rows][1].legend({'Mean':df[var].mean(),'Median':df[var].median(),'Mode':df[v
        rows += 1
    plt.show()
```

In [36]:

```
col_num = [ 'Income', 'Miles']
outlier_detect(new_data,col_num,2,2,14,12)
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
```



Observation

Both Miles and Income have significant outliers based on the above boxblot.

Also both are "right-skewed distribution" which means the mass of the distribution is concentrated on the left of the figure.

Majority of Customers fall within the USD 45,000 - USD 60,000 range

There are outliers over USD 85,000

Only a few of our customers run more than 180 miles per week

handling outliers

In [32]:

```
new_data = df.copy()
```

In [ ]:

```

```

Removing outliers for Income Feature

In [33]:

```
#Outlier Treatment: Remove top 5% & bottom 1% of the Column Outlier values
Q3 = new_data['Income'].quantile(0.75)
Q1 = new_data['Income'].quantile(0.25)
IQR = Q3-Q1
new_data = new_data[(new_data['Income'] > Q1 - 1.5*IQR) & (new_data['Income'] < Q3 + 1.
plt.show()
```

In [ ]:

```

```

Removing outliers for the Mile Feature

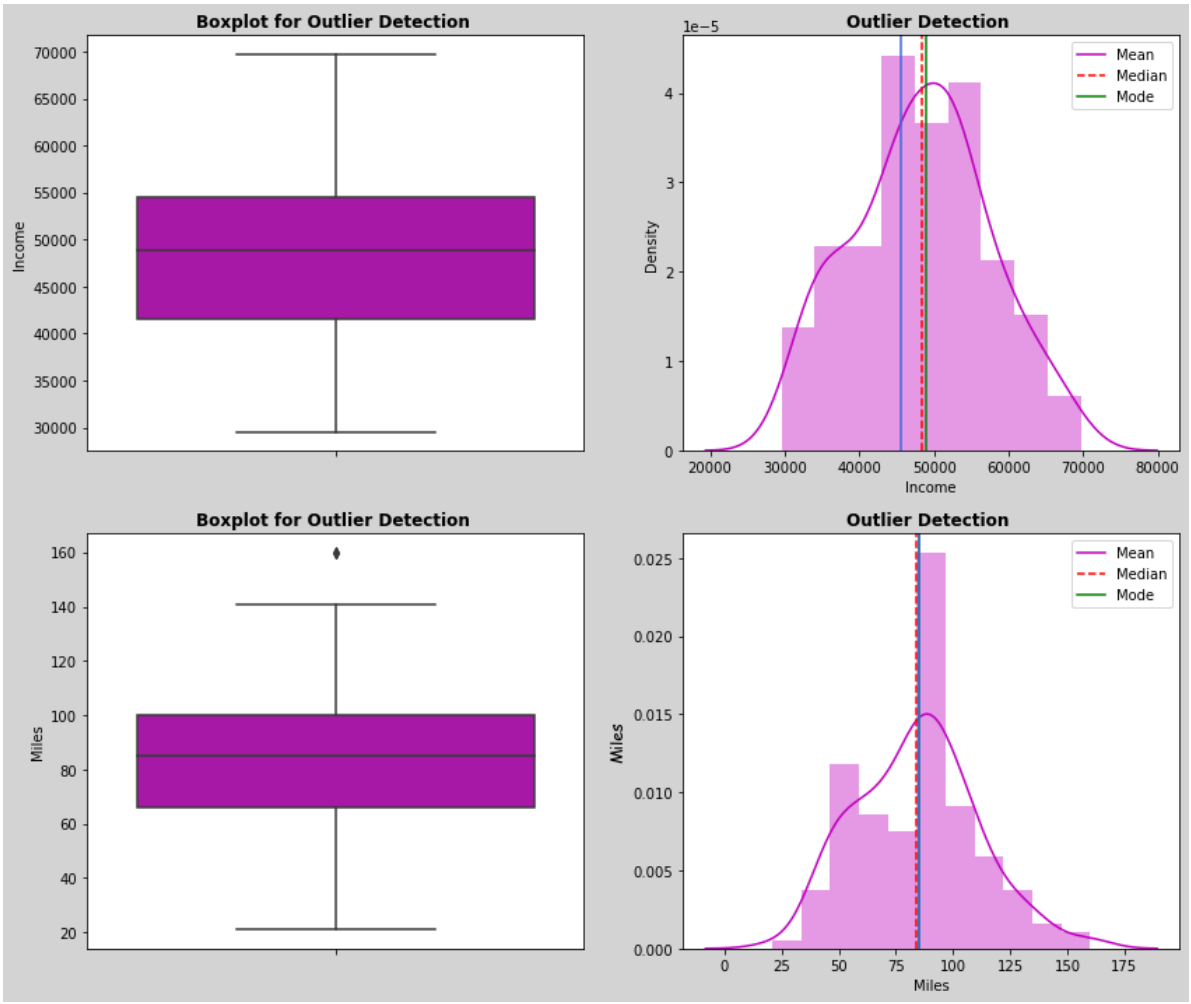
In [34]:

```
#Outlier Treatment: Remove top 5% & bottom 1% of the Column Outlier values
Q3 = new_data['Miles'].quantile(0.75)
Q1 = new_data['Miles'].quantile(0.25)
IQR = Q3-Q1
new_data = new_data[(new_data['Miles'] > Q1 - 1.5*IQR) & (new_data['Miles'] < Q3 + 1.5*
plt.show()
```

In [37]:

```
col_num = [ 'Income', 'Miles']
outlier_detect(new_data,col_num,2,2,14,12)
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
```



In [38]:

```
1 new_data.shape
```

Out[38]:

```
(147, 9)
```

Observation

It's true that there are outliers, but they may provide many insights for high-end models that can benefit companies more. Therefore, they should not be removed for further analysis.

Examine Data

In [16]:

```
1 #This is to look at what all unique values have . Just trying to use python
2 list_col=['Product','MaritalStatus','Usage','Fitness','Education','Age']
3 for col in list_col:
4     print('{} :{} ' . format(col.upper(),df[col].unique()))
```

```
PRODUCT :['KP281', 'KP481', 'KP781']
Categories (3, object): ['KP281', 'KP481', 'KP781']
MARITALSTATUS :['Single', 'Partnered']
Categories (2, object): ['Partnered', 'Single']
USAGE :[3 2 4 5 6 7]
FITNESS :[4 3 2 1 5]
EDUCATION :[14 15 12 13 16 18 20 21]
AGE :[18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 4
1
 43 44 46 47 50 45 48 42]
```

Observation:

There are 3 different treadmills products.

There are both Partnered and single customers

Age of customers ranges from 18 to 50

Education in years is from 12 -21

Usage is from 2 days to 7 days a week

Fitness level of customers from 1 -5

In [40]:

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

Out[40]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	
count	180	180.000000	180	180.000000	180	180.000000	180.000000	18
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	5371
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	1650
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	2956
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	4405
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	5059
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	5866
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	10458

Observation:

Age of customer using treadmill is between range 18 - 50 . Average age is 28.78 and median is 26.

Maximum income of treadmill user is 100K , Average income approx. 54K ,while median is is approx. 51K.

Expected Treadmill usage is atleast Once a week , maximum is 7 times a week and on Average 3 times a week

Customer education is between 12 -21 years, with average and median of 16 years and maximum of 21 years

Customer expects to runs on an average of 103.19 miles per week, median 94 miles per week.

Average self rated fitness is 3.

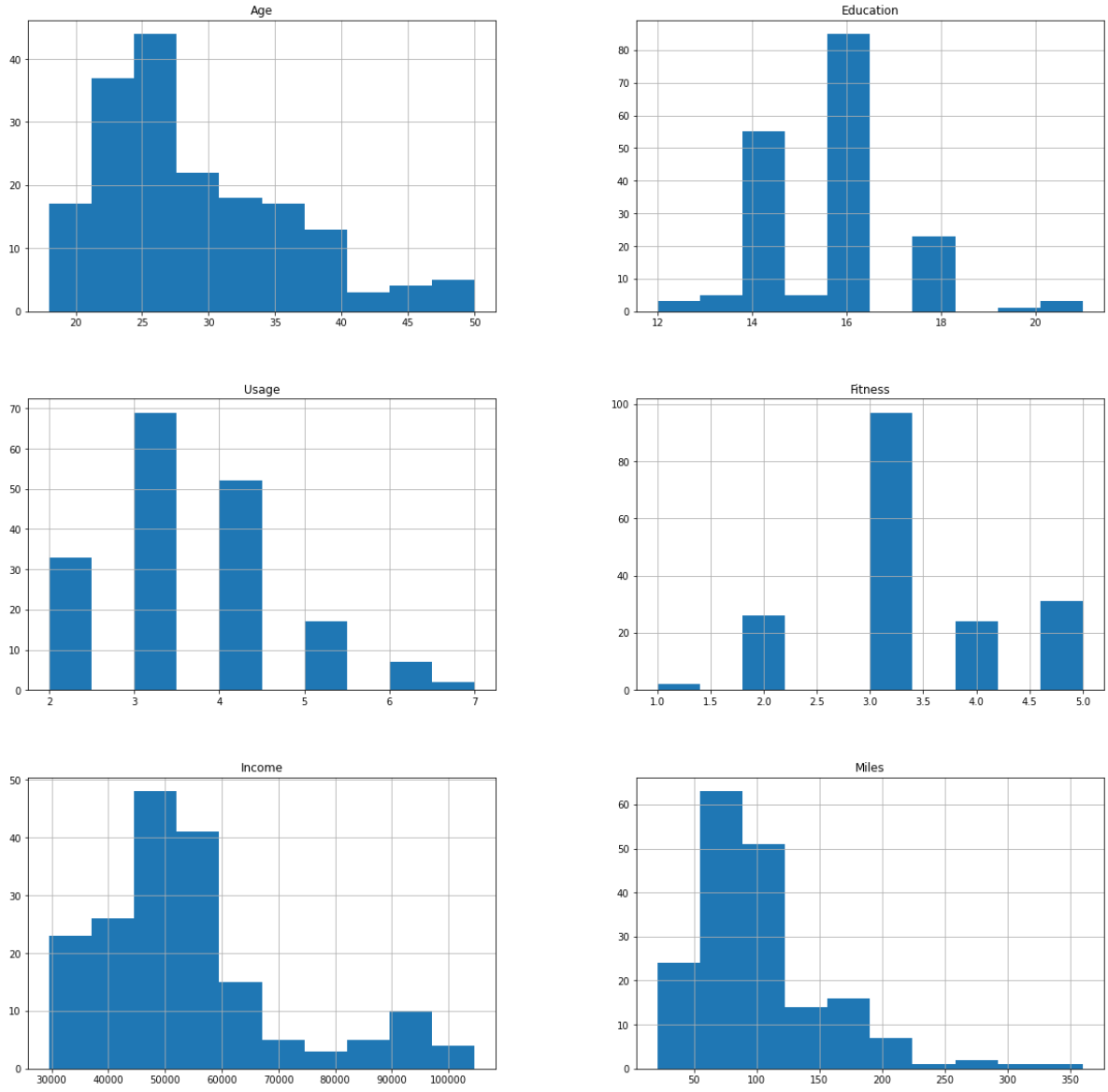
Histogram

In [41]:

```
1 df.hist(figsize=(20,20))
```

Out[41]:

```
array([[<AxesSubplot:title={'center':'Age'}>,
        <AxesSubplot:title={'center':'Education'}>],
       [<AxesSubplot:title={'center':'Usage'}>,
        <AxesSubplot:title={'center':'Fitness'}>],
       [<AxesSubplot:title={'center':'Income'}>,
        <AxesSubplot:title={'center':'Miles'}>]], dtype=object)
```



#Which is most sold Model?

In [42]:

```
1 df.Product.value_counts()
```

Out[42]:

```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

Observation:

KP281 treadmill model is most sold model.

In [ ]:

```
1
```

Are Male customers buying treadmill more than female customers?

In [43]:

```
1 df.Gender.value_counts()
```

Out[43]:

```
Male      104
Female     76
Name: Gender, dtype: int64
```

Observation:

There are 76 female and 104 males customers. More Male customers are buying treadmill compared to female customer

Are married customer buying Treadmill more than Single customers?

In [44]:

```
1 df.MaritalStatus.value_counts()
```

Out[44]:

```
Partnered    107
Single        73
Name: MaritalStatus, dtype: int64
```

Observation:

There are 107 Partnered and 73 single customers. Customers who are Partnered are buying treadmill more compared to single customer.

In [ ]:

```
1
```

In [45]:

```
1 df[df['Product'] == 'KP281'].describe().T
```

Out[45]:

	count	mean	std	min	25%	50%	75%	max
Age	80.0	28.5500	7.221452	18.0	23.0	26.0	33.0	50.0
Education	80.0	15.0375	1.216383	12.0	14.0	16.0	16.0	18.0
Usage	80.0	3.0875	0.782624	2.0	3.0	3.0	4.0	5.0
Fitness	80.0	2.9625	0.664540	1.0	3.0	3.0	3.0	5.0
Income	80.0	46418.0250	9075.783190	29562.0	38658.0	46617.0	53439.0	68220.0
Miles	80.0	82.7875	28.874102	38.0	66.0	85.0	94.0	188.0

Observation

80 customers bought KP281 model

Average age of customer who purchases KP281 is 28.5 , Median is 26 . Data is right skewed.

Average Education is 15 and median is 16.

Expected usage is 3 day a week

Expected Miles to run is on an Average 82.78 miles per week and median is 85.

Self rated fitness is 3 that is average fitness level

Average income and median is around $46K.

In [ ]:

```
1
```

In [46]:

```
1 df[df['Product'] ==  'KP481'].describe().T
```

Out[46]:

	count	mean	std	min	25%	50%	75%	max
Age	60.0	28.900000	6.645248	19.0	24.0	26.0	33.25	48.0
Education	60.0	15.116667	1.222552	12.0	14.0	16.0	16.00	18.0
Usage	60.0	3.066667	0.799717	2.0	3.0	3.0	3.25	5.0
Fitness	60.0	2.900000	0.629770	1.0	3.0	3.0	3.00	4.0
Income	60.0	48973.650000	8653.989388	31836.0	44911.5	49459.5	53439.00	67083.0
Miles	60.0	87.933333	33.263135	21.0	64.0	85.0	106.00	212.0

Observations

There are 60 customers who purchased KP481 Model

Average age of customer who purchases KP481 is 28.9 , Median is 26 . Age is right skewed. Customer range is between 24-33.

Average Education is 15 and median is 16.

Expected usage is 3 day a week

Expected Miles to run is on an Average 60 miles per week and median is 85.

Average Income is 48973.

Median Income is 49459

In [47]:

```
df[df['Product'] == 'KP781'].describe().T
```

Out[47]:

	count	mean	std	min	25%	50%	75%	max
Age	40.0	29.100	6.971738	22.0	24.75	27.0	30.25	48.0
Education	40.0	17.325	1.639066	14.0	16.00	18.0	18.00	21.0
Usage	40.0	4.775	0.946993	3.0	4.00	5.0	5.00	7.0
Fitness	40.0	4.625	0.667467	3.0	4.00	5.0	5.00	5.0
Income	40.0	75441.575	18505.836720	48556.0	58204.75	76568.5	90886.00	104581.0
Miles	40.0	166.900	60.066544	80.0	120.00	160.0	200.00	360.0

Observations

Average age of customer who purchases KP781 is 29 , Median is 27 .

Average Education is 17 and median is 18.

Expected usage is 4-5 day a week

Expected Miles to run is on an Average 166 miles per week and median is 160.

Average Income is 75K and median is 76K

In [ ]:

```

```

Let Visualize the Data and get more insights

Univariate Analysis

In [48]:

```
def analysis(data):
 # function plots a combined graph for univariate analysis of continous variable
 #to check spread, central tendency , dispersion and outliers
    Name=data.name.upper()
    fig, axes =plt.subplots(1,3,figsize=(17, 7))
    fig.suptitle("SPREAD OF DATA FOR "+ Name  , fontsize=18, fontweight='bold')
    sns.distplot(data,kde=False,color='Blue',ax=axes[0])
    axes[0].axvline(data.mean(), color='y', linestyle='--',linewidth=2)
    axes[0].axvline(data.median(), color='r', linestyle='dashed', linewidth=2)
    axes[0].axvline(data.mode()[0],color='g',linestyle='solid',linewidth=2)
    axes[0].legend({'Mean':data.mean(),'Median':data.median(),'Mode':data.mode()})
    sns.boxplot(x=data,showmeans=True, orient='h',color="purple",ax=axes[1])
    #just exploring violin plot
    sns.violinplot(data,ax=axes[2],showmeans=True)
```

In [49]:

```
analysis(df.Income)
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```

Observations:

Income is skewed towards right , Median is 50K , Mean is 55k and mode is $45K.

Most of the customers are in lower pay range and earn less than 70K.

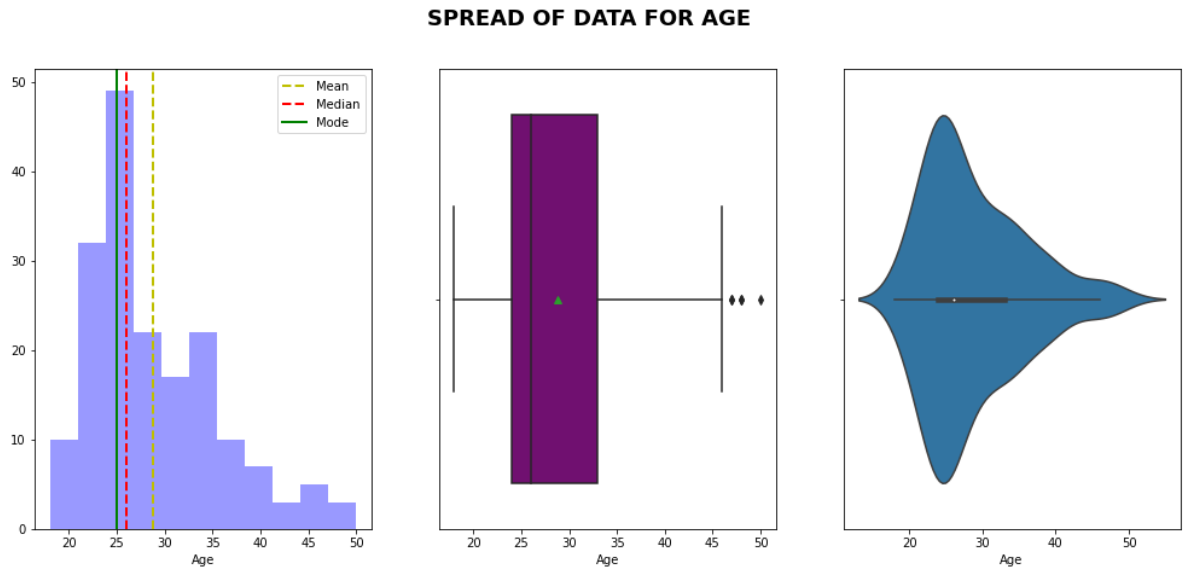
Income has some outliers. Few customers earn beyond 80K.

Type *Markdown* and LaTeX: α^2

In [50]:

```
1 analysis(df.Age)
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```



Observations:

Age is skewed towards right.

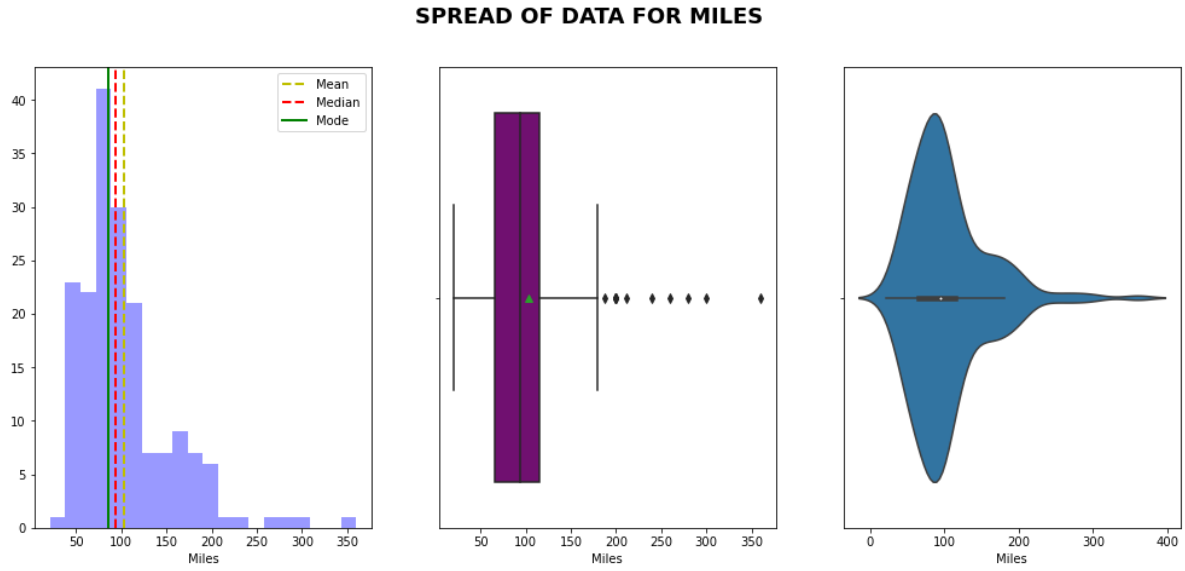
Customers buying treadmill are younger and average age of customer is 28 , median is 26 and mode is 25

Customers buying treadmill after age of 40 and before 20 are very less.

In [51]:

```
1 analysis(df.Miles)
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```



Observations:

Miles is skewed towards right.

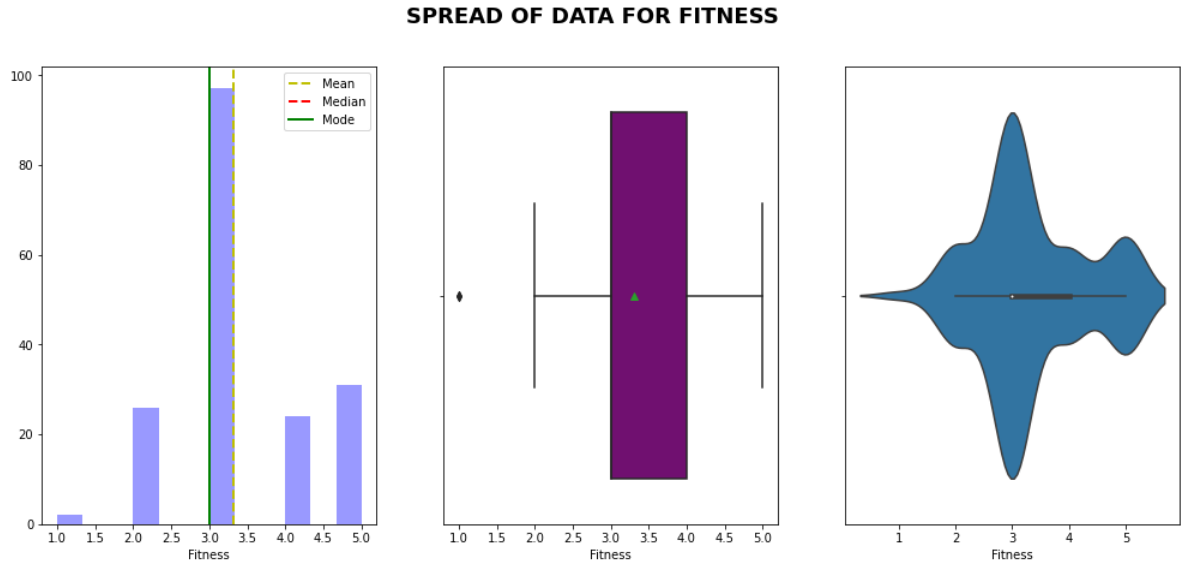
Customers expect to run on an average 80 miles per week.

There are some outliers, where customers are expecting to run more than 200 miles per weak.

In [52]:

```
1 analysis(df.Fitness)
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```



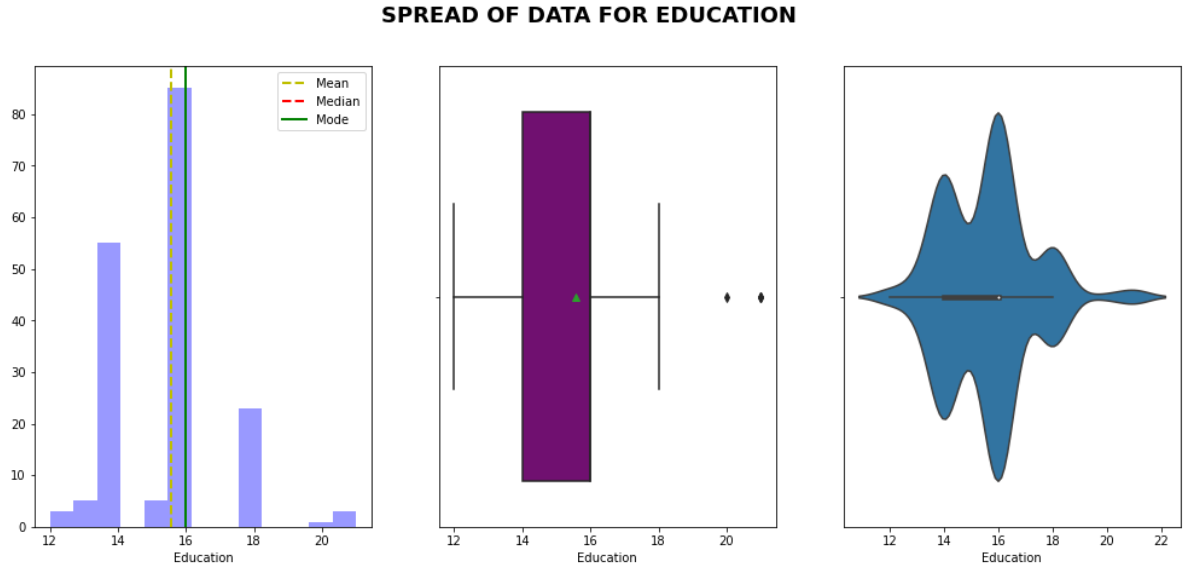
Observations

Most of the customers have self-rated their fitness as 3(average).

In [53]:

```
1 analysis(df.Education)
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```



Observations

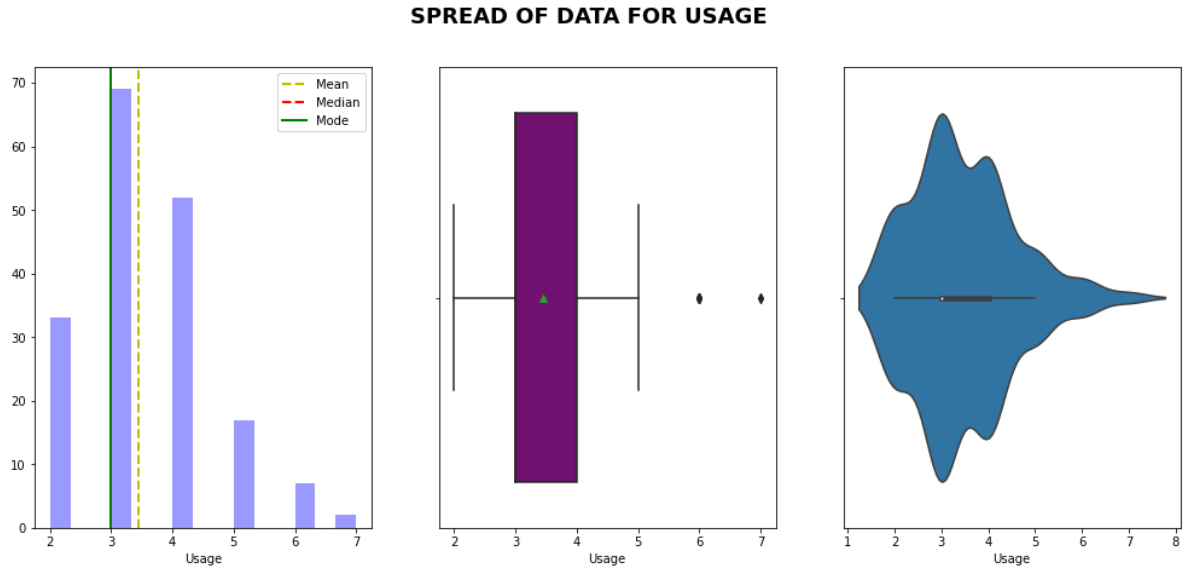
Most of the customers have 16 year of education (assuming them to be college graduates or bachelors).

There are few outliers.

In [54]:

```
1 analysis(df.Usage)
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\distributions.py:261
9: FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure-l
evel function with similar flexibility) or `histplot` (an axes-level functio
n for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```



Observations

Most of customers expect they will be using the treadmill 3-4 days per week.

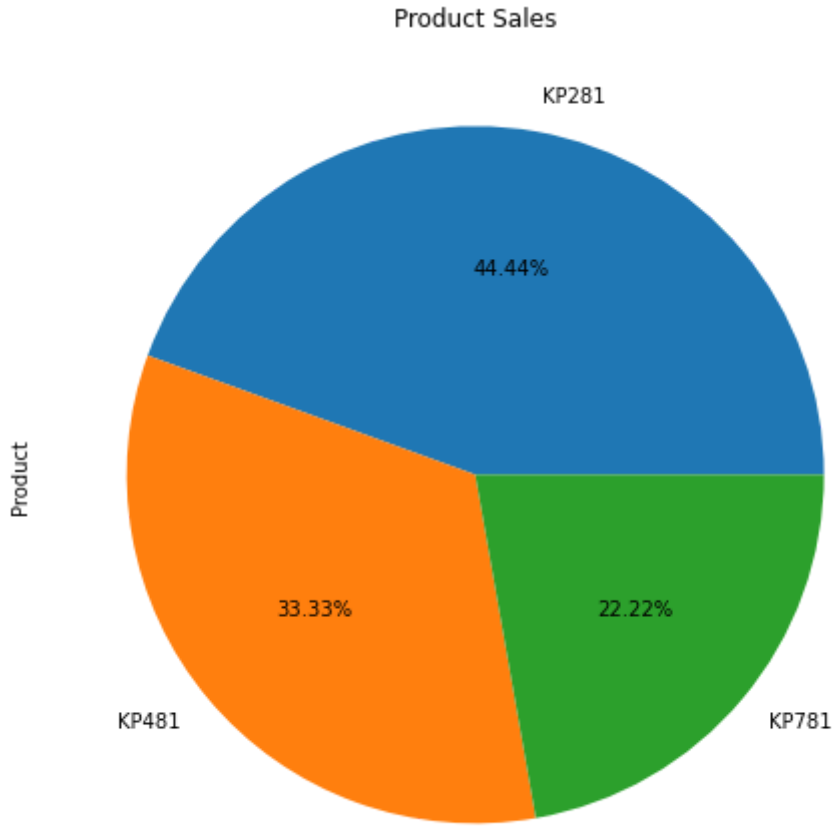
There are few outliers where customer are expecting to use treadmill for 6 or 7 times a week

Univariate Analysis

categorical variables

In [61]:

```
plt.figure(figsize=(10,5))
df['Product'].value_counts().plot.pie(autopct='%1.2f%%',figsize=(8,8))
plt.title("Product Sales")
plt.show()
```

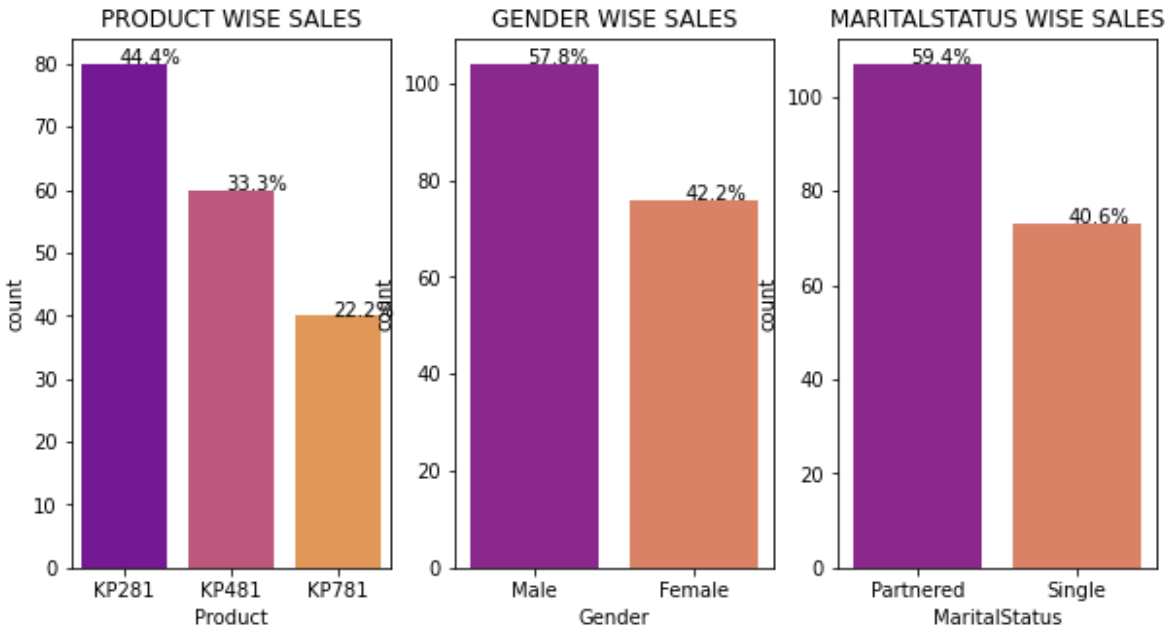


In [62]:

```
# Function to create barplots that indicate percentage for each category.
def bar(plot, feature):
    '''
    plot
    feature: 1-d categorical feature array
    '''
    total = len(feature) # length of the column
    for p in plot.patches:
        percentage = '{:.1f}%'.format(100 * p.get_height()/total) # percentage of each
        x = p.get_x() + p.get_width() / 2 - 0.05 # width of the plot
        y = p.get_y() + p.get_height()           # hieght of the plot
        plot.annotate(percentage, (x, y), size = 10) # annotate the percentage
```

In [63]:

```
fig1, axes1 =plt.subplots(1,3,figsize=(10, 5))
list_col=['Product','Gender','MaritalStatus']
j=0
for i in range(len(list_col)):
    order = df[list_col[i]].value_counts(ascending=False).index # to display bar in asc
    axis=sns.countplot(x=list_col[i], data=df , order=order,ax=axes1[i],palette='plasma
    bar(axes1[i],df[list_col[i]])
```



Observation:

44.4% customers brought KP281. KP281 model is the most purchased model. KP481 was purchased more than KP781.

57.8% male brought Treadmill.There are more Male customers than Female customers.

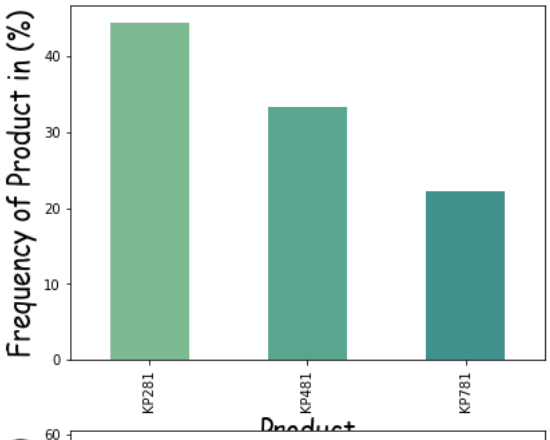
59.4% of the customers who purchased treadmill are Married.

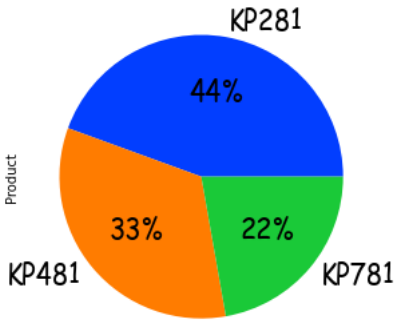
In [96]:

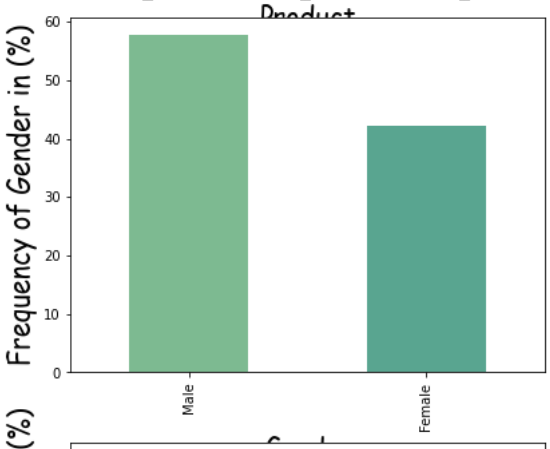
```
# Frequency of each feature in percentage.
def cat_analysis(df, colnames, nrows=2,mcols=2,width=20,height=30, sortbyindex=False):
    fig , ax = plt.subplots(nrows,mcols,figsize=(width,height))
    fig.set_facecolor(color = 'white')
    string = "Frequency of "
    rows = 0
    for colname in colnames:
        count = (df[colname].value_counts(normalize=True)*100)
        string += colname + ' in (%)'
        if sortbyindex:
                count = count.sort_index()
        count.plot.bar(color=sns.color_palette("crest"),ax=ax[rows][0])
        ax[rows][0].set_ylabel(string, fontsize=20,family = "Comic Sans MS")
        ax[rows][0].set_xlabel(colname, fontsize=20,family = "Comic Sans MS")
        count.plot.pie(colors = sns.color_palette("bright"),autopct='%0.0f%%',
                       textprops={'fontsize': 20,'family':"Comic Sans MS"},ax=ax[rows]|
        string = "Frequency of "
        rows += 1
```

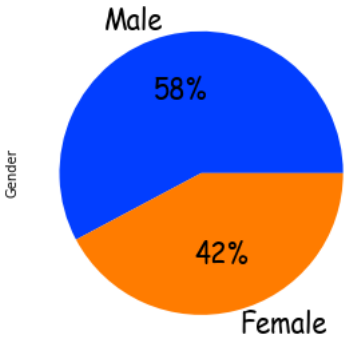
In [97]:

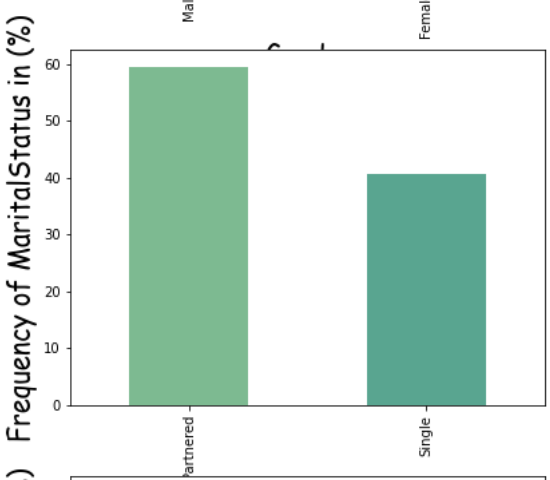
```
cat_colnames = ['Product', 'Gender', 'MaritalStatus', 'AgeGroup', 'AgeCategory','Income
cat_analysis(df,cat_colnames,7,2,14,40)
```

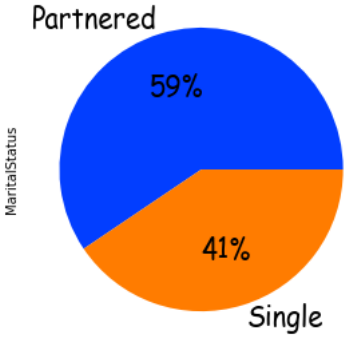


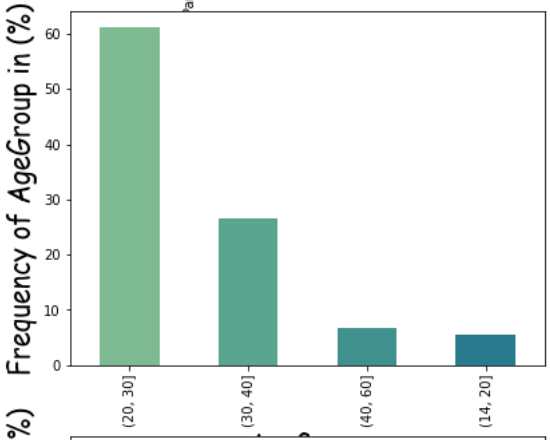


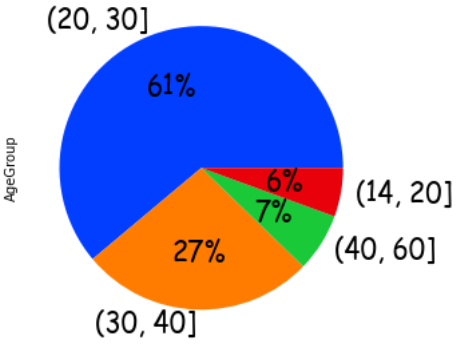


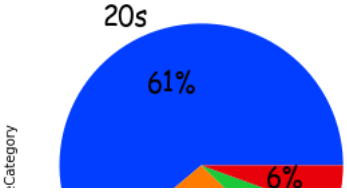


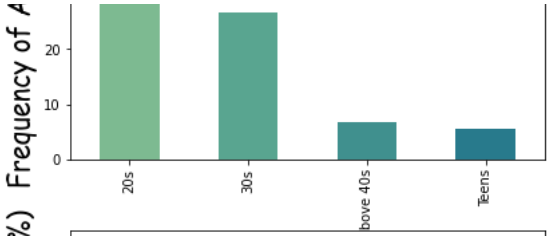


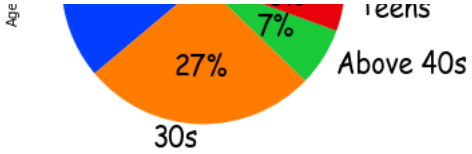


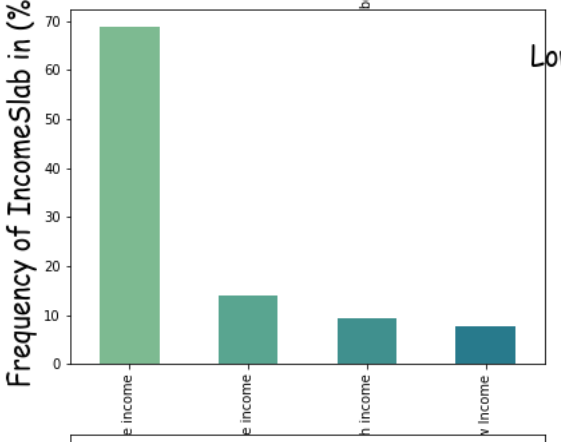


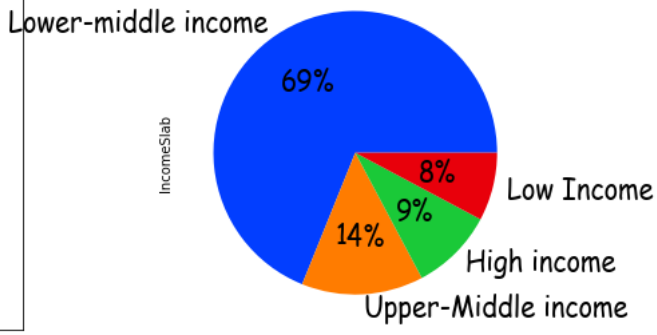


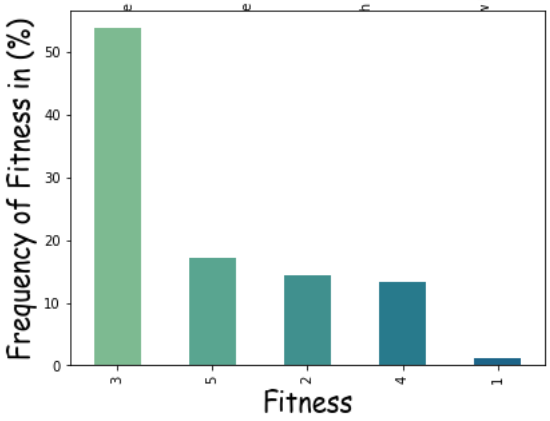


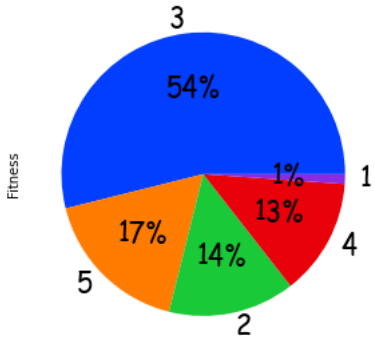












Observation

83% of treadmills are bought by customers with incomes between USD dollars 35000-60000, and USD dollars 60,000-85000.

88% of treadmills are purchased by customers aged 20 to 40.

The treadmills are more likely to be purchased by married people

Model KP281 is the best-selling product

Customer with fitness level 3 buy major chuck of treadmills. (54%)

Bi variate analysis

In [99]:

```
#Average age of customer buying each model
df.groupby('Product')['Age'].mean()
```

Out[99]:

```
Product
KP281    28.55
KP481    28.90
KP781    29.10
Name: Age, dtype: float64
```

In [100]:

```
#Average Income of customer buying each model
df.groupby('Product')['Income'].mean()
```

Out[100]:

```
Product
KP281    46418.025
KP481    48973.650
KP781    75441.575
Name: Income, dtype: float64
```

In [101]:

```
#Average Income of customer buying each model
df.groupby('Product')['Miles'].mean()
```

Out[101]:

```
Product
KP281     82.787500
KP481     87.933333
KP781    166.900000
Name: Miles, dtype: float64
```

In [103]:

```
plt.figure(figsize=(10,10))
prd_gender=pd.crosstab(df['Product'],df['Gender'] )
print(prd_gender)

ax=prd_gender.plot(kind='bar')

plt.title("PRODUCT BY GENDER")
```

. . .

Observation

KP281 model was equally bought my Male and Female

Compared to females, male bought KP481 model .

KP781 model is popular in Males than in female.

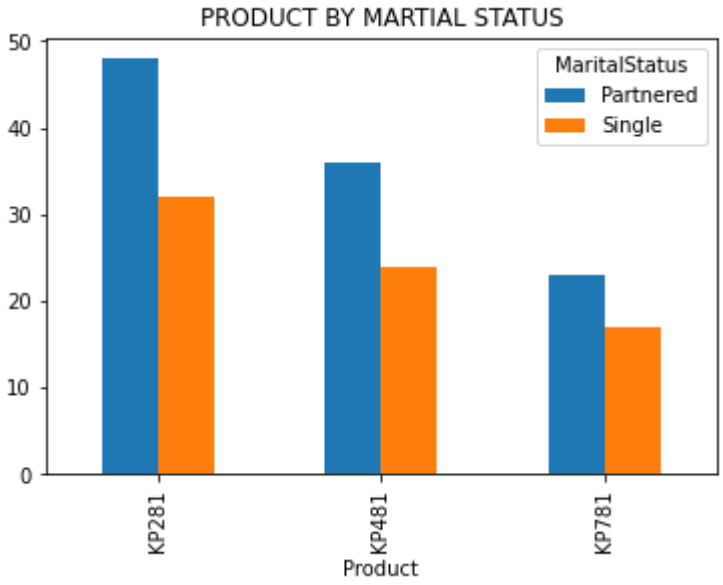
In [104]:

```
prd_mar_status=pd.crosstab(df['Product'],df['MaritalStatus'] )
print(prd_mar_status)
prd_mar_status.plot(kind='bar')
plt.title("PRODUCT BY MARTIAL STATUS")
```

```
MaritalStatus  Partnered  Single
Product
KP281                 48      32
KP481                 36      24
KP781                 23      17
```

Out[104]:

```
Text(0.5, 1.0, 'PRODUCT BY MARTIAL STATUS')
```

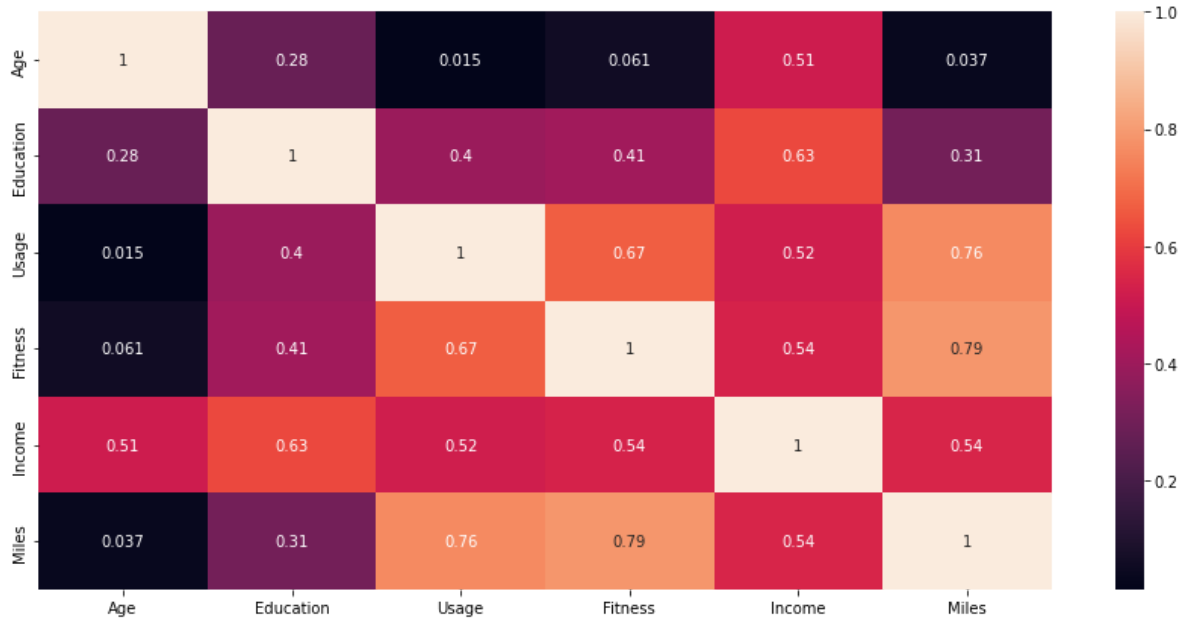


In [105]:

```
plt.figure(figsize=(15,7))
sns.heatmap(df.corr(), annot=True)
```

Out[105]:

<AxesSubplot:>



In [41]:

```
corr_pairs = df.corr().unstack() # give pairs of correlation
print( corr_pairs[abs(corr_pairs)>0.5]) # Gives us correlated data
```

```
Age        Age          1.000000
           Income       0.513414
Education  Education    1.000000
           Income       0.625827
Usage      Usage        1.000000
           Fitness      0.668606
           Income       0.519537
           Miles        0.759130
Fitness    Usage        0.668606
           Fitness      1.000000
           Income       0.535005
           Miles        0.785702
Income     Age          0.513414
           Education    0.625827
           Usage        0.519537
           Fitness      0.535005
           Income       1.000000
           Miles        0.543473
Miles      Usage        0.759130
           Fitness      0.785702
           Income       0.543473
           Miles        1.000000
dtype: float64
```

Observation

Age and Income has some in significant correlation

Education and Income has very little correlation

There is some corelation between Usage and Income

Fitness and miles are corelated

KP781 model is correlated to Education, Usage,Fitness, Income and Miles.

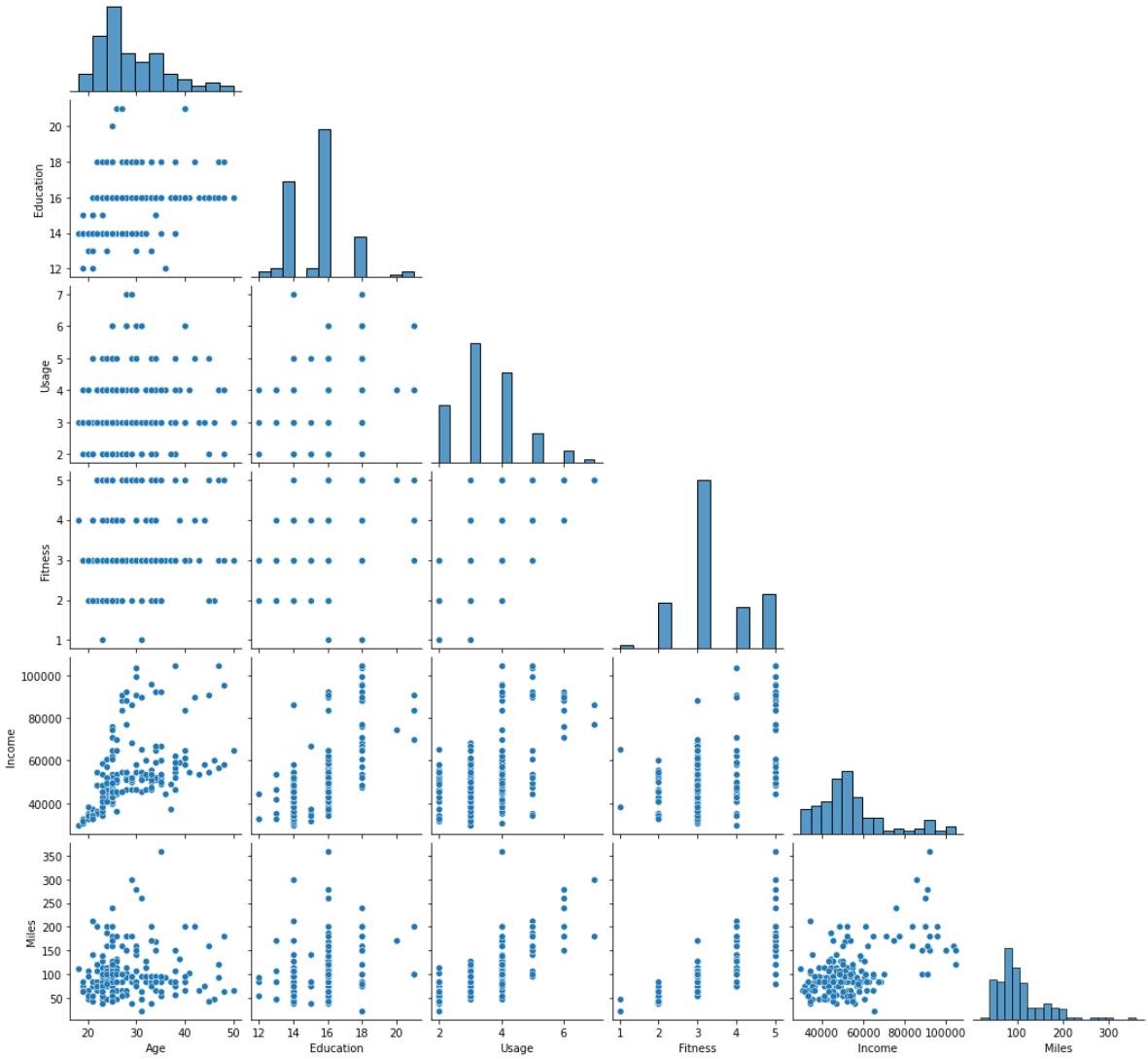
Miles and usage are positively correlated

In [106]:

```
1 plt.figure(figsize=(15,7))
2 sns.pairplot(data=df,corner=True)
```

Out[106]:

```
<seaborn.axisgrid.PairGrid at 0x1f54a039a00>

<Figure size 1080x504 with 0 Axes>
```

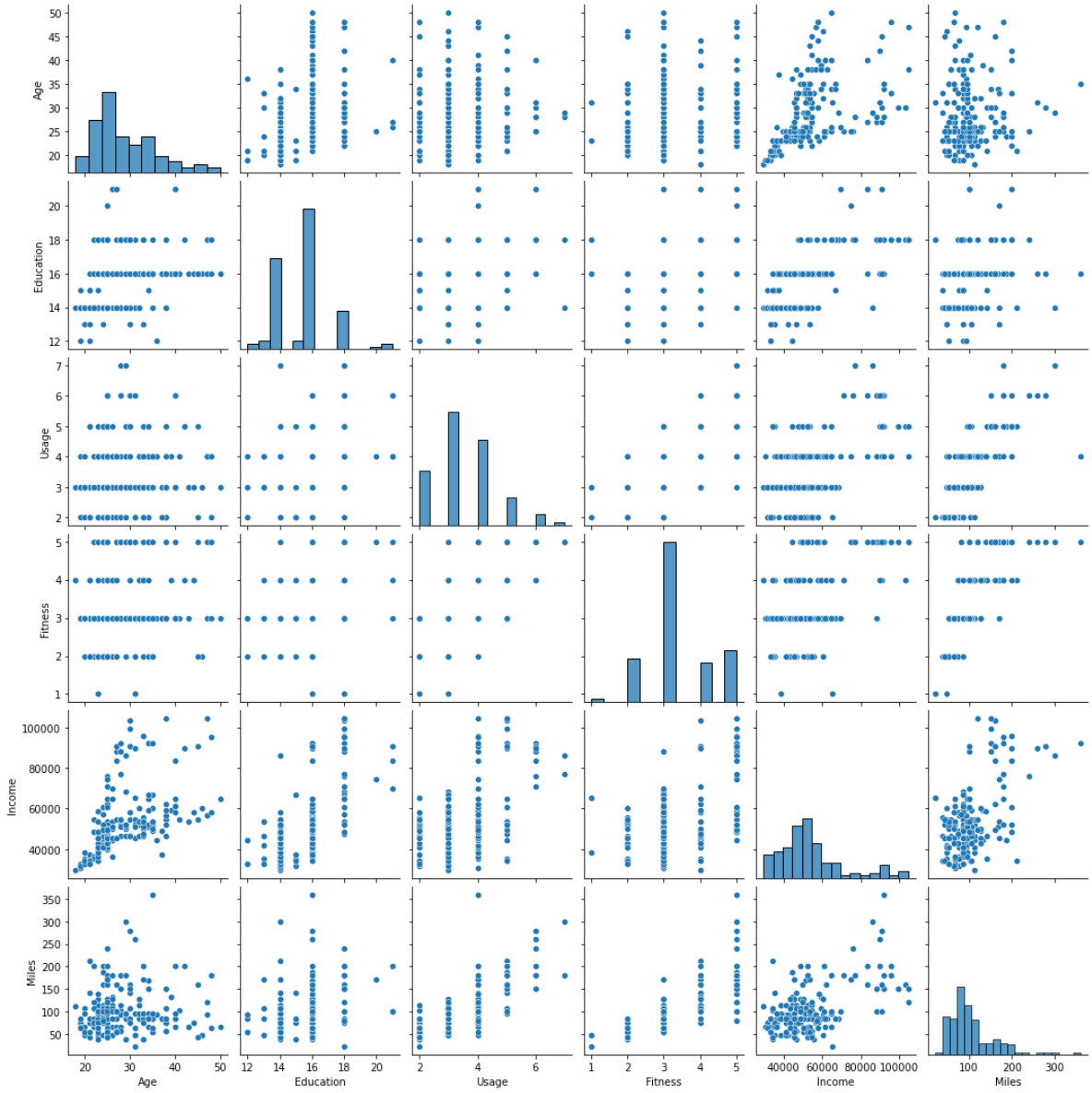


In [43]:

```
1 sns.pairplot(df)
```

Out[43]:

```
<seaborn.axisgrid.PairGrid at 0x28c690c25e0>
```



Observation:-

we get the same observation as from the correlation plot

Bi Varaite Analysis for

1.Product & Age

2.Product & Income

3.Product & Education

4.Product & Usage

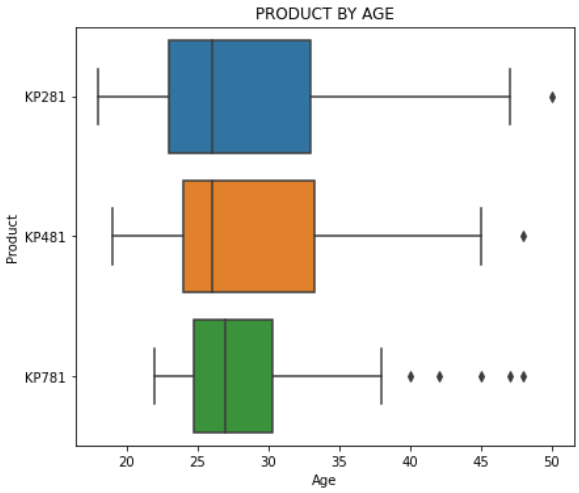
5.Product & Fitness

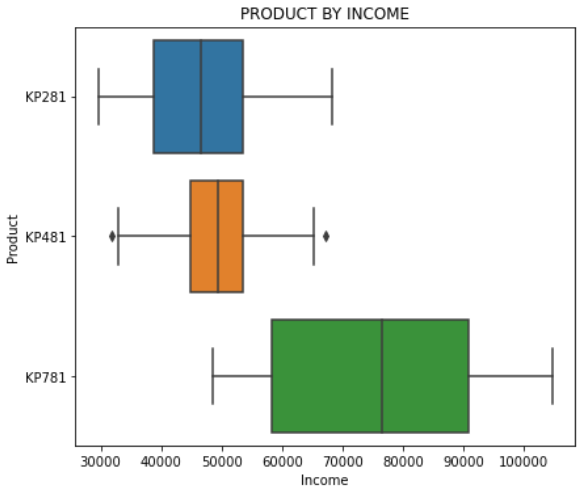
6.Product & Miles

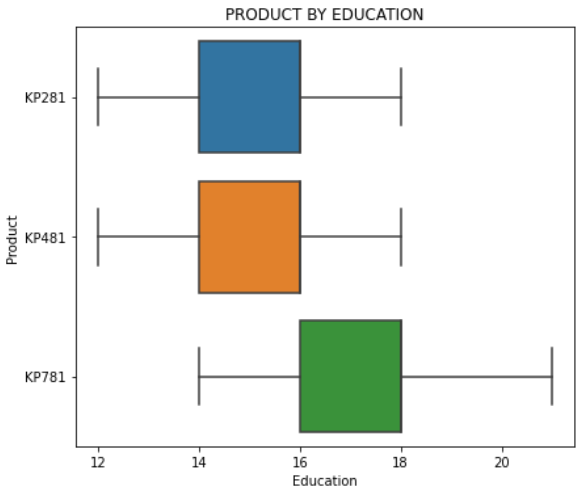
In [107]:

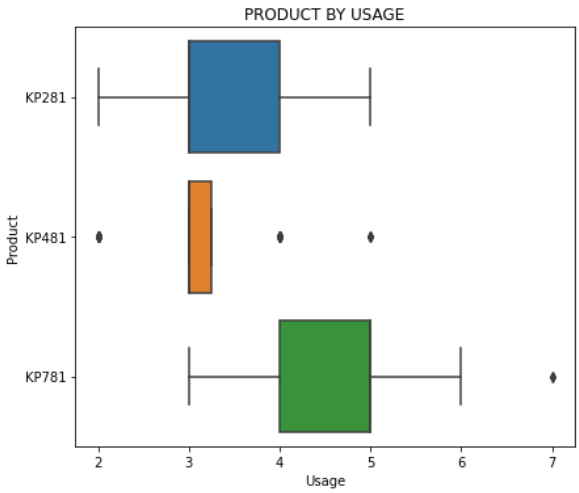
```
fig1, axes1 =plt.subplots(3,2,figsize=(15, 20))
list1_col=['Age','Income','Education','Usage','Fitness','Miles']
#instead of writing  boxplot 6 times using for loop
for i in range(len(list1_col)):
    row=i//2
    col=i%2
    ax=axes1[row,col]

    sns.boxplot(df[list1_col[i]],df['Product'],ax=ax).set(title='PRODUCT BY ' + list1_c
```

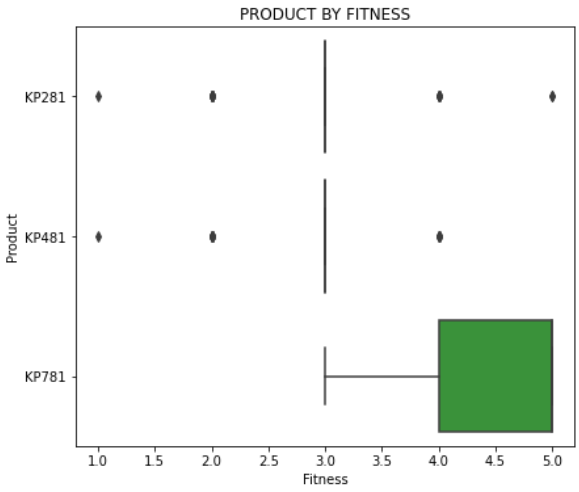
```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
```

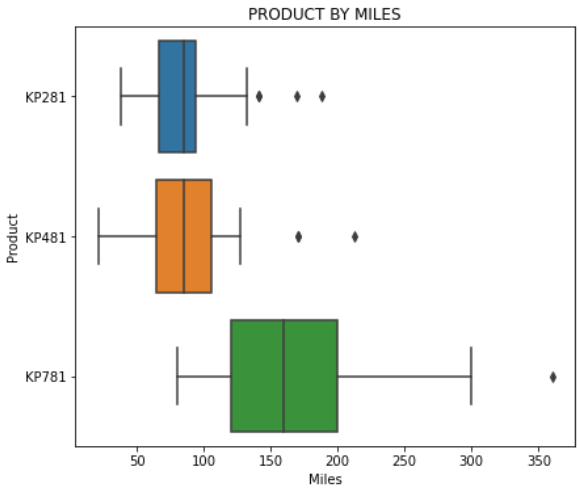

PRODUCT BY AGE
KP281
KP481
KP781
Product
20
25
30
35
40
45
50
Age


PRODUCT BY INCOME
KP281
KP481
KP781
Product
30000
40000
50000
60000
70000
80000
90000
100000
Income


PRODUCT BY EDUCATION
KP281
KP481
KP781
Product
12
14
16
18
20
Education


PRODUCT BY USAGE
KP281
KP481
KP781
Product
2
3
4
5
6
7
Usage


PRODUCT BY FITNESS
KP281
KP481
KP781
Product
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Fitness


PRODUCT BY MILES
KP281
KP481
KP781
Product
50
100
150
200
250
300
350
Miles

Observations:-

There are many outliers for KP781 ,customers are more than age of 40 .

Age of customers buying KP281 and KP481 is between 20-35, where as customers buying KP781 are primarily in 25-30

Customers with higher income and more education have purchased KP781 model.

Customers with lower income purchase KP281 and TM498 model may be because of cost of the Treadmill

Customer with KP281 expect to use treadmill 3-4 times a week and have average self rated fitness as 3 and some unfits.

Customers who bought KP481 model expecting to use Treadmill less frequently but to run more miles a week.

Customer buying KP781 plan to use it more frequently , run more miles and have high self rated fitness .They seem to be more health conscious or professionals.

KP 781 model was purchased more by males customer than female customers .

More partnered customer tend to buy KP781 than Single customers

Bi Varaite Analysis for

1.Gender & Age

2.Gender & Income

3.Gender & Education

4.Gender & Usage

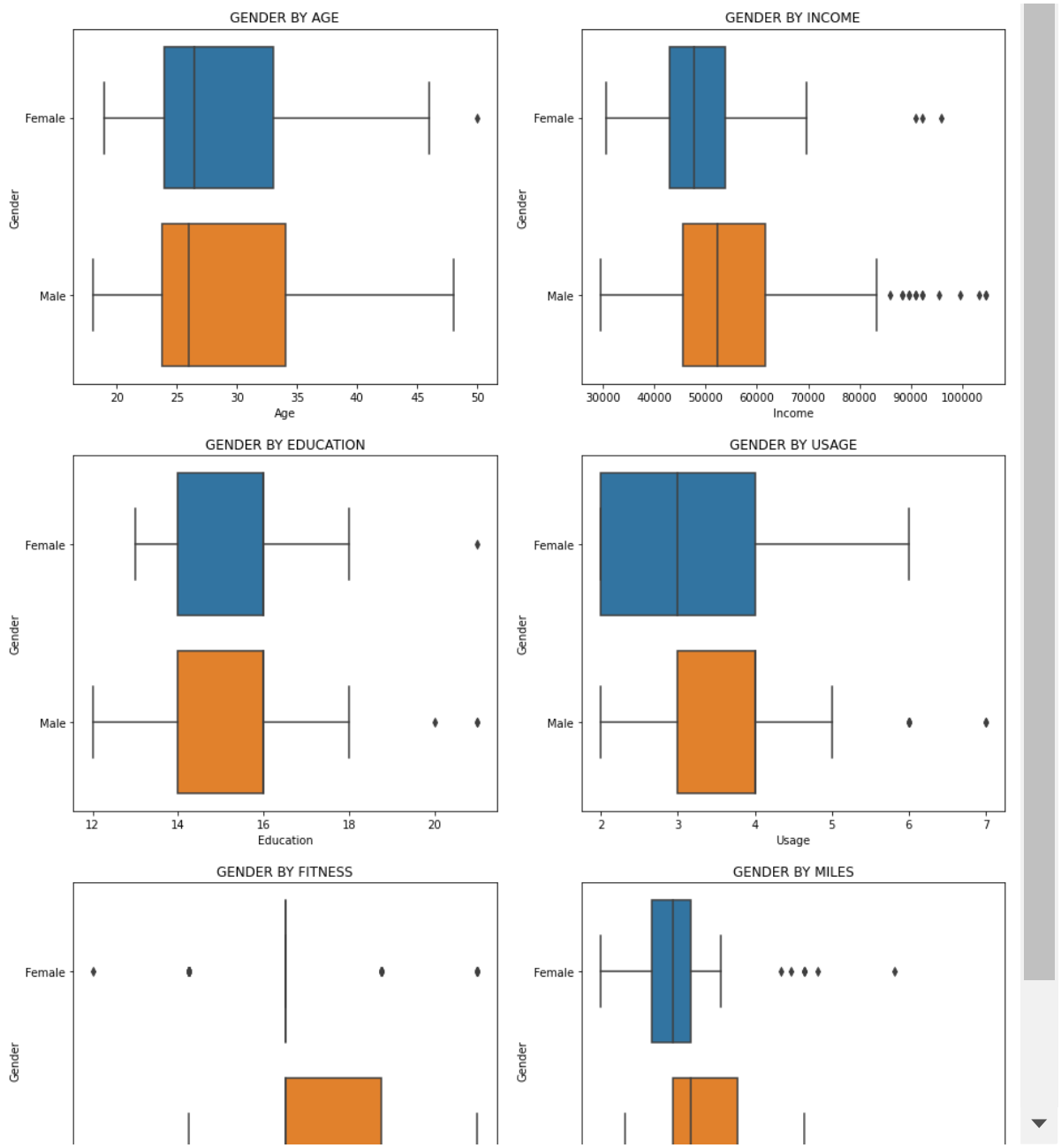
5.Gender & Fitness

6.Gender & Miles

In [109]:

```
fig1, axes1 =plt.subplots(3,2,figsize=(15, 20))
list1_col=['Age','Income','Education','Usage','Fitness','Miles']
# to plot graph side by side.
for i in range(len(list1_col)):
    row=i//2
    col=i%2
    ax=axes1[row,col]
    sns.boxplot(df[list1_col[i]],df['Gender'],ax=ax).set(title='GENDER BY ' + list1_co
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
```



Observations:-

Male customers earn more than Female customers.

Males Customer have higher self rated fitness than female.

Expected Usage and miles covered on tread mill is less in Female customers than male customers.

Female in age range 23-33 purchased the treadmill.

Education of Male and Female customers is same.

In [ ]:

```
1
```

Bi Varaite Analysis for

1.Martial Status & Age

2.Martial Status & Income

3.Martial Status & Education

4.Martial Status & Usage

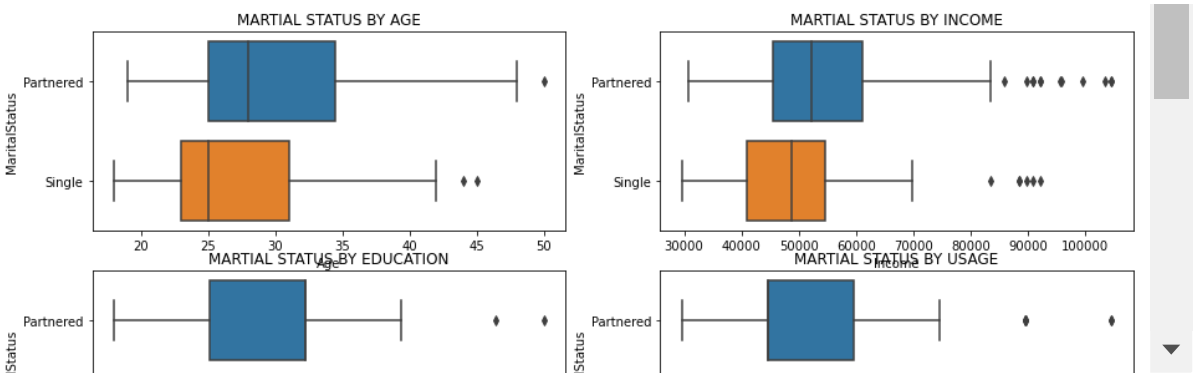
5.Martial Status & Fitness

6.Martial Status & Miles

In [110]:

```python
plt.figure(figsize=(7,7))
fig1, axes1 =plt.subplots(3,2,figsize=(15, 10))
list1_col=['Age','Income','Education','Usage','Fitness','Miles']
for i in range(len(list1_col)):
    row=i//2
    col=i%2
    ax=axes1[row,col]
    sns.boxplot(df[list1_col[i]],df['MaritalStatus'],ax=ax).set(title='MARTIAL STATUS
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variables as keyword args: x, y. From versio
n 0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or misinterpre
tation.
  warnings.warn(

<Figure size 504x504 with 0 Axes>
```



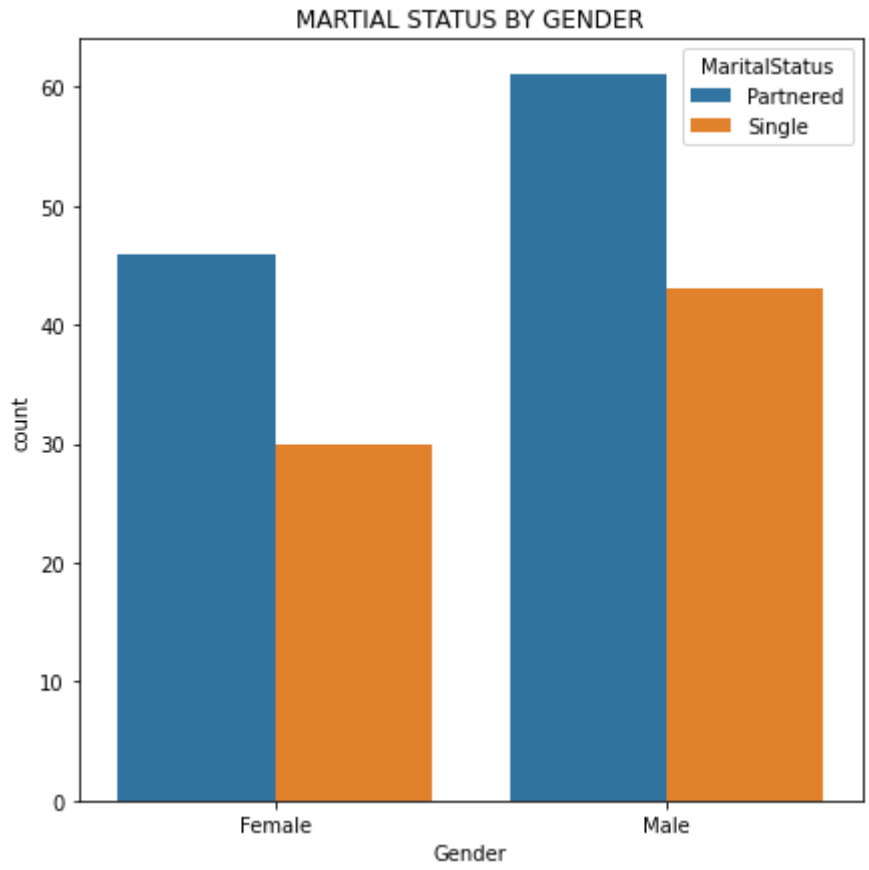
In [111]:

```
1 plt.figure(figsize=(7,7))
2 sns.countplot(df['Gender'],hue=df["MaritalStatus"]).set(title='MARTIAL STATUS BY GENDER
```

```
C:\Users\Shelendra\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Fu
tureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other a
rguments without an explicit keyword will result in an error or misinterpret
ation.
  warnings.warn(
```

Out[111]:

```
[Text(0.5, 1.0, 'MARTIAL STATUS BY GENDER')]
```



Observations

Partnered customer expects to run more miles compared to single

Income of Partnered customer is more than income of single customer.

Age of Partnered customer is more than Age of single customer

There are more single males buying Treadmill than single Females

Self rated Fitness of both Partnered and Single customer are same.

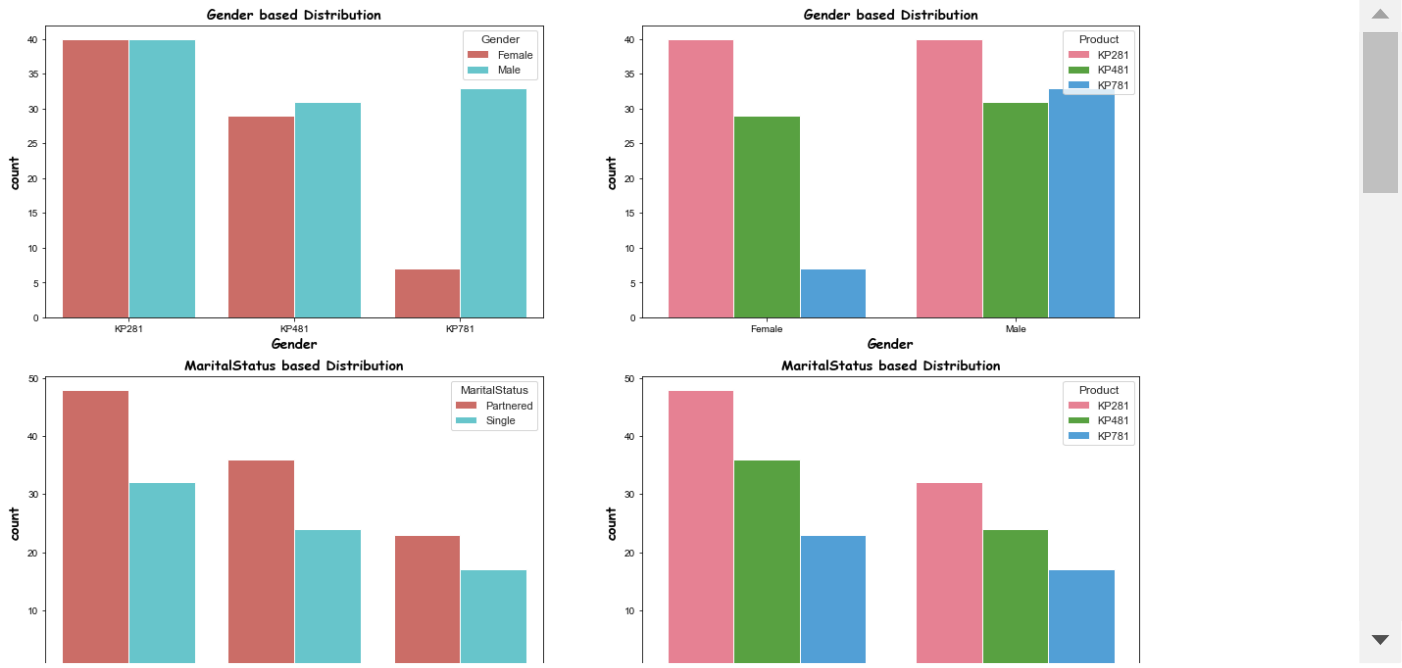
Education of both Partnered and Single customer is same

In [112]:

```
def cat_bi_analysis(df,colname,depend_var,nrows=2,mcols=2,width=20,height=15):
    fig , ax = plt.subplots(nrows,mcols,figsize=(width,height))
    sns.set(style='white')
    rows = 0
    string = " based Distribution"
    for var in colname:
        string = var + string
        sns.countplot(data=df,x=depend_var, hue=var, palette="hls",ax=ax[rows][0])
        sns.countplot(data=df, x=var, hue=depend_var, palette="husl",ax=ax[rows][1])
        ax[rows][0].set_title(string, fontweight="bold",fontsize=14,family = "Comic Sar
        ax[rows][1].set_title(string, fontweight="bold",fontsize=14,family = "Comic Sar
        ax[rows][0].set_ylabel('count', fontweight="bold",fontsize=14,family = "Comic S
        ax[rows][0].set_xlabel(var,fontweight="bold", fontsize=14,family = "Comic Sans
        ax[rows][1].set_ylabel('count', fontweight="bold",fontsize=14,family = "Comic S
        ax[rows][1].set_xlabel(var,fontweight="bold", fontsize=14,family = "Comic Sans
        rows += 1
        string = " based Distribution"
    plt.show()
```

In [113]:

```
col_names = ['Gender', 'MaritalStatus', 'AgeGroup', 'AgeCategory','IncomeSlab','Fitness
cat_bi_analysis(df,col_names,'Product',7,2,20,45)
```



Observation

Gender

KP781 model is the most popular among males

KP281 is equally preferred by men and women

AgeCategory

The most useful treadmills product for people over 40s is the KP281 & KP781.However, they buy fewer treadmills.

Income

Customer with high income only buy high end model. (KP781)

##Fitness Level

Customers with 5 fitness level prefer using KP781.(High end Model)

With moderate fitness level , customer prefer using KP281.

Education

Customer above 20 years education, purchase only KP781 model.

The other categorical features show no specific trends.

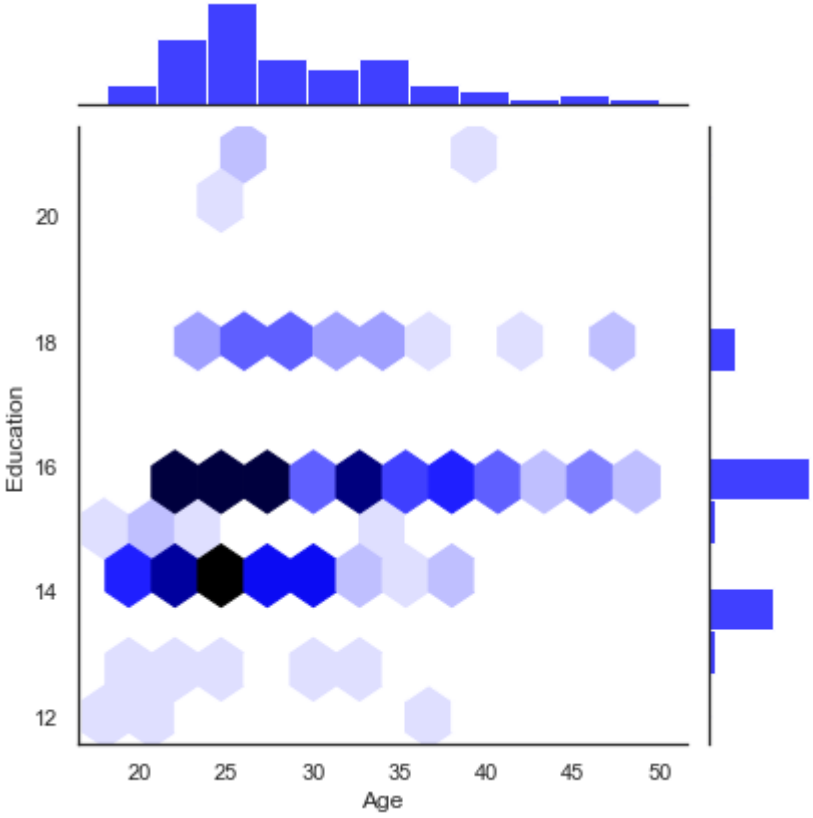
Bivariate Analysis Age & Education

In [115]:

```
1
2 sns.jointplot(x = 'Age',y = 'Education',data = df,color="blue",kind='hex')
```

Out[115]:

```
<seaborn.axisgrid.JointGrid at 0x1f54e02a5e0>
```



Observation:-

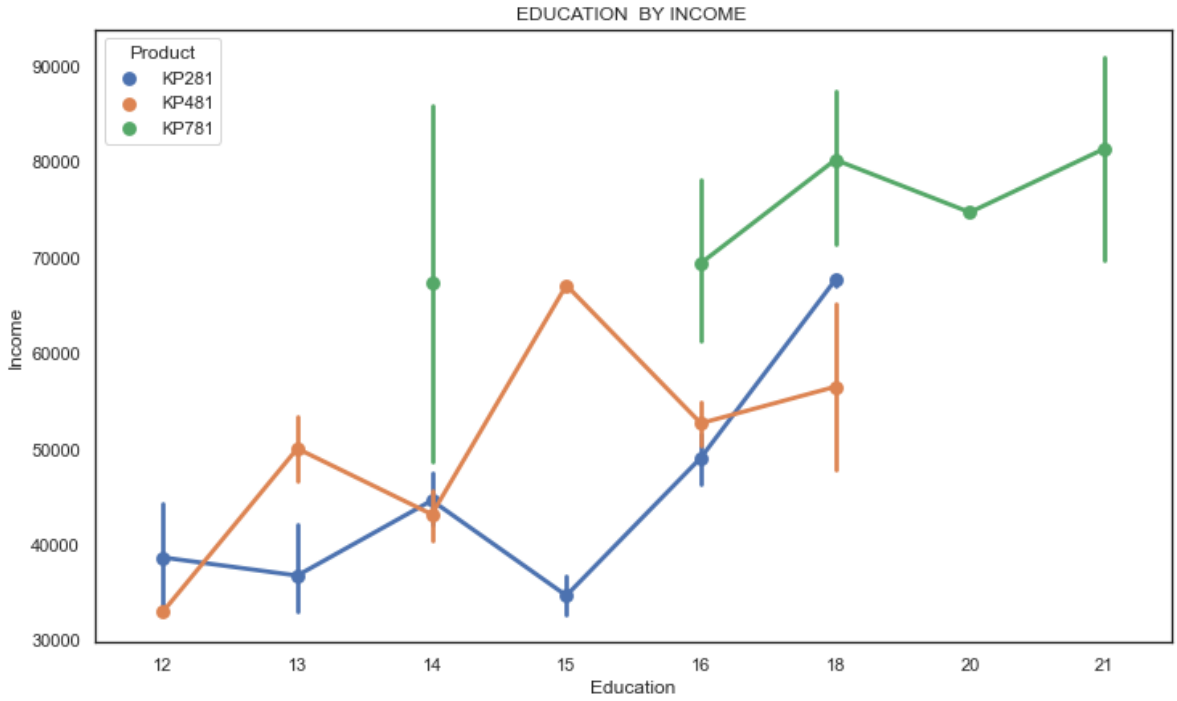
Customer between age 20-40 have 14 -16 years of education

In [116]:

```
plt.figure(figsize=(12,7))
sns.pointplot(x=df["Education"],y=df["Income"],hue=df['Product']).set(title='EDUCATION
```

Out[116]:

```
[Text(0.5, 1.0, 'EDUCATION  BY INCOME ')]
```



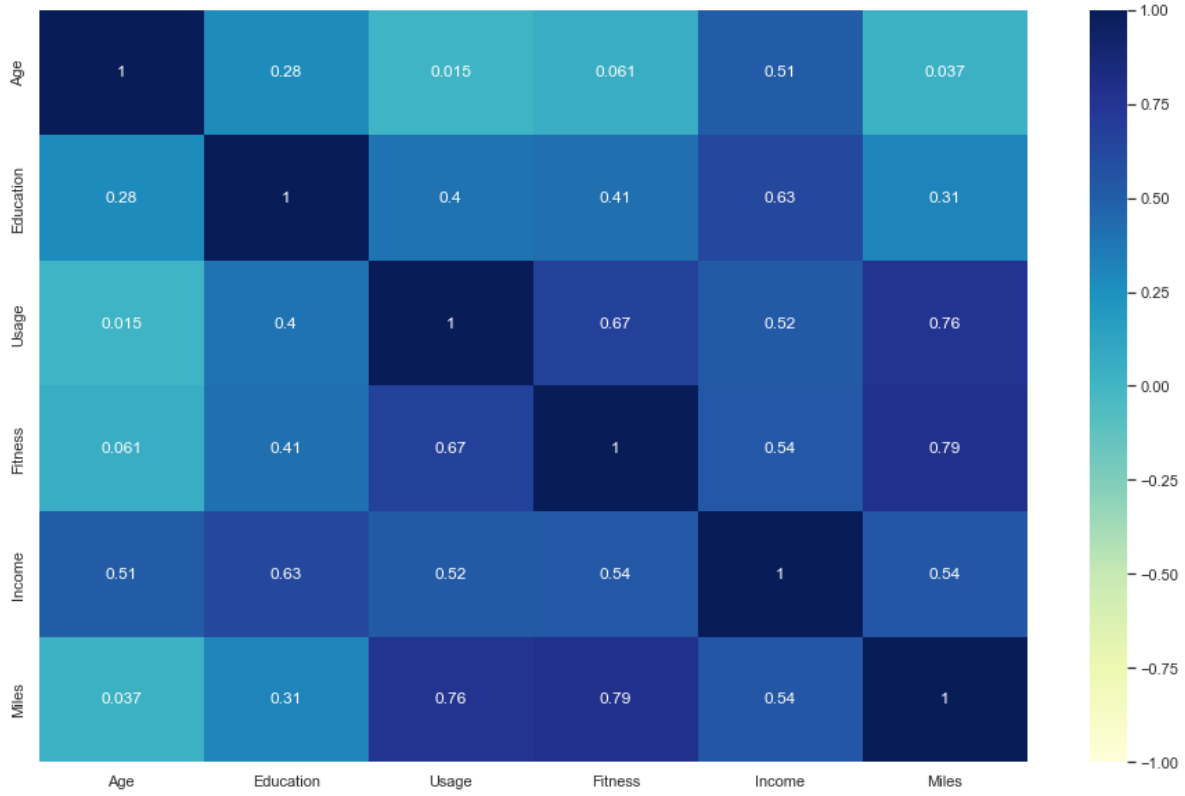
Observation:-

Education and Income are correlated.

KP781 has higher income and higher education

In [117]:

```
plt.figure(figsize = (16, 10))
sns.heatmap(df.corr(), annot=True, vmin=-1, vmax = 1,cmap="YlGnBu")
plt.show()
```



Observation

Miles and Fitness and Miles and Usage are highly correlated, which means if a customer's fitness level is high they use more treadmills.

Income and education show a strong correlation. High-income and highly educated people prefer high-end models (KP781), as mentioned during Bivariant analysis of Categorical variables.

There is no corelation between Usage & Age or Fitness & Age which mean Age should not be barrier to use treadmills or specific model of treadmills.

In [119]:

```
1 sns.pairplot(df, hue='Product')
2 plt.show()
```

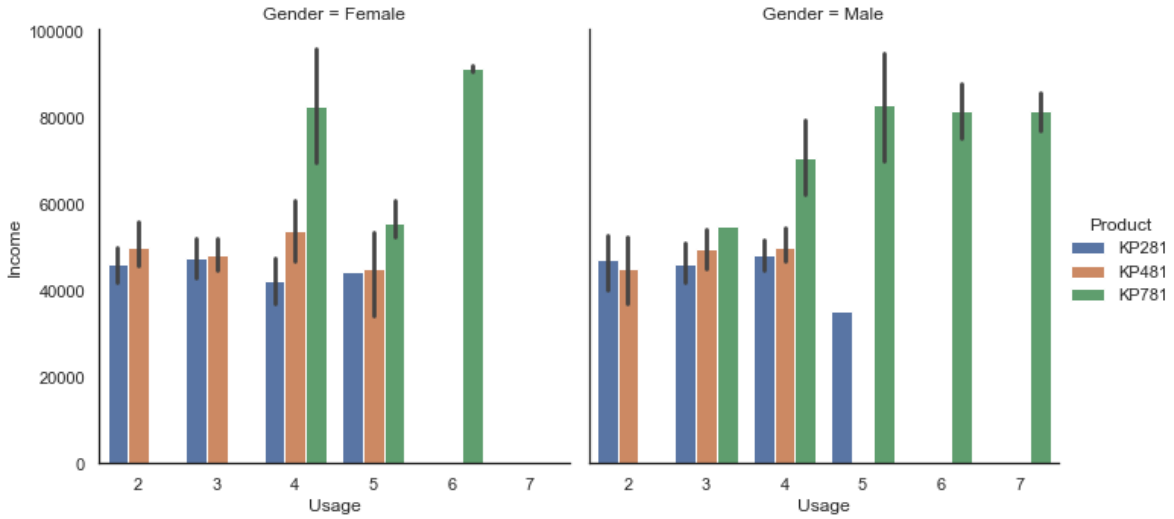
Multivariate Analysis

In [120]:

```
plt.figure(figsize=(15,10))
sns.catplot(x='Usage', y='Income', col='Gender',hue='Product' ,kind="bar", data=df)
```

Out[120]:

```
<seaborn.axisgrid.FacetGrid at 0x1f54f337430>

<Figure size 1080x720 with 0 Axes>
```



Observations

Male customer with higher income ,bought KP781 Model and expect to use treadmill 4-6 /week

Customer who bought KP281 and KP481 are in same income range and expect to use treadmill 3-4 /week

In [121]:

```
prd_mar_gen= pd.crosstab(index=df["Product"], columns=[df["MaritalStatus"],df["Gender"]
prd_mar_gen
```

Out[121]:

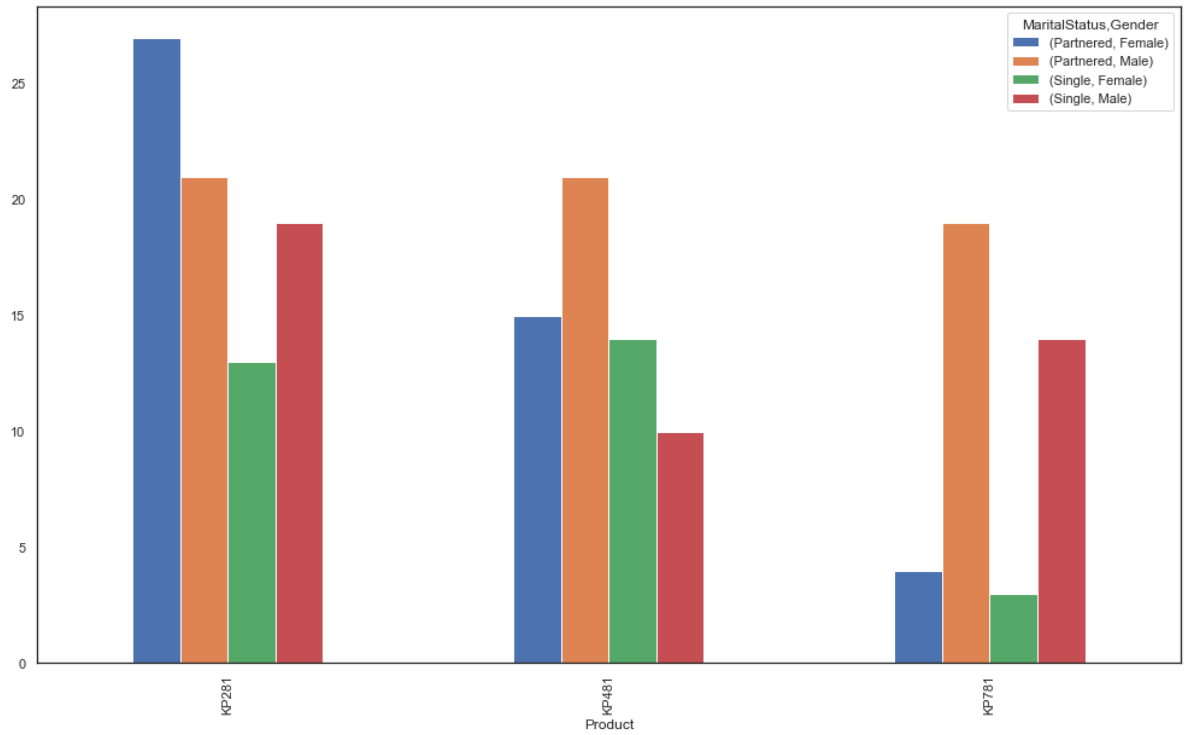
MaritalStatus	Partnered		Single	
Gender	Female	Male	Female	Male
Product				
KP281	27	21	13	19
KP481	15	21	14	10
KP781	4	19	3	14

In [122]:

```
prd_mar_gen.plot(kind='bar',figsize=(17,10))
```

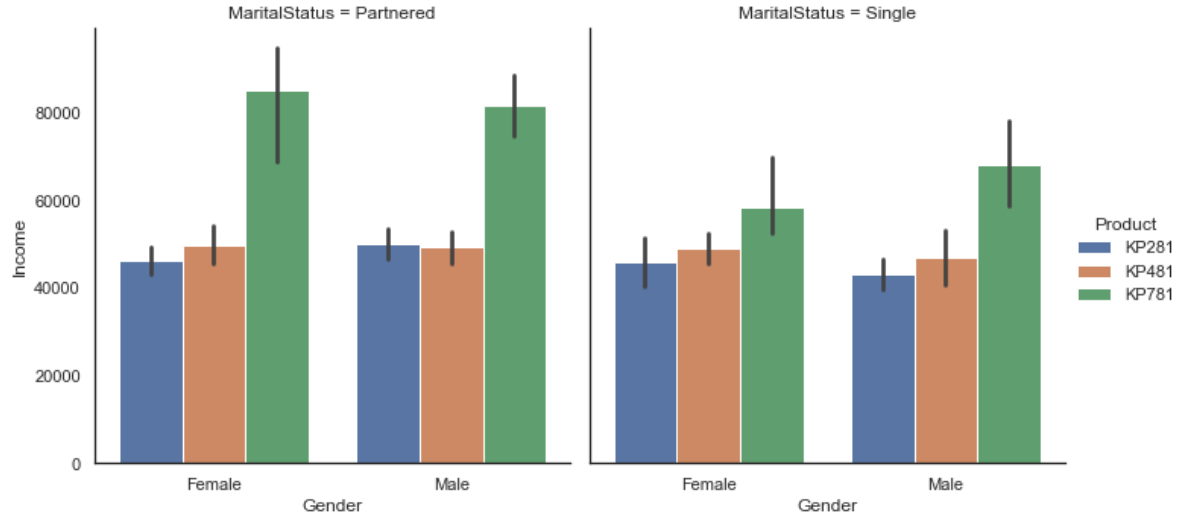
Out[122]:

<AxesSubplot:xlabel='Product'>



In [123]:

```
# Income by gender by product and by marital status
sns.catplot(x='Gender',y='Income', hue='Product', col='MaritalStatus', data=df,kind='ba
```



Observations

Partnered Female bought KP281 Model compared to Partnered male.

Single Female customers bought KP481 model more than Single male customers.

Partnered Male customers bought KP781 model more than Single Male customers.

There are more single males buying Treadmill than single Females.

Single Male customers bought KP281 Model compared to Single Female.

Majority of people who buy the KP781 are man & partnered.

The majority of our buyers are man.

In [124]:

```
1 prod_gen_fit=pd.crosstab(index=df['Product'],columns=[df['Gender'],df['Fitness']])
2 prod_gen_fit
```

Out[124]:

Gender	Female					Male				
Fitness	1	2	3	4	5	1	2	3	4	5
Product										
KP281	0	10	26	3	1	1	4	28	6	1
KP481	1	6	18	4	0	0	6	21	4	0
KP781	0	0	1	1	5	0	0	3	6	24

In [125]:

```
#scatter plot between income, age ,product and usage
sns.relplot(x="Age", y="Income", hue="Product", size="Usage",
            sizes=(40, 400), alpha=.5, palette="plasma",
            height=6, data=df).set(title='INCOME BY AGE ,PRODUCT AND USAGE');
```

In [ ]:

```
1
```

Observations:

Products KP281 and KP481 are bought by people with lower than 70K as income and age is concentrated more in range of 23-35

Product KP781 is mainly bought by people with higher than 70K income and age falls in range of 23-30. -Majority of people who buys the KP781 expect that they will run more than consumers of the other two products, on average.

Analysis using Contingency Tables to Calculate Probabilities¶

(Marginal Probabilities, Joint Probabilities, Conditional Probabilities)

Product - Incomeslab

Product - Gender

Product - Fitness

Product - AgeCategory

Product - Marital Status

In [ ]:

```
1 # Product - Incomeslab
```

In [129]:

```
1 pd.crosstab(index=df['Product'], columns=[df['IncomeSlab']],margins=True)
```

Out[129]:

IncomeSlab	Low Income	Lower-middle income	Upper-Middle income	High income	All
Product					
KP281	8	66	6	0	80
KP481	6	47	7	0	60
KP781	0	11	12	17	40
All	14	124	25	17	180

In [ ]:

```
1
```

Percentage of a high-income customer purchasing a treadmill (Marginal Probability)

In [130]:

```
1 # Sum of the treadmill purchased by high income customer by total no. of customers.
2 round(14/180,2)*100
```

Out[130]:

```
8.0
```

Percentage of a High-income customer purchasing KP781 treadmill (Joint Probability)

In [131]:

```
# Sum of the treadmill with model TM798 purchased by high income customer by total no.
round(17/180,2)*100
```

Out[131]:

9.0

Percentage of customer with high-Income salary buying treadmill given that Product is KP781 (Conditional Probability)

In [132]:

```
round(17/17,2)*100
```

Out[132]:

100.0

Observation

Customers having salary more than USD dollar 85,000 buys only KP781 (high-end Model).

In [134]:

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

Out[134]:

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

Percentage of a Male customer purchasing a treadmill

In [135]:

```
prob = round((104/180),2)
pct = round(prob*100,2)
pct
```

Out[135]:

58.0

Percentage of a Female customer purchasing KP781 treadmill

In [136]:

```
prob = round((7/180),2)
pct = round(prob*100,2)
pct
```

Out[136]:

4.0

Percentage of Female customer buying treadmill given that Product is KP281

P(A|B) = P(A,B)/P(B)

P(Female|KP281) = P(Female,KP281)/P(KP281)

In [138]:

```
prob = round((40/80),2)
pct = round(prob*100,2)
pct
```

Out[138]:

50.0

Observation

Female customer prefer to buy KP281 & KP481

50% of female tend to purchase treadmill model KP281

In [ ]:

```
# Product - Fitness
```

In [140]:

```
pd.crosstab(index=df['Product'], columns=[df['Fitness']],margins=True)
```

Out[140]:

Fitness	1	2	3	4	5	All
Product						
KP281	1	14	54	9	2	80
KP481	1	12	39	8	0	60
KP781	0	0	4	7	29	40
All	2	26	97	24	31	180

Percentage of a customers having fitness level5 are

In [141]:

```
prob = round((31/180),2)
pct = round(prob*100,2)
pct
```

Out[141]:

```
17.0
```

Percentage of a customer with Fitness Level 5 purchasing KP781 treadmill

In [142]:

```
prob = round((29/180),2)
pct = round(prob*100,2)
pct
```

Out[142]:

```
16.0
```

Percentage of customer with fitness level-5 buying KP781 treadmill given that Product is KP781

In [143]:

```
prob = round((29/31),2)
pct = round(prob*100,2)
pct
```

Out[143]:

```
94.0
```

Observation

94% of customers with fitness level 5, purchased KP781

In [ ]:

```
# Product - AgeCategory
```

In [145]:

```
pd.crosstab(index=df['Product'], columns=[df['AgeCategory']],margins=True)
```

Out[145]:

AgeCategory	Teens	20s	30s	Above 40s	All
Product					
KP281	6	49	19	6	80
KP481	4	31	23	2	60
KP781	0	30	6	4	40
All	10	110	48	12	180

Percentage of customers with Age between 20s and 30s use treadmills

In [146]:

```
prob = round((110/180),2)
pct = round(prob*100,2)
pct
```

Out[146]:

```
61.0
```

Observation

Teen doesnot prefer to buy KP781

61% of customer with Age group between 20 and 30 purchase treadmills.

In [ ]:

```
# Product - Marital Status
```

In [147]:

```
pd.crosstab(index=df['Product'], columns=[df['MaritalStatus']],margins=True)
```

Out[147]:

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

In [148]:

```
prob = round((107/180),2)
pct = round(prob*100,2)
pct
```

Out[148]:

```
59.0
```

Observation

59 percent of customer with maritial Stuatus as Partnered by the treadmills.

Observation productwise

KP781

Average age of customer who purchases KP781 is 29 , Median is 27 . There are some outliers , suggesting we need explore more closely customers who are above 40 for any possibility of new customers.

Average Education is 17 and median is 18, suggest they have some advanced education

Expected usage is 4-5 day a week

Expected Miles to run is on an Average 166 miles per week and median is 160.

Average Income is 75K and median is 76K

Product made only 22 % of sales.

KP481

This Model is sold more than KP781 model

Average Income of the customer is 48,973

Customers with lower income purchase KP281 and KP481 model may be because of cost of the Treadmill

Average age of customer who purchases KP481 is 28.9 , Median is 26 . Customer range is between 24-33.

Average years of Education of customers is 16 assuming it to be bachelor's

Sale was 33%.This was the 2nd most sold model. The income of this group is almost same as KP281 model. KP481 model expecting to use Treadmill less frequently but to run more miles a week.Single Female customers bought TM498 model more than Single male customers, may be cause of some feature difference.

KP281

44.4% customers brought KP281. Making it most popular model.

Average customer income is 46K

Customers who bought this treadmill have income less than 60k with an average of 55K.

There are same numbers of Male and Female customers

Average age of customer who purchases KP281 is 28.5, Median is 26.

Average years of Education of customers is 15, and median is 16 assuming it to be bachelors.

VSelf rate fitness level of customer is average.

They expect to use treadmill 3-4 times a week.

Our 44.4% sale has come from this model. Majority of people whose income is around 55K has purchased this model assuming it's because of its appealing price and affordability. Equal amount of males and females bought this model suggesting this model is not gender specfic. Majority of the customers who purchased this model are Partnered Females and Single Males compared to Single females and Partnered male. This may be cause of the features this treadmill provides and the cost of treadmill. Customers who bought this treadmill believe there fitness is average, and might be looking for a basic treadmill that does the job.

In [ ]:

```
1
```

Conclusion (Important Observations):

Model KP281 is the best-selling product. 44.0% of all treadmill sales go to model KP281.

The majority of treadmill customers fall within the USD 45,000 - USD 80,000 income bracket. 83% of treadmills are bought by individuals with incomes between USD dollor 35000 and 85000.

There are only 8% of customers with incomes below USD 35000 who buy treadmills.

88% of treadmills are purchased by customers aged 20 to 40.

Miles and Fitness & Miles and Usage are highly correlated, which means if a customer's fitness level is high they use more treadmills.

KP781 is the only model purchased by a customer who has more than 20 years of education and an income of over USD dollor 85,000.

With Fitness level 4 and 5, the customers tend to use high-end models and the average number of miles is above 150 per week

In [ ]:

```
1
```

Recommendations

KP281 & KP481 are popular with customers earning USD 45,000 and USD 60,000 and can be offered by these companies as affordable models.

KP781 should be marketed as a Premium Model and marketing it to high income groups and educational over 20 years market segments could result in more sales.

Aerofit should conduct market research to determine if it can attract customers with income under USD 35,000 to expand its customer base.

The KP781 is a premium model, so it is ideally suited for sporty people who have a high average weekly mileage.

KP281 & KP481 attracts people with income less than 60k , may be because of cost of both models. We should market these models as a budget Treadmill for all.

KP781 should be marketed as a high end Treadmill for professionals and athletes. Create a luxiurous brand image for this Treadmill.

Assuming KP781 provides high margin of profit, we should brand it as Treadmill for athletes. We can also endorse some athlete to promote this Treadmill. This might increase there sales.

Considering above observations, We can attract customers to upgrade from their existing treadmill and switch to KP781 ,highlighting extra features this Treadmill provides.

To expand our sales with Female customers, We could run a marketing campaign during Women's days, Mothers days emphasizing on fitness and exercise.

The age of our customers are in the range of 35 years old and 18 years old. We need to research if there is any scope to increase sale with customers who are more than 35 years old.

KP281 & KP481 attracts people with income less than 60k , may be because of cost of both models. We should market these models as a budget Treadmill for all.

KP781 should be marketed as a high end Treadmill for professionals and athletes. Create a luxiurous brand image for this Treadmill.

Assuming KP781 provides high margin of profit, we should brand it as Treadmill for athletes. We can also endorse some athlete to promote this Treadmill. This might increase there sales.

Considering above observations, We can attract customers to upgrade from their existing treadmill and switch to KP781 ,highlighting extra features this Treadmill provides.

To expand our sales with Female customers, We could run a marketing campaign during Women's days, Mothers days emphasizing on fitness and exercise.

The age of our customers are in the range of 35 years old and 18 years old. We need to research if there is any scope to increase sale with customers who are more than 35 years old.