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

In [2]:
```
data = pd.read_csv('aerofit_treadmill.csv')
```

In [3]:
```
data.head()
```

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

In [4]:
```
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
```

In [5]:
```
data.shape
```

Out[5]: (180, 9)

In [6]:
```
print(f"TOTAL ROWS : {data.shape[0]}")
print(f"TOTAL COLUMNS : {data.shape[1]}")
```

```
TOTAL ROWS : 180
TOTAL COLUMNS : 9
```

In [7]:
```
print(f"SIZE OF DataFrame : {data.size}")
```

```
SIZE OF DataFrame : 1620
```

In [9]:
```
print(f"Index of the DataFrame : {data.index}")
```

```
Index of the DataFrame : RangeIndex(start=0, stop=180, step=1)
```

In [10]:
```
print(f"Coulumns : {data.columns}")
```

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

In [11]:
```
data.describe()
```

Out[11]:

	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 [12]:
```
data.describe().T
```

Out[12]:

	count	mean	std	min	25%	50%	75%	max
Age	180.0	28.788889	6.943498	18.0	24.00	26.0	33.00	50.0
Education	180.0	15.572222	1.617055	12.0	14.00	16.0	16.00	21.0
Usage	180.0	3.455556	1.084797	2.0	3.00	3.0	4.00	7.0
Fitness	180.0	3.311111	0.958869	1.0	3.00	3.0	4.00	5.0
Income	180.0	53719.577778	16506.684226	29562.0	44058.75	50596.5	58668.00	104581.0
Miles	180.0	103.194444	51.863605	21.0	66.00	94.0	114.75	360.0

In [14]:
```
data.isnull()
```

Out[14]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
175	False	False	False	False	False	False	False	False	False
176	False	False	False	False	False	False	False	False	False
177	False	False	False	False	False	False	False	False	False
178	False	False	False	False	False	False	False	False	False
179	False	False	False	False	False	False	False	False	False

180 rows × 9 columns

In [15]:
```
data.isnull().sum()
```

Out[15]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

In [16]:
```
#There are zero null values in all the columns (DataFrame don't have any null values)
```

In [17]:
```
data.duplicated()
```

Out[17]:
```
0      False
1      False
2      False
3      False
4      False
       ...
175    False
176    False
177    False
178    False
179    False
Length: 180, dtype: bool
```

In [18]:
```
data.duplicated().sum()
```

Out[18]:
```
0
```

In [19]:
```
#There are zero duplicate values in the DataFrame (DataFrame don't have any duplicate values)
```

In [20]:
```
#Taking a copy of data into data_copy
data_copy = data.copy()
```

In [22]:
```
data.nunique()
```

Out[22]:
```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

In [23]:
```
data['Product'].unique()
```

Out[23]:
```
array(['KP281', 'KP481', 'KP781'], dtype=object)
```

In [24]:
```
#'KP281', 'KP481', 'KP781' are the Products available
```

In [25]:
```
data['Age'].unique()
```

Out[25]:
```
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 [26]:
```
#There are age group data between 18 to 42 years
```

In [27]:
```
data['Gender'].unique()
```

Out[27]:
```
array(['Male', 'Female'], dtype=object)
```

In [28]:
```
#There are data of both Male as well as Female
```

In [29]:
```
data['Education'].unique()
```

Out[29]:
```
array([14, 15, 12, 13, 16, 18, 20, 21], dtype=int64)
```

In [30]:
```
#Education in years are between 14 to 21 years
```

In [31]:
```
data['MaritalStatus'].unique()
```

Out[31]:
```
array(['Single', 'Partnered'], dtype=object)
```

In [32]:
```
#There are data of both married and unmarried 'Single', 'Partnered'
```

In [33]:
```
data['Usage'].unique()
```

Out[33]:
```
array([3, 2, 4, 5, 6, 7], dtype=int64)
```

In [34]:
```
#The average number of times the customer plans to use the treadmill each week are between 2 to 7
```

In [35]:
```
data['Fitness'].unique()
```

Out[35]:
```
array([4, 3, 2, 1, 5], dtype=int64)
```

In [36]:
```
#Rating is between 1 to 5
```

In [37]:
```
data['Income'].unique()
```

Out[37]:
```
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 [38]:
```
data['Income'].min()
```

Out[38]:
```
29562
```

In [39]:
```
data['Income'].max()
```

Out[39]:
```
104581
```

In [40]:
```
#Minimum salary is 29562 and maximum salary 104581 of the customer
```

In [41]:
```
data['Miles'].unique()
```

Out[41]:
```
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)
```

In [42]:
```
data['Miles'].min()
```

Out[42]:
```
21
```

In [43]:
```
data['Miles'].max()
```

Out[43]:
```
360
```

In [49]:
```
# Convert categorical columns to 'category' data type
categorical_columns = ['Product', 'Gender', 'MaritalStatus']
for col in categorical_columns:
    data[col] = data[col].astype('category')
```

In [47]:
```
type_counts = data['Product'].value_counts()
type_counts
```

Out[47]:
```
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
```

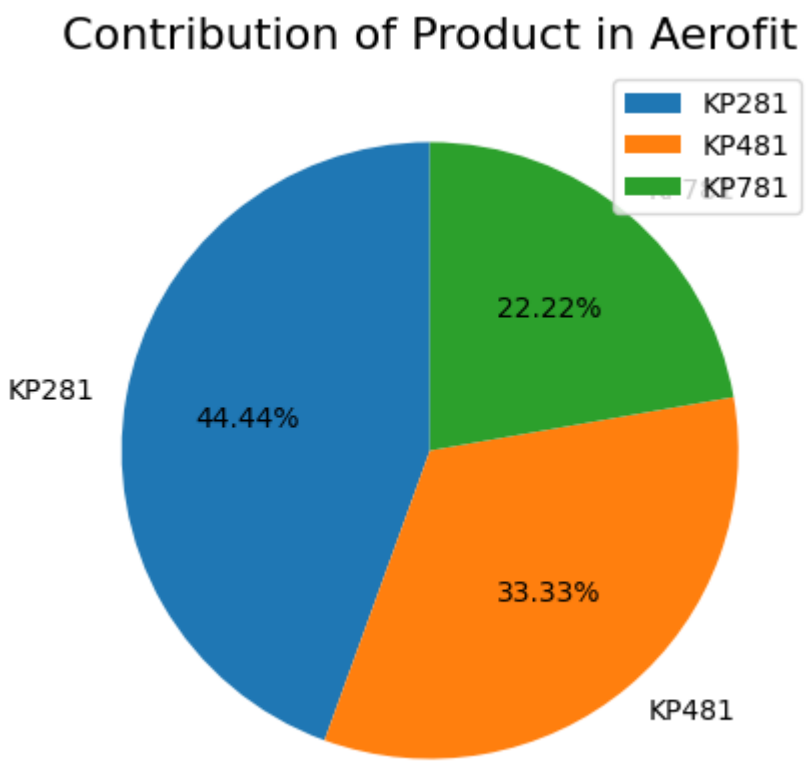
In [51]:
```
type_counts = data['Product'].value_counts(normalize=True)*100
type_counts.round(2)
type_counts
```

Out[51]:
```
Product
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: proportion, dtype: float64
```

In [52]:
```
#Majority of the customers use Product KP281, then KP481, then KP781 respectively
#Order of price of items are KP281 < KP481 < KP781
```

In [53]:
```
#Customers prefered to use less priced products
```

In [60]:
```
type_counts = data['Product'].value_counts(normalize=True)*100
plt.figure(figsize=(5,5))
plt.pie(type_counts.values, labels = type_counts.index, autopct = '%.2f%%', startangle = 90)
plt.legend()
plt.title('Contribution of Product in Aerofit', fontsize=16)
plt.show()
```



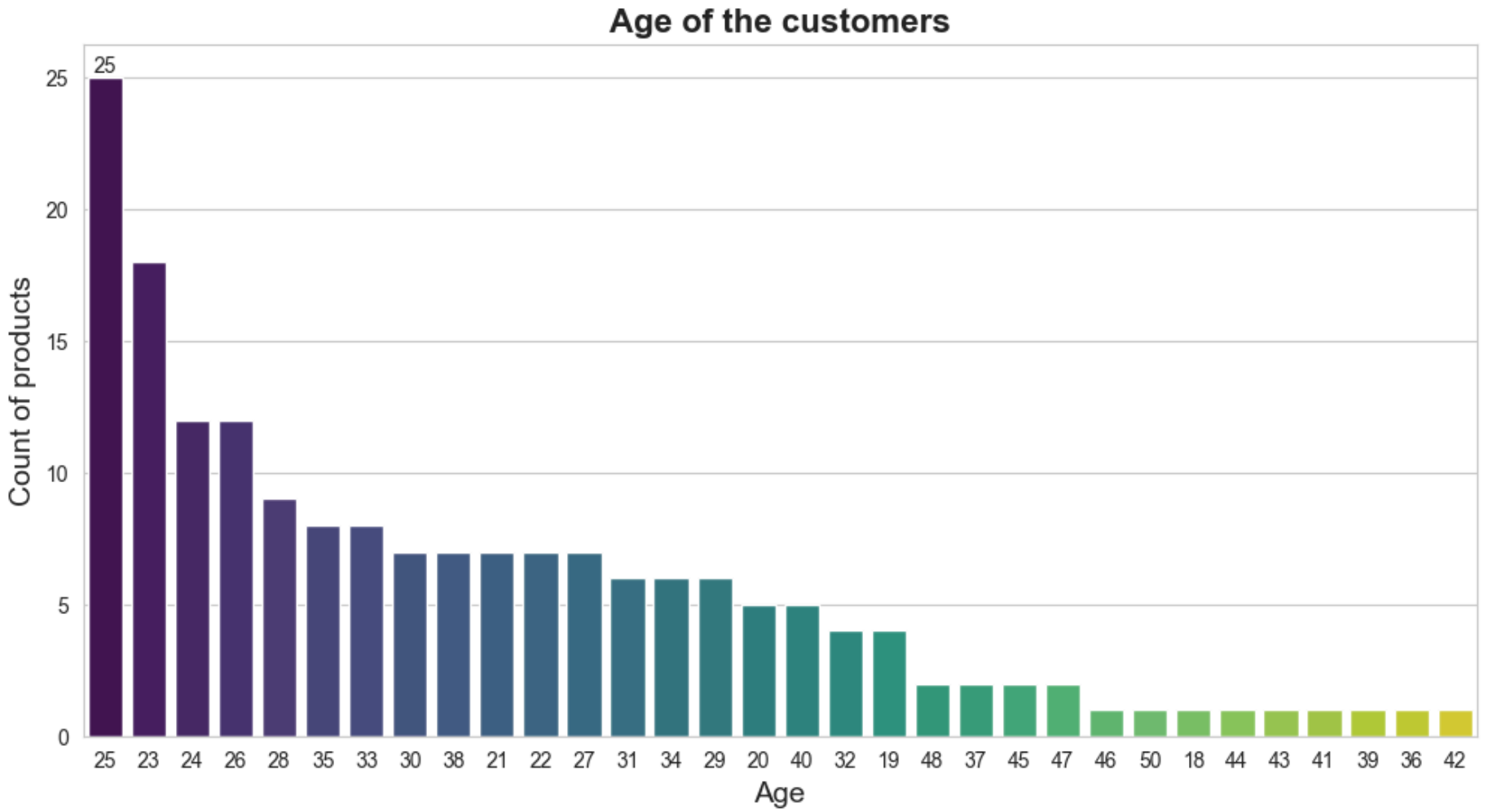
In [64]:
```
age_counts = data['Age'].value_counts()
age_counts.round(2)
age_counts
```

Out[64]:
```
Age
25    25
23    18
24    12
26    12
28     9
35     8
33     8
30     7
38     7
21     7
22     7
27     7
31     6
34     6
29     6
20     5
40     5
32     4
19     4
48     2
37     2
45     2
47     2
46     1
50     1
18     1
44     1
43     1
41     1
39     1
36     1
42     1
Name: count, dtype: int64
```

In [66]:
```
import warnings

warnings.filterwarnings("ignore", category=FutureWarning)
```

In [68]:
```
sns.set_style("whitegrid")
fig, ax = plt.subplots(figsize=(12, 6))
ax = sns.countplot(data=data, x="Age", order=data["Age"].value_counts().index, saturation=0.75, palette="viridis", legend=Fals
ax.set_title("Age of the customers", fontsize=16, fontweight='bold')
ax.set_xlabel("Age", fontsize=14)
ax.set_ylabel("Count of products", fontsize=14)
ax.bar_label(ax.containers[0], fontsize=10)
plt.show()
```

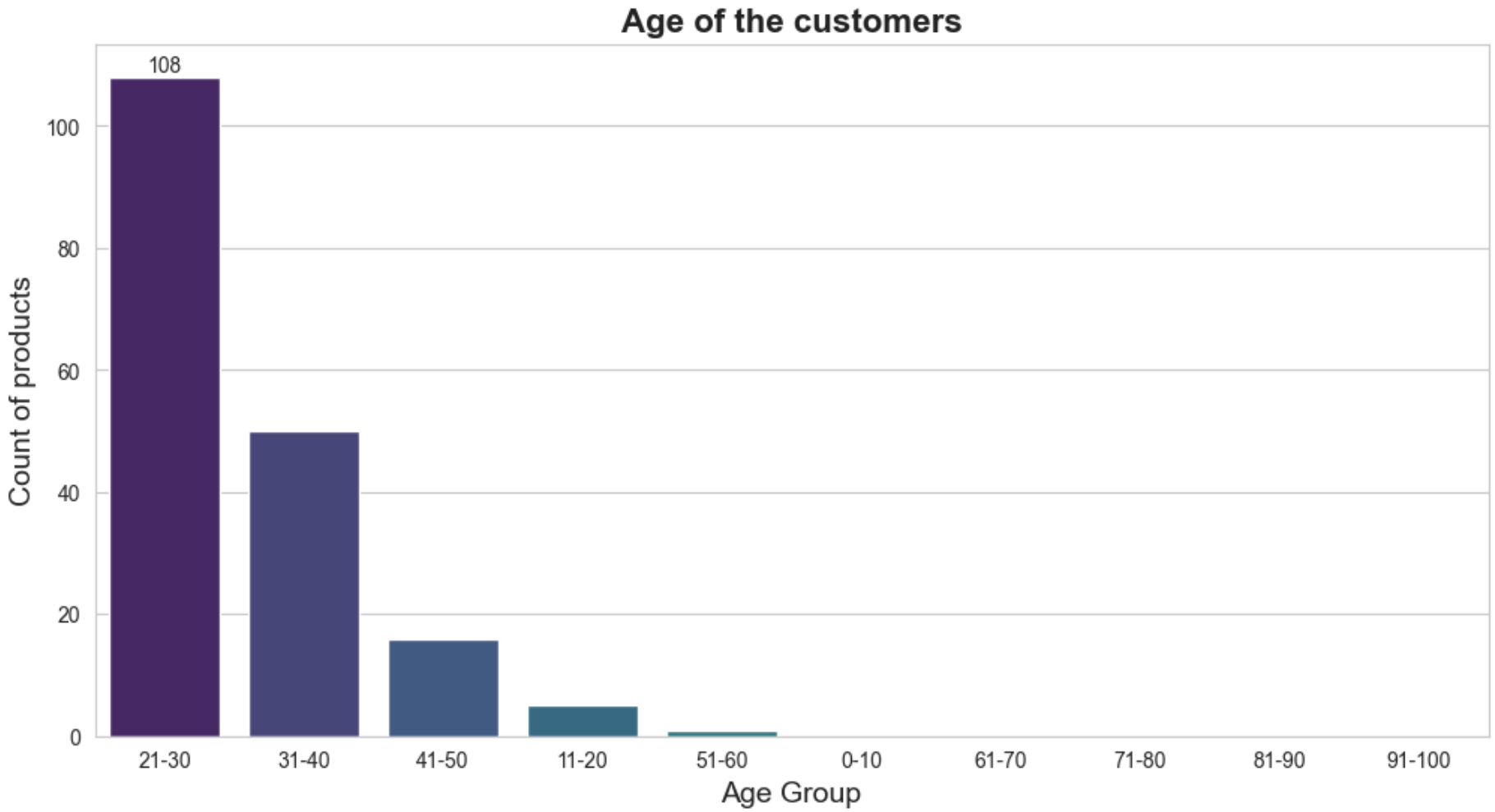


In [69]:
```
bins = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
labels = ['0-10', '11-20', '21-30', '31-40', '41-50', '51-60', '61-70', '71-80', '81-90', '91-100']
data['AgeGroup'] = pd.cut(data['Age'], bins=bins, labels=labels, right=False)

data['AgeGroup'].value_counts()
```

Out[69]:
```
AgeGroup
21-30     108
31-40      50
41-50      16
11-20       5
51-60       1
0-10        0
61-70       0
71-80       0
81-90       0
91-100      0
Name: count, dtype: int64
```

In [70]:
```
sns.set_style("whitegrid")
fig, ax = plt.subplots(figsize=(12, 6))
ax = sns.countplot(data=data, x="AgeGroup", order=data['AgeGroup'].value_counts().index, saturation=0.75, palette="viridis", l
ax.set_title("Age of the customers", fontsize=16, fontweight='bold')
ax.set_xlabel("Age Group", fontsize=14)
ax.set_ylabel("Count of products", fontsize=14)
ax.bar_label(ax.containers[0], fontsize=10)
plt.show()
```

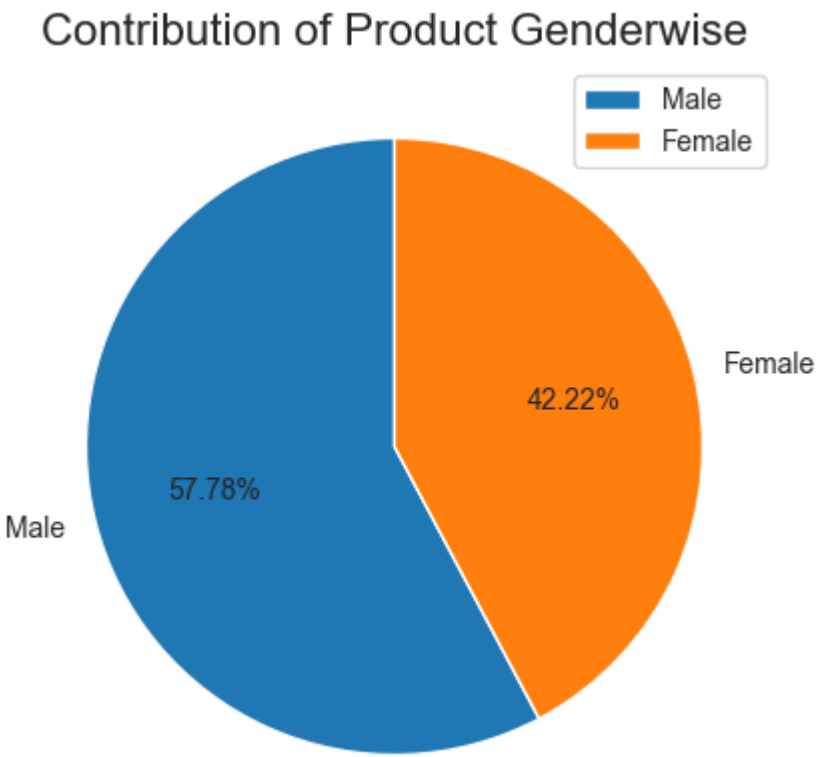


In [71]:
```
#People in age group 21-30 use more products, then 31-40 then 41-50, then 11-20
```

In [73]:
```
gender_counts = data['Gender'].value_counts(normalize=True)*100
gender_counts.round(2)
gender_counts
```

Out[73]:
```
Gender
Male      57.777778
Female    42.222222
Name: proportion, dtype: float64
```

In [74]:
```
gender_counts = data['Gender'].value_counts(normalize=True)*100
plt.figure(figsize=(5,5))
plt.pie(gender_counts.values, labels = gender_counts.index, autopct = '%.2f%%', startangle = 90)
plt.legend()
plt.title('Contribution of Product Genderwise', fontsize=16)
plt.show()
```

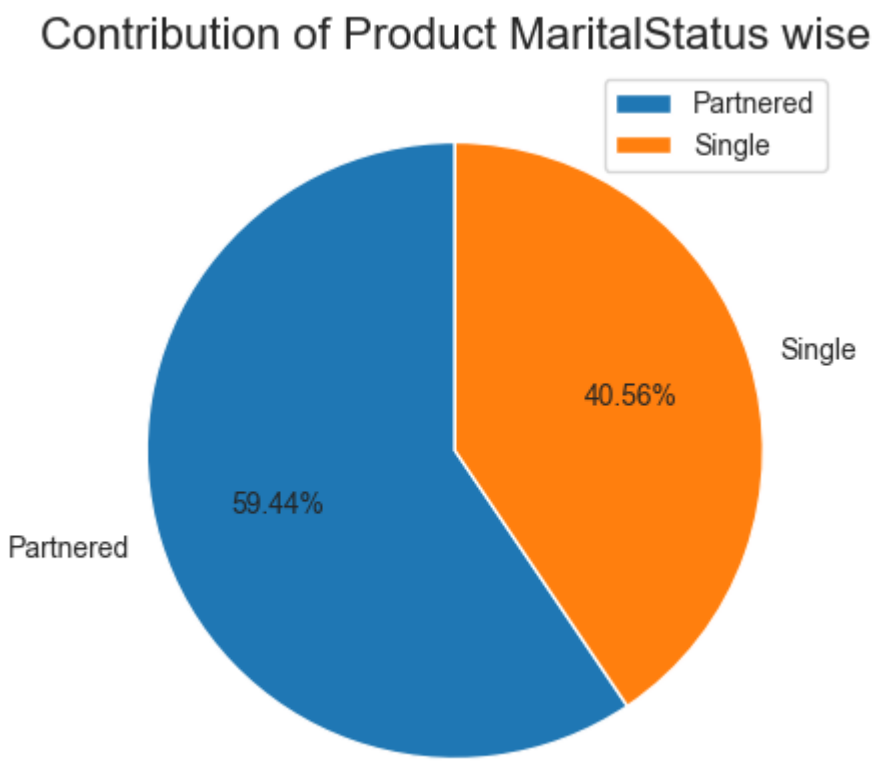


In [75]:
```
#57.78% of Customers are Male and remaining are female
```

In [76]:
```
ms_counts = data['MaritalStatus'].value_counts(normalize=True)*100
ms_counts.round(2)
ms_counts
```

Out[76]:
```
MaritalStatus
Partnered    59.444444
Single       40.555556
Name: proportion, dtype: float64
```

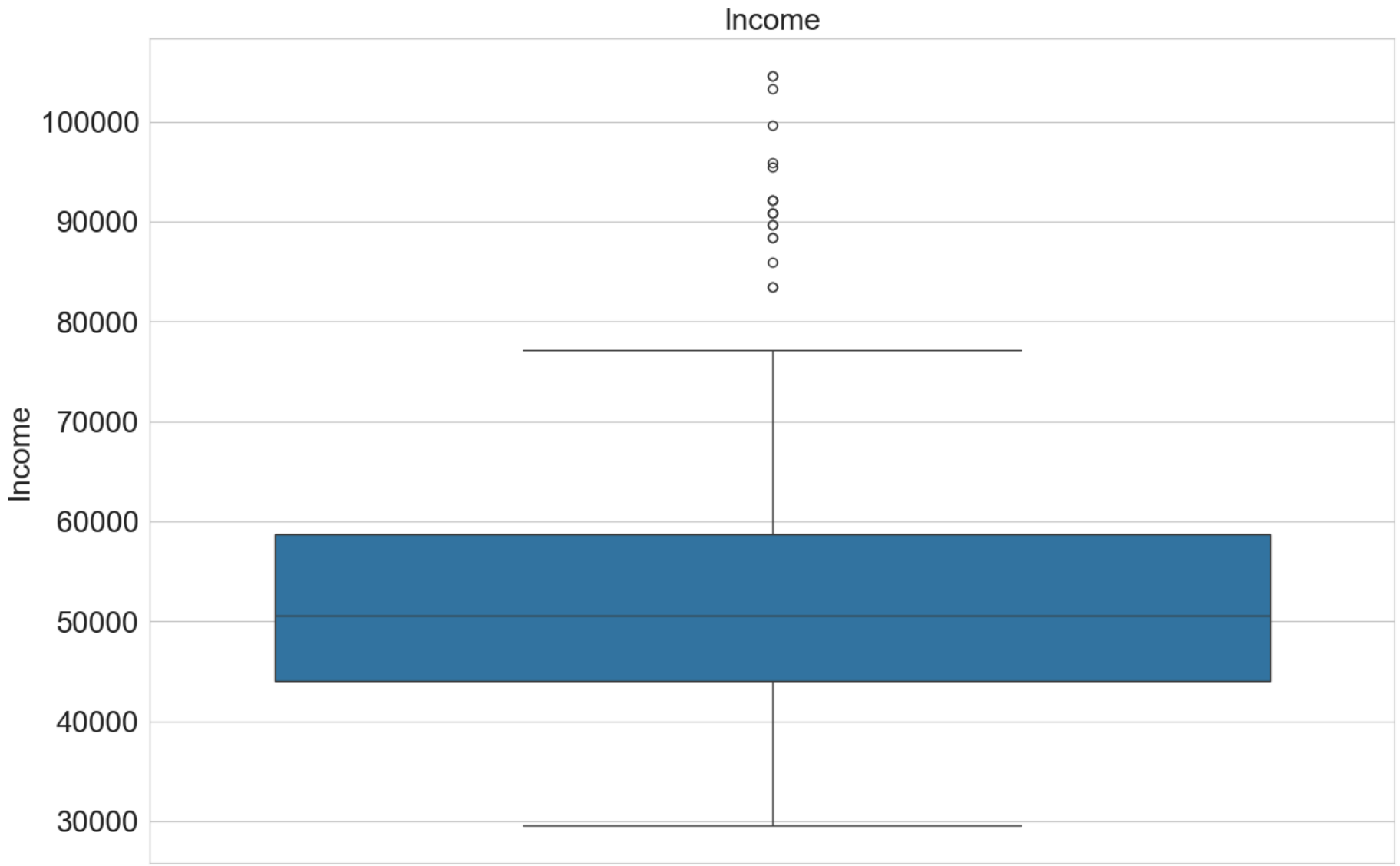
In [77]:
```
ms_counts = data['MaritalStatus'].value_counts(normalize=True)*100
plt.figure(figsize=(5,5))
plt.pie(ms_counts.values, labels = ms_counts.index, autopct = '%.2f%%', startangle = 90)
plt.legend()
plt.title('Contribution of Product MaritalStatus wise', fontsize=16)
plt.show()
```



In [78]:
```
#59.44% of people are marries people and others are Single
```

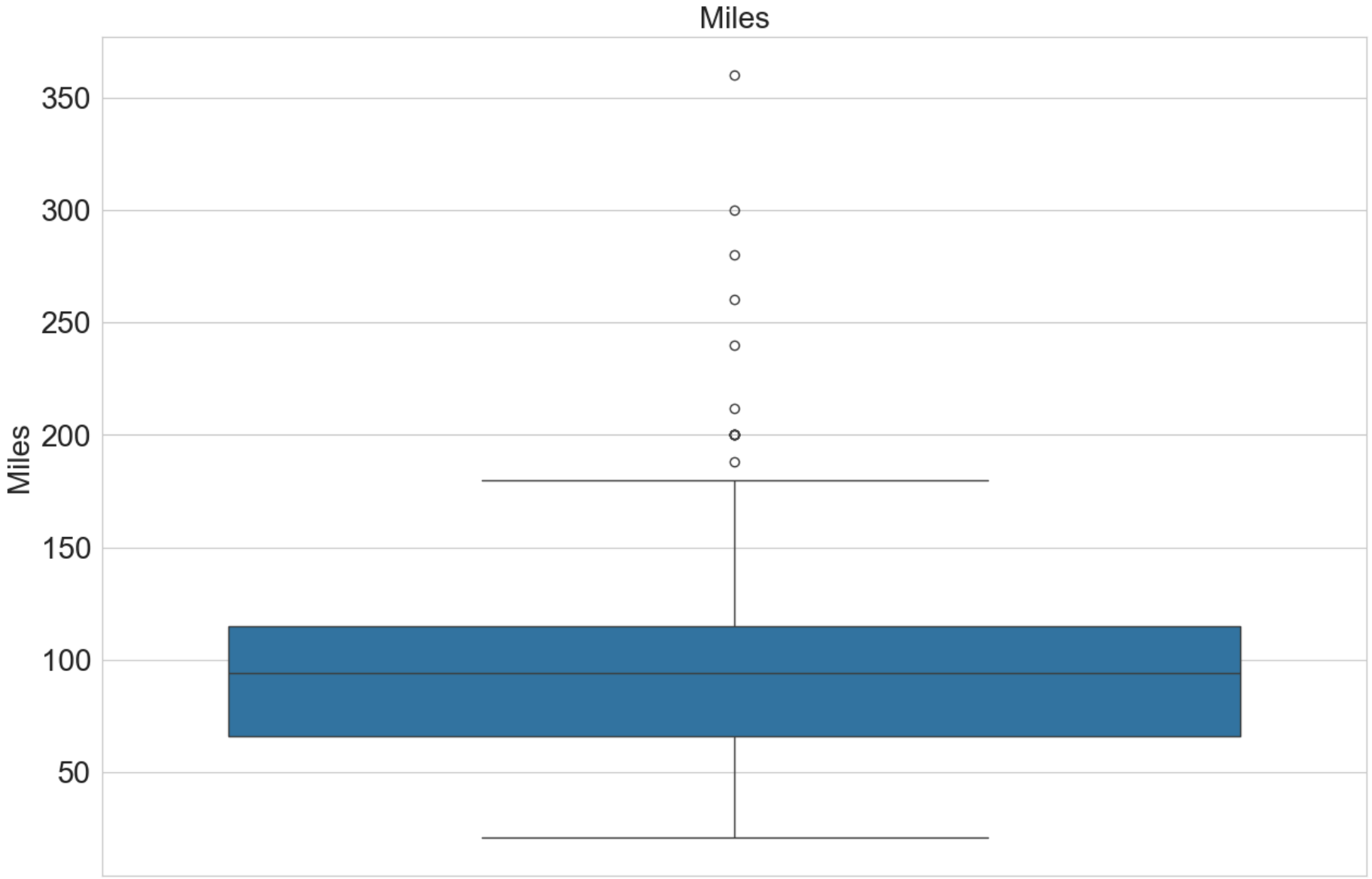
In [83]:
```
plt.figure(figsize=(15,10))
sns.boxplot(y = data["Income"])
plt.yticks(fontsize=20)
plt.ylabel('Income', fontsize=20)
plt.title('Income', fontsize=20)
```

Out[83]:
```
Text(0.5, 1.0, 'Income')
```



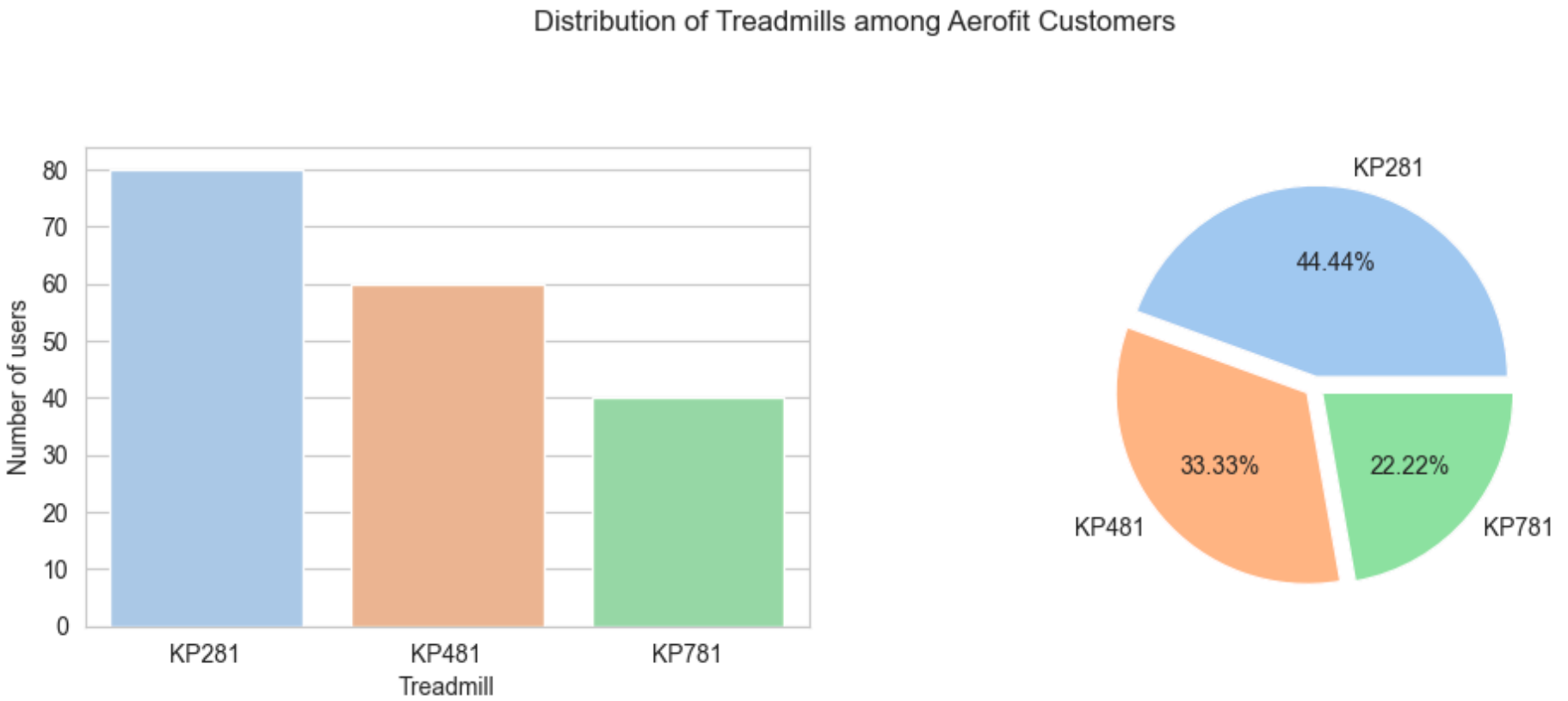
In [84]:
```
plt.figure(figsize=(15,10))
sns.boxplot(y = data["Miles"])
plt.yticks(fontsize=20)
plt.ylabel('Miles', fontsize=20)
plt.title('Miles', fontsize=20)
```

Out[84]: Text(0.5, 1.0, 'Miles')



In [89]:
```
plt.figure(figsize = (12,8))
plt.subplot(2,2,1)
sns.countplot(data = data, x = data['Product'], palette = 'pastel')
```

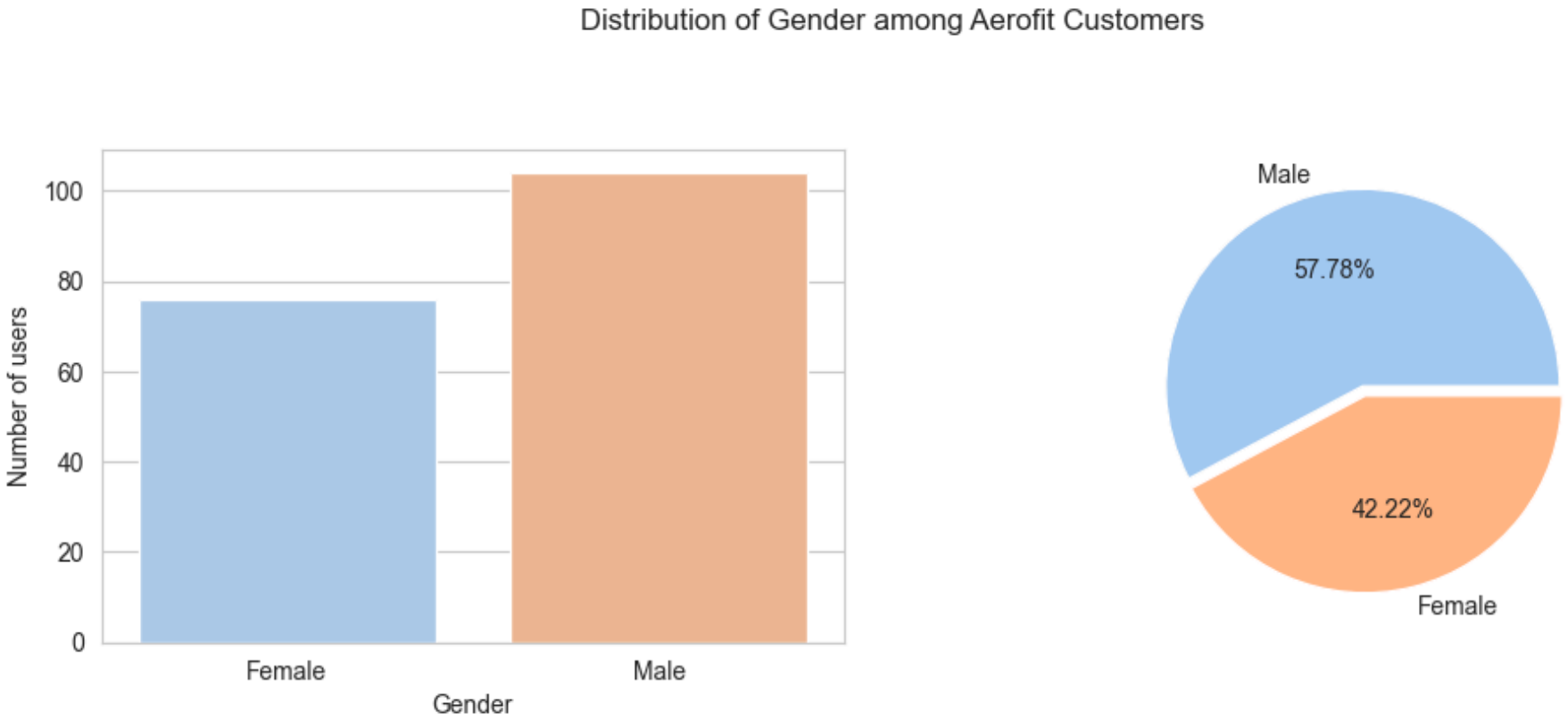
```
plt.xlabel('Treadmill')
plt.ylabel('Number of users')
plt.subplot(2,2,2)
plt.pie(data['Product'].value_counts(), labels = data['Product'].unique(), explode = (0.05,0.05,0.05), colors = sns.color_pale
plt.suptitle('Distribution of Treadmills among Aerofit Customers')
plt.show()
```



In [90]:
```
plt.figure(figsize = (12,8))
plt.subplot(2,2,1)
sns.countplot(data = data, x = data['Gender'], palette = 'pastel')
plt.xlabel('Gender')
plt.ylabel('Number of users')

plt.subplot(2,2,2)
plt.pie(data['Gender'].value_counts(), labels = data['Gender'].unique(), explode = (0.05,0), colors = sns.color_palette('paste
```

```
plt.suptitle('Distribution of Gender among Aerofit Customers')
plt.show()
```

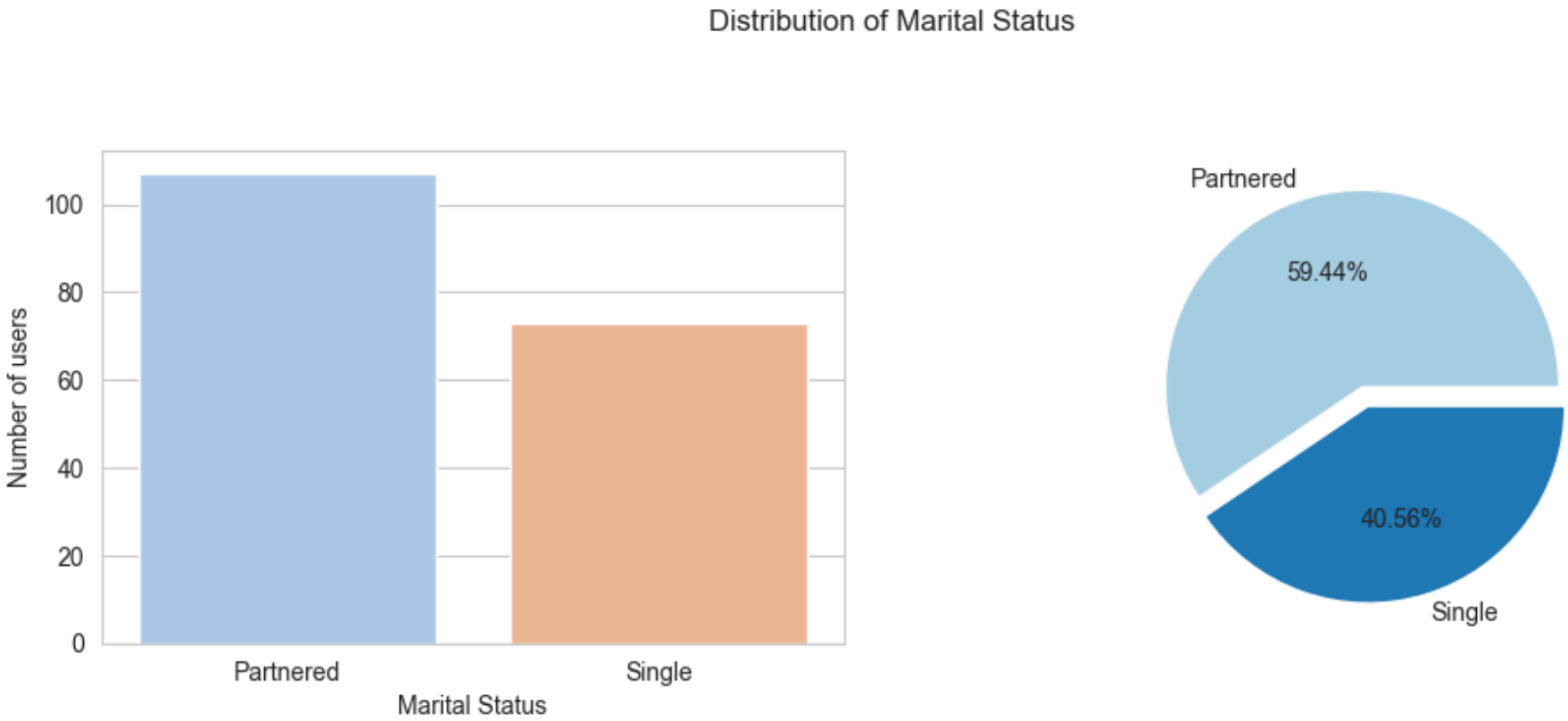


In [91]:
```
plt.figure(figsize = (12,8))

plt.subplot(2,2,1)
sns.countplot(data = data, x = 'MaritalStatus', palette = 'pastel')
plt.xlabel('Marital Status')
plt.ylabel('Number of users')

plt.subplot(2,2,2)
plt.pie(data['MaritalStatus'].value_counts(), labels = ['Partnered','Single'], explode = (0.05,0.05), colors = sns.color_palet
```

```
plt.suptitle('Distribution of Marital Status')
plt.show()
```



In [97]:
```
data["Income"].max()
```

Out[97]: 104581

In [98]:
```
bins = [0, 25000, 50000, 75000, 100000, 150000]
labels = ['0-25k', '25k-50k', '50k-75k', '75k-100k', '100k-125k']
data['IncomeGroup'] = pd.cut(data['Income'], bins=bins, labels=labels, right=False)

data['IncomeGroup'].value_counts()
```

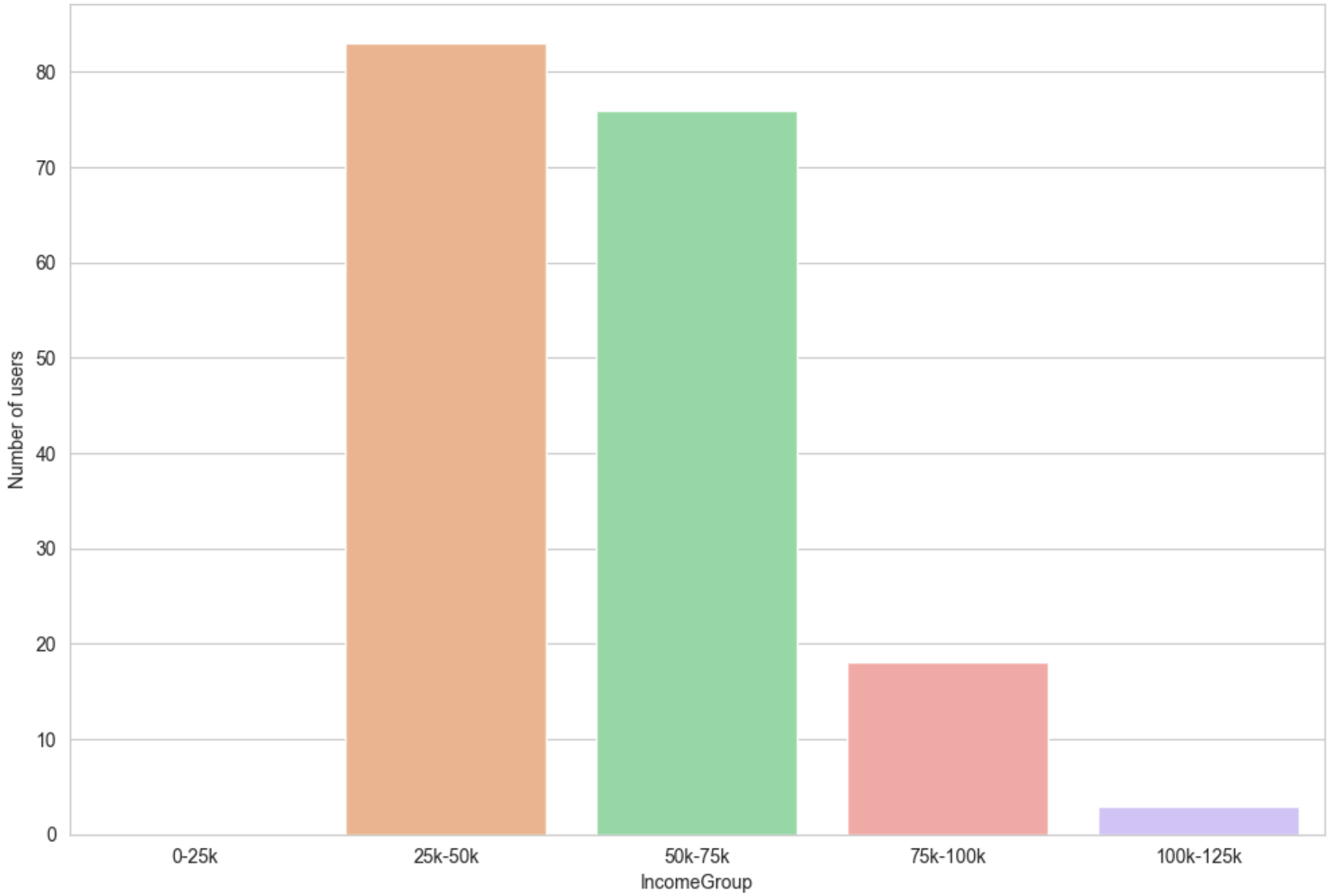
Out[98]:
```
IncomeGroup
25k-50k      83
50k-75k      76
75k-100k     18
100k-125k     3
0-25k         0
Name: count, dtype: int64
```

In [100…

```
plt.figure(figsize = (12,8))

sns.countplot(data = data, x = 'IncomeGroup', palette = 'pastel')
plt.xlabel('IncomeGroup')
plt.ylabel('Number of users')

plt.show()
```



In [101…

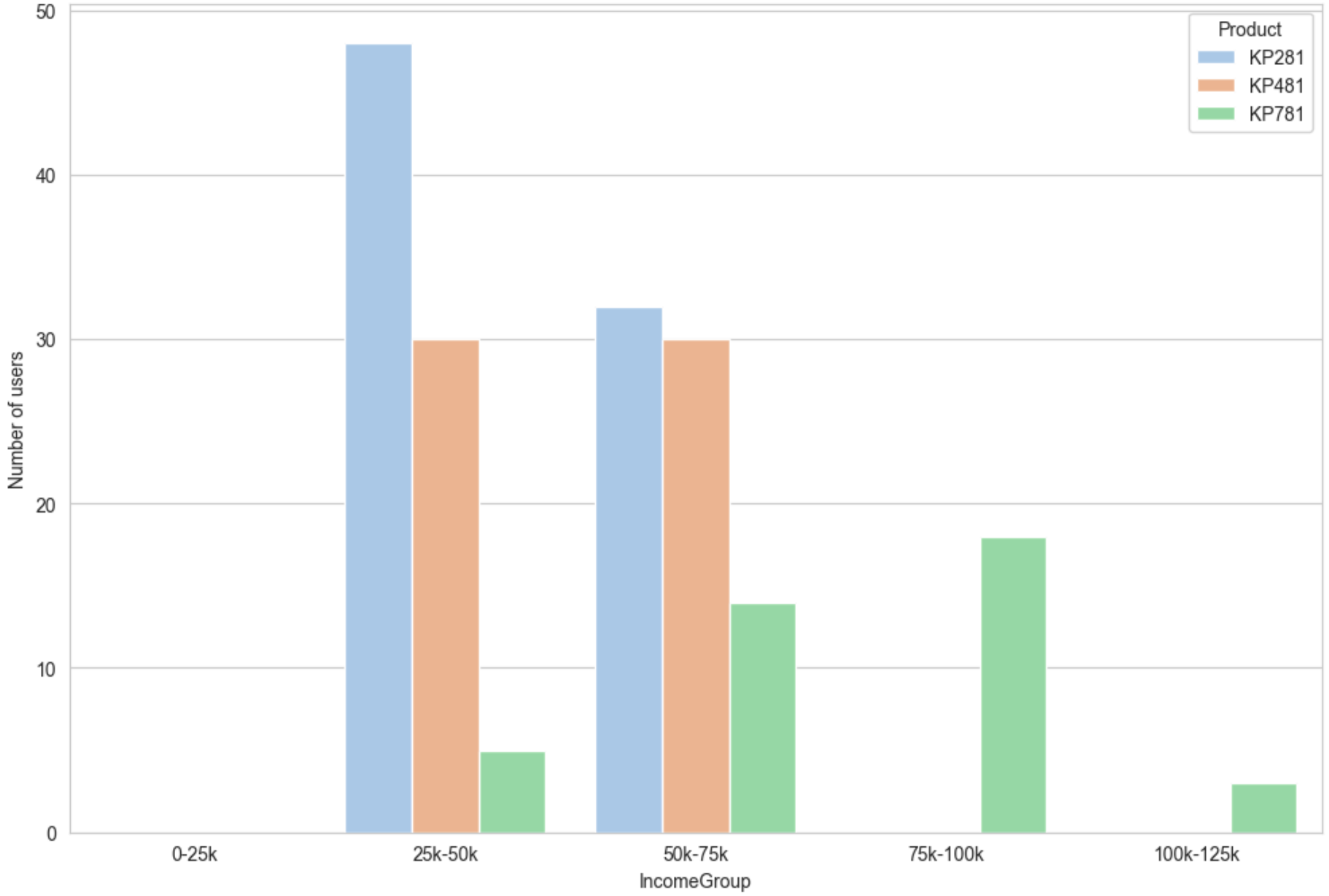
```
#People between 25k to 75k are mostly use the Products
```

In [102…

```
plt.figure(figsize = (12,8))

sns.countplot(data = data, x = 'IncomeGroup', hue = "Product", palette = 'pastel')
plt.xlabel('IncomeGroup')
plt.ylabel('Number of users')

plt.show()
```



In [104…

```
#Customer with Income more than 75 mostly using the higherend product which is KP781 (Higher price)
#Customer with Income less than 75 mostly using the lowerend product which is KP281 (Higher price)
```

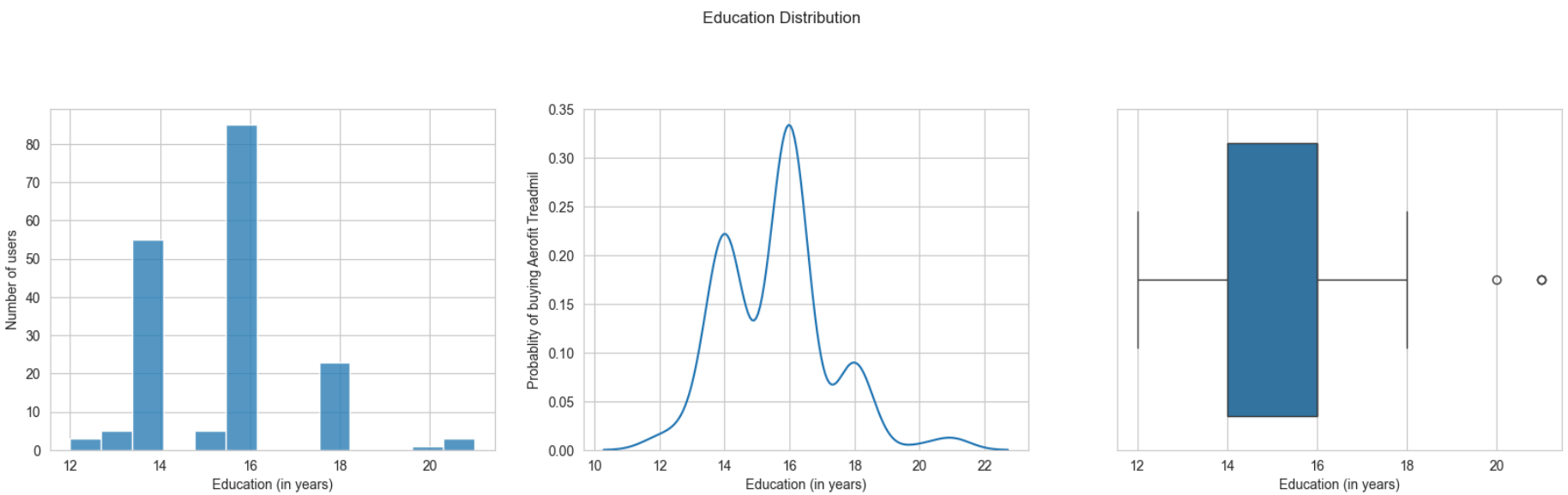
In [105…

```
plt.figure(figsize = (20,10))

# Histogram
plt.subplot(2,3,1)
sns.histplot(data = data, x = 'Education')
plt.xlabel('Education (in years)')
plt.ylabel('Number of users')

# KDE plot
plt.subplot(2,3,2)
sns.kdeplot(data = data, x = 'Education')
plt.xlabel('Education (in years)')
plt.ylabel('Probablity of buying Aerofit Treadmil')

#Boxplot
plt.subplot(2,3,3)
sns.boxplot(data = data, x = 'Education')
plt.xlabel('Education (in years)')
plt.suptitle('Education Distribution')
plt.show()
```



In [106…
```
#Customer with education year between 14 to 16 are the more users of the Product
```

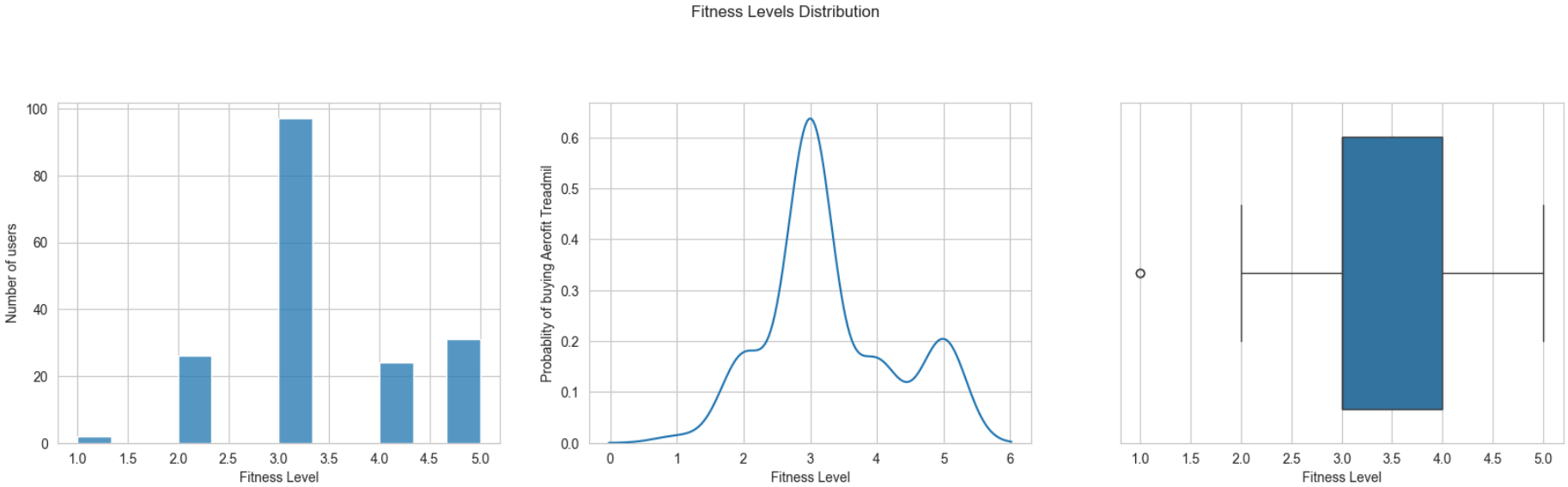
In [107…
```
#Customer with education more than 18 are less likely to purchase a Product
```

In [109…
```
plt.figure(figsize = (20,10))

# Histogram
plt.subplot(2,3,1)
sns.histplot(data = data, x = 'Fitness')
plt.xlabel('Fitness Level')
plt.ylabel('Number of users')

# KDE plot
plt.subplot(2,3,2)
sns.kdeplot(data = data, x = 'Fitness')
plt.xlabel('Fitness Level')
plt.ylabel('Probablity of buying Aerofit Treadmil')

#Boxplot
plt.subplot(2,3,3)
sns.boxplot(data = data, x = 'Fitness')
plt.xlabel('Fitness Level')
plt.suptitle('Fitness Levels Distribution')
plt.show()
```



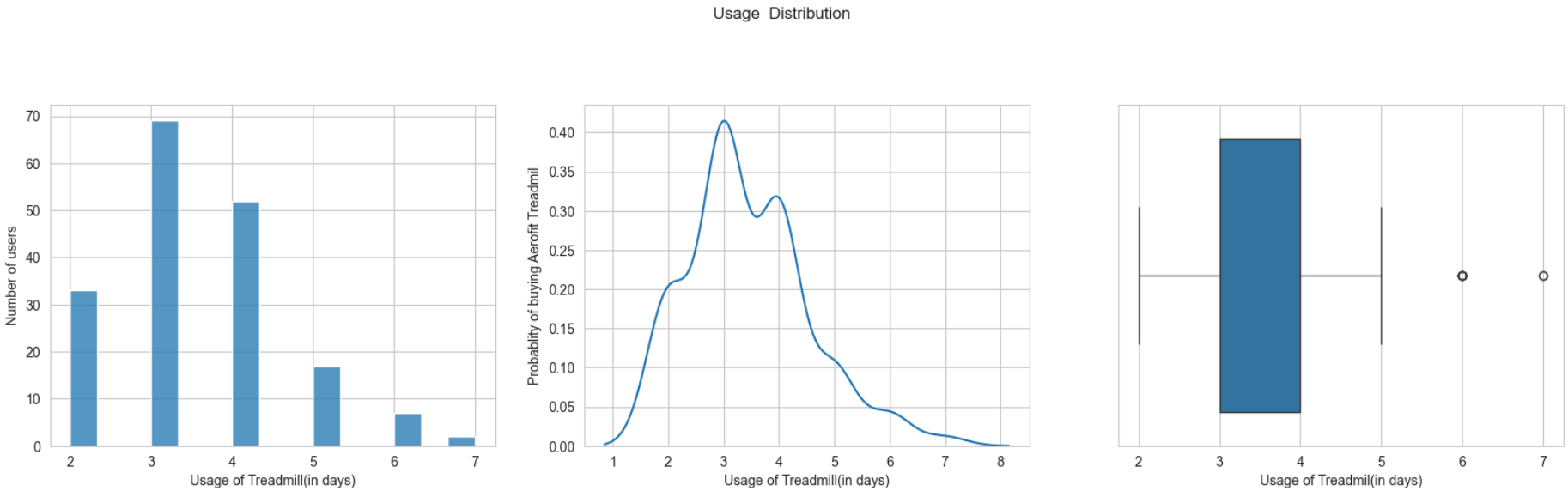
In [110…
```
#Majority of the customer are possess fitness level 3
```

In [112…
```
plt.figure(figsize = (20,10))

# Histogram
plt.subplot(2,3,1)
sns.histplot(data = data, x = 'Usage')
plt.xlabel('Usage of Treadmill(in days)')
plt.ylabel('Number of users')

# KDE plot
plt.subplot(2,3,2)
sns.kdeplot(data = data, x = 'Usage')
plt.xlabel('Usage of Treadmill(in days)')
plt.ylabel('Probablity of buying Aerofit Treadmil')

#Boxplot
plt.subplot(2,3,3)
sns.boxplot(data = data, x = 'Usage')
plt.xlabel('Usage of Treadmil(in days)')
plt.suptitle('Usage  Distribution')
plt.show()
```



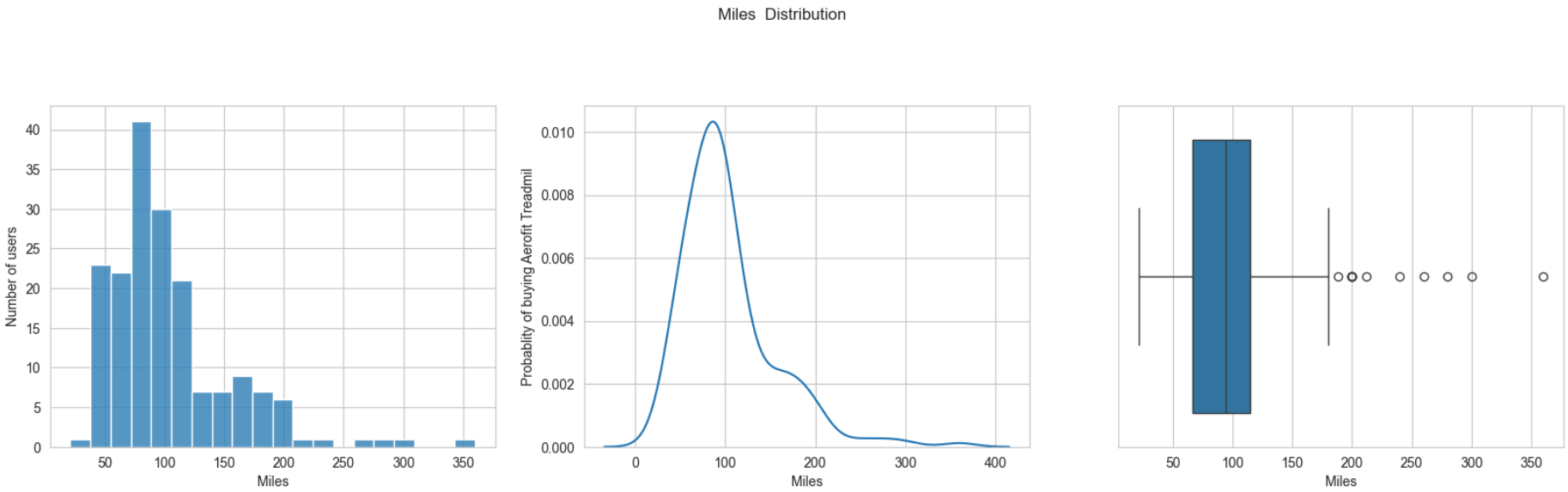
In [113…
```
#The majority of customers use treadmills three times a week
```

In [114…
```
plt.figure(figsize = (20,10))

# Histogram
plt.subplot(2,3,1)
sns.histplot(data = data, x = 'Miles')
plt.xlabel('Miles')
plt.ylabel('Number of users')

# KDE plot
plt.subplot(2,3,2)
sns.kdeplot(data = data, x = 'Miles')
plt.xlabel('Miles')
plt.ylabel('Probablity of buying Aerofit Treadmil')

#Boxplot
plt.subplot(2,3,3)
sns.boxplot(data = data, x = 'Miles')
plt.xlabel('Miles')
plt.suptitle('Miles  Distribution')
plt.show()
```



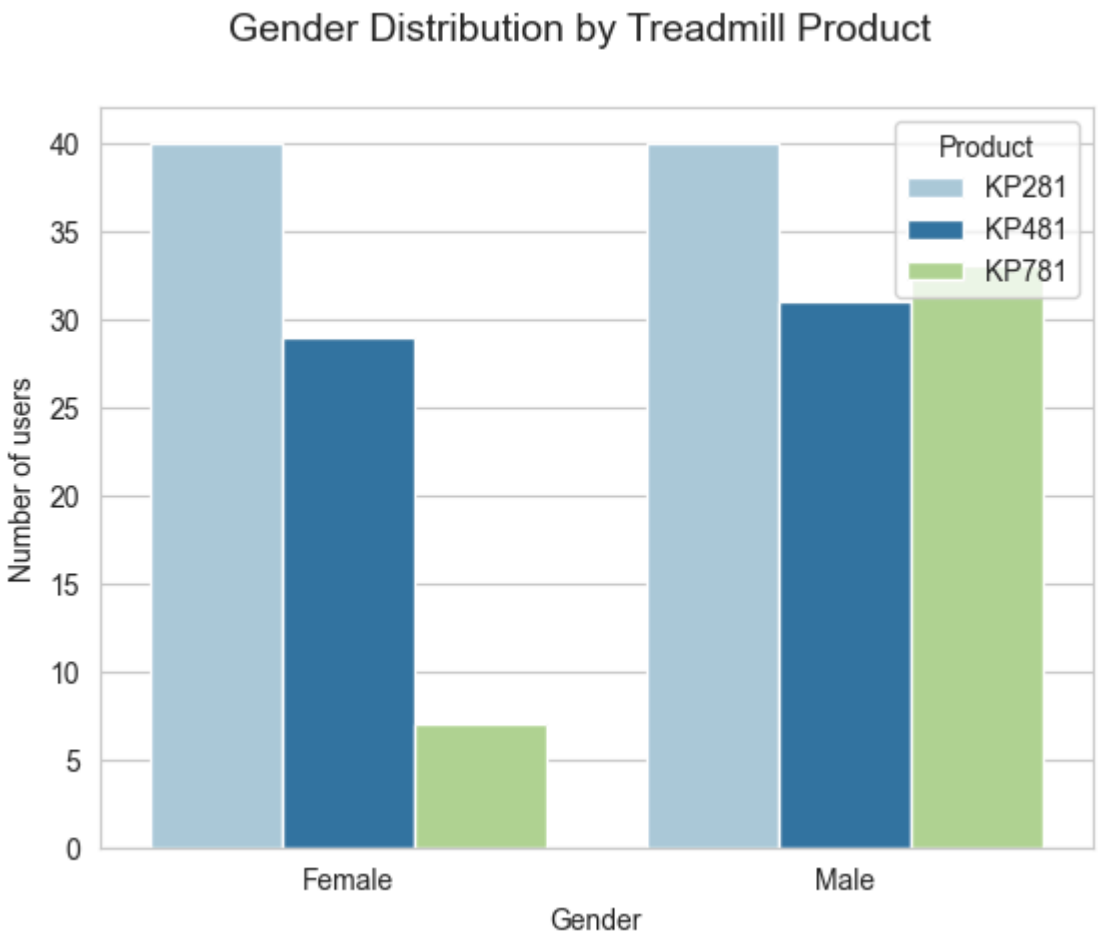
In [115…
```
#Majority of the customer prefere 90-100 miles per week
```

In [117…
```
gender_data = data.groupby(['Product', 'Gender']).size().unstack()
gender_data
```

Out[117…

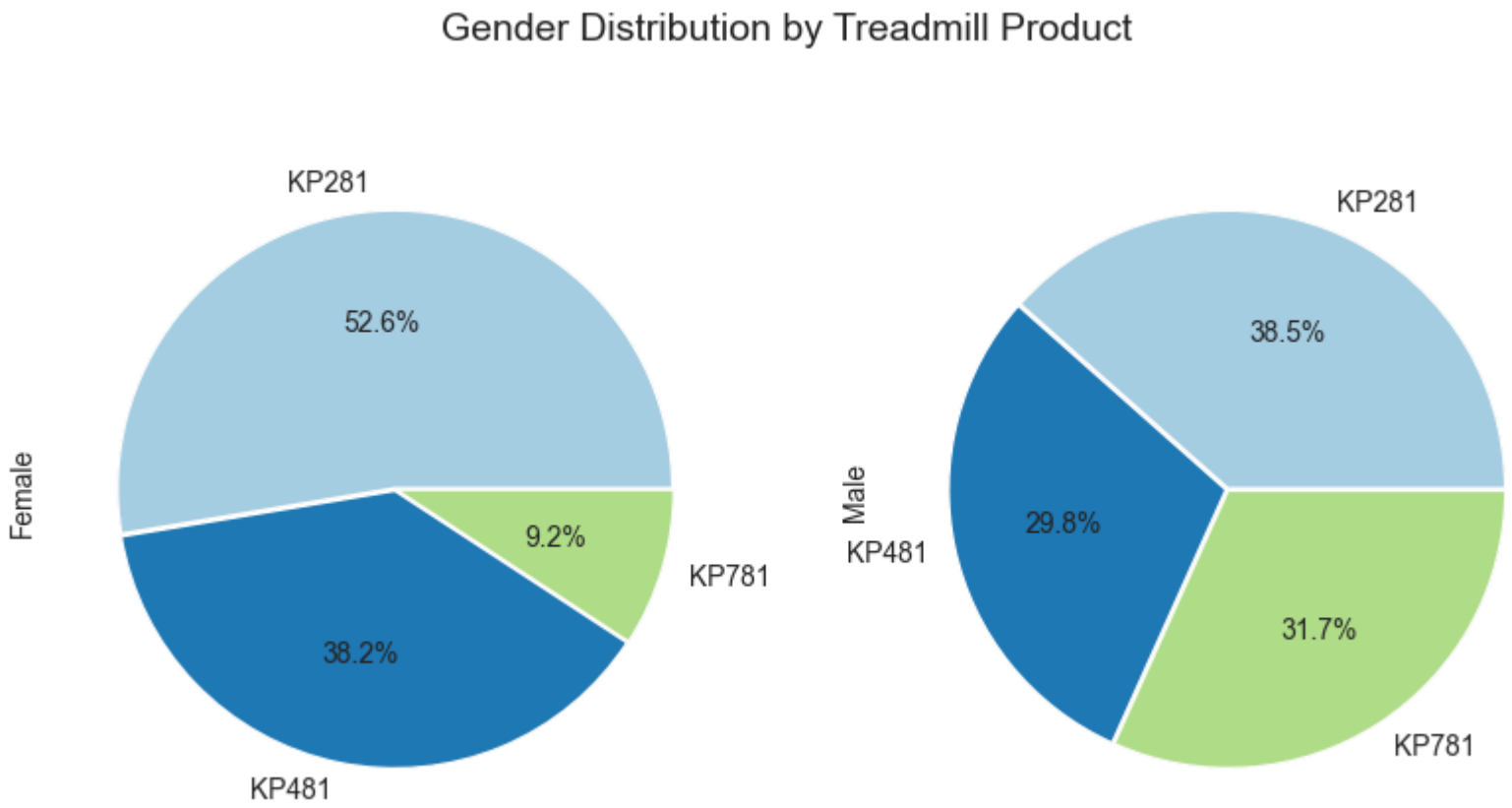
Product	Female	Male
KP281	40	40
KP481	29	31
KP781	7	33

In [119…
```
sns.countplot(data=data,x='Gender',hue='Product',palette='Paired')
plt.suptitle('Gender Distribution by Treadmill Product', fontsize=14)
plt.xlabel('Gender')
plt.ylabel('Number of users')
plt.show()
```



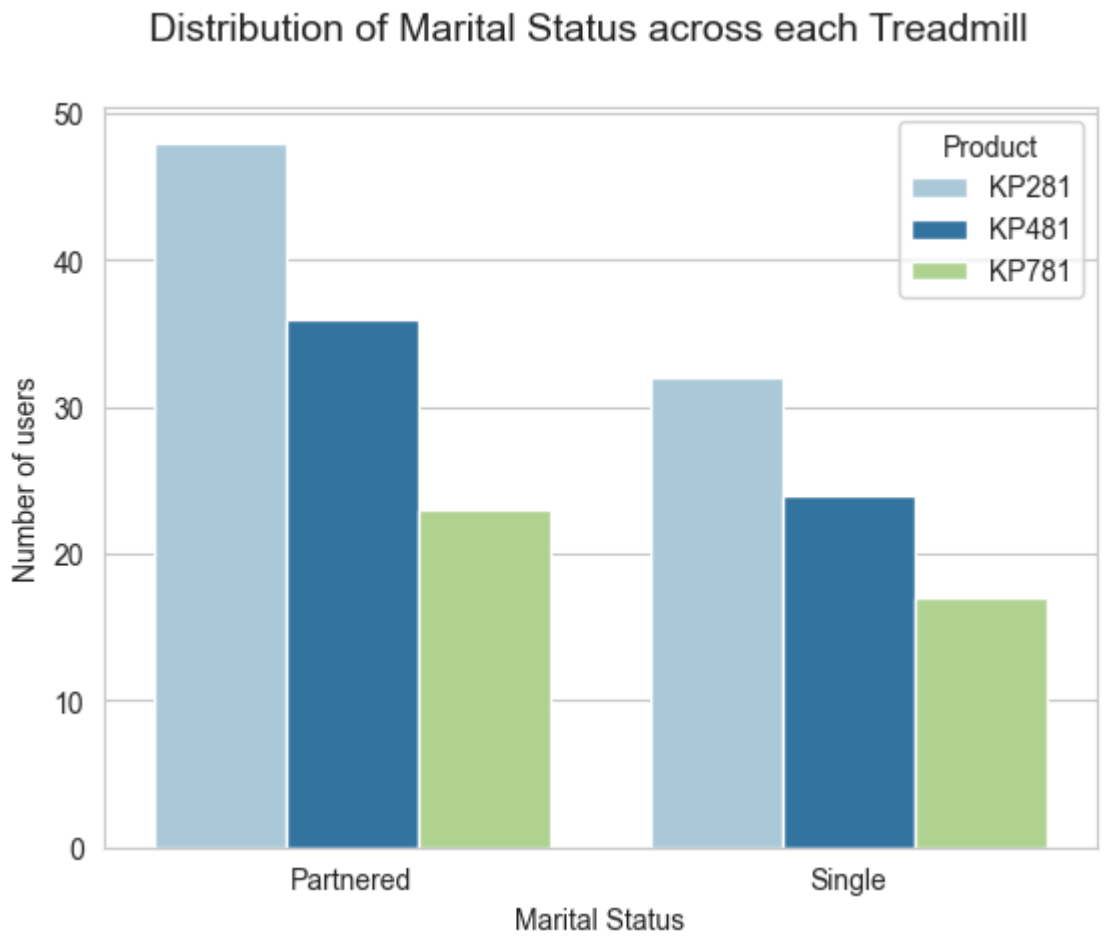
In [120…
```
#Product KP281 is more prefered by Male and Female
#Product KP781 is more prefered by Male than Female
```

In [121…
```
gender_data.plot(kind='pie',subplots=True ,figsize=(10,5),explode=(0.005,0.005,0.005), autopct='%1.1f%%', legend=False,colors=
plt.suptitle('Gender Distribution by Treadmill Product', fontsize=14)
plt.show()
```



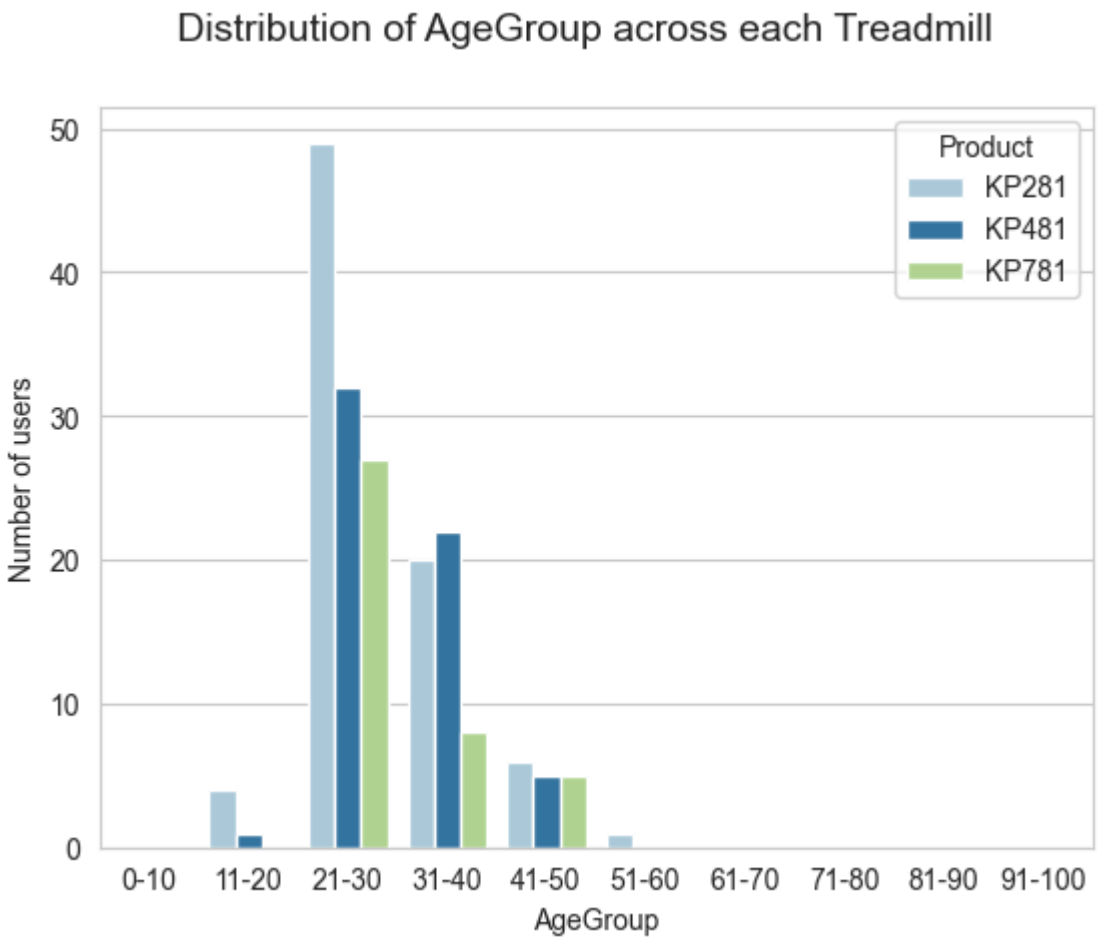
In [122…

```
sns.countplot(data=data, x='MaritalStatus',hue='Product',palette='Paired')
plt.suptitle(' Distribution of Marital Status across each Treadmill', fontsize=14)
plt.xlabel('Marital Status')
plt.ylabel('Number of users')
plt.show()
```



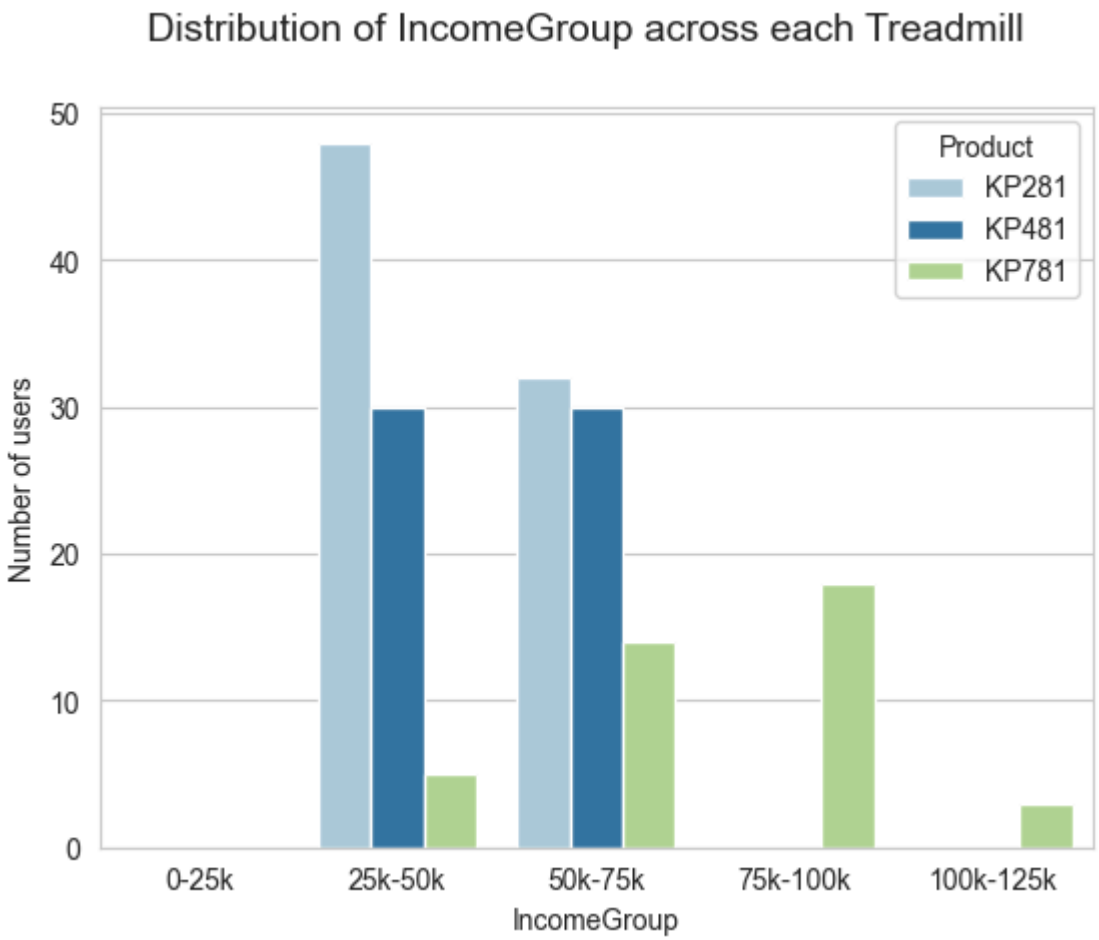
In [123…
```
#Married customers have a higher frequency of purchasing all treadmills compared to single customers.
```

In [172…
```
sns.countplot(data=data, x='AgeGroup',hue='Product',palette='Paired')
plt.suptitle(' Distribution of AgeGroup across each Treadmill', fontsize=14)
plt.xlabel('AgeGroup')
plt.ylabel('Number of users')
plt.show()
```



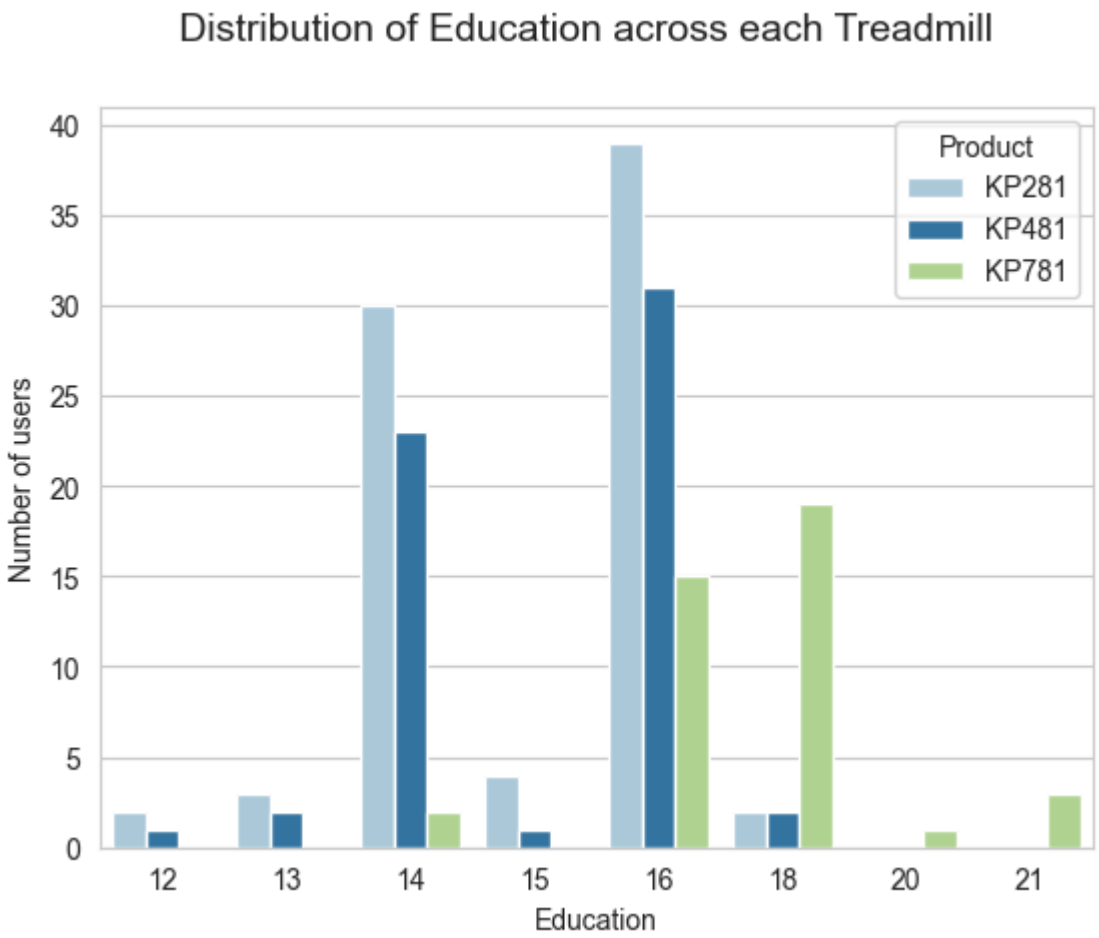
In [176…

```
sns.countplot(data=data, x='IncomeGroup',hue='Product',palette='Paired')
plt.suptitle(' Distribution of IncomeGroup across each Treadmill', fontsize=14)
plt.xlabel('IncomeGroup')
plt.ylabel('Number of users')
plt.show()
```



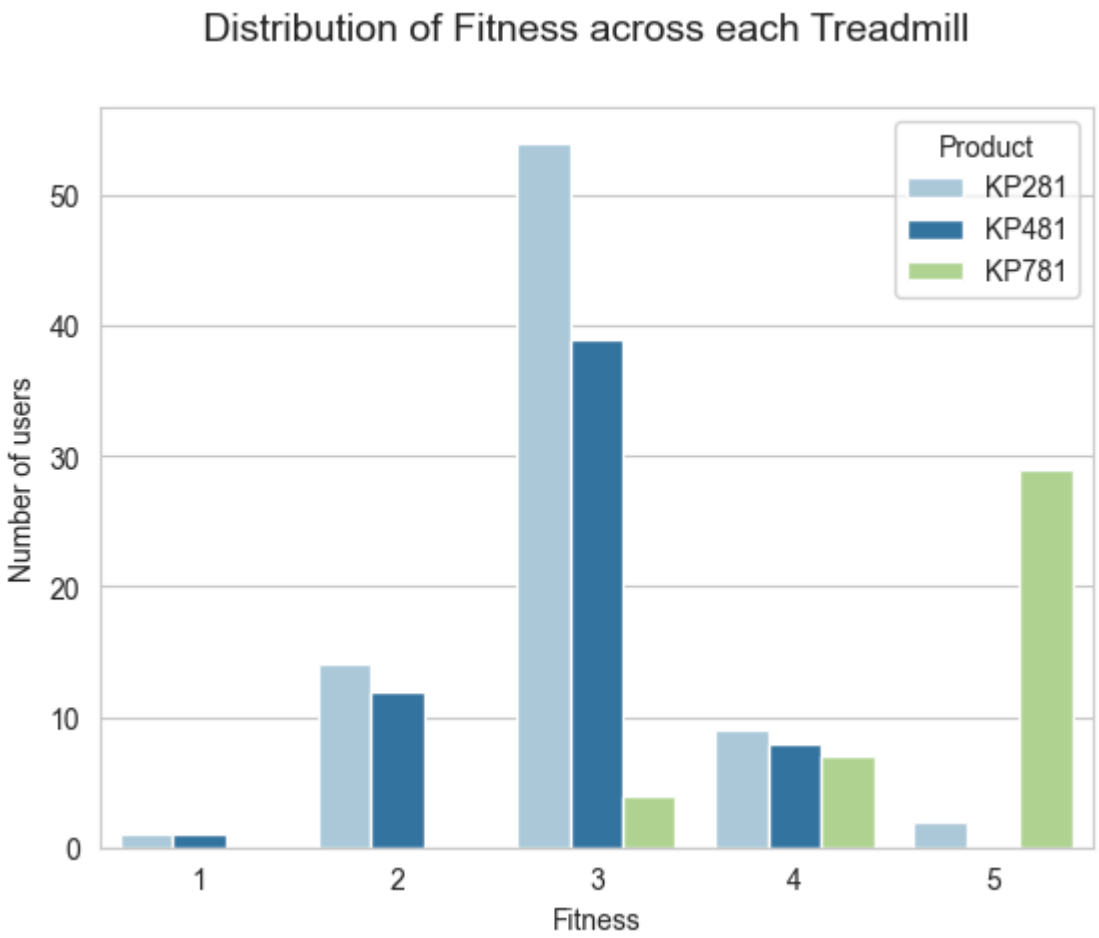
In [177…

```
sns.countplot(data=data, x='Education',hue='Product',palette='Paired')
plt.suptitle(' Distribution of Education across each Treadmill', fontsize=14)
plt.xlabel('Education')
plt.ylabel('Number of users')
plt.show()
```



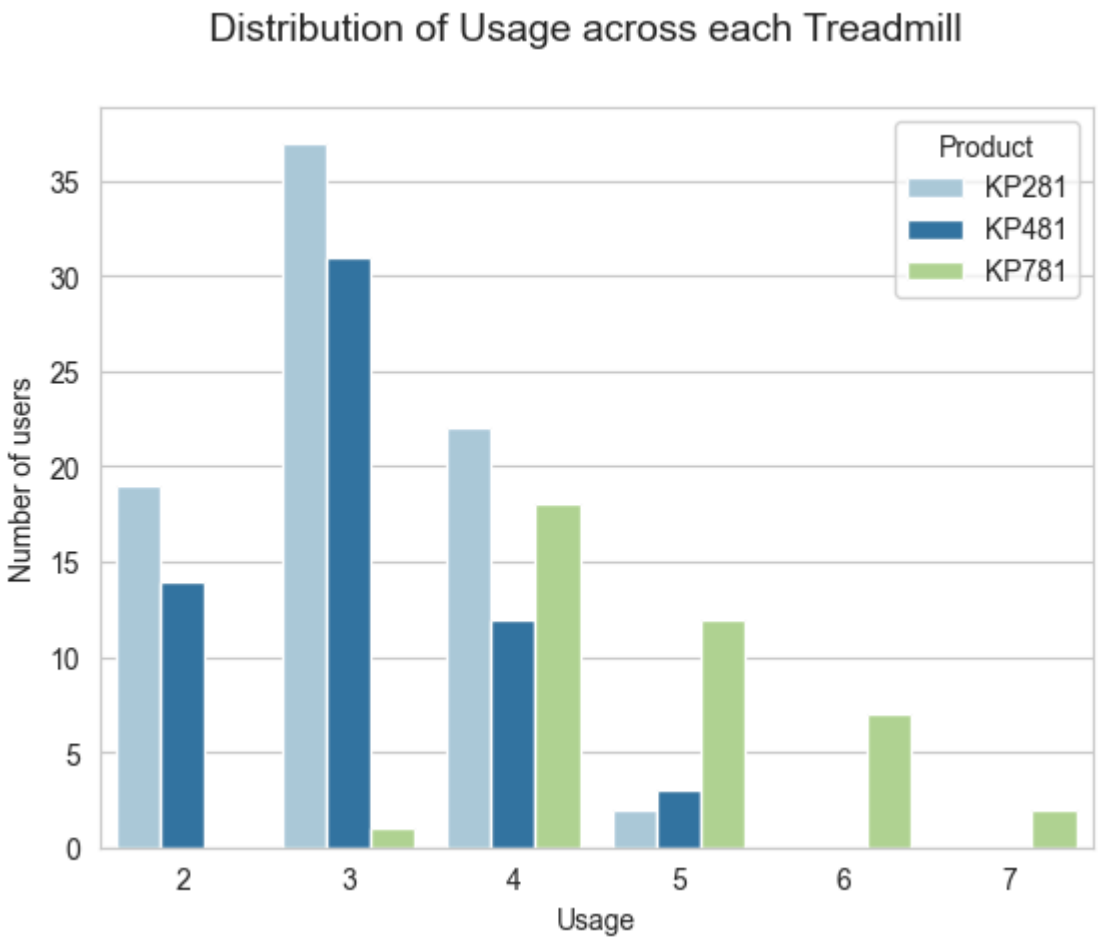
In [179…

```
sns.countplot(data=data, x='Fitness',hue='Product',palette='Paired')
plt.suptitle(' Distribution of Fitness across each Treadmill', fontsize=14)
plt.xlabel('Fitness')
plt.ylabel('Number of users')
plt.show()
```



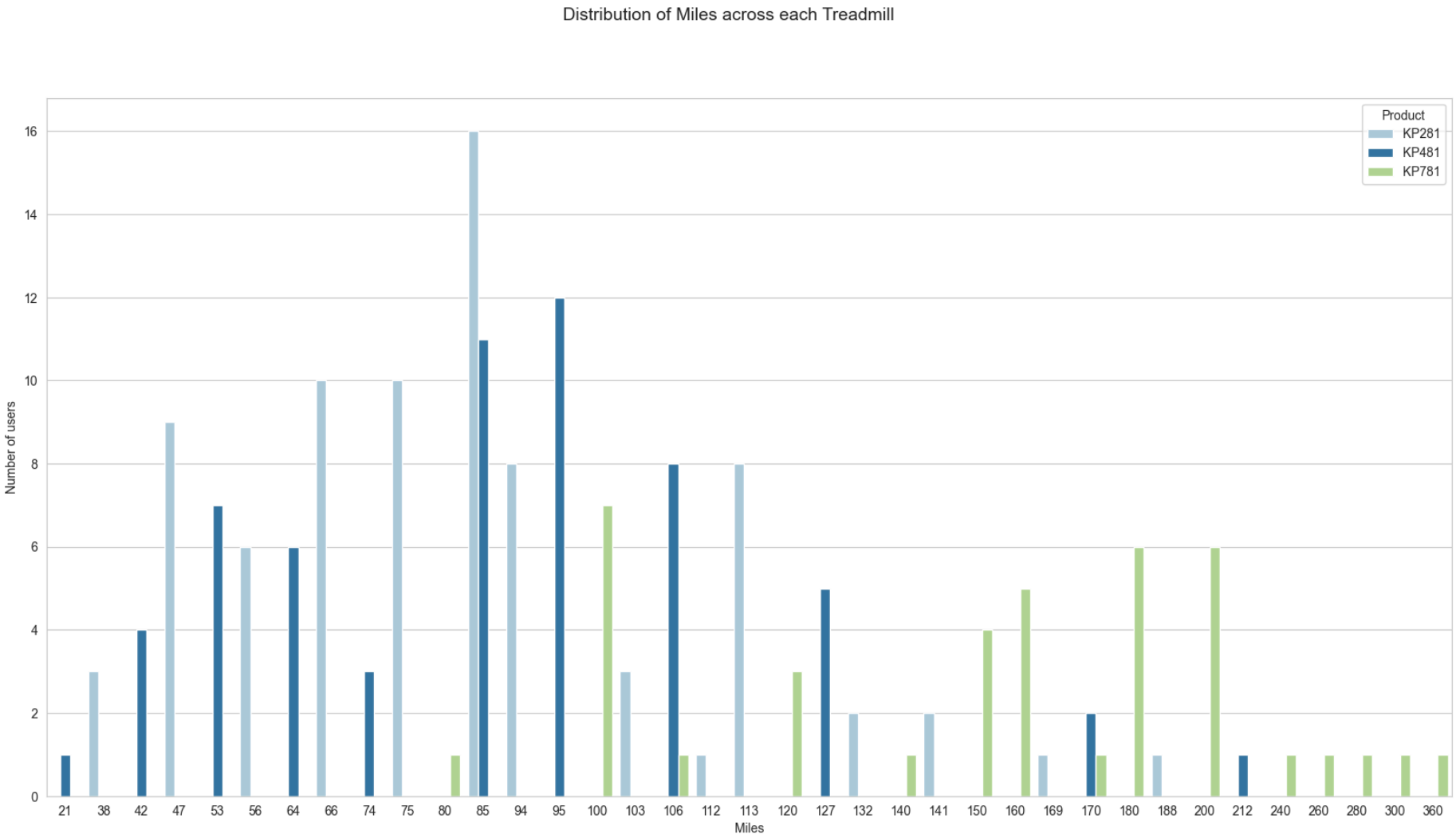
In [180…

```
sns.countplot(data=data, x='Usage',hue='Product',palette='Paired')
plt.suptitle(' Distribution of Usage across each Treadmill', fontsize=14)
plt.xlabel('Usage')
plt.ylabel('Number of users')
plt.show()
```



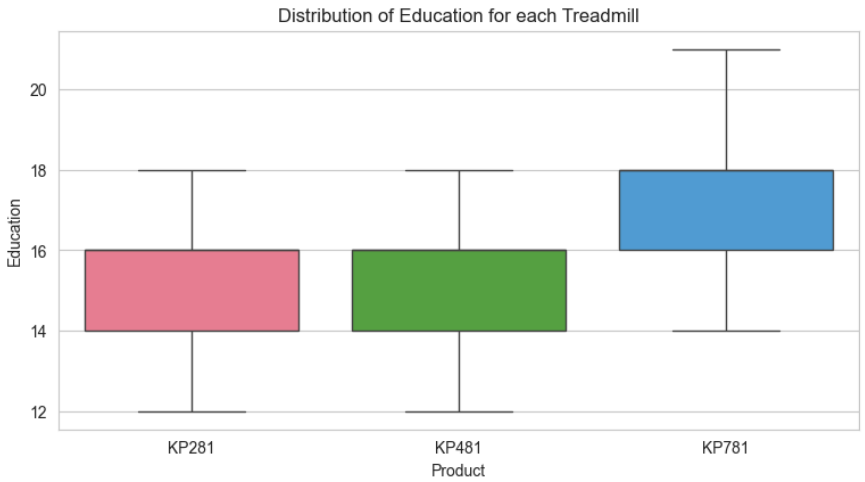
In [182…

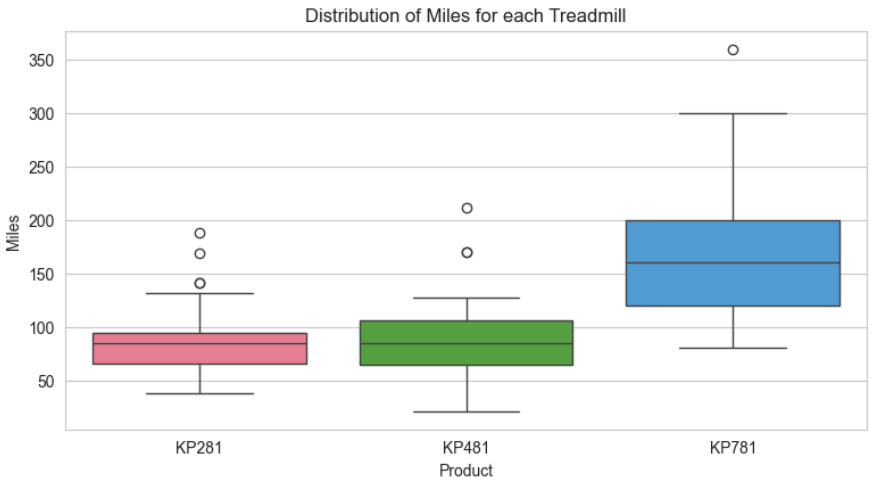
```
plt.figure(figsize=(20,10))
sns.countplot(data=data, x='Miles',hue='Product',palette='Paired')
plt.suptitle(' Distribution of Miles across each Treadmill', fontsize=14)
plt.xlabel('Miles')
plt.ylabel('Number of users')
plt.show()
```



In [124…

```
columns=['Education','Miles']
plt.figure(figsize=(20,10))
for i,col in enumerate(columns,1):
    plt.subplot(2,2,i)
    sns.boxplot(data=data,x='Product',y=col,palette='husl')
    plt.title(f'Distribution of {col} for each Treadmill')
plt.show()
```



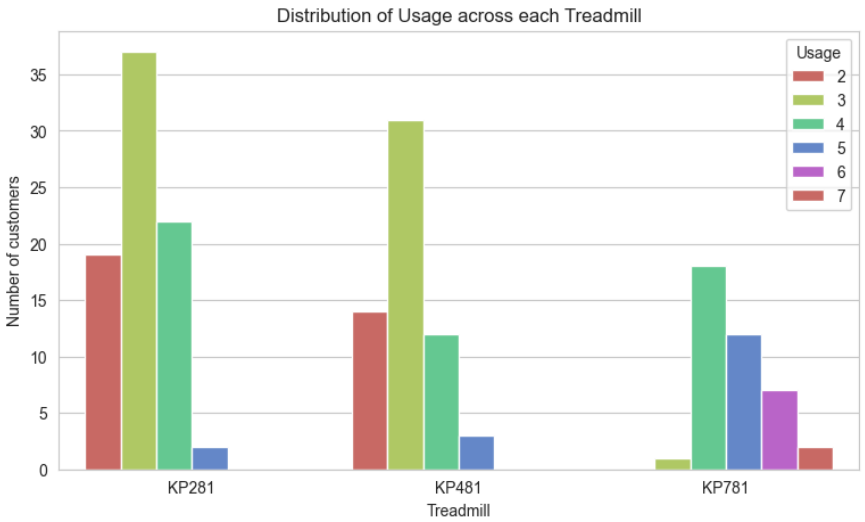


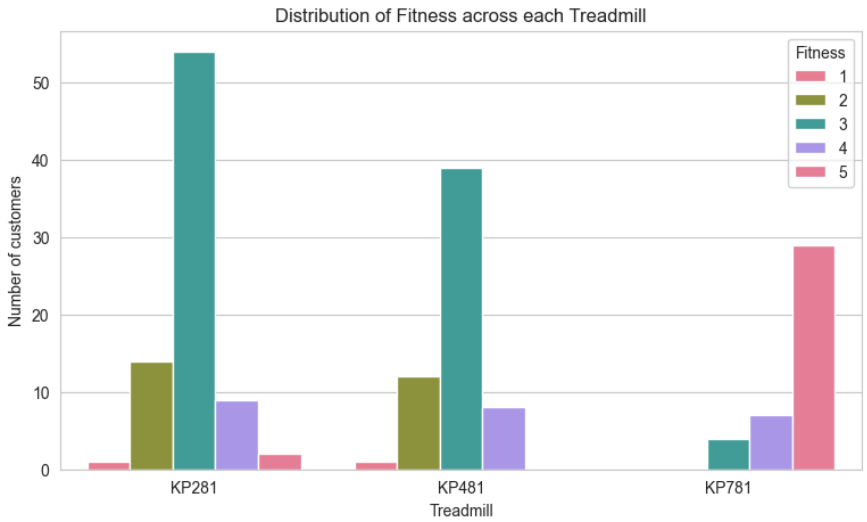
In [125…

```
plt.figure(figsize = (20,5))
# Usage column
plt.subplot(1,2,1)
sns.countplot(data = data, x = 'Product', hue = 'Usage', palette = 'hls')
plt.xlabel('Treadmill')
plt.ylabel('Number of customers')
plt.title('Distribution of Usage across each Treadmill')

# Fitness column
plt.subplot(1,2,2)
sns.countplot(data = data, x = 'Product', hue = 'Fitness', palette = 'husl')
plt.xlabel('Treadmill')
plt.ylabel('Number of customers')
plt.title('Distribution of Fitness across each Treadmill')

plt.show()
```





In [126…
```
#Customers who use treadmills 3 times a week prefer both KP281 and KP481
```

In [127…
```
#Customers who use treadmills 4-5 times a week favor the KP781 treadmill
```

In [128…
```
#Customers with fitness level 3 prefer both KP281 and KP481 treadmills
```

In [129…
```
#Customers with fitness level 5 predominantly use the most advanced KP781 treadmill
```

In [131…
```
#Conditional and Marginal Probablities
```

In [130…
```
pd.crosstab(index=data['Product'],columns=data['Gender'],margins=True,margins_name='Total',normalize=True).round(2)
```

Out[130…

Gender	Female	Male	Total
Product			
KP281	0.22	0.22	0.44
KP481	0.16	0.17	0.33
KP781	0.04	0.18	0.22
Total	0.42	0.58	1.00

In [133…

```
#Marginal Probablities

#    P(KP281) = 0.44
#    P(KP481) = 0.33
#    P(KP781) = 0.22
#    P(Male)  = 0.58
#    P(Female)= 0.42
```

In [135…

```
#Conditional Probablities

#    P(KP281|Male)  = 0.22
#    P(KP281|Female)= 0.22
#    P(KP481|Male)  = 0.17
#    P(KP481|Female)= 0.16
#    P(KP781|Male)  = 0.18
#    P(KP781|Female)= 0.04
```

In [136…

```
pd.crosstab(index=data['Product'],columns=data['MaritalStatus'],margins=True,margins_name='Total',normalize=True).round(2)
```

Out[136…

MaritalStatus	Partnered	Single	Total
Product			
KP281	0.27	0.18	0.44
KP481	0.20	0.13	0.33
KP781	0.13	0.09	0.22
Total	0.59	0.41	1.00

In [138…

```
#Marginal Probablities

#    P(KP281)   = 0.44
#    P(KP481)   = 0.33
#    P(KP781)   = 0.22
#    P(Patnered)= 0.59
#    P(Single)  = 0.41
```

In [139…

```
#Conditional Probablities:

#    P(KP281|Partnered)= 0.27
#    P(KP281|Single)   = 0.18
#    P(KP481|Partnered)= 0.20
#    P(KP481|Single)   = 0.13
#    P(KP781|Partnered)= 0.13
#    P(KP781|Single)   = 0.09
```

In [145…

```
data.head()
```

Out[145...

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	AgeGroup	IncomeGroup
0	KP281	18	Male	14	Single	3	4	29562	112	11-20	25k-50k
1	KP281	19	Male	15	Single	2	3	31836	75	11-20	25k-50k
2	KP281	19	Female	14	Partnered	4	3	30699	66	11-20	25k-50k
3	KP281	19	Male	12	Single	3	3	32973	85	11-20	25k-50k
4	KP281	20	Male	13	Partnered	4	2	35247	47	21-30	25k-50k

In [146...
```
pd.crosstab(index=data['Product'],columns=data['AgeGroup'],margins=True,margins_name='Total',normalize=True).round(2)
```

Out[146...

AgeGroup	11-20	21-30	31-40	41-50	51-60	Total
Product						
KP281	0.02	0.27	0.11	0.03	0.01	0.44
KP481	0.01	0.18	0.12	0.03	0.00	0.33
KP781	0.00	0.15	0.04	0.03	0.00	0.22
Total	0.03	0.60	0.28	0.09	0.01	1.00

In [147...
```
pd.crosstab(index=data['Product'],columns=data['IncomeGroup'],margins=True,margins_name='Total',normalize=True).round(2)
```

Out[147...

IncomeGroup	25k-50k	50k-75k	75k-100k	100k-125k	Total
Product					
KP281	0.27	0.18	0.0	0.00	0.44
KP481	0.17	0.17	0.0	0.00	0.33
KP781	0.03	0.08	0.1	0.02	0.22
Total	0.46	0.42	0.1	0.02	1.00

In [150…
```
pd.crosstab(index=data['Product'],columns=data['Education'],margins=True,margins_name='Total',normalize=True).round(2)
```

Out[150…

Education	12	13	14	15	16	18	20	21	Total
Product									
KP281	0.01	0.02	0.17	0.02	0.22	0.01	0.00	0.00	0.44
KP481	0.01	0.01	0.13	0.01	0.17	0.01	0.00	0.00	0.33
KP781	0.00	0.00	0.01	0.00	0.08	0.11	0.01	0.02	0.22
Total	0.02	0.03	0.31	0.03	0.47	0.13	0.01	0.02	1.00

In [151…
```
pd.crosstab(index=data['Product'],columns=data['Usage'],margins=True,margins_name='Total',normalize=True).round(2)
```

Out[151…

Usage	2	3	4	5	6	7	Total
Product							
KP281	0.11	0.21	0.12	0.01	0.00	0.00	0.44
KP481	0.08	0.17	0.07	0.02	0.00	0.00	0.33
KP781	0.00	0.01	0.10	0.07	0.04	0.01	0.22
Total	0.18	0.38	0.29	0.09	0.04	0.01	1.00

In [152…
```
pd.crosstab(index=data['Product'],columns=data['Fitness'],margins=True,margins_name='Total',normalize=True).round(2)
```

Out[152...

Fitness	1	2	3	4	5	Total
Product						
KP281	0.01	0.08	0.30	0.05	0.01	0.44
KP481	0.01	0.07	0.22	0.04	0.00	0.33
KP781	0.00	0.00	0.02	0.04	0.16	0.22
Total	0.01	0.14	0.54	0.13	0.17	1.00

In [154...

```
pd.crosstab(index=data['Product'],columns=data['Miles'],margins=True,margins_name='Total',normalize=True).round(2)
```

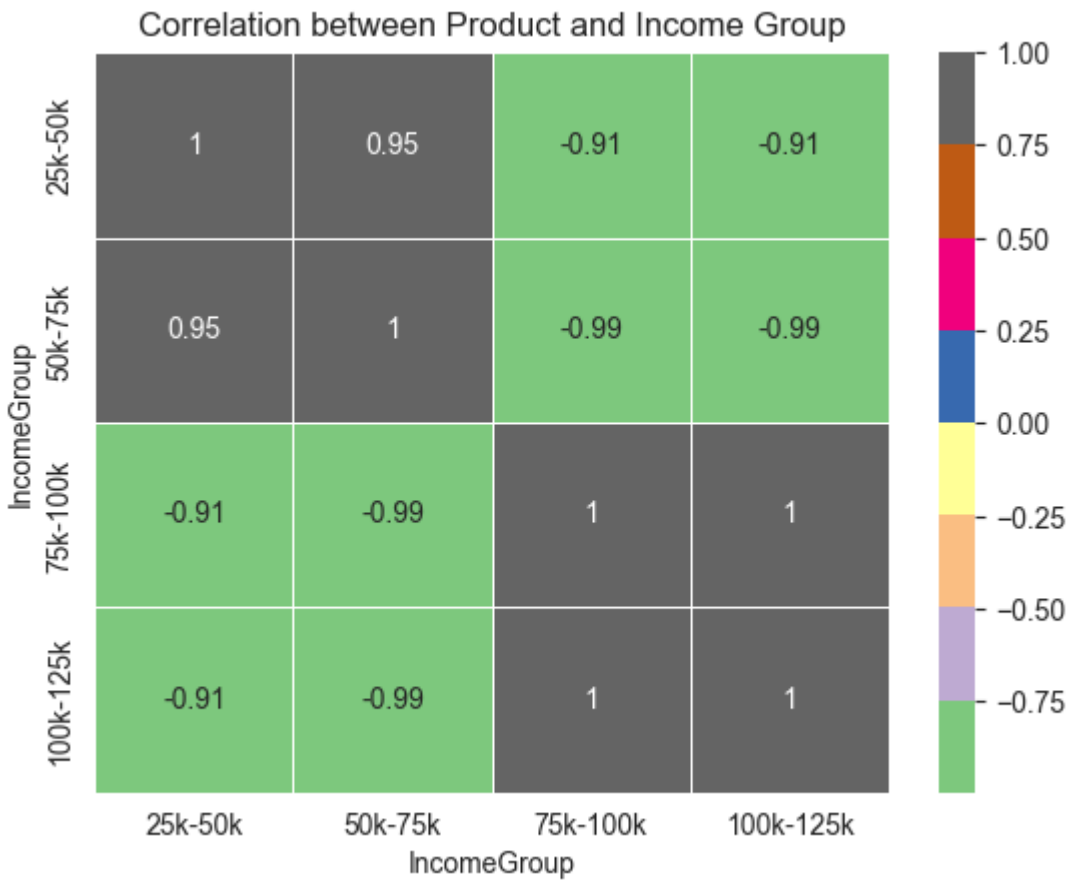
Out[154...

Miles	21	38	42	47	53	56	64	66	74	75	...	180	188	200	212	240	260	280	300	360	Total
Product																					
KP281	0.00	0.02	0.00	0.05	0.00	0.03	0.00	0.06	0.00	0.06	...	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.44
KP481	0.01	0.00	0.02	0.00	0.04	0.00	0.03	0.00	0.02	0.00	...	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.33
KP781	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.03	0.00	0.03	0.00	0.01	0.01	0.01	0.01	0.01	0.22
Total	0.01	0.02	0.02	0.05	0.04	0.03	0.03	0.06	0.02	0.06	...	0.03	0.01	0.03	0.01	0.01	0.01	0.01	0.01	0.01	1.00

4 rows × 38 columns

In [169...

```
pivot_table = pd.crosstab(data['Product'], data['IncomeGroup'])
correlation_matrix = pivot_table.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='Accent', linewidths=.5)
plt.title('Correlation between Product and Income Group')
plt.show()
```



In [192…

```
#Gender: Male and Female
#Marital status: Partnered and Single
#Age: AgeGroup created with bins of 10
#Income: IncomeGroup Created with bins of 25k
```

In [193…

```
#There are no duplicated and no null values observed
```

In [194…

```
#Customer Profiles
#Product: KP281
#Gender : Male and Female
#Marital status: Both Partnered and single
#Age: All the age groups uses this product, but more prefered between 21-40
```

```
#Income: IncomeGroup between 25k to 75k uses the product
#Education: Year between 11 to 18 using the product, more prefered by 14 an 16
#Fitness level: Prefered by fitness level 3
#Usage: Prefered by usage 3 times
#Miles: Prefered for lower miles

#Product: KP481
#Gender : Male and Female
#Marital status: Both Partnered and single
#Age: Agegroup between 11-50 used this product, more prefered in 21-40
#Income: IncomeGroup between 25k to 75k uses the product
#Education: Year between 11 to 18 using the product, more prefered by 14 and 16
#Fitness level: Prefered by fitness level 3
#Usage: Prefered by usage 3 times
#Miles: Prefered for lower miles

#Product: KP781
#Gender : Male and Female
#Marital status: Both Partnered and single
#Age: Agegroup between 21-50 used this product, more prefered in 21-30
#Income: IncomeGroup between 25k to 125k uses the product, more prefered by high income customers
#Education: More prefered by 16 and 18
#Fitness level: Prefered by fitness level 5
#Usage: Prefered by usage 4 and above times
#Miles: Prefered for higher miles
```

In [195…

```
#Insights
#1. 44.44% prefer using the KP281, 33.33% prefer using the KP481 and 22.22% prefer using the KP781
#2. Price of the product increase in the order of KP281, KP481, KP781 respectively
#3. Aerofit has more Male customers (57.78%) than female customers (42.22%)
#4. 'KP281', 'KP481', 'KP781' are the Products available
#5. Male and Female are prefered to by all the 3 products
#6. Both Partnered and single prefered to by all the 3 products
#7. There are age group data between 18 to 42 years
#8. Education in years are between 14 to 21 years
#9. The average number of times the customer plans to use the treadmill each week are between 2 to 7
#10. Fitness rating is between 1 to 5
#11. Minimum salary is 29562 and maximum salary 104581 of the customer
#12. Majority of the customers use Product KP281, then KP481, then KP781 respectively
#13. Customers prefered to use less priced products
```

```
#14. People in age group 21-30 use more products, then 31-40 then 41-50, then 11-20
#15. 59.44% of customers are marries people and others are Single
#16. People between 25k to 75k are the most customer use the Products
#17. Customer with Income more than 75 mostly using the higherend product which is KP781 (Higher price)
#18. Customer with Income less than 75 mostly using the lowerend product which is KP281 (Higher price)
#19. Customer with education year between 14 to 16 are the more users of the Product
#20. Customer with education more than 18 are less likely to purchase a Product
#21. Majority of the customer are possess fitness level 3
#22. The majority of customers use treadmills three times a week
#23. Majority of the customer prefere 90-100 miles per week
#24. Product KP281 is more prefered by Male and Female
#25. Product KP781 is more prefered by Male than Female
#26. Married customers have a higher frequency of purchasing all treadmills compared to single customers
#27. Customers who use treadmills 3 times a week prefer both KP281 and KP481
#28. Customers who use treadmills 4-5 times a week favor the KP781 treadmill
#29. Customers with fitness level 3 prefer both KP281 and KP481 treadmills
#30. Customers with fitness level 5 predominantly use the most advanced KP781 treadmill
#31. Probability of buying the product is mentioned completely in the Jupyter Notebook
#32. Probability of buying KP281 is higher
#33. Most of the Aerofit customer falls under young age-group (18-29)
```

In [196…

```
#Recommendations:
#1. Need to invest more on Budgest friendly products which more customers prefer
#2. Products must satisfy the usage of 3 times a week which more customers prefer
#3. Aerofit must focus more on products which more customers prefer which mentioned in the insights
#4. The products must be gender friendly as Male/Female customers are almost equal in count
#5. Target on all the category of people which mentioned in the insights
#6. Create more engaging products to attract more customers
#7. Add more products which satisfy most of the categories, which attracts more customers
```

In [ ]: