Case Study-2 (Aerofit Treadmil)

October 4, 2024

[34]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme()
from IPython.display import display
sns.factorplot = sns.catplot
# seaborn.factorplot was renamed to seaborn.catplot in seaborn 0.9
# and has been marked as deprecated since then.
# It was definitely removed in seaborn 0.12 (see release notes).
```

[3]:
```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/
↪original/aerofit_treadmill.csv
```

```
Downloading…
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/ori
ginal/aerofit_treadmill.csv
To: C:\Users\Anjali Sharma\Untitled Folder 1\Untitled
Folder\aerofit_treadmill.csv

  0%|          | 0.00/7.28k [00:00<?, ?B/s]
100%|##########| 7.28k/7.28k [00:00<?, ?B/s]
```

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

[2]: (180, 9)

[3]:
```
data.describe()
```

[3]:

	Age	Education	Usage	Fitness	Income	\
count	180.000000	180.000000	180.000000	180.000000	180.000000	
mean	28.788889	15.572222	3.455556	3.311111	53719.577778	
std	6.943498	1.617055	1.084797	0.958869	16506.684226	
min	18.000000	12.000000	2.000000	1.000000	29562.000000	
25%	24.000000	14.000000	3.000000	3.000000	44058.750000	
50%	26.000000	16.000000	3.000000	3.000000	50596.500000	
75%	33.000000	16.000000	4.000000	4.000000	58668.000000	

```
max      50.000000    21.000000     7.000000     5.000000   104581.000000

            Miles
count  180.000000
mean   103.194444
std     51.863605
min     21.000000
25%     66.000000
50%     94.000000
75%    114.750000
max    360.000000
```

[5]:
```python
data.dtypes
```

[5]:
```
Product          object
Age               int64
Gender           object
Education         int64
MaritalStatus    object
Usage             int64
Fitness           int64
Income            int64
Miles             int64
dtype: object
```

[6]:
```python
# Number of unique values in each column
for i in data.columns:
  print(i, ':', data[i].nunique())
```

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

From the above observation, we can conclude that only Income, Miles and Age can be considered as Continuous, the rest of the columns though integers/floats should be considered as categories.

[7]:
```python
# Checking for null values -
data.isnull().sum()
```

[7]:
```
Product          0
Age              0
Gender           0
```

```
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

There aren't any missing values in the dataset.

1 Business Problem

The market research team at AeroFit wants to identify the characteristics of the target audience for each type of treadmill offered by the company, to provide a better recommendation of the treadmills to the new customers. The team decides to investigate whether there are differences across the product with respect to customer characteristics.

Perform descriptive analytics to create a customer profile for each AeroFit treadmill product by developing appropriate tables and charts. For each AeroFit treadmill product, construct two-way contingency tables and compute all conditional and marginal probabilities along with their insights/impact on the business. Dataset

The company collected the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months. The dataset has the following features:

Product Purchased: KP281, KP481, or KP781

Age: In years

Gender: Male/Female

Education: In years

MaritalStatus: Single or partnered

Usage: The average number of times the customer plans to use the treadmill each week.

Income: Annual income (in $)

Fitness: Self-rated fitness on a 1-to-5 scale, where 1 is the poor shape and 5 is the excellent shape.

Miles: The average number of miles the customer expects to walk/run each week

2 Product Portfolio:

The KP281 is an entry-level treadmill that sells for $1,500.

The KP481 is for mid-level runners that sell for $1,750.

The KP781 treadmill is having advanced features that sell for $2,500.

[9]:
```
data['Product'].value_counts()
```

[9]:
```
KP281    80
KP481    60
KP781    40
Name: Product, dtype: int64
```

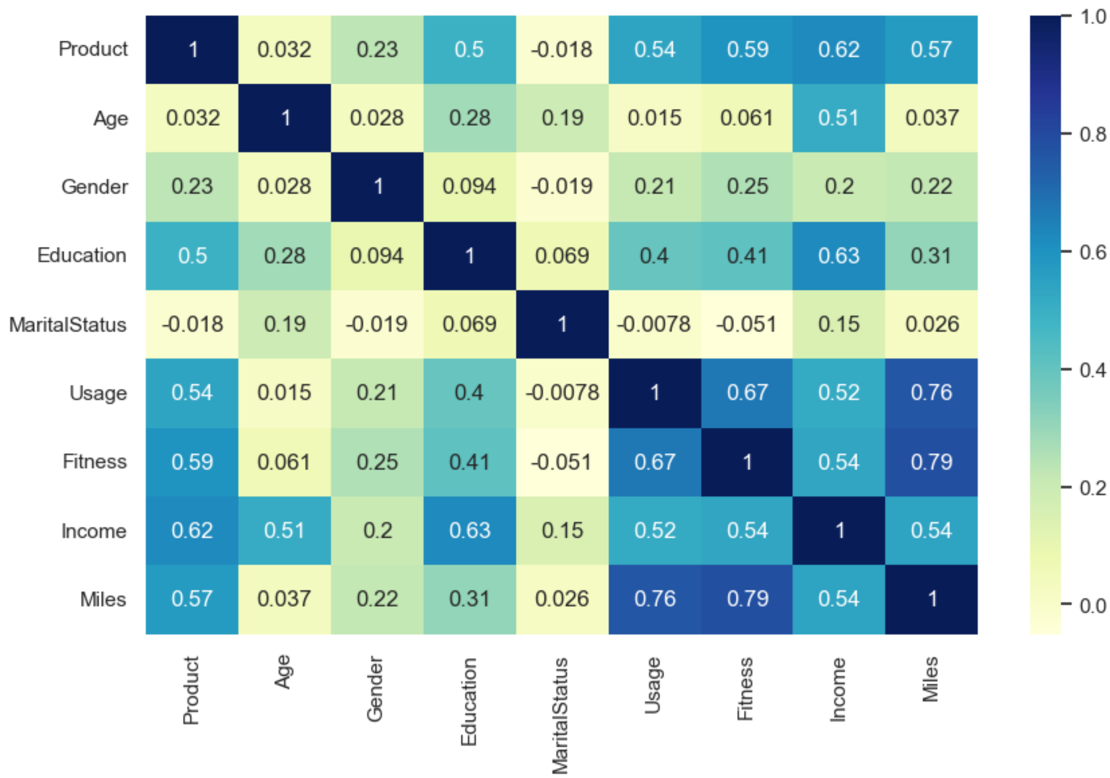
[10]:
```python
# A broader look at correlation between the columns of dataframe

# Creating a copy of the dataframe -
df_copy=data.copy()

df_copy['Gender'].replace(['Male', 'Female'], [1, 0], inplace=True)

df_copy['MaritalStatus'].replace(['Single', 'Partnered'], [0, 1], inplace=True)

df_copy['Product'].replace(['KP281', 'KP481', 'KP781'], [0, 1, 2], inplace=True)

df_copy.corr()
```

[10]:

	Product	Age	Gender	Education	MaritalStatus
Product	1.000000	0.032225	0.230653	0.495018	-0.017602
Age	0.032225	1.000000	0.027544	0.280496	0.192152
Gender	0.230653	0.027544	1.000000	0.094089	-0.018836
Education	0.495018	0.280496	0.094089	1.000000	0.068569
MaritalStatus	-0.017602	0.192152	-0.018836	0.068569	1.000000
Usage	0.537447	0.015064	0.214424	0.395155	-0.007786
Fitness	0.594883	0.061105	0.254609	0.410581	-0.050751
Income	0.624168	0.513414	0.202053	0.625827	0.150293
Miles	0.571596	0.036618	0.217869	0.307284	0.025639

	Usage	Fitness	Income	Miles
Product	0.537447	0.594883	0.624168	0.571596
Age	0.015064	0.061105	0.513414	0.036618
Gender	0.214424	0.254609	0.202053	0.217869
Education	0.395155	0.410581	0.625827	0.307284
MaritalStatus	-0.007786	-0.050751	0.150293	0.025639
Usage	1.000000	0.668606	0.519537	0.759130
Fitness	0.668606	1.000000	0.535005	0.785702
Income	0.519537	0.535005	1.000000	0.543473
Miles	0.759130	0.785702	0.543473	1.000000

[12]:
```python
# Correlation Plot above as a Heatmap -

plt.figure(figsize=(10,6))
sns.heatmap(df_copy.corr(), cmap="YlGnBu", annot=True)
plt.show()
```



3 Noteworthy Points

1. The product/treadmill purchased highly correlates with Education, Income, Usage, Fitness and Miles
2. Age is highly correlated to Income (0.51) which definitely seems reasonable. It's also correlated with Education and Marital Status which stands completely alright.
3. Gender certainly has some correlation to Usage, Fitness, Income and Miles.
4. Education is correlated to Age and Miles. It's highly correlated to Income (as expected). It's sufficiently correlated to Usage and Fitness too.
5. Marital Status has some correlation to Income and Age (as expected).
6. Usage is extremely correlated to Fitness and Miles and has a higher correlation with Income as well.
7. Fitness has a great correlation with Income.

4 More Observations and Possibilities:-

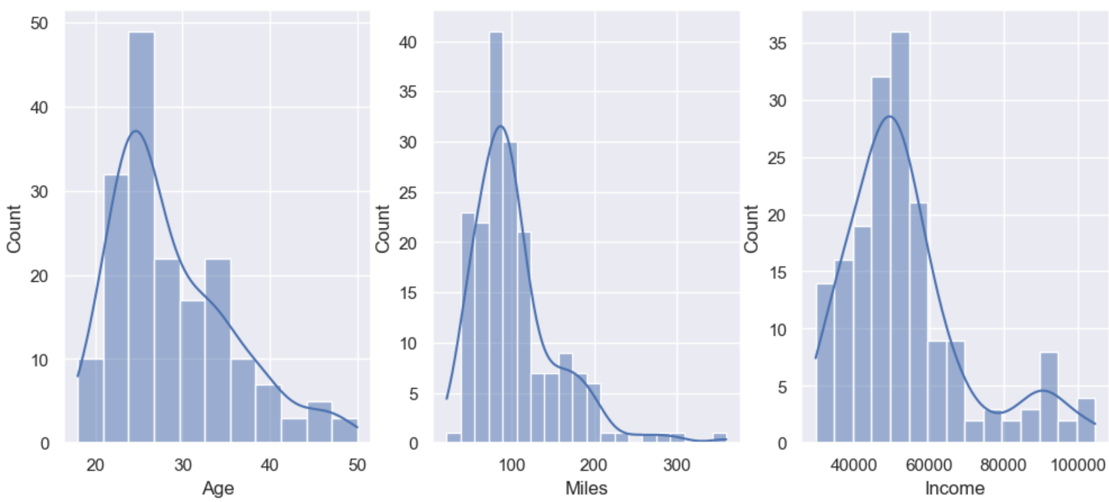
1. Product, Fitness, Usage and Miles depict a ridiculously higher correlation among themselves which looks as expected since more the usage implying more miles run and certainly more

fitness.

2. Also a story which seems reasonable is that Age and Education (predominately) are indicators of Income which affects the products bought. The more advanced the product is, the more its usage and hence more the miles run which in turns improves the fitness.

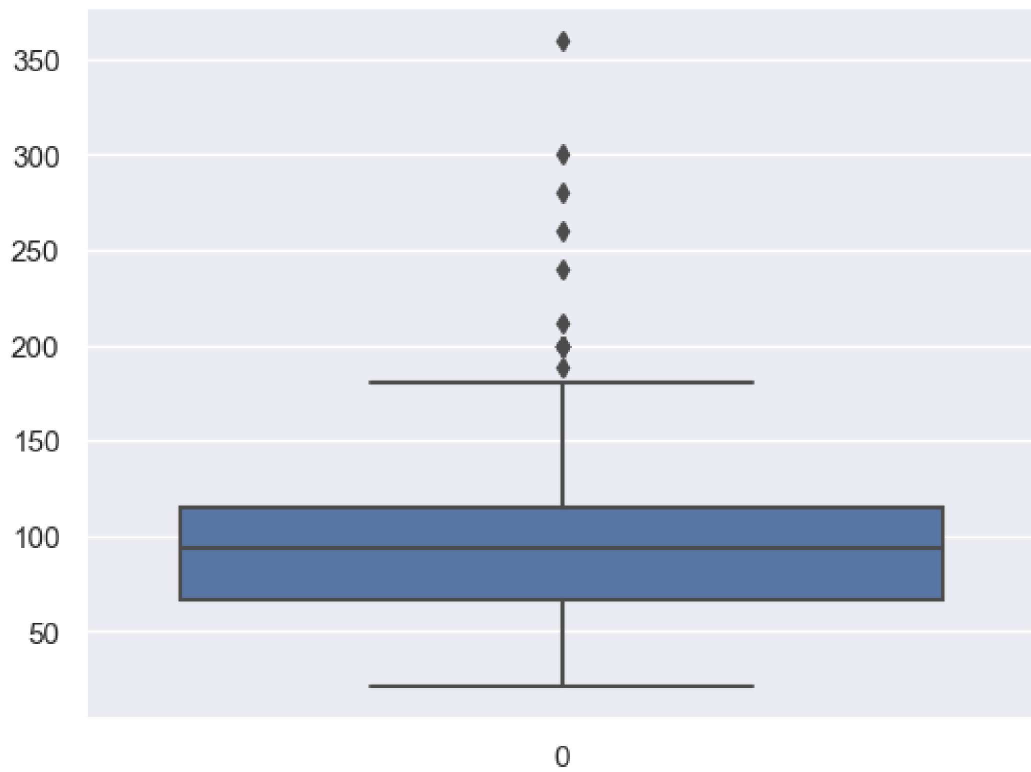
5 Detect outliers

[23]:
```
plt.figure(figsize=(12,5))
plt.subplot(1,3,1)
sns.histplot(data['Age'], kde=True)                    # Skewed distribution which␣
↪shows outliers

plt.subplot(1,3,2)
sns.histplot(data['Miles'], kde=True)

plt.subplot(1,3,3)
sns.histplot(data['Income'], kde=True)
plt.show()
```



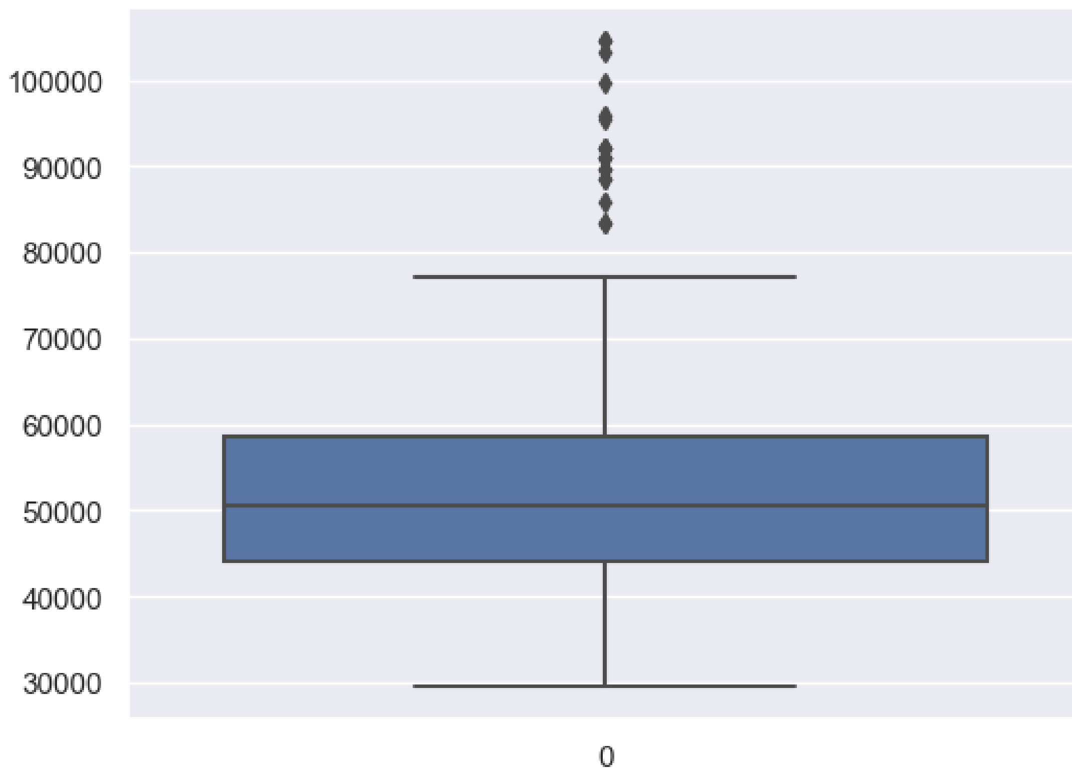
[25]:
```
sns.boxplot(data['Miles'])
```

[25]: <Axes: >



[26]:
```
sns.boxplot(data['Income'])
```

[26]: <Axes: >



[29]:
```
data['Miles'].skew()             # right skewed
```

[29]: 1.7244965928707188

[31]:
```
# Handle outliers
data['Income'].describe()
```

[31]:
```
count       180.000000
mean      53719.577778
std       16506.684226
min       29562.000000
25%       44058.750000
50%       50596.500000
75%       58668.000000
max      104581.000000
Name: Income, dtype: float64
```

[44]:
```
# Finding the IQR
percentile25= data['Income'].quantile(0.25)
percentile75= data['Income'].quantile(0.75)
IQR= percentile75 - percentile25
print(percentile25)
```

```
print(percentile75)
print(IQR)

upper_limit= percentile75 + 1.5*IQR
lower_limit= percentile25 - 1.5*IQR
print(upper_limit)
print(lower_limit)
```

```
44058.75
58668.0
14609.25
80581.875
22144.875
```

6 Finding Outliers

[ ]:
```
data[data['Income']>upper_limit].head(3)
```

[37]:
```
 data[data['Income']<lower_limit]
```

[37]:
```
Empty DataFrame
Columns: [Product, Age, Gender, Education, MaritalStatus, Usage, Fitness,
Income, Miles]
Index: []
```

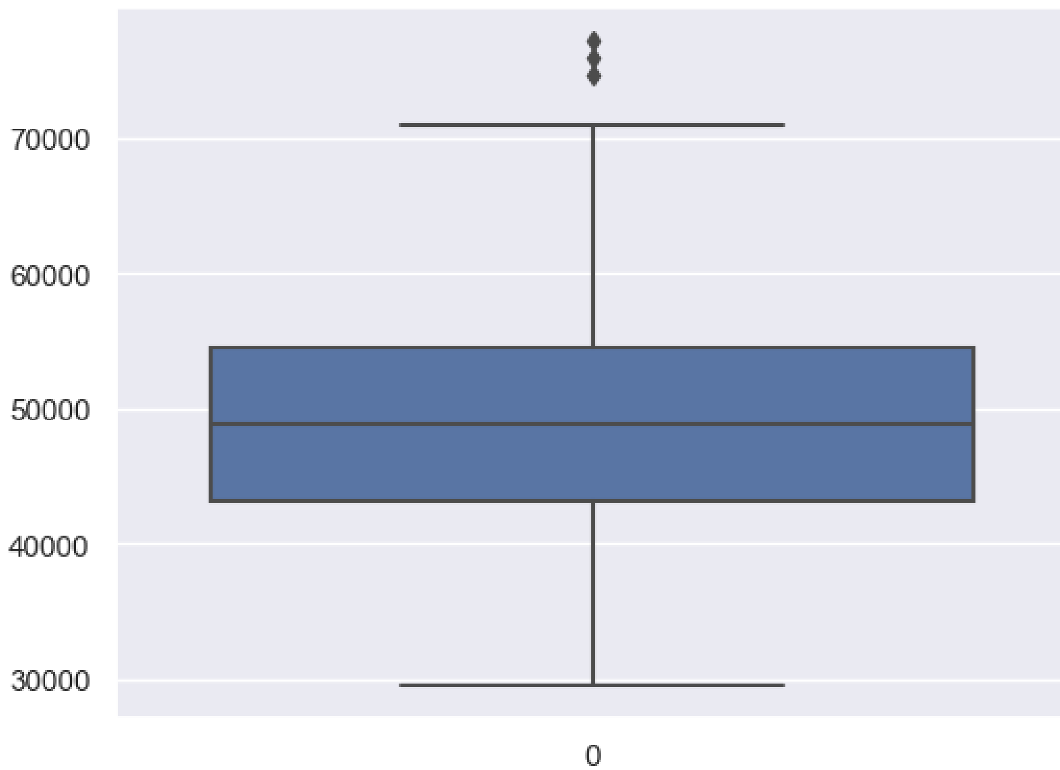
[38]:
```
new_df= data[data['Income']< upper_limit]
new_df
```

[38]:
```
    Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
0     KP281   18    Male         14        Single      3        4   29562
1     KP281   19    Male         15        Single      2        3   31836
2     KP281   19  Female         14     Partnered      4        3   30699
3     KP281   19    Male         12        Single      3        3   32973
4     KP281   20    Male         13     Partnered      4        2   35247
..      …   …     …        …          …     …        …      …
156   KP781   25    Male         20     Partnered      4        5   74701
157   KP781   26  Female         21        Single      4        3   69721
158   KP781   26    Male         16     Partnered      5        4   64741
163   KP781   28    Male         18     Partnered      7        5   77191
165   KP781   29    Male         18        Single      5        5   52290

     Miles
0      112
1       75
2       66
3       85
4       47
```

```
..     …
156    170
157    100
158    180
163    180
165    180

[161 rows x 9 columns]
```

[39]:
```
sns.boxplot(new_df['Income'])
```

[39]:
```
<Axes: >
```



[41]:
```
new_df[new_df['Income']>70000]
```

[41]:
```
    Product  Age Gender  Education MaritalStatus  Usage  Fitness  Income  \
154   KP781   25   Male         18     Partnered      6        4   70966
155   KP781   25   Male         18     Partnered      6        5   75946
156   KP781   25   Male         20     Partnered      4        5   74701
163   KP781   28   Male         18     Partnered      7        5   77191

     Miles
```

```
154    180
155    240
156    170
163    180
```

[48]:
```
new_df['Income'].mean()
```

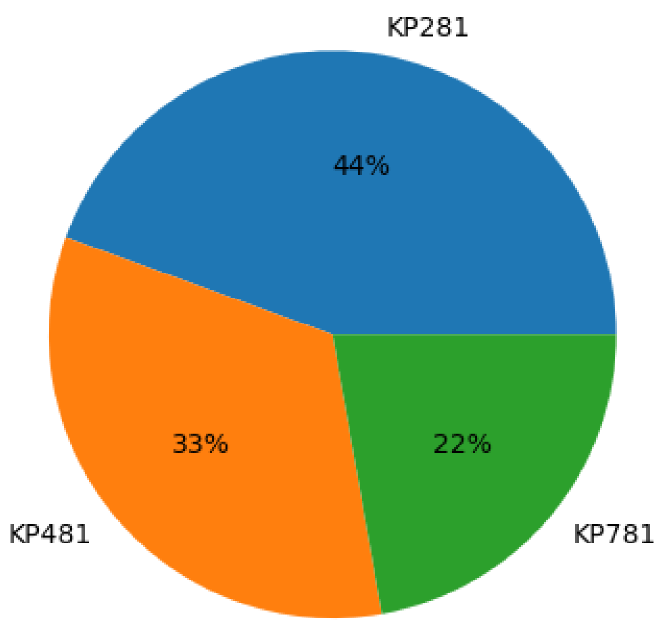
[48]: 49119.1801242236

[49]:
```
data['Income'].mean()
```

[49]: 53719.57777777778

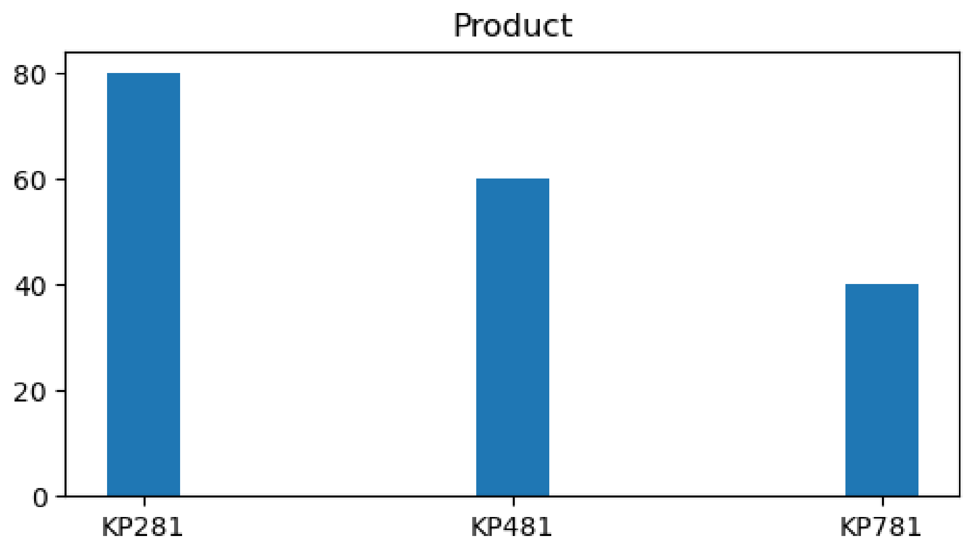
[13]:
```
dist= data[['Product']]
dist.groupby(['Product']).value_counts().plot(kind='pie', autopct='%1.0f%%')
```

[13]: <Axes: >



[22]:
```
cat_counts= data['Product'].value_counts()
x_bar= cat_counts.index
y_bar= cat_counts

plt.figure(figsize=(6,3))
```

```
plt.bar(x_bar, y_bar, width=0.2)
plt.title('Product', fontsize=12)
plt.show()
```

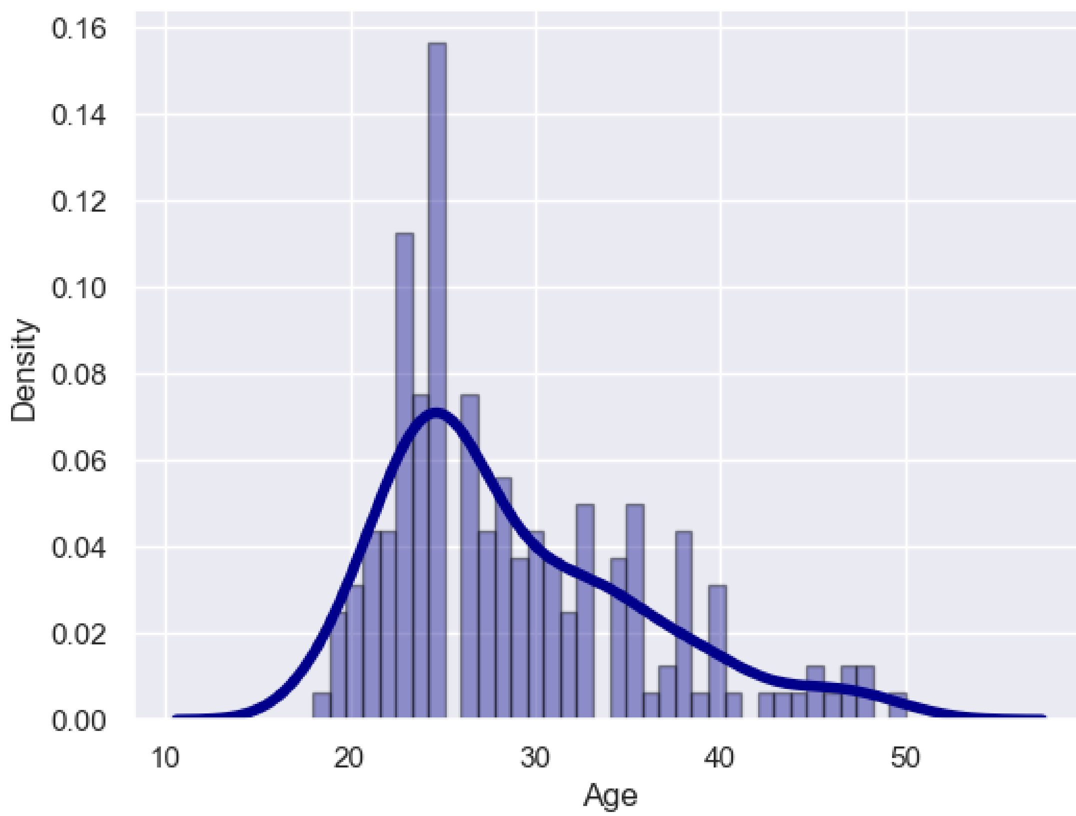


7 Analysis of Categorical Columns with the Product -

For this section, We'll be converting the Ages, Incomes and Miles to bins for better analysis.

[14]:
```
# Observing the ages to create bins -

sns.distplot(data['Age'], hist=True, kde=True,
bins=int(36), color = 'darkblue',
hist_kws={'edgecolor':'black'},
kde_kws={'linewidth': 4})
plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\363168779.py:3:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['Age'], hist=True, kde=True,
```



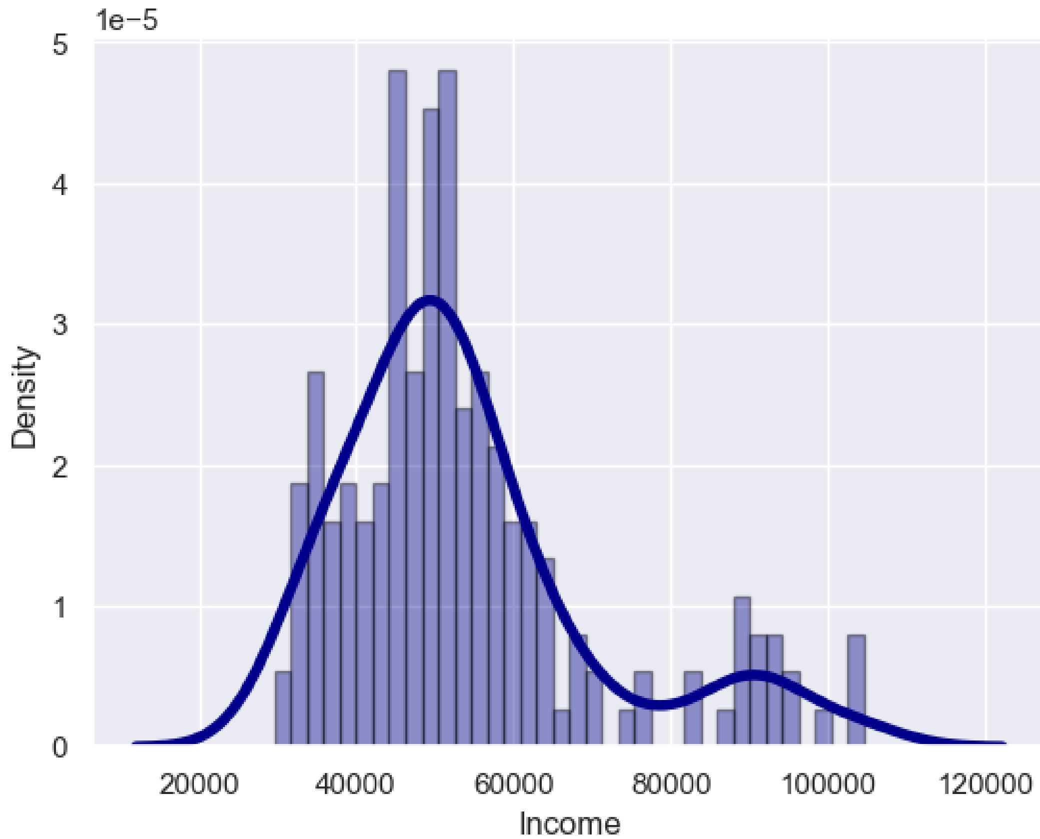

```
[29]: bins = [-1,20,25,30,35,40,55]
      labels = ['<20','20-25','25-30','30-35','35-40','40+']
      data['Age_bins'] = pd.cut(data['Age'], bins=bins, labels=labels)
```

```
[16]: # Observing the incomes to create bins -

      sns.distplot(data['Income'], hist=True, kde=True,
      bins=int(36), color = 'darkblue',
      hist_kws={'edgecolor':'black'},
      kde_kws={'linewidth': 4})
      plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\53825801.py:3:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
```

https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

sns.distplot(data['Income'], hist=True, kde=True,



[17]:
```
data['Income'].describe()
```

[17]:
```
count       180.000000
mean      53719.577778
std       16506.684226
min       29562.000000
25%       44058.750000
50%       50596.500000
75%       58668.000000
max      104581.000000
Name: Income, dtype: float64
```

[18]:
```
# Creating bins for income -
bins = [-1,35000,45000,50000,60000,70000,90000,120000]
labels =␣
↪['<35000','35000-45000','45000-50000','50000-60000','60000-70000','70000-90000','90000+']
```

```
data['Income_bins'] = pd.cut(data['Income'], bins=bins, labels=labels)
data.head()
```

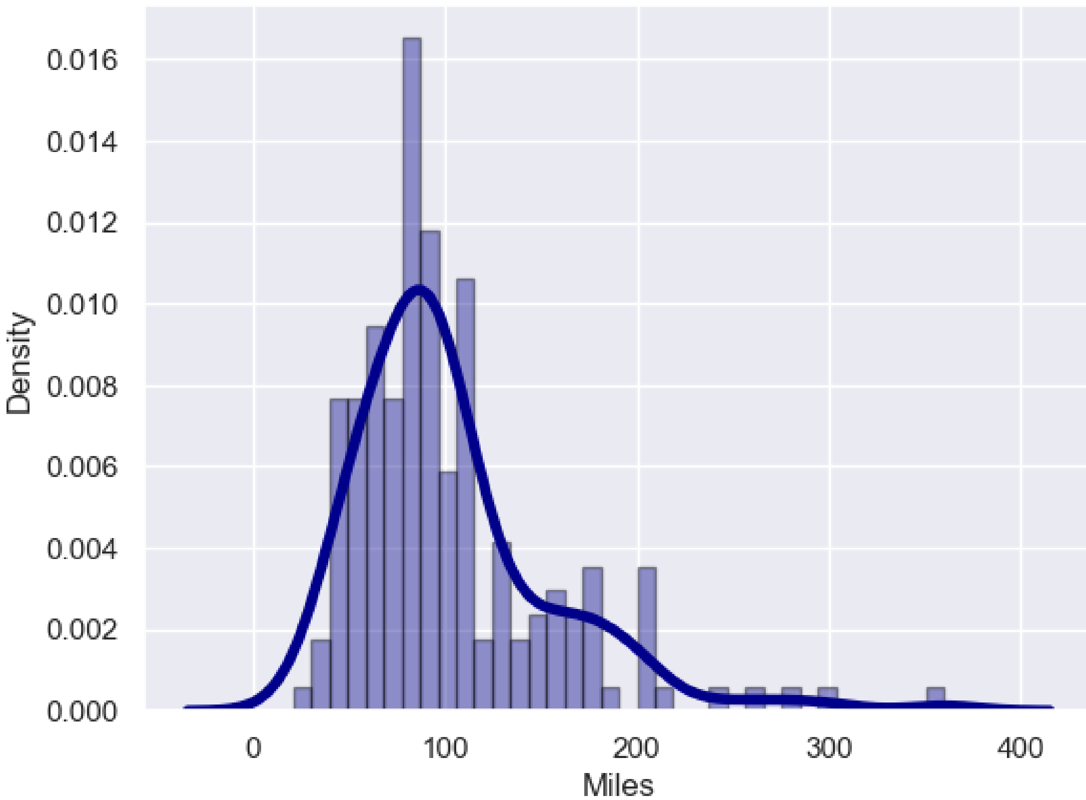
[18]:

```
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
0   KP281   18    Male         14        Single      3        4   29562
1   KP281   19    Male         15        Single      2        3   31836
2   KP281   19  Female         14     Partnered      4        3   30699
3   KP281   19    Male         12        Single      3        3   32973
4   KP281   20    Male         13     Partnered      4        2   35247

   Miles  Income_bins
0    112       <35000
1     75       <35000
2     66       <35000
3     85       <35000
4     47  35000-45000
```

[19]:

```
# Observing the miles to create bins -

sns.distplot(data['Miles'], hist=True, kde=True,
bins=int(36), color = 'darkblue',
hist_kws={'edgecolor':'black'},
kde_kws={'linewidth': 4})
plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\3037726482.py:3:
UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['Miles'], hist=True, kde=True,
```

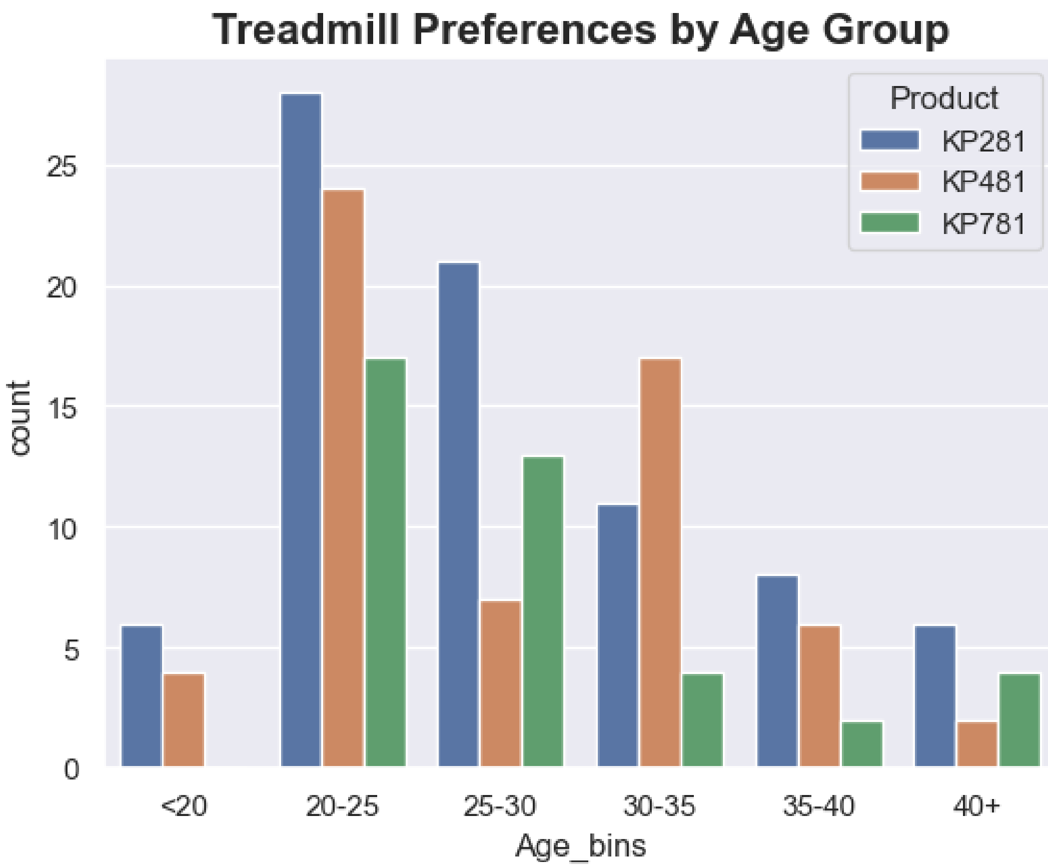


[20]:
```
# Creating bins for miles -
bins = [-1,50,100,150,400]
labels = ['<50','50-100','100-150','150+']
data['Mile_bins'] = pd.cut(data['Miles'], bins=bins, labels=labels)
data.head(3)
```

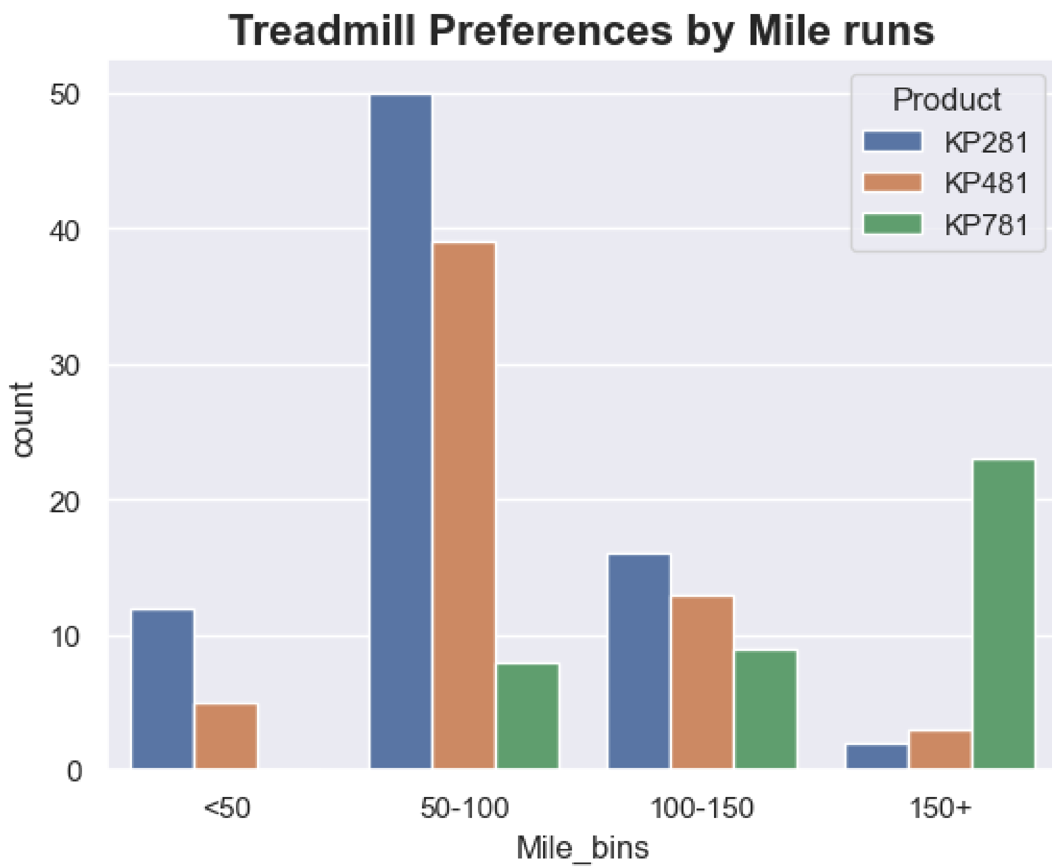
[20]:
```
  Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
0   KP281   18    Male         14        Single      3        4   29562
1   KP281   19    Male         15        Single      2        3   31836
2   KP281   19  Female         14     Partnered      4        3   30699

   Miles Income_bins Mile_bins
0    112      <35000   100-150
1     75      <35000    50-100
2     66      <35000    50-100
```

[9]:
```
sns.countplot(x='Age_bins', hue='Product', data=data)
plt.title("Treadmill Preferences by Age Group", fontsize=16, fontweight='bold')
plt.xlabel("Age_bins", fontsize=12)

# Show the graph
```

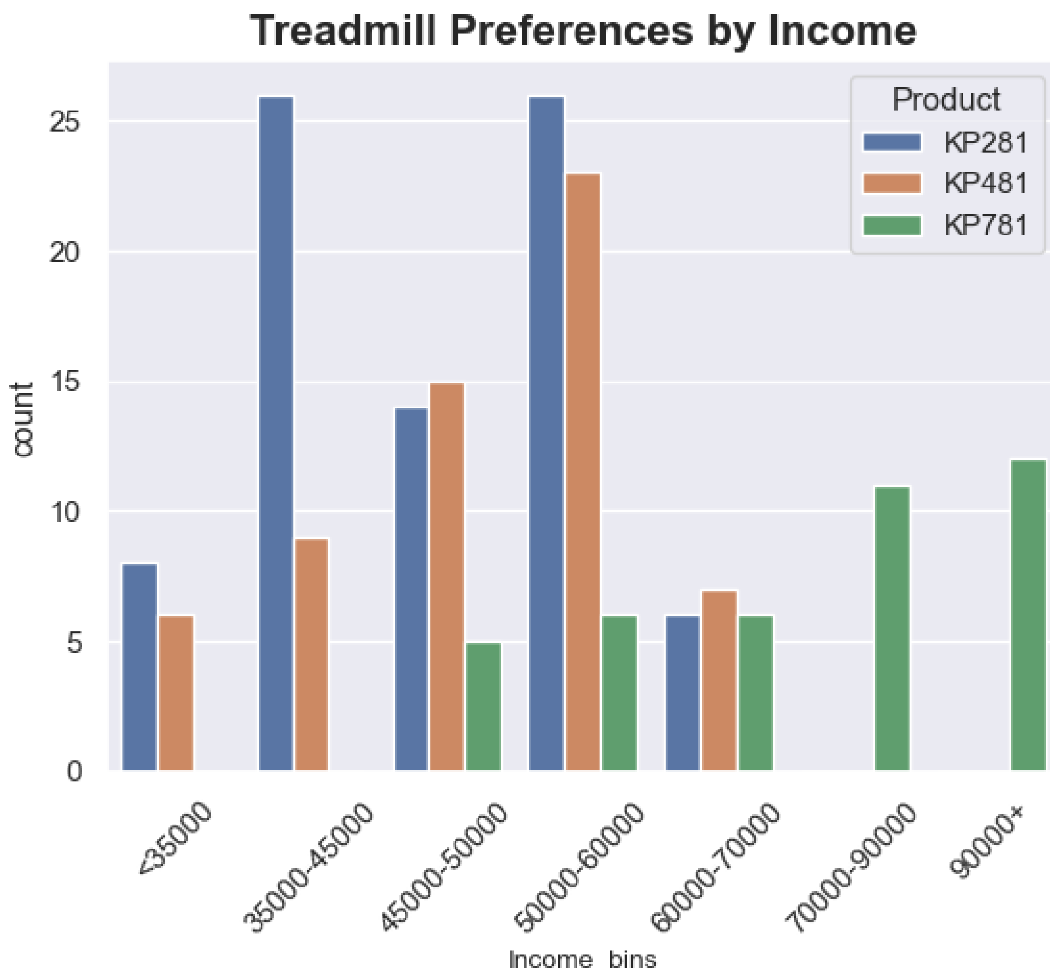
```
plt.show()
```



[13]:
```
sns.countplot(x='Mile_bins', hue='Product', data=data)
plt.title("Treadmill Preferences by Mile runs", fontsize=16, fontweight='bold')
plt.xlabel("Mile_bins", fontsize=12)
plt.show()
```



[16]:
```
sns.countplot(x='Income_bins', hue='Product', data=data)
plt.title("Treadmill Preferences by Income", fontsize=16, fontweight='bold')
plt.xlabel("Income_bins", fontsize=10)
plt.xticks(rotation=45)
plt.show()
```



[22]:
```python
i = 'Gender'
print(f"Proportion of different {'Gender'} byuing different Treadmills")
```

```
Proportion of different Gender byuing different Treadmills
```

[23]:
```python
pd.crosstab(data['Gender'], data['Product'])
```

[23]:
```
Product  KP281  KP481  KP781
Gender
Female      40     29      7
Male        40     31     33
```

[24]:
```python
pd.crosstab(data['Gender'],data['Product']).sum(axis=1)
```

[24]:
```
Gender
Female     76
Male      104
```

```
dtype: int64
```

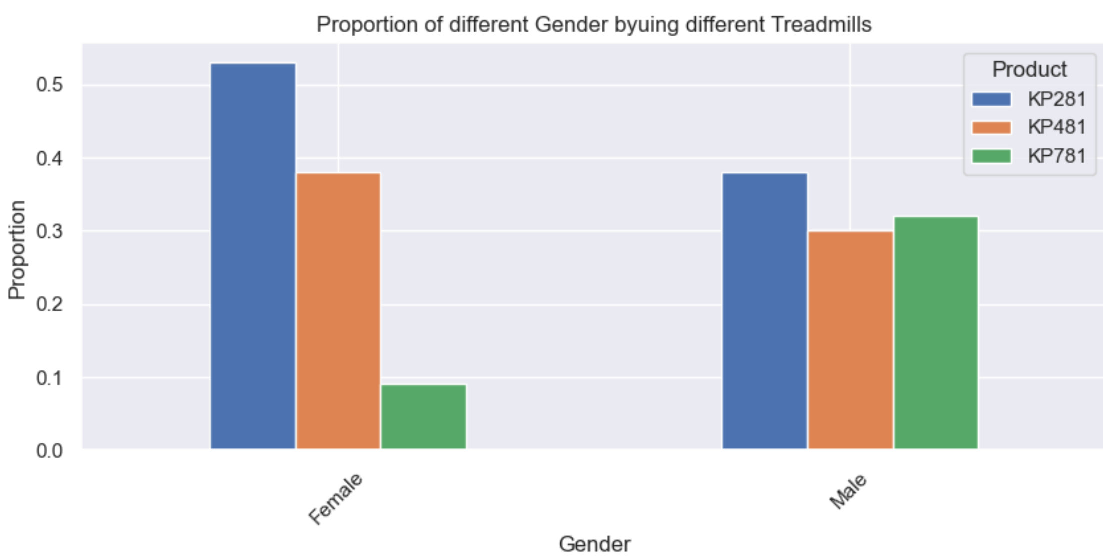
[25]:
```
pd.crosstab(data['Gender'], data['Product']).div(pd.
 ↪crosstab(data['Gender'],data['Product']).sum(axis=1),axis=0)
```

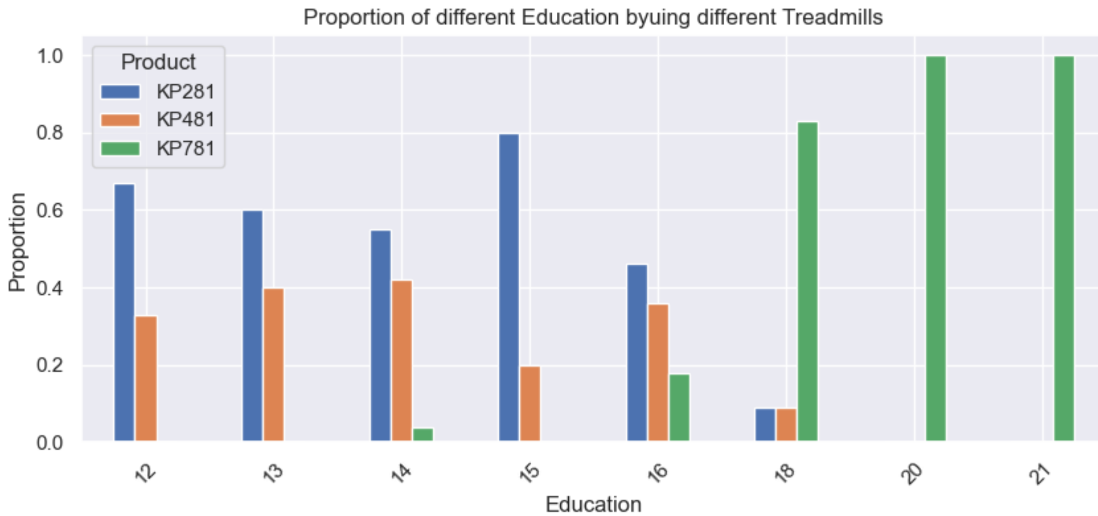
[25]:
```
Product      KP281     KP481     KP781
Gender
Female    0.526316  0.381579  0.092105
Male      0.384615  0.298077  0.317308
```

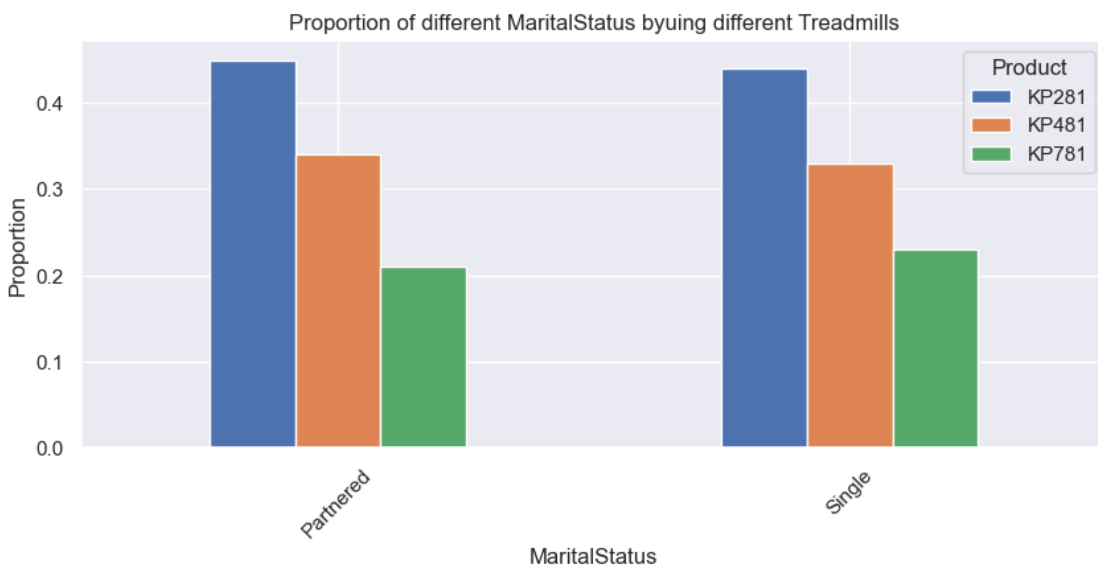
[27]:
```
round(pd.crosstab(data['Gender'], data['Product']).div(pd.
 ↪crosstab(data['Gender'],data['Product']).sum(axis=1),axis=0),2)
```

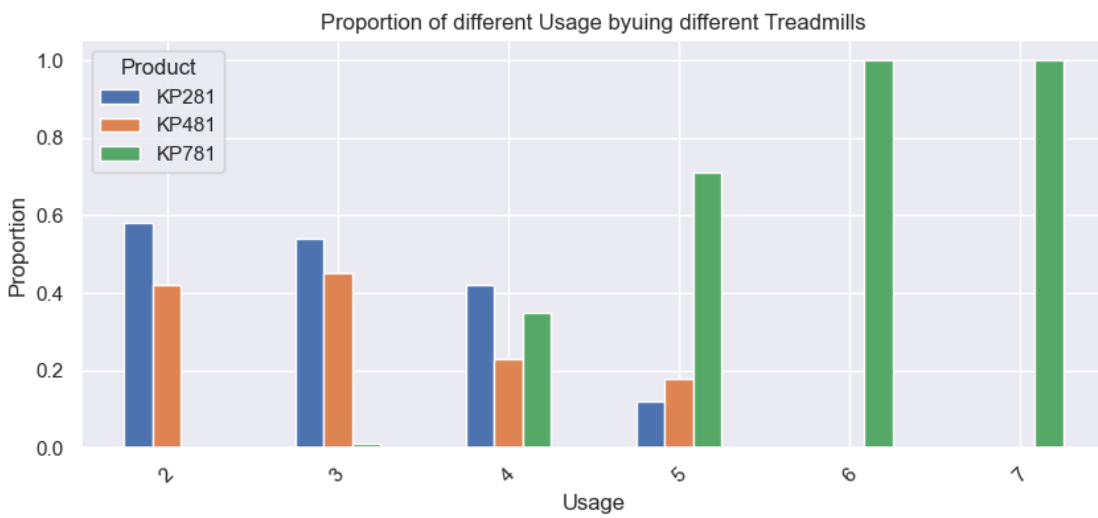
[27]:
```
Product  KP281  KP481  KP781
Gender
Female    0.53   0.38   0.09
Male      0.38   0.30   0.32
```

[30]:
```
# Crosstabs -

cat_cols=['Gender','Education', 'MaritalStatus',␣
 ↪'Usage','Fitness','Age_bins','Income_bins','Mile_bins']
for i in cat_cols:
    other= round(pd.crosstab(data[i], data['Product']).div(pd.
 ↪crosstab(data[i],data['Product']).sum(axis=1),0),2)
    ax = other.plot(kind ='bar', title = i, figsize = (10,4))
    ax.set_xlabel(i)
    ax.set_ylabel('Proportion')
    plt.xticks(rotation=45)
    plt.title(f"Proportion of different {i} byuing different Treadmills")
    plt.show()
```

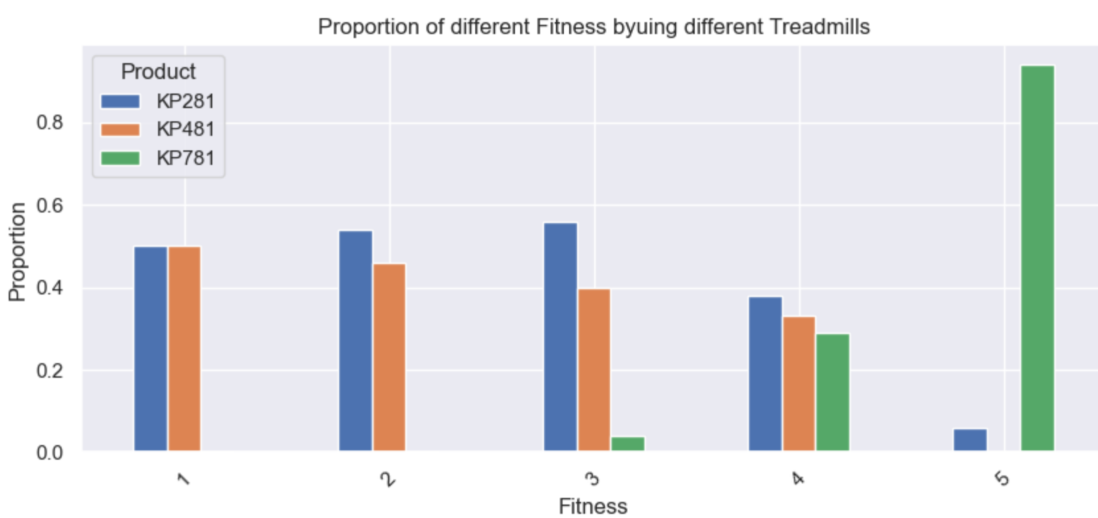



Proportion of different Education byuing different Treadmills
Product
KP281
KP481
KP781
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
12
13
14
15
16
18
20
21
Education

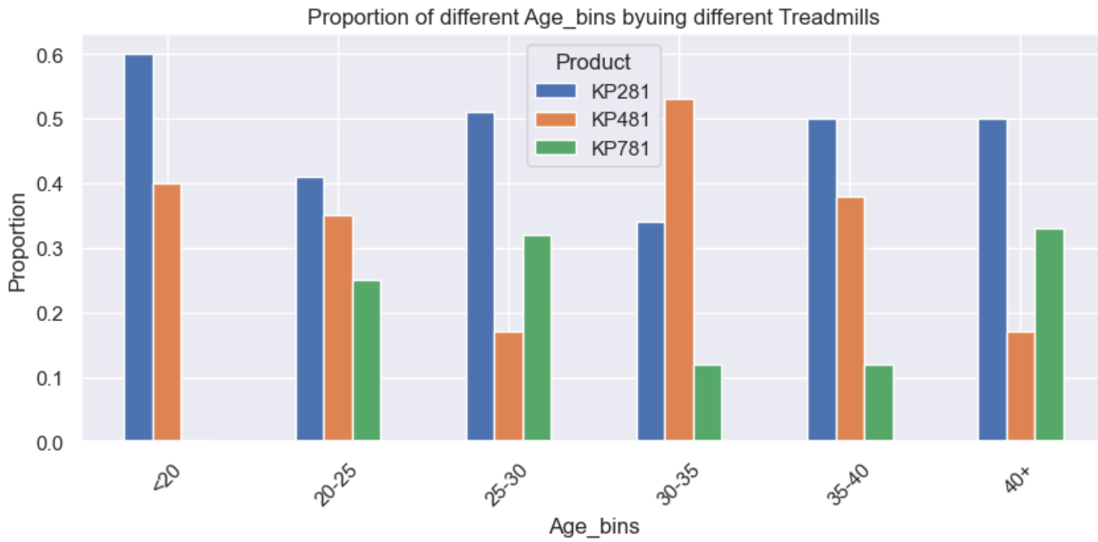

Proportion of different MaritalStatus byuing different Treadmills
Product
KP281
KP481
KP781
Proportion
0.4
0.3
0.2
0.1
0.0
Partnered
Single
MaritalStatus

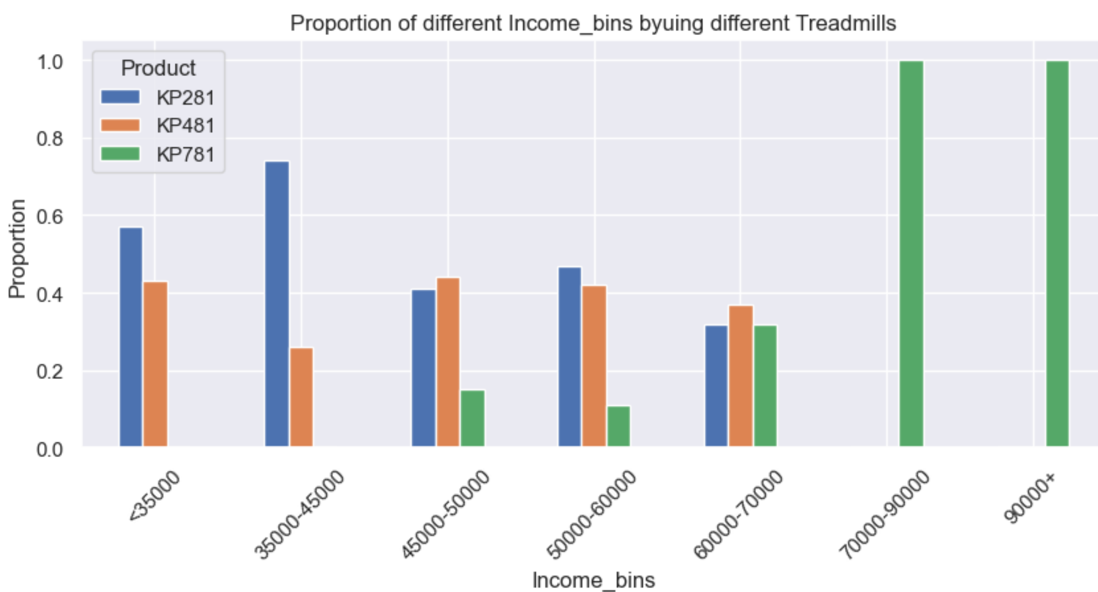


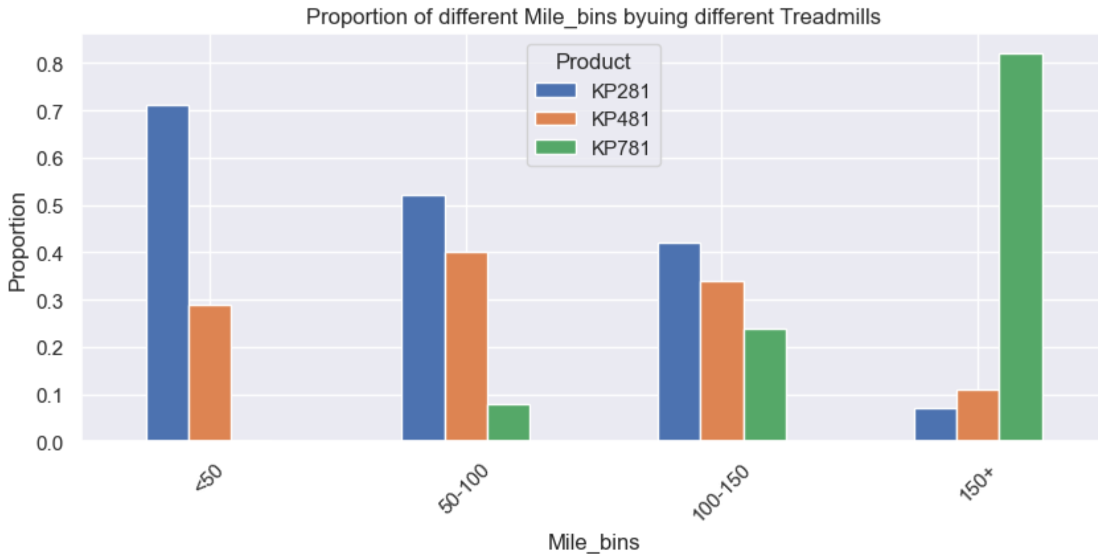
Proportion of different Usage byuing different Treadmills
Product
KP281
KP481
KP781
1.0
0.8
0.6
0.4
0.2
0.0
Proportion
2
3
4
5
6
7
Usage



Proportion of different Fitness byuing different Treadmills
Product
KP281
KP481
KP781
0.8
0.6
0.4
0.2
0.0
Proportion
1
2
3
4
5
Fitness


Proportion of different Age_bins byuing different Treadmills
Product
KP281
KP481
KP781
Proportion
Age_bins
<20
20-25
25-30
30-35
35-40
40+


Proportion of different Income_bins byuing different Treadmills
Product
KP281
KP481
KP781
Proportion
Income_bins
<35000
35000-45000
45000-50000
50000-60000
60000-70000
70000-90000
90000+



8 Observations on the basis of above Categorical Plots

1. Around 55% of women prefer KP281 and only 10% prefer KP781. While around 35% of men prefer KP781.
2. 80% in Eduation level of 18 and everyone in Education levels of 20 or 21 use KP781 while below 14 level, no one uses KP781.
3. Marital Status implies no significant information on the usages of different treadmills.
4. Those who workout 6 or 7 days a week use KP781 while 60% of those who workout 5 days a week use KP781.
5. 95% of customers having fitness level of 5 use KP781 and none of those having fitness level below 3 use KP781.
6. No one below 20 years of age use KP781.
7. Above 70000 units of Income, people only use KP781 while in Incomes below 45000, no one uses KP781.
8. Almost 80% of people who run over 200 miles and those who run above 150 miles use KP781 and no one who runs below 50 miles use KP781. The usage of KP281 decreases with the increase in miles while that of KP781 increases with the increase in miles.

[ ]:
```
# Let's deal with Probabilities -
```

[32]:
```
for i in cat_cols:
  print('Table for',str(i),'vs Treadmill Product')
  display(pd.crosstab(data[i], data['Product'], margins=True,␣
 ↪normalize='index'))
```

```
print("\n")
```

```
Table for Gender vs Treadmill Product

Product     KP281     KP481     KP781
Gender
Female   0.526316  0.381579  0.092105
Male     0.384615  0.298077  0.317308
All      0.444444  0.333333  0.222222



Table for Education vs Treadmill Product

Product       KP281     KP481     KP781
Education
12         0.666667  0.333333  0.000000
13         0.600000  0.400000  0.000000
14         0.545455  0.418182  0.036364
15         0.800000  0.200000  0.000000
16         0.458824  0.364706  0.176471
18         0.086957  0.086957  0.826087
20         0.000000  0.000000  1.000000
21         0.000000  0.000000  1.000000
All        0.444444  0.333333  0.222222



Table for MaritalStatus vs Treadmill Product

Product           KP281     KP481     KP781
MaritalStatus
Partnered      0.448598  0.336449  0.214953
Single         0.438356  0.328767  0.232877
All            0.444444  0.333333  0.222222



Table for Usage vs Treadmill Product

Product     KP281     KP481     KP781
Usage
2        0.575758  0.424242  0.000000
3        0.536232  0.449275  0.014493
4        0.423077  0.230769  0.346154
5        0.117647  0.176471  0.705882
6        0.000000  0.000000  1.000000
7        0.000000  0.000000  1.000000
All      0.444444  0.333333  0.222222
```

Table for Fitness vs Treadmill Product

Product Fitness	KP281	KP481	KP781
1	0.500000	0.500000	0.000000
2	0.538462	0.461538	0.000000
3	0.556701	0.402062	0.041237
4	0.375000	0.333333	0.291667
5	0.064516	0.000000	0.935484
All	0.444444	0.333333	0.222222

Table for Age_bins vs Treadmill Product

Product Age_bins	KP281	KP481	KP781
<20	0.600000	0.400000	0.000000
20-25	0.405797	0.347826	0.246377
25-30	0.512195	0.170732	0.317073
30-35	0.343750	0.531250	0.125000
35-40	0.500000	0.375000	0.125000
40+	0.500000	0.166667	0.333333
All	0.444444	0.333333	0.222222

Table for Income_bins vs Treadmill Product

Product Income_bins	KP281	KP481	KP781
<35000	0.571429	0.428571	0.000000
35000-45000	0.742857	0.257143	0.000000
45000-50000	0.411765	0.441176	0.147059
50000-60000	0.472727	0.418182	0.109091
60000-70000	0.315789	0.368421	0.315789
70000-90000	0.000000	0.000000	1.000000
90000+	0.000000	0.000000	1.000000
All	0.444444	0.333333	0.222222

Table for Mile_bins vs Treadmill Product

Product Mile_bins	KP281	KP481	KP781
<50	0.705882	0.294118	0.000000
50-100	0.515464	0.402062	0.082474
100-150	0.421053	0.342105	0.236842
150+	0.071429	0.107143	0.821429
All	0.444444	0.333333	0.222222

9 Brief depiction of Probabilities Inferred from the above tables

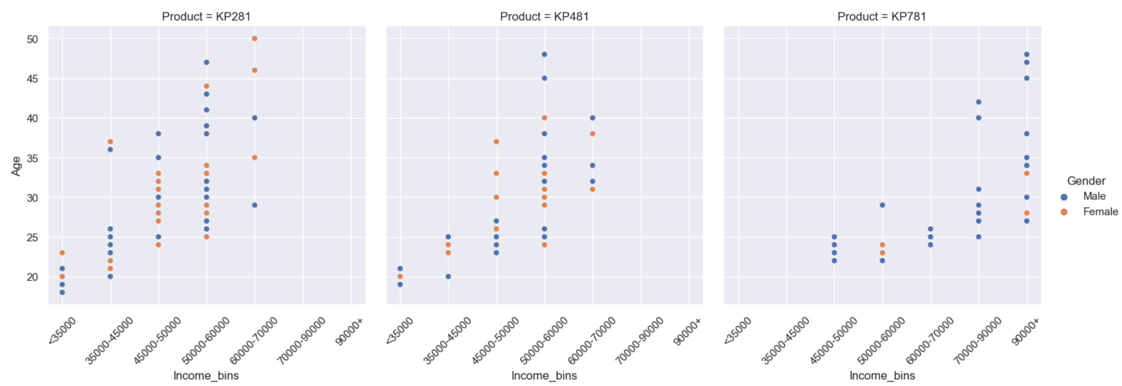
- In all the tables, one can see the last row named All, it consists of the overall probabilities of purchases of those 3 treadmills, i.e. - Probability of purchase of KP281= 44.44%, KP481= 33.33% and KP781=22.22%
- P(KP281|Education=12) = 66.66% and P(KP781|Education=18) = 82.6% P(KP781|Education=20) = P(KP781|Education=21) = 100%
- P(KP281|Usage=2) = 57.57%, P(KP781|Usage=6)=P(KP781|Usage=7) = 100%
- P(KP481|Fitness=2) = 46.15%
- P(KP481|Age_bins=30-35) = 53.12%
- P(KP781|Income>70000) = 100% and P(KP481|Income_bins=45000-50000) = 44.11%
- P(KP281|Mile_bins<50) = 70.5% and P(KP781|Mile_bins>150)=82.1%

Till now we've got great insights for the customers peratining to KP781 while clarity regarding KP281 and KP481 seems to be missing

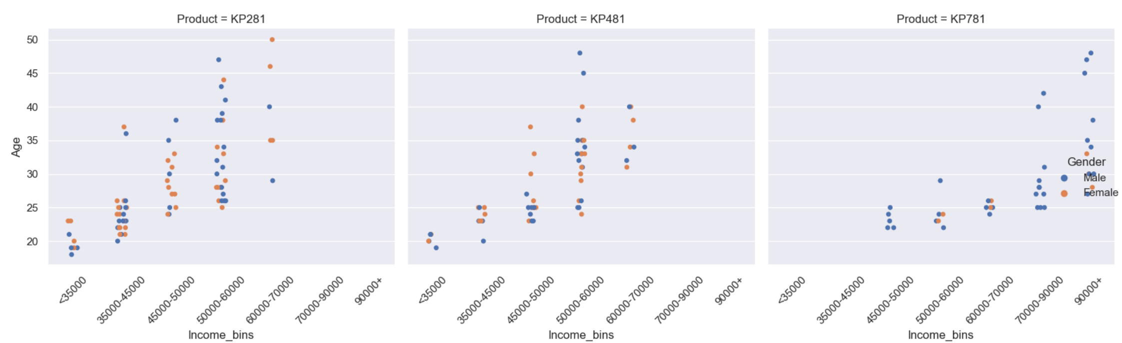
10 Multivariate Analysis using Scatterplots and Factorplots for different Products (predominately to understand target audience for KP281 and KP481 which appear quite similar).

[35]:
```
plot=sns.relplot(data=data, x='Income_bins', y='Age', col='Product',␣
 ↪hue='Gender', palette="deep", kind="scatter")
plt.figure(figsize=(10,6))
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plot=sns.factorplot(x ='Income_bins', y ='Age',
               hue='Gender',col = 'Product', data = data) #jitterplot
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\4197790395.py:4:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\4197790395.py:9:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
```



<Figure size 1000x600 with 0 Axes>

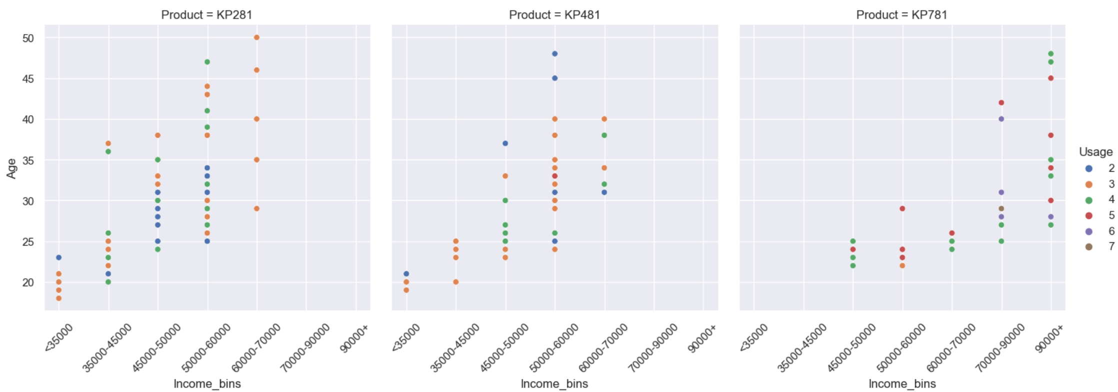


1. For women having incomes below 70k, the average age of those who use KP281 is 40 while it's 35 for those who use KP481.
2. Only 2 women have incomes over 70k which is certainly the reason for a large proportion of them not buying KP781 (affordability).
3. Moreover the variances are higher in case of KP281. Though there are less data points for this observation, so it requires more data to be verified.

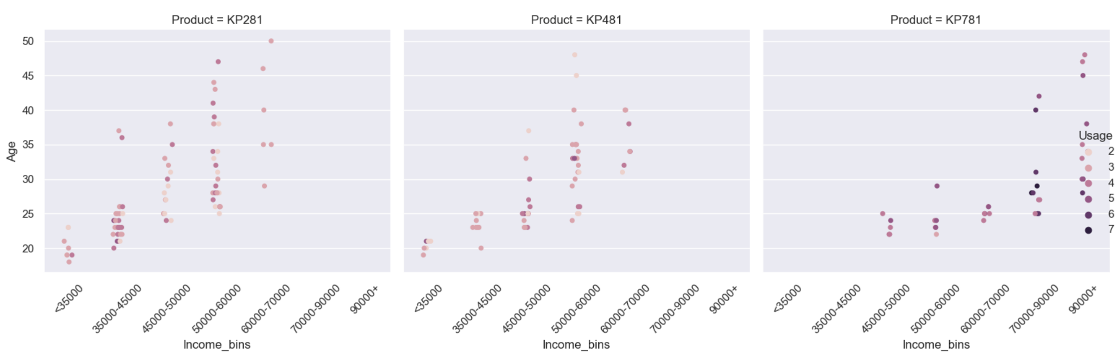
[38]:
```
plot=sns.relplot(data=data, x='Income_bins', y='Age', col='Product',␣
 ↪hue='Usage', palette="deep", kind="scatter")
plt.figure(figsize=(10,6))
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plot=sns.factorplot(x ='Income_bins', y ='Age',
               hue='Usage',col = 'Product', data = data)
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
```

```
plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\3587304225.py:4:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\3587304225.py:9:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
```



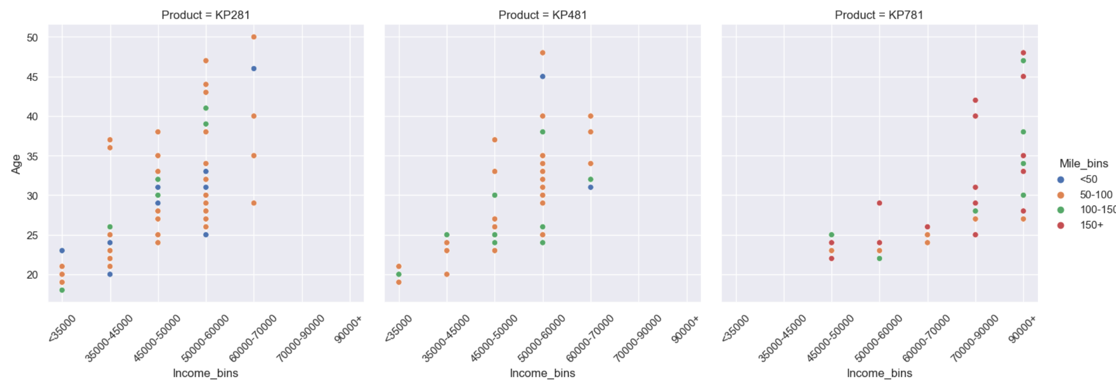

```
<Figure size 1000x600 with 0 Axes>
```



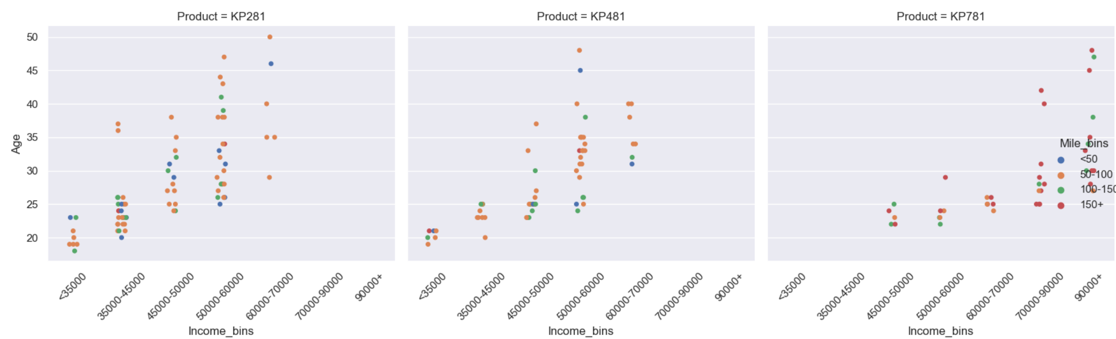
1. For Usage=3 and Income in the range 60k-70k, we are very much certain of the user to be buying the KP281 Treadmill.
2. For Usage=2 and Income in the range 45k-50k, we are very much certain of the user to be buying the KP281 Treadmill.
3. For income range 45k-50k and Usage=4, we are very much certain of the user to be buying KP281 Treadmill.
4. For income range 50k-60k and Usage=4, we are very much certain of the user to be buying KP281 Treadmill.

[39]:
```
plot=sns.relplot(data=data, x='Income_bins', y='Age', col='Product',␣
↪hue='Mile_bins', palette="deep", kind="scatter")
plt.figure(figsize=(10,6))
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plot=sns.factorplot(x ='Income_bins', y ='Age',
                hue='Mile_bins',col = 'Product', data = data)
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\1559582105.py:4:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\1559582105.py:9:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
```



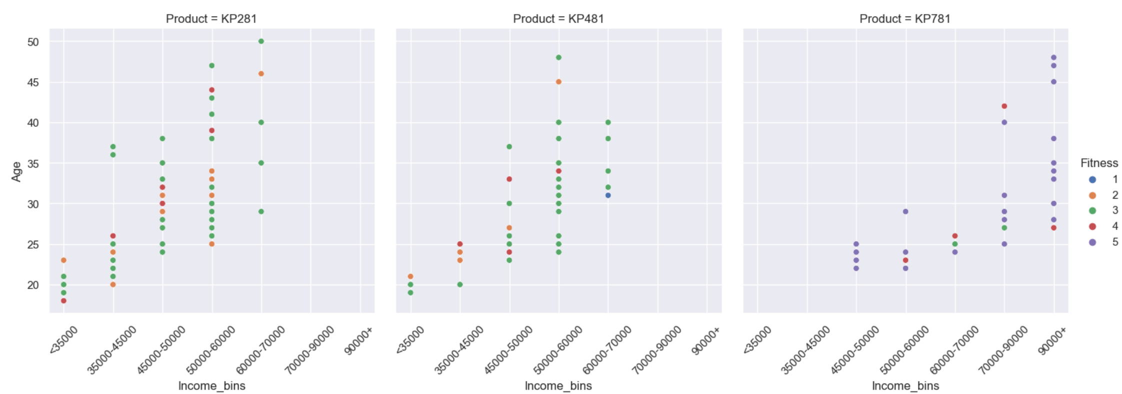

```
<Figure size 1000x600 with 0 Axes>
```



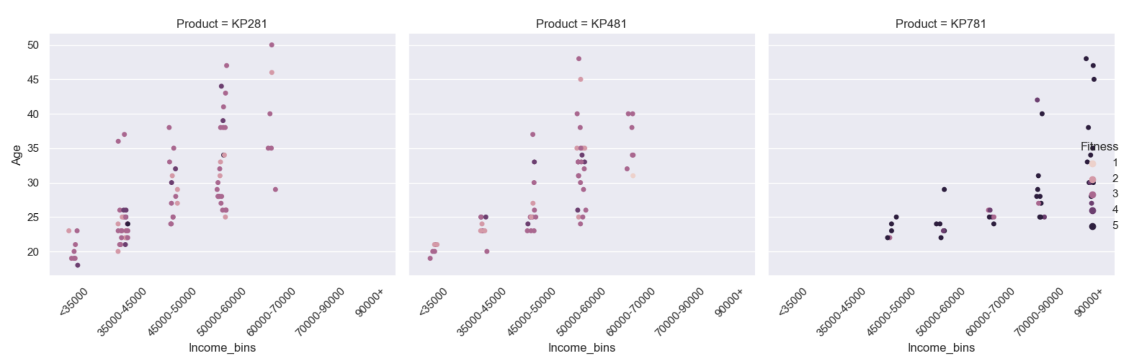
- For no. of miles in range 100-150, those customers whose incomes are in range of 50k-60k and age between 25 to 30 tend to use KP481.

[40]:
```
plot=sns.relplot(data=data, x='Income_bins', y='Age', col='Product', hue='Fitness', palette="deep", kind="scatter")
plt.figure(figsize=(10,6))
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plot=sns.factorplot(x ='Income_bins', y ='Age',
                    hue='Fitness',col = 'Product', data = data)
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\1375695413.py:4:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\1375695413.py:9:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
```



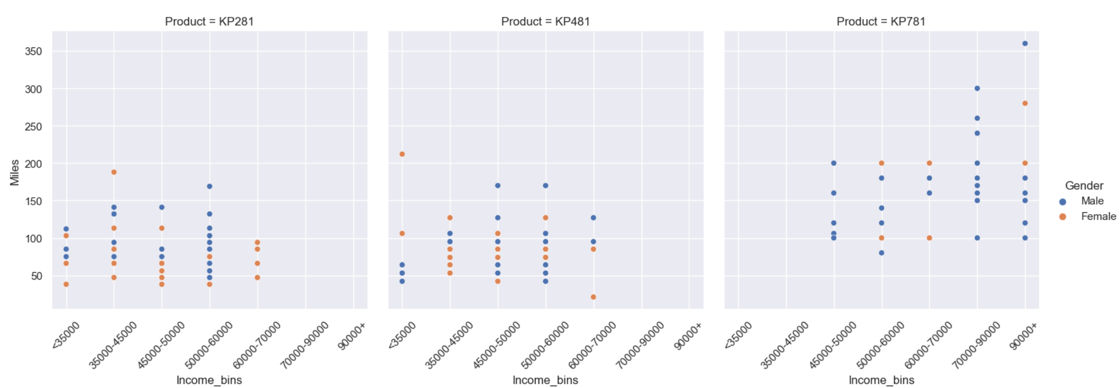

```
<Figure size 1000x600 with 0 Axes>
```



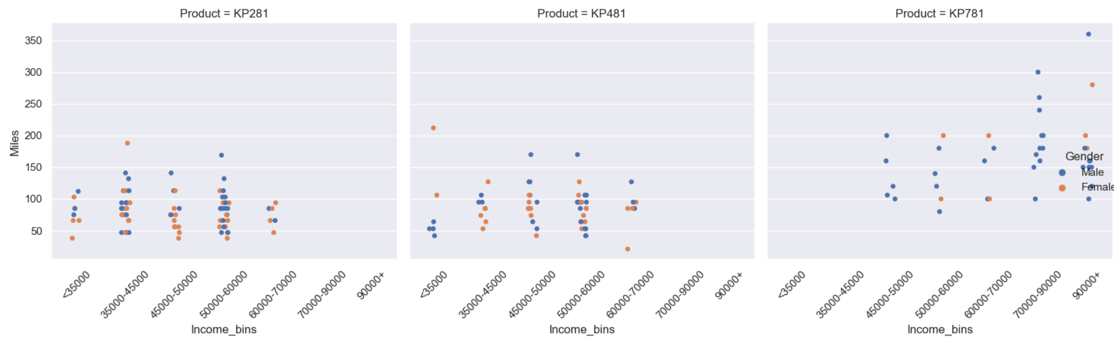
- For Education level of 16, above 32 years of age with Income between 45k-50k will tend to use KP281 and below 22 will tend to use KP481.
- Also for the same Education level customers but in Income range 60k-70k, above 45 years of age will tend to use KP281 while customers below 35 years of age will tend to use KP481.

[41]:
```
plot=sns.relplot(data=data, x='Income_bins', y='Miles', col='Product',␣
 ↪hue='Gender', palette="deep", kind="scatter")
plt.figure(figsize=(10,6))
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plot=sns.factorplot(x ='Income_bins', y ='Miles',
               hue='Gender',col = 'Product', data = data)
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\3780353717.py:4:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\3780353717.py:9:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
```




```
<Figure size 1000x600 with 0 Axes>
```

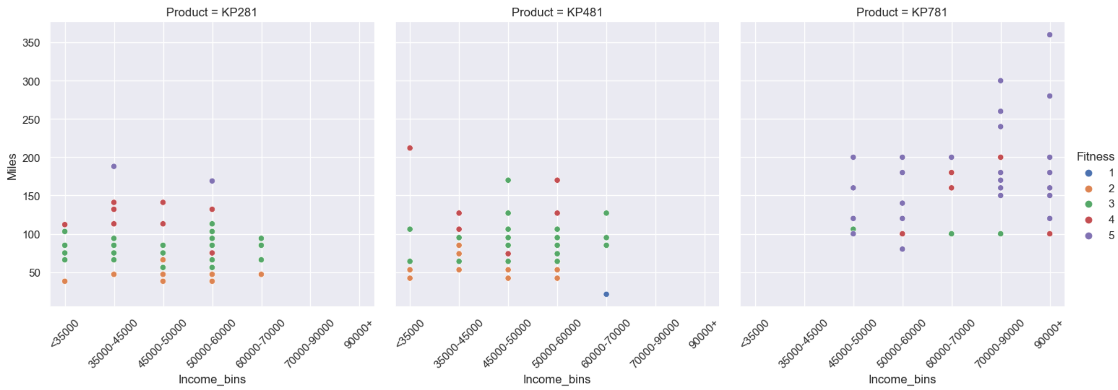


- For income level below 35k, women who tread over 105 miles tend to use KP481 while those who tread below 105 tend to use KP281.
- Men with income level in 60k-70k, those who run in the range of 100-150 miles tend to use KP481.

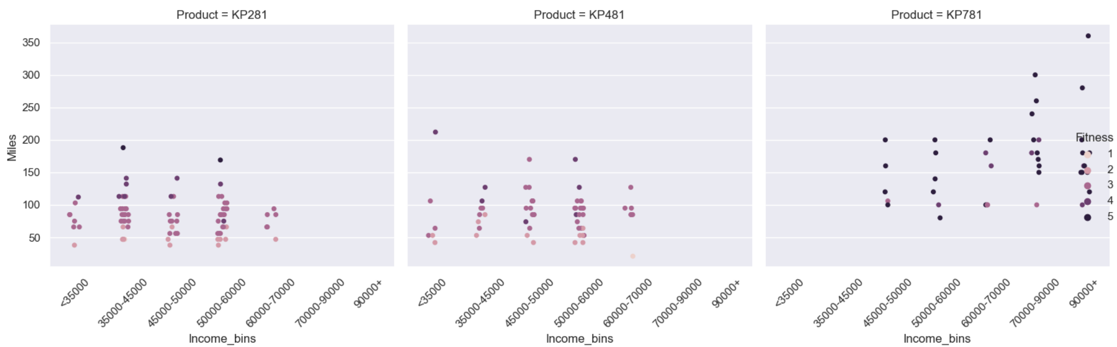
[42]:

```
plot=sns.relplot(data=data, x='Income_bins', y='Miles', col='Product', hue='Fitness', palette="deep", kind="scatter")
plt.figure(figsize=(10,6))
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plot=sns.factorplot(x ='Income_bins', y ='Miles',
                    hue='Fitness',col = 'Product', data = data)
for axes in plot.axes.flat:
    _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
plt.tight_layout()
plt.show()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\649949351.py:4:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_25244\649949351.py:9:
UserWarning: FixedFormatter should only be used together with FixedLocator
  _ = axes.set_xticklabels(axes.get_xticklabels(), rotation=45)
```




```
<Figure size 1000x600 with 0 Axes>
```



- If Fitness level=4 and incomes between 50k-60k, then these customers will tend to use KP281
- If Fitness level=4 in Income level of 50k-60k, if the person runs more than 100 miles, they tend to use KP481.

11 Target Audience

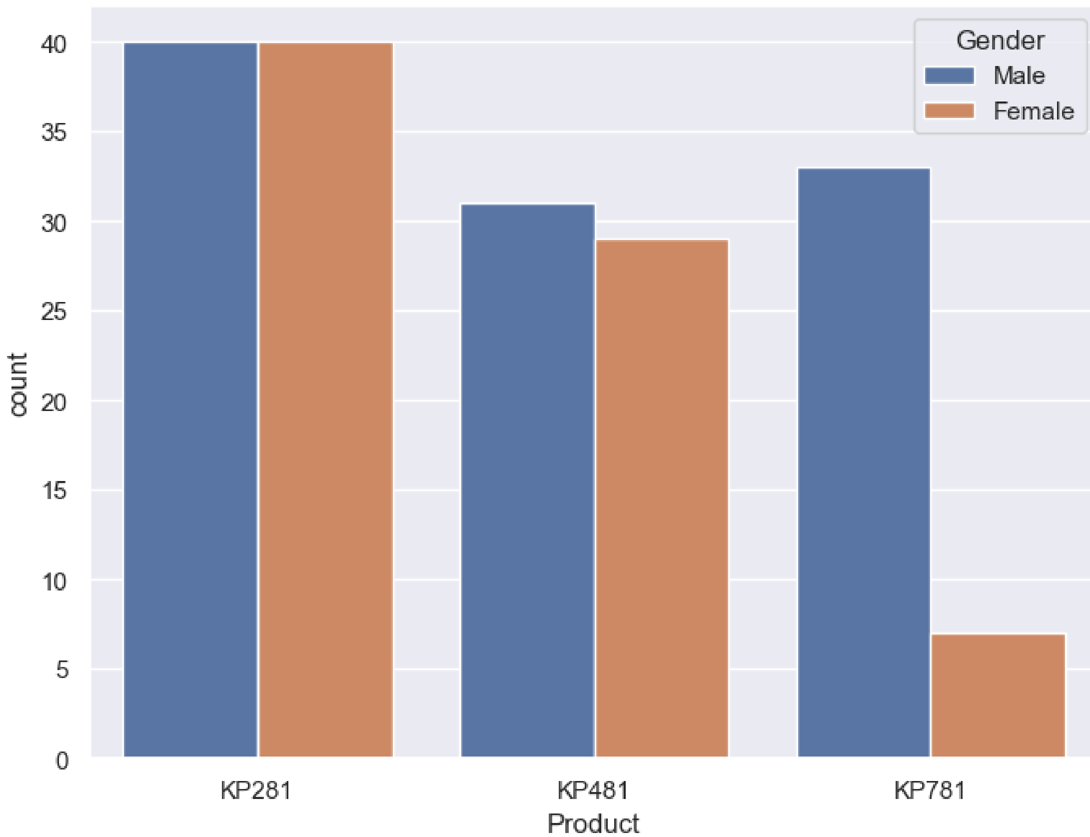
mostly from age 20 to 28, purchase most of our product.

[25]:
```
data['Age'].min()
```

[25]:
```
18
```

[21]:
```
plt.figure(figsize=(8,6))
sns.countplot(x='Product',hue='Gender',data=data)
```

[21]:
```
<Axes: xlabel='Product', ylabel='count'>
```



- Men are most interested in KP781 product

[4]:
```
data.groupby('Gender').mean()
```

```
C:\Users\Anjali Sharma\AppData\Local\Temp\ipykernel_2176\4284277325.py:1:
FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is
deprecated. In a future version, numeric_only will default to False. Either
specify numeric_only or select only columns which should be valid for the
function.
  data.groupby('Gender').mean()
```

[4]:

	Age	Education	Usage	Fitness	Income	Miles
Gender						
Female	28.565789	15.394737	3.184211	3.026316	49828.907895	90.013158
Male	28.951923	15.701923	3.653846	3.519231	56562.759615	112.826923

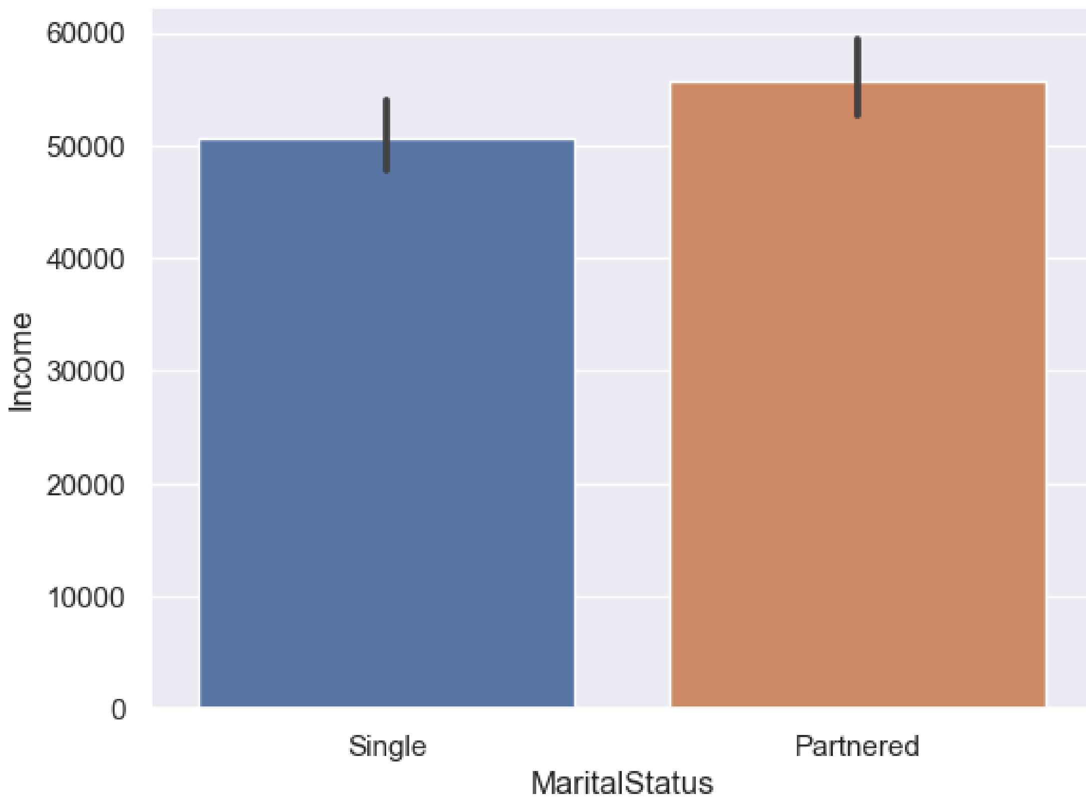
12 people with higher income greater than average tend to buy KP781

[60]:
```
data['Income'].mean()
```

[60]: 53719.57777777778

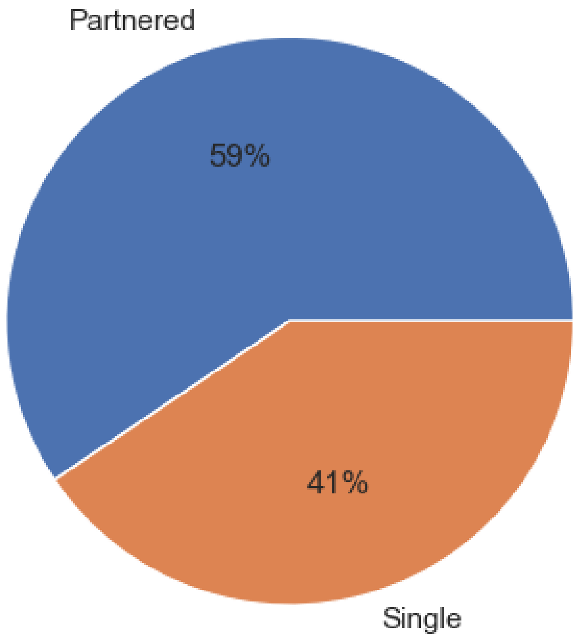
[65]:
```
sns.barplot(data=data,x='MaritalStatus', y='Income',estimator=np.mean)
```

[65]: <Axes: xlabel='MaritalStatus', ylabel='Income'>



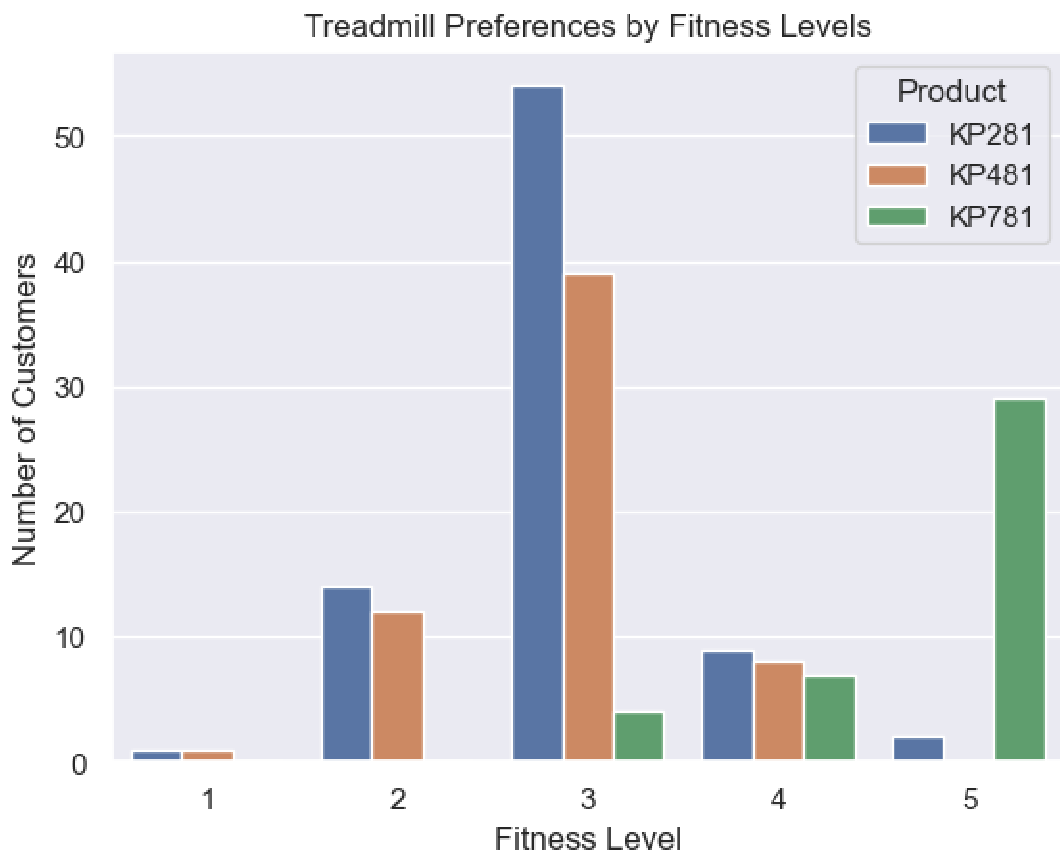
[66]:
```
dist2= data[['MaritalStatus']]
dist2.groupby(['MaritalStatus']).value_counts().plot(kind='pie', autopct='%1.
↪0f%%')
```

[66]: <Axes: >



- People who have partners with income greater than average are likely purchase product KP281 AND KP481

[7]:
```
sns.countplot(x='Fitness', hue='Product', data=data)
plt.title('Treadmill Preferences by Fitness Levels')
plt.xlabel('Fitness Level')
plt.ylabel('Number of Customers')
plt.show()
```



13 Customer with certain fitness levels prefer a specific product

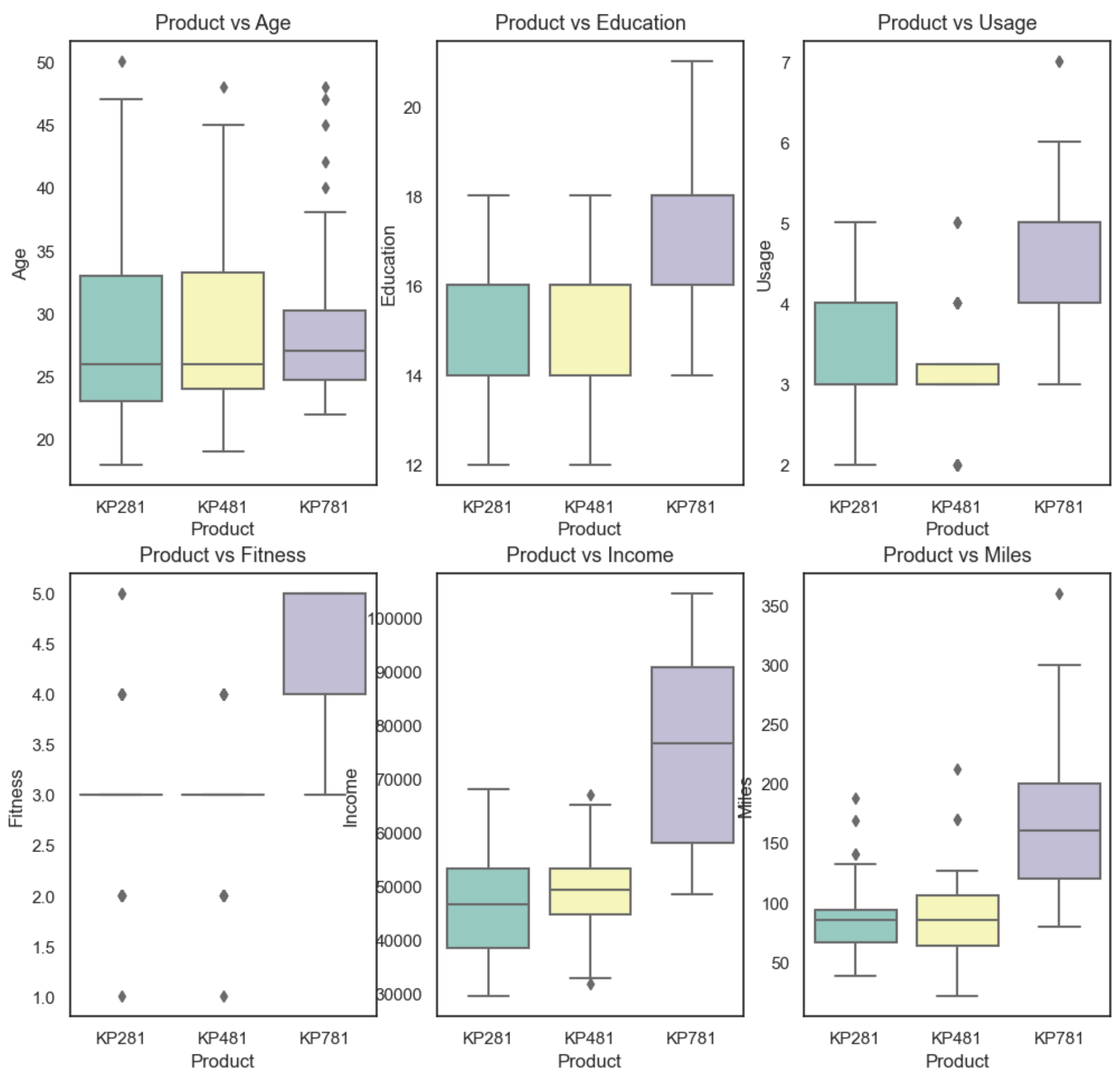
- People who purchased KP781 most expensive product has given most rating to themselves is 5
- People with fitness of 4-5 prefer KP781 product
- People with fitness > 4 prefer KP281 and KP481 product

14 Checking if following features have any effect on the product purchased:

1. Age
2. Education
3. Usage
4. Fitness
5. Income
6. Miles

[94]:

```
attrs= ['Age','Education','Usage','Fitness','Income','Miles']
fig,axs= plt.subplots(nrows=2, ncols=3, figsize=(12,8))
fig.subplots_adjust(top=1.2)
count=0
for i in range(2):
    for j in range(3):
        sns.boxplot(data=data, x='Product', y=attrs[count],ax=axs[i,j],␣
 ↪palette='Set3')
        axs[i,j].set_title(f"Product vs {attrs[count]}",pad=8, fontsize=13)
        count+=1
```



[93]:

```
filter_data= data[(data['Age']>40) & (data['Product']=='KP781')]
filter_data
```

[93]:
```
    Product  Age Gender  Education MaritalStatus  Usage  Fitness  Income  \
176   KP781   42   Male         18        Single      5        4   89641
177   KP781   45   Male         16        Single      5        5   90886
178   KP781   47   Male         18     Partnered      4        5  104581
179   KP781   48   Male         18     Partnered      4        5   95508

     Miles
176    200
177    160
178    120
179    180
```

15 Obsevation for:

16 Product vs Age

- People who purchased KP281 and KP481 have same median age
- min age of people who purchased most expensive product were starting from 23
- There are certain people who buy KP781 product after the age of 38

17 Product vs Education

- Customers whose Education is greater than 16, have more chances to purchase the KP781 product.
- till 18 years of education is max in which customer will buy product of KP281 and KP481

18 Product vs Usage

- customers who are planning to use treadmil 4 or more than 4 times a week is more likely to buy KP781 product
- 75% of people are planning to use the product greater than 4 times a week will buy KP781 product

19 Product vs Fitness

- Customers who have given themselves 4-5 rating of fitness or consider themselves fit are more likey to buy KP781

20 Product vs Income

- customers who have average income approx 50,000, prefer to buy KP281 and KP481 product
- customers who have income greater than 60 or 60 can buy KP781 product

21 Product vs Miles

- If customers expect to walk/run more than 120 miles per week, there are chances than customers can KP781 product

22 Computing Marginal Probabilities:

[101]:
```
data['Product'].value_counts(normalize=True)*100
```

[101]:
```
KP281    44.444444
KP481    33.333333
KP781    22.222222
Name: Product, dtype: float64
```

[100]:
```
data['Gender'].value_counts(normalize=True)
```

[100]:
```
Male      0.577778
Female    0.422222
Name: Gender, dtype: float64
```

[177]:
```
data['MaritalStatus'].value_counts(normalize=True)
```

[177]:
```
Partnered    0.594444
Single       0.405556
Name: MaritalStatus, dtype: float64
```

23 Computing Conditional Probabilities:

[174]:
```
# Contingency table for Product Purchased and Gender
product_gender = pd.crosstab(data['Product'], data['Gender'], margins=True)
prod_gender = pd.crosstab(data['Product'], data['Gender'], margins=True,␣
    ↪normalize='index')
print(prod_gender)
```

```
Gender     Female      Male
Product
KP281    0.500000  0.500000
KP481    0.483333  0.516667
KP781    0.175000  0.825000
All      0.422222  0.577778
```

24 Insights

- There are 82% chance if any customer is a male with a income of greater than 50,000 will buy KP781 product
- there are only 17% chance that female purchase KP781 product
- There are equal chance that customer will buy both KP281 and KP481 product

[176]:
```python
# Contingency table for Product Purchased and MaritalStatus
product_marital = pd.crosstab(data['Product'], data['MaritalStatus'],␣
↪margins=True)
product_marital
```

[176]:

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

[180]:
```python
48/80*100
```

[180]: 60.0

25 Insights

- There are 57% chance that customers who have partner can buy KP781 also around 60% partners prefer to buy KP281

26 Find probability that customers with certain income level group tends to use which product frequently?

[102]:
```python
data.Income.min()
data.Income.max()
```

[102]: 104581

[183]:
```python
bin_edges = [0, 30000, 50000, 70000, 90000, 110000]  # Adjust the bin edges␣
↪based on the minimum and maximum income
bin_labels = ['<30000', '30000-50000', '50000-70000', '70000-90000', '>90000']

data['Income_Bins'] = pd.cut(data['Income'], bins=bin_edges, labels=bin_labels,␣
↪right=False)

data[['Income', 'Income_Bins']].head(4)
```

[183]:

	Income	Income_Bins
0	29562	<30000
1	31836	30000-50000
2	30699	30000-50000
3	32973	30000-50000

[188]:
```python
cust_income = pd.crosstab(data['Income_Bins'], data['Product'],␣
↪margins=True,normalize=True)
```

```
cust_income
```

[188]:

Product	KP281	KP481	KP781	All
Income_Bins				
<30000	0.005556	0.000000	0.000000	0.005556
30000-50000	0.261111	0.166667	0.027778	0.455556
50000-70000	0.177778	0.166667	0.066667	0.411111
70000-90000	0.000000	0.000000	0.061111	0.061111
>90000	0.000000	0.000000	0.066667	0.066667
All	0.444444	0.333333	0.222222	1.000000

[190]:
```
# Probability of Purchasing Product 'KP781' AND income is between 30-50000
5/180
```

[190]: 0.027777777777777776

[187]:
```
# probabilty of customer purchase product KP281 whose income is between 30-50000
47/82
```

[187]: 0.573170731707317

27 Customer Profiling -

Customer Profiles for KP781

1. Only people having incomes greater than 70k have run over 220 miles and all of then use KP781.
2. Recommend KP781 if one or more conditions are satisfied along with a necessary condition of Income > 70000:-

a) Education Level >= 18 b) Usage days > = 5 c) Fitness Levels = 5 d) The person runs more than 150 miles(80% of them use KP781)

3. Never Recommend KP781 if one or more of these conditions are satisfied:-

a) Education Levels < 14 b) Fitness < 3 c) Age < 20 d)Income < 45000 e) Miles run < 50

Why very few women have bought the luxurious KP781 treadmill?

- Only 2 women have incomes over 70k which is certainly the reason for a large proportion of them not buying KP781(affordability).

Customer Profiles for KP281:

1. Women having incomes below 70k and age > 40
2. Customers having income in range 60k-70k and usage days=3
3. Customers having income in range 45k-50k and usage days=2
4. Customers having income in range 35k-45k and usage days=4
5. Customers having income in range 50k-60k and usage days=4
6. Customers with Fitness=4, age closer to 40 and income 50k-60k
7. Customers with Education Level=16, Age>32 and income 45k-50k

8. Customers with Education Level=16, Age>45 and income 60k-70k
9. Customers with Age in 25-30 and 35-40 having incomes in range 35k-45k
10. Customers with 40+ Age and 60k-70k income
11. Women with incomes < 35k and whose miles run < 105
12. Customers with usages=5, incomes in range 35k-45k and who run more than 140 miles
13. Customers with Fitness=5, incomes < 70k and Incomes in 45k-50k
14. Customers with Education level=15 having incomes less than 35k
15. Customers with Usages=3, miles run < 70 and Age>40
16. Customers with Usages=2 and Age between 25-30

Customer Profiles for KP481:

1. Women having incomes below 70k and age between 32-37
2. Customers with age < 25, incomes in range 50-60k and the miles run is in the range 100-150
3. Customers with Fitness=4, age in range 25-32 and income 50k-60k
4. Customers with Education Level=16, Age< 22 and income 45k-50k
5. Customers with Education Level=16, Age< 35 and income 60k-70k
6. Customers with 35-40 Age and 60k-70k income
7. Women with incomes < 35k and whose miles run >105
8. Men with incomes 60k-70k and who tread in range 100-150 miles
9. Customers with Fitness=4, incomes < 45k-50k and who run more than 100 miles
10. Customers with Education level=13 having incomes in ranges 45-60k
11. Customers with Usages=2 and Age>40

28 Insights

- Target Audience: mostly from age 22 to 30, purchase most of our product.
- Men are most interested in KP781 product
- people with higher income greater than average are buying KP781
- People who have partners with income greater than average are likely purchase product KP281 AND KP481
- Customer with certain fitness levels prefer a specific product - People who purchased KP781 most expensive product has given most rating to themselves is - People with fitness of 4-5 prefer KP781 product - People with fitness > 4 prefer KP281 and KP481 product
- Obsevation for: - Product vs Age - People who purchased KP281 and KP481 have same median age - min age of people who purchased most expensive product were starting from 23 - There are certain people who buy KP781 product after the age of 38
- Product vs Education - Customers whose Education is greater than 16, have more chances to purchase the KP781 product. - till 18 years of education is max in which customer will buy product of KP281 and KP481
- Product vs Usage - customers who are planning to use treadmil 4 or more than 4 times a week is more likely to buy KP781 product - 75% of people are planning to use the product greater than 4 times a week will buy KP781 product

- Product vs Fitness - Customers who have given themselves 4-5 rating of fitness or consider themselves fit are more likey to buy KP781
- Product vs Income - customers who have average income approx 50,000, prefer to buy KP281 and KP481 product - customers who have income greater than 60 or 60 can buy KP781 product
- Product vs Miles - If customers expect to walk/run more than 120 miles per week, there are chances than customers can KP781 product

29 More Insights

- There are 82% chance if any customer is a male with a income of greater than 50,000 (avg) will buy KP781 product
- there are only 17% chance that female purchase KP781 product
- There are equal chance that customer will buy both KP281 and KP481 product
- there are 57% chance that customer purchase product KP281 whose income is between 30-50000
- Also there are high chances that if customer has greater than 90000 income will buy KP781 product

30 Recommendations:

- Market Expansion: Evaluate opportunities to expand into new markets or distribution channels based on the identified target audience characteristics. This could involve partnerships with fitness clubs, online retailers, or targeting specific demographics in different geographic regions.
- Targeted Marketing Campaigns: Utilize the information about the target audience to tailor marketing campaigns specifically for each product. For instance, focus on highlighting the advanced features and benefits of the KP781 treadmill to appeal to customers with higher income and fitness levels.
- Customer Engagement and Retention: Implement strategies to engage with existing customers and encourage repeat purchases. This could include loyalty programs, personalized recommendations based on past purchases, and after-sales support to ensure customer satisfaction.
- Social Proof and Testimonials: Showcase testimonials and success stories from satisfied customers who have benefited from using AeroFit treadmills, especially the KP781 model. Positive reviews and word-of-mouth recommendations can help build trust and credibility, driving more sales.