OBJECTIVE

- Investigate whether there are differences across the product with respect to customer characteristics.
- Find the target audience for each type of treadmill with analysis and provide a better recommendation to the new customers.

In [463]:

```
# !pip install pandas-profiling
```

...

In [464]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

```

In [5]:

```
af  = pd.read_csv(r"C:\Users\Acer\Downloads\aerofit_treadmill.csv")
```

In [6]:

```
af
```

Out[6]:

	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

```
#### Challenges
Age: Given in years which can be changed into different bins for easy classification
```

```
Fitness: Given as a numerical value which is defined in the problem statement which can be changes to a catagorical value
```

Observations on the shape of data & Data types of all the attributes

In [12]:

```python
af.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
```

Observations

- There are 9 columns and 180 rows in the dataframe
- There seems to be a mix of catagorical and numerical values in the data
- Cannot find any null or missing vsalues in primary check

...

Null value detection

In [41]:

```python
null_value = af.loc[af.isnull().values.any(axis =1)]
null_value
```

Out[41]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles

Obeservation

- there is no null value , confirmed

In [13]:

```
af.describe(include = object)
```

Out[13]:

	Product	Gender	MaritalStatus
count	180	180	180
unique	3	2	2
top	KP281	Male	Partnered
freq	80	104	107

Observations

```
- There are mainly 3 products : KP281, KP481, or KP781
- 2 Genders - Male and Female
- In the primary look there seems to be a male dominance in usage and KP281 being
 the most preferred product.
```

....

Converting Objects into Catagories

In [16]:

```
for col in ['Product', 'Gender', 'MaritalStatus']:
    af[col] = af[col].astype('category')
```

In [17]:

```
af.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
```

Observations

- Converted Product, Gender, MaritalStatus to catagorical values inorder to work hasslefree.

In [469]:

```
af['Education'].mean()
```

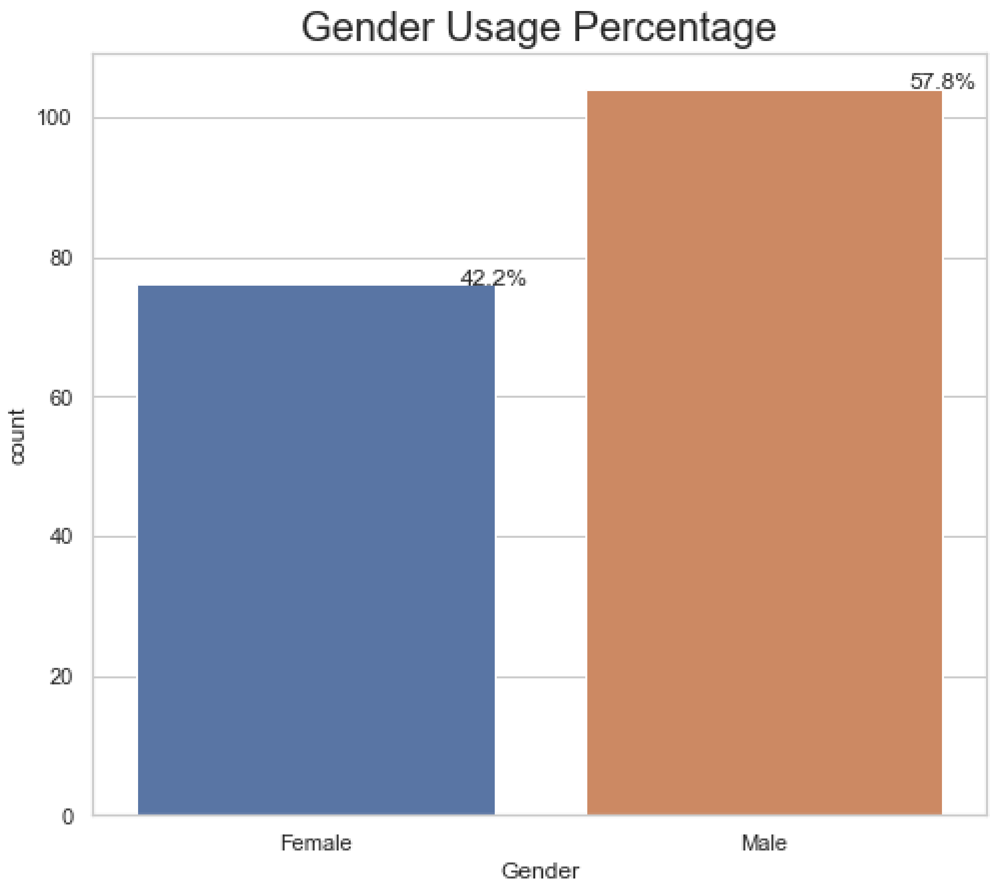
Out[469]:

```
15.572222222222223
```

Visual Analysis - Univariate

In [278]:

```
# sns.countplot(x = 'Gender' , data = af)
# print(af['Gender'].value_counts())

sns.set(style="whitegrid")
plt.figure(figsize=(8,7))
total = float(len(af))
ax = sns.countplot(x="Gender", data=af)
plt.title('Gender Usage Percentage', fontsize=20)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height()/total)
    x = p.get_x() + p.get_width()
    y = p.get_height()
    ax.annotate(percentage, (x, y),ha='center')
plt.show()
```

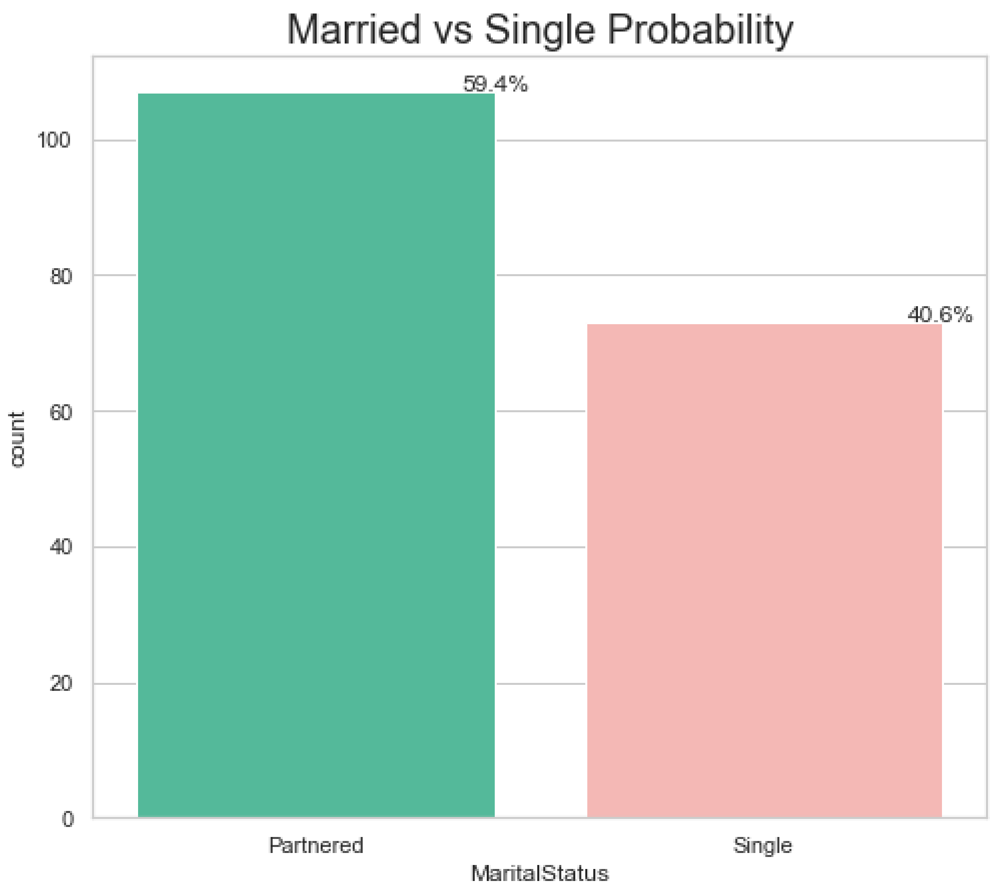


Observations

- Its evident that more males use the products than females
- Around 15% more males use the product

In [321]:

```
sns.set(style="whitegrid")
plt.figure(figsize=(8,7))
total = float(len(af))
ax = sns.countplot(x="MaritalStatus", data=af , palette=['#43CAA1',"#FFAEAB"])
plt.title('Married vs Single Probability', fontsize=20)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height()/total)
    x = p.get_x() + p.get_width()
    y = p.get_height()
    ax.annotate(percentage, (x, y),ha='center')
plt.show()
```

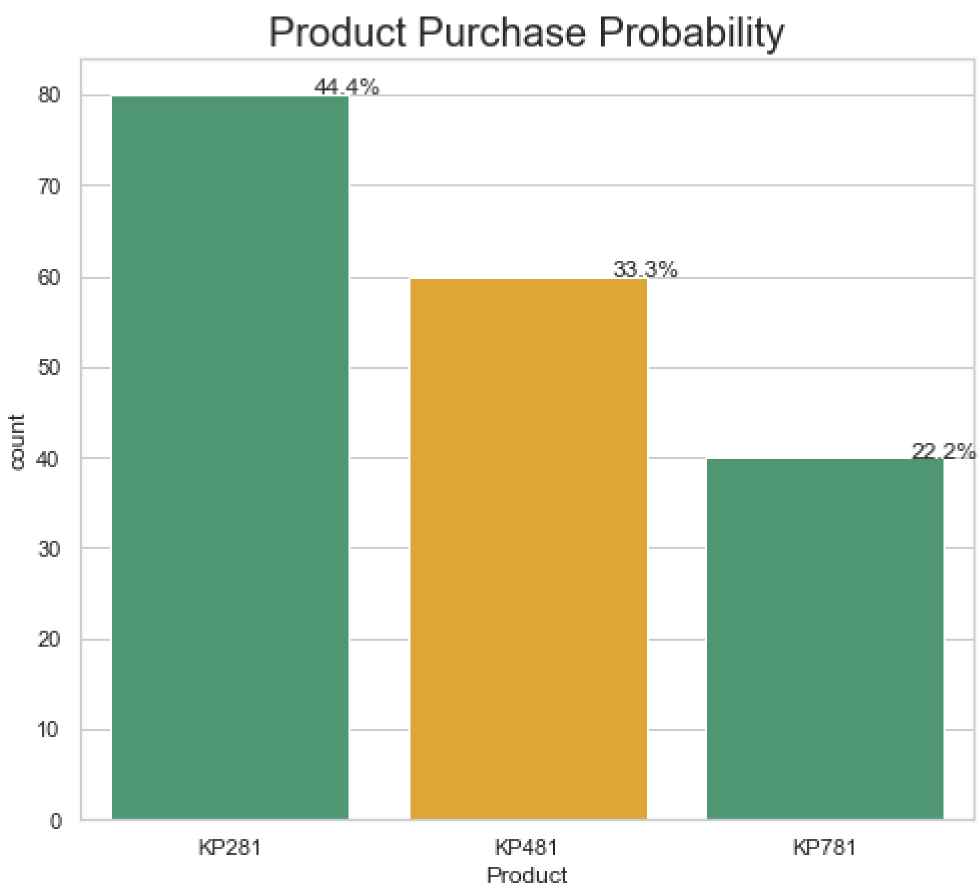


Observations

- more users are partnered / married.
- 20% more users are married : this data can be used more.

In [316]:

```
sns.set(style="whitegrid")
plt.figure(figsize=(8,7))
total = float(len(af))
ax = sns.countplot(x="Product", data=af,   palette=['#43A371',"#FAAE1B"])
plt.title('Product Purchase Probability', fontsize=20)
for p in ax.patches:
    percentage = '{:.1f}%'.format(100 * p.get_height()/total)
    x = p.get_x() + p.get_width()
    y = p.get_height()
    ax.annotate(percentage, (x, y),ha='center')
plt.show()


print('**************************************************')

print(df.groupby('Product')['Price'].sum())

print('**************************************************')
```




```
**************************************************
Product
KP281    9327200
KP481    8160600
KP781    7772560
Name: Price, dtype: int64
**************************************************
```

Observations

- the data clearly shows KP 281 as the leading product which happen to be the cheapest product and the most sold item
- KP 481 as the second leading product followed by KP 781

KP 281

In [101]:

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

Out[101]:

	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

Observations on KP281

```
- Average customer age is 28 with most falls in the age range of 23 - 33
- its used atleast 3 times a week
- a total of 80 customers purchased the product
```

KP 481

In [102]:

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

Out[102]:

	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 on KP781

```
- Average customer age is 29 with most falls in the age range of 22 - 30.
- its used atleast 4 times a week
- a total of 40 customers purchased the product making it least purchased item
```

KP 781

In [30]:

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

Out[30]:

	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

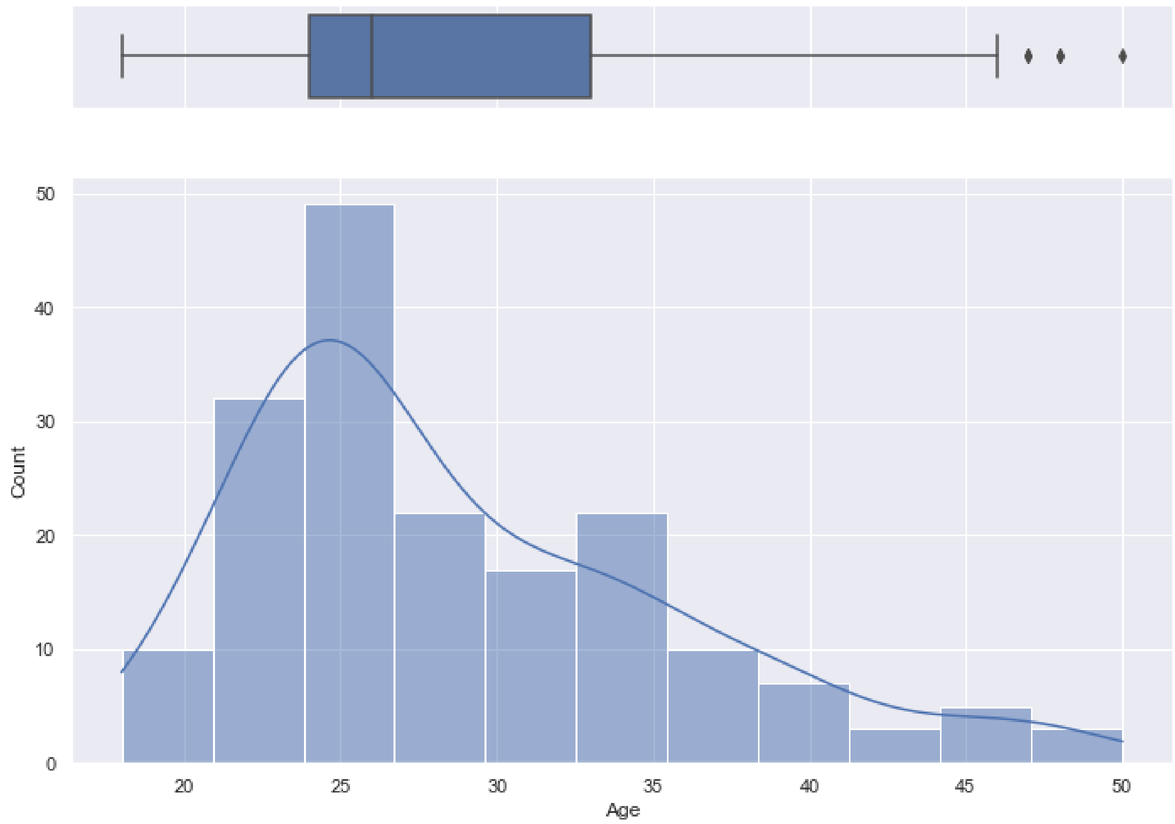
Observations on KP481

- Average customer age is 28 with most falls in the age range of 24 - 33.
- its used atleast 3 times a week
- a total of 60 customers purchased the product making it least purchased item

Treating Outliers

In [402]:

```
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15,
sns.boxplot(x = af['Age'] , data = af , orient = 'h' , ax = ax_box )
sns.set(rc={'figure.figsize':(5,8.27)})
sns.histplot(x = af['Age'] , data = af  , ax = ax_hist , kde = True )
ax_box.set(xlabel='')
plt.show()
```



In [403]:

```
Q3 = af["Age"].quantile(0.75)
Q1 = af["Age"].quantile(0.25)

IQR = Q3 - Q1

upper = Q3 + (1.5 * IQR)
lower = Q1 - (1.5 * IQR)

print(upper , lower)
```

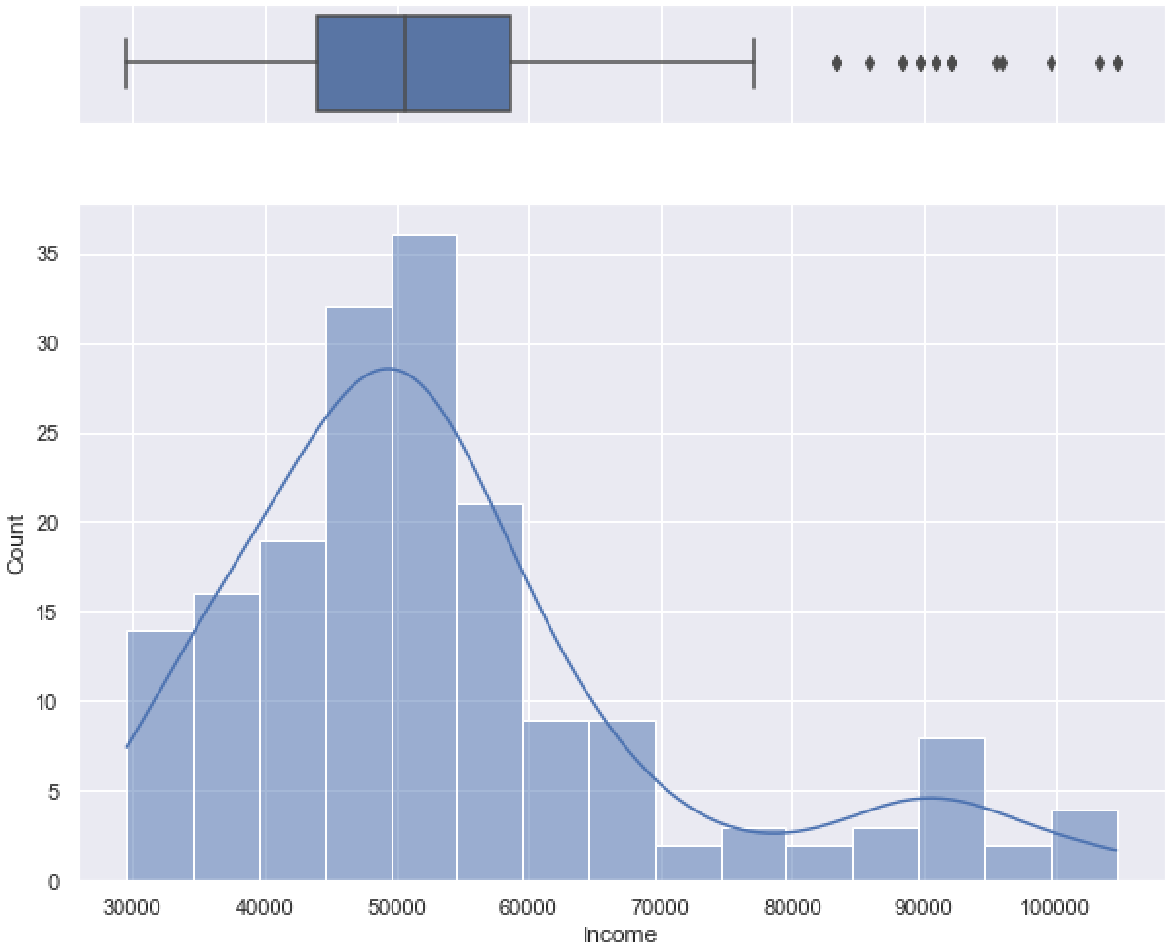
```
46.5 10.5
```

In [ ]:

```

```

In [406]:

```
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15,
sns.boxplot(x = af['Income'] , data = af , orient = 'h' , ax = ax_box )
sns.set(rc={'figure.figsize':(5,8.27)})
sns.histplot(x = af['Income'] , data = af  , ax = ax_hist , kde = True )
ax_box.set(xlabel='')
plt.show()
```



In [407]:

```
Q3 = af["Income"].quantile(0.75)
Q1 = af["Income"].quantile(0.25)

IQR = Q3 - Q1

upper = Q3 + (1.5 * IQR)
lower = Q1 - (1.5 * IQR)

print(upper , lower)
```

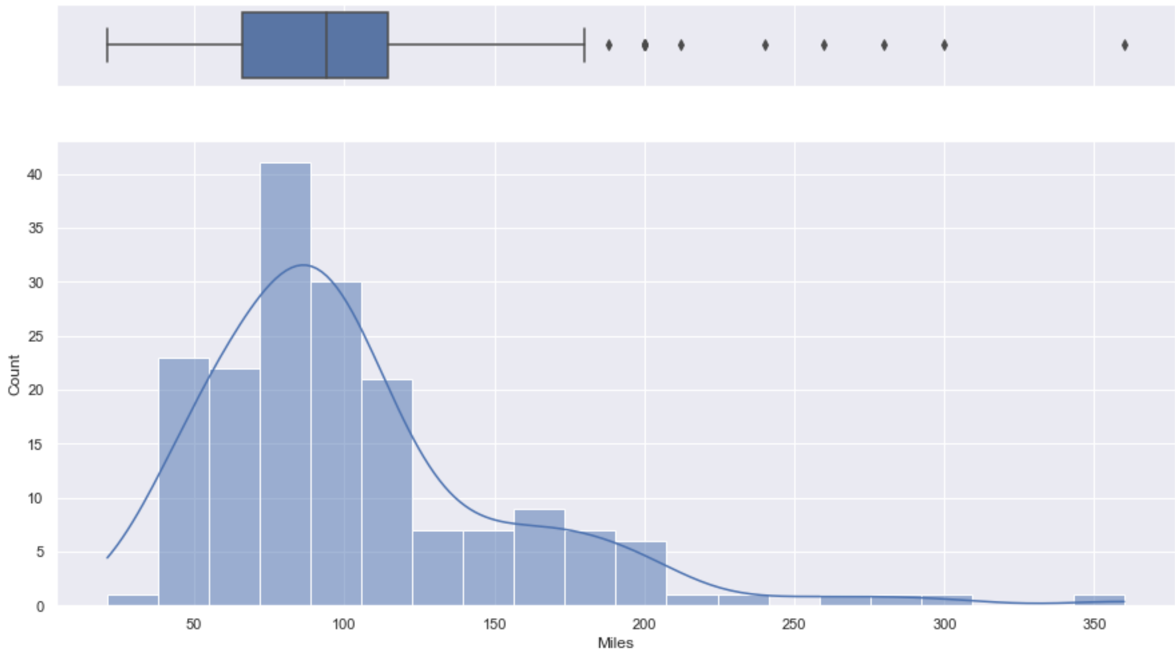
```
80581.875 22144.875
```

In [ ]:

```
Observations
```

In [420]:

```
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True, gridspec_kw={"height_ratios": (.15,
sns.boxplot(x = af['Miles'] , data = af , orient = 'h' , ax = ax_box )
sns.set(rc={'figure.figsize':(5,8.27)})
sns.histplot(x = af['Miles'] , data = af  , ax = ax_hist , kde = True )
ax_box.set(xlabel='')
plt.show()
```



In [422]:

```
Q3 = af["Miles"].quantile(0.75)
Q1 = af["Miles"].quantile(0.25)

IQR = Q3 - Q1

upper = Q3 + (1.5 * IQR)
lower = Q1 - (1.5 * IQR)

print(upper , lower)
```

```
187.875 -7.125
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

Bivariate Visual Analysis

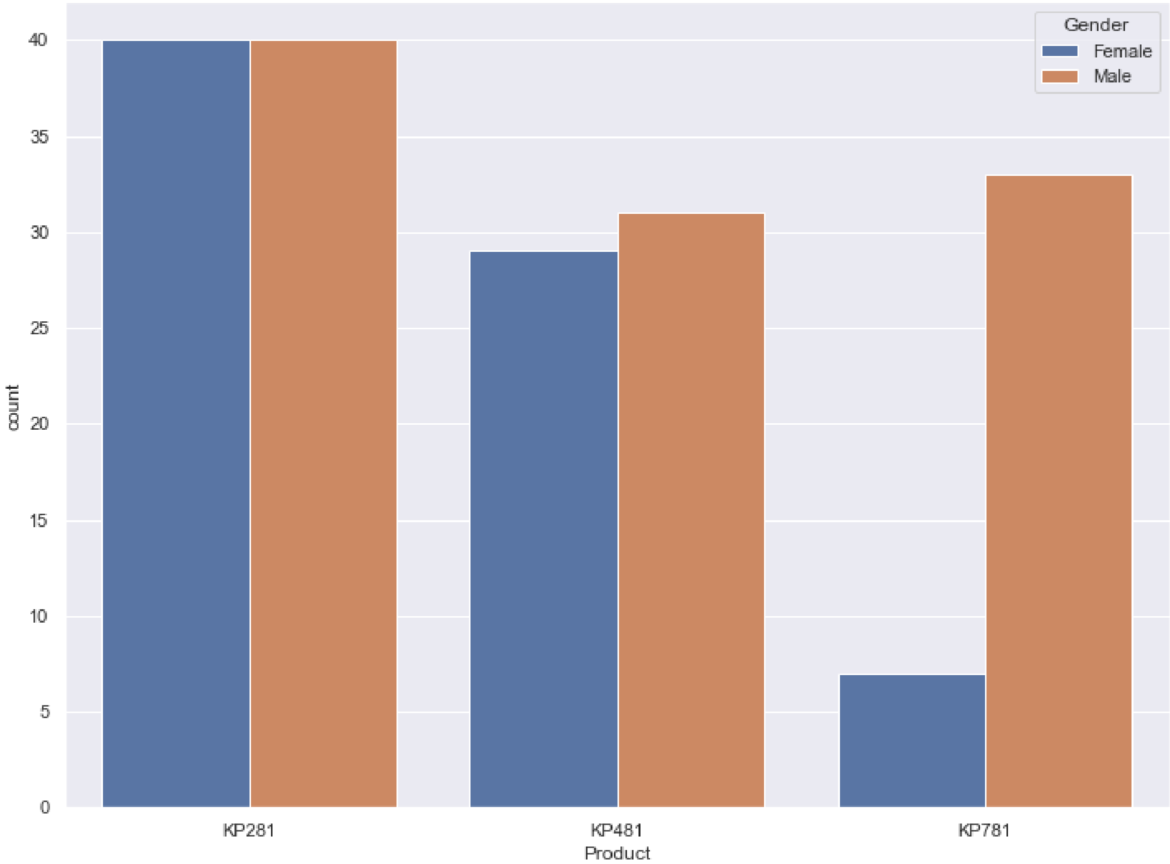
In [292]:

```
print('**************************************************')
print(pd.crosstab(af['Product'] , af['Gender']))
print('**************************************************')

sns.set(rc={'figure.figsize':(12,9)})
sns.countplot(x = 'Product'  , hue = 'Gender' , data = af)

```

```
**************************************************
Gender   Female  Male
Product
KP281        40    40
KP481        29    31
KP781         7    33
**************************************************
```

Out[292]:

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



Observation

- We can see that in all the products , Male users dominate the usage

- In KP 281 we can see that the number of female users are equal to that of male, making it most favourable for couples or females.

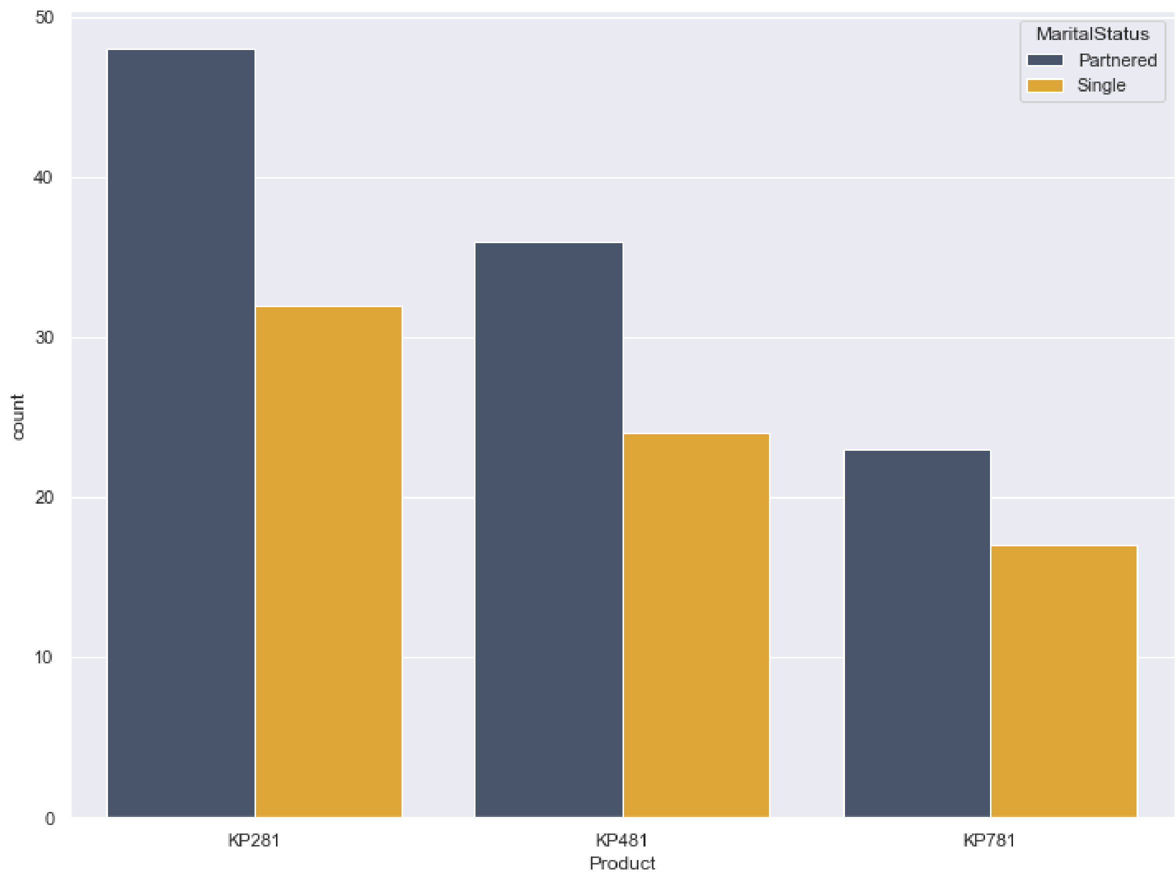
In [304]:

```
print('----------------------------------------------------')
print(pd.crosstab(af['Product'] , af['MaritalStatus']))
print('----------------------------------------------------')


sns.countplot(x = 'Product'  , hue = 'MaritalStatus' , data = af ,  palette=['#435371',
```

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

Out[304]:

<AxesSubplot:xlabel='Product', ylabel='count'>



Observation

- For every Product the numbers are dominated by Partnered when compared to single.
- This shows that more than focusing on the relationship status we can focus on their fitness and usage to draw more information and recommendation

In [54]:

```
1 pd.crosstab(index = [af['Product'] , af['MaritalStatus']] , columns = af['Gender'] , ma
```

Out[54]:

	Gender	Female	Male	All
Product	**MaritalStatus**			
KP281	**Partnered**	27	21	48
	Single	13	19	32
KP481	**Partnered**	15	21	36
	Single	14	10	24
KP781	**Partnered**	4	19	23
	Single	3	14	17
All		76	104	180

Observations

- this data shows that of leading Women who use the product KP 281 most are married
- For every product the usage is dominated by Married couples than Singles

converting age to certain bins

In [87]:

```
1 af['Age_Group'] = af['Age']
```

In [88]:

```
1 af["Age_Group"] = pd.cut(af["Age_Group"], bins =[0,21,35,45,60], include_lowest=True, l
```

In [89]:

```
af
```

Out[89]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Age_Gr
0	KP281	18	Male	14	Single	3	4	29562	112	Teen(0
1	KP281	19	Male	15	Single	2	3	31836	75	Teen(0
2	KP281	19	Female	14	Partnered	4	3	30699	66	Teen(0
3	KP281	19	Male	12	Single	3	3	32973	85	Teen(0
4	KP281	20	Male	13	Partnered	4	2	35247	47	Teen(0
...	...	...	...	...	...	...	...	...	...	
175	KP781	40	Male	21	Single	6	5	83416	200	Adult(35
176	KP781	42	Male	18	Single	5	4	89641	200	Adult(35
177	KP781	45	Male	16	Single	5	5	90886	160	Adult(35
178	KP781	47	Male	18	Partnered	4	5	104581	120	Towards_ age(45
179	KP781	48	Male	18	Partnered	4	5	95508	180	Towards_ age(45

180 rows × 10 columns

In [90]:

```
pd.crosstab(af['Product'] , af['Age_Group'])
```

Out[90]:

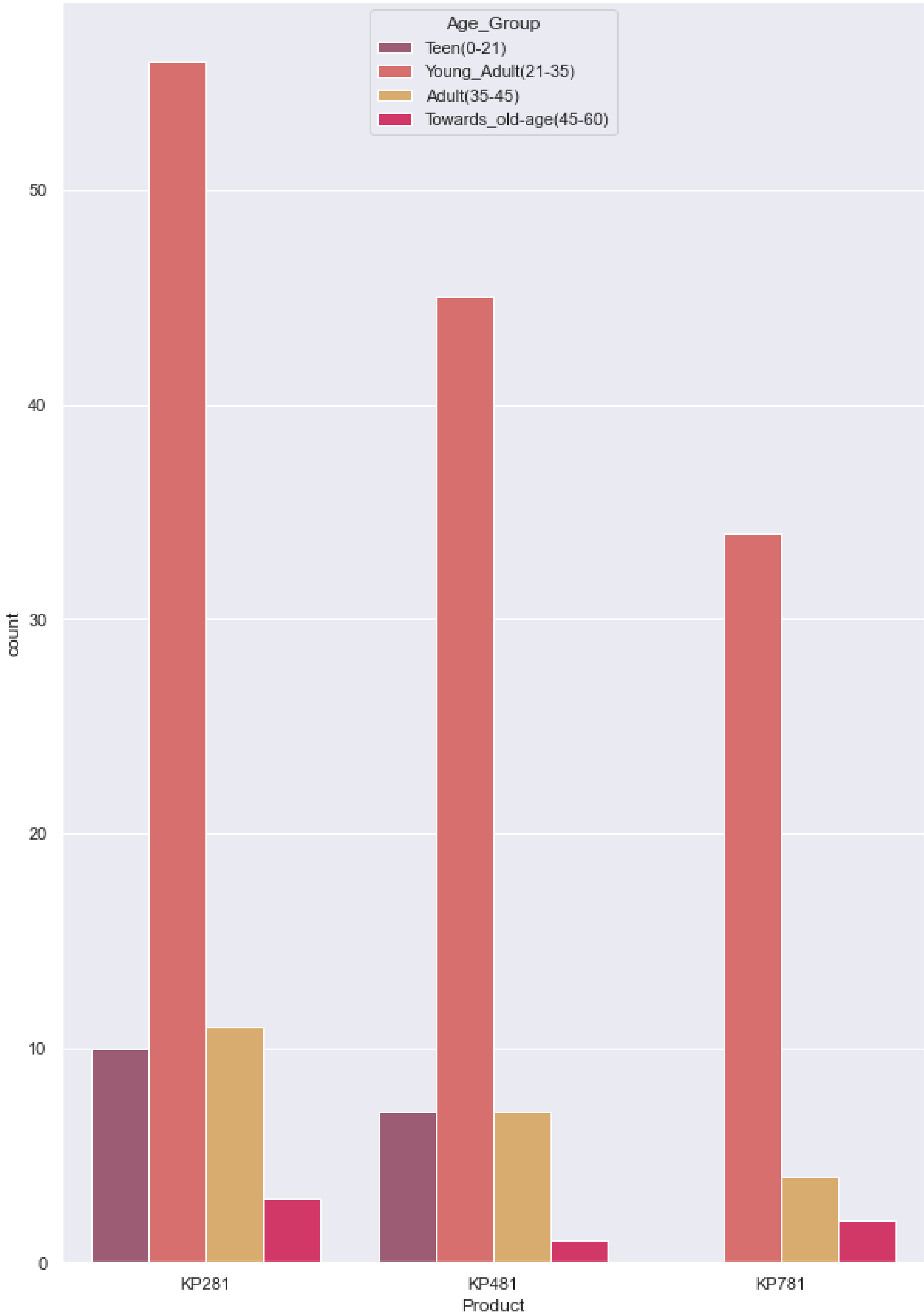
Age_Group	Teen(0-21)	Young_Adult(21-35)	Adult(35-45)	Towards_old-age(45-60)
Product				
KP281	10	56	11	3
KP481	7	45	7	1
KP781	0	34	4	2

In [331]:

```
1 sns.countplot(x = 'Product'  , hue = 'Age_Group' , data = af ,  palette=['#A75171',"#EA
```

Out[331]:

<AxesSubplot:xlabel='Product', ylabel='count'>



Observation

- In every product the number of Young Adult between age 21 - 35 is dominated
- signifying the amount of couples and singles usage and preferences in the catagory

Changing Fitness Numericals Into Catagorical Variables

In [92]:

```
df = af.copy()
```

In [93]:

```
df['Fitness_level'] = df['Fitness'].astype('object')
```

In [94]:

```
df['Fitness_level'] = df['Fitness_level'].replace({1: 'poor fitness',
                                                    2: 'low fitness',
                                                    3: 'marginal fitness',
                                                    4: 'good fitness',
                                                    5: 'high performance'})
```

In [95]:

```
df.drop(columns = ['Age' , 'Fitness'])
```

Out[95]:

	Product	Gender	Education	MaritalStatus	Usage	Income	Miles	Age_Group	Fitness_le
0	KP281	Male	14	Single	3	29562	112	Teen(0-21)	good fitn
1	KP281	Male	15	Single	2	31836	75	Teen(0-21)	marg fitn
2	KP281	Female	14	Partnered	4	30699	66	Teen(0-21)	marg fitn
3	KP281	Male	12	Single	3	32973	85	Teen(0-21)	marg fitn
4	KP281	Male	13	Partnered	4	35247	47	Teen(0-21)	low fitn
...	...	...	...	...	...	...	...	...	
175	KP781	Male	21	Single	6	83416	200	Adult(35-45)	I performa
176	KP781	Male	18	Single	5	89641	200	Adult(35-45)	good fitn
177	KP781	Male	16	Single	5	90886	160	Adult(35-45)	I performa
178	KP781	Male	18	Partnered	4	104581	120	Towards_old-age(45-60)	I performa
179	KP781	Male	18	Partnered	4	95508	180	Towards_old-age(45-60)	I performa

180 rows × 9 columns

In [100]:

```
1 pd.crosstab(index = [df['Product'] , df['Fitness_level']] , columns = df['Gender'] , ma
```

Out[100]:

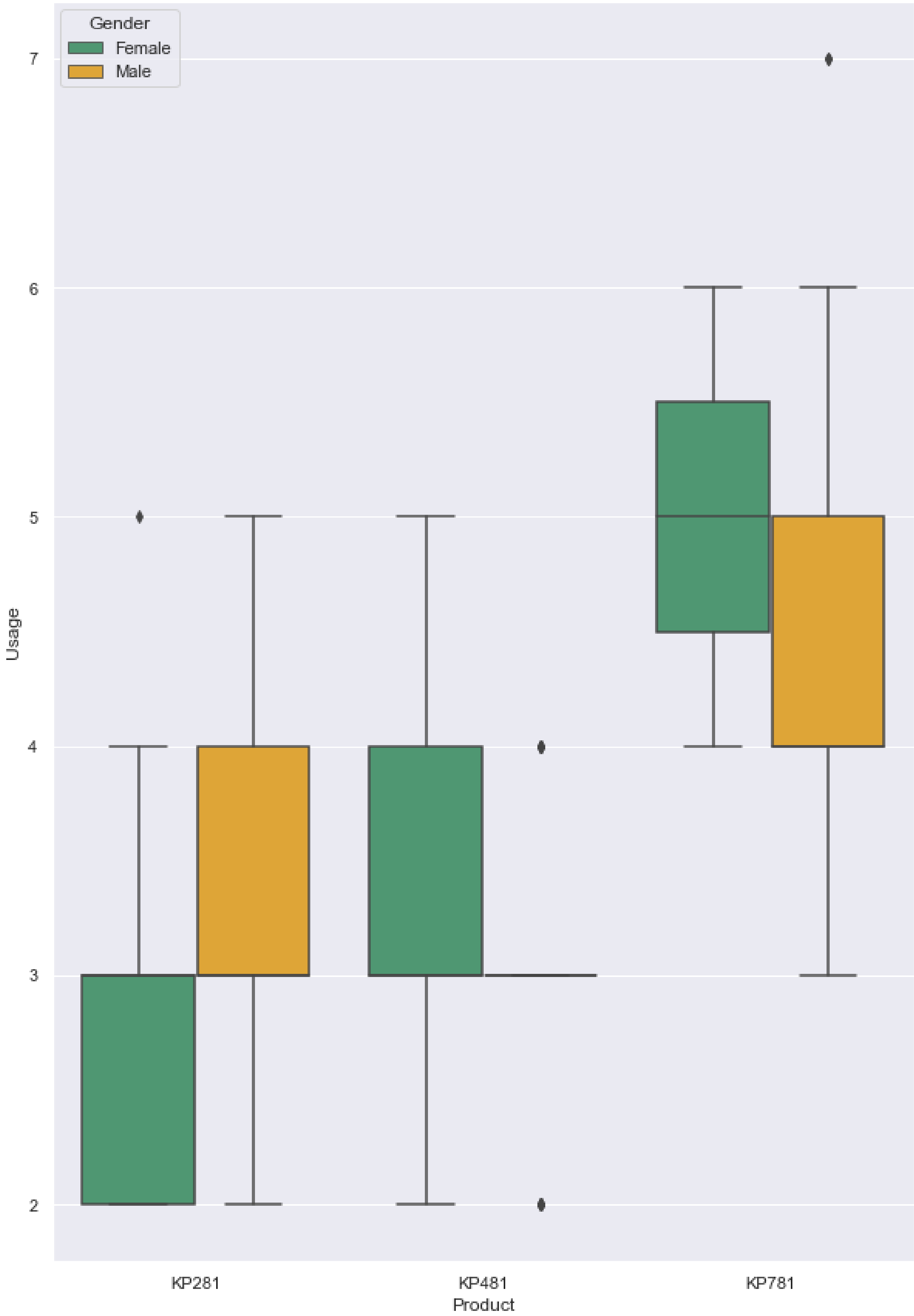
	Gender	Female	Male	All
Product	**Fitness_level**			
KP281	**good fitness**	3	6	9
	high performance	1	1	2
	low fitness	10	4	14
	marginal fitness	26	28	54
	poor fitness	0	1	1
KP481	**good fitness**	4	4	8
	low fitness	6	6	12
	marginal fitness	18	21	39
	poor fitness	1	0	1
KP781	**good fitness**	1	6	7
	high performance	5	24	29
	marginal fitness	1	3	4
All		76	104	180

Observations

- Product KP 781 has a base of only Good , Marginal and High performers sowing its mostly used by athletes or performers than day to day users.
- KP 281 has the most Users in the low and marginal catagory showing these are the products used by customers who is trying to get into shape.
- KP 481 has not high performance users which can be focused for improvement in the future

In [386]:

```
sns.set(style="darkgrid")
sns.boxplot(x="Product", y="Usage", hue ='Gender', data = af, palette=['#43A371',"#FAAE
sns.set(rc={'figure.figsize':(10,15)})
```



Observations

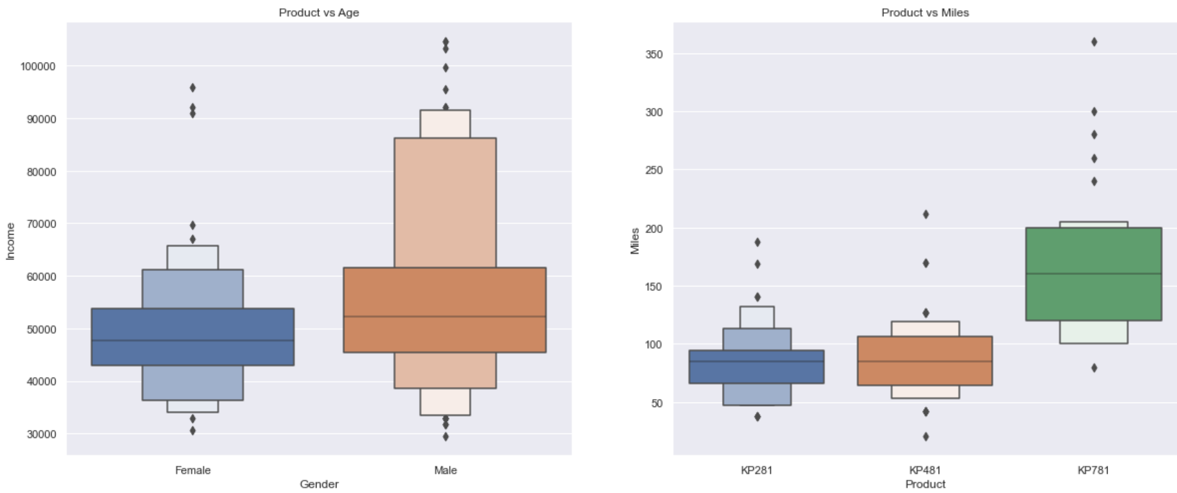
- for KP 281 the Male users are in much more of frequent users than the females.
- KP 781 females are the more frequent users here compared to males

In [428]:

```
fig, axes = plt.subplots(figsize=(20, 8), nrows=1, ncols=2)
sns.boxenplot(x='Gender',y='Income',data=af, ax = axes[0])
axes[0].set_title("Product vs Age")
# ax = sns.stripplot(x="Gender", y="Income", data=af,size=4, color=".26")

sns.boxenplot(x='Product',y='Miles',data=af, ax=axes[1])
axes[1].set_title("Product vs Miles")
# ax = sns.stripplot(x="Product", y="Miles", data=af, size=4, color=".26")
```

Out[428]:

```
Text(0.5, 1.0, 'Product vs Miles')
```



Observations

- From the fig - Females Income is mostly focused between 4000 - 5000 range where as males is between 5000 - 6000 with large spread from 3500 to 8500 - this shows that the male customers are more euipped to buy the high end products
- In figure 2 - the KP 781 leads the miles run by a fair distance while compared to other products

In [144]:

```
df['Price'] = df['Product']
```

In [154]:

```
df['Price'].replace({'KP281' : 116590,
                     'KP781' : 194314,
                     'KP481' : 136010} , inplace = True)
```

In [155]:

```
df
```

Out[155]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Age_Gr
0	KP281	18	Male	14	Single	3	4	29562	112	Teen(0
1	KP281	19	Male	15	Single	2	3	31836	75	Teen(0
2	KP281	19	Female	14	Partnered	4	3	30699	66	Teen(0
3	KP281	19	Male	12	Single	3	3	32973	85	Teen(0
4	KP281	20	Male	13	Partnered	4	2	35247	47	Teen(0
...	...	...	...	...	...	...	...	...	...	
175	KP781	40	Male	21	Single	6	5	83416	200	Adult(35
176	KP781	42	Male	18	Single	5	4	89641	200	Adult(35
177	KP781	45	Male	16	Single	5	5	90886	160	Adult(35
178	KP781	47	Male	18	Partnered	4	5	104581	120	Towards_ age(45
179	KP781	48	Male	18	Partnered	4	5	95508	180	Towards_ age(45

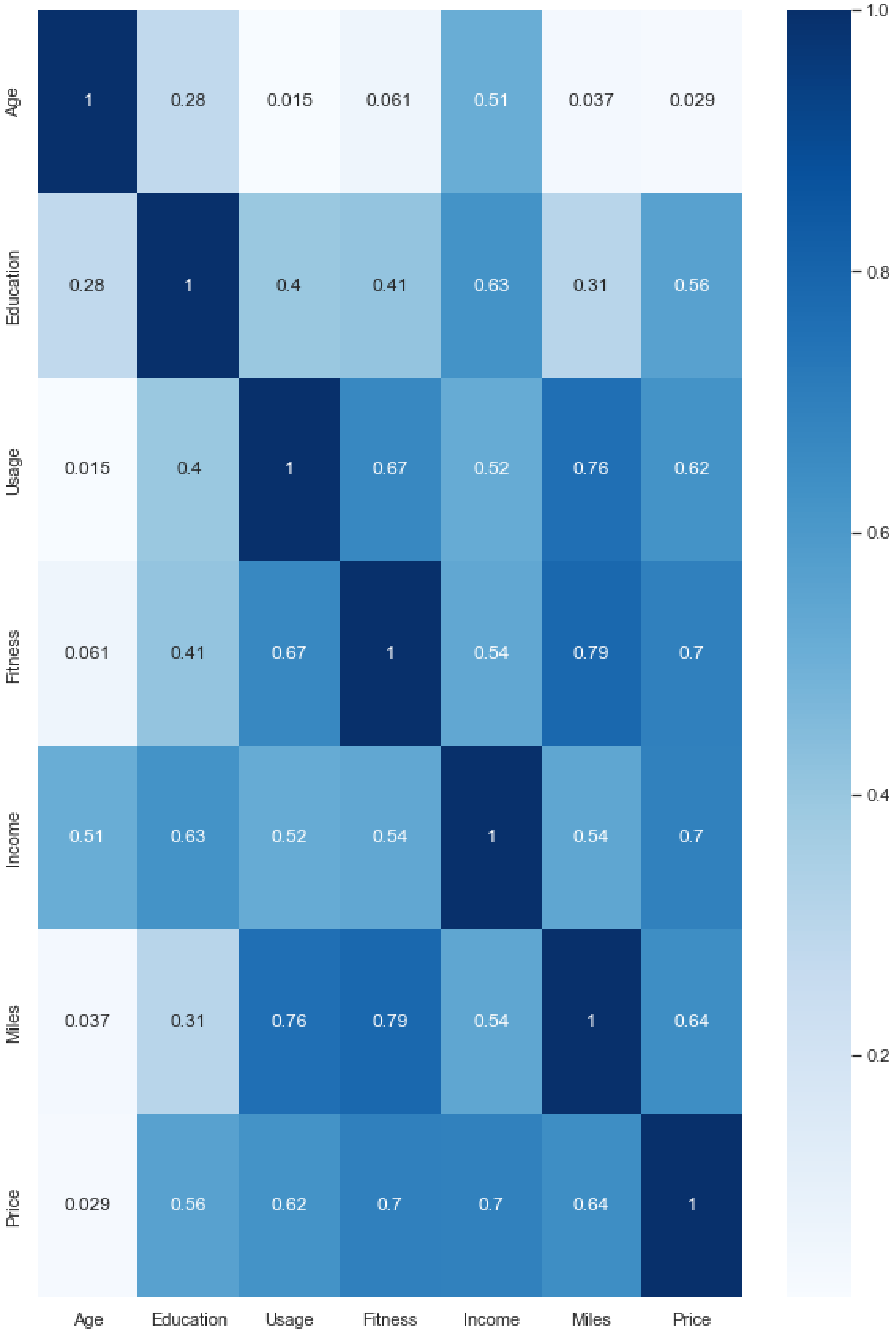
180 rows × 12 columns

In [158]:

```
sns.heatmap(df.corr(), cmap = 'Blues' , annot = True)
```

Out[158]:

<AxesSubplot:>



In [198]:

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

```
Age        Age          1.000000
           Income       0.513414
Education  Education    1.000000
           Income       0.625827
           Price        0.563487
Usage      Usage        1.000000
           Fitness      0.668606
           Income       0.519537
           Miles        0.759130
           Price        0.623157
Fitness    Usage        0.668606
           Fitness      1.000000
           Income       0.535005
           Miles        0.785702
           Price        0.696657
Income     Age          0.513414
           Education    0.625827
           Usage        0.519537
           Fitness      0.535005
           Income       1.000000
           Miles        0.543473
           Price        0.695870
Miles      Usage        0.759130
           Fitness      0.785702
           Income       0.543473
           Miles        1.000000
           Price        0.643948
Price      Education    0.563487
           Usage        0.623157
           Fitness      0.696657
           Income       0.695870
           Miles        0.643948
           Price        1.000000
dtype: float64
```

Observations

- Unstacked all the correlation points which are greater than 0.5 so that it gets easy to work on the probability and multi variate analysis
- will be focussing on
  - USAGE : FITNESS WRT (PRODUCT , GENDER , AGE , MARITAL STATUS)

- MILES : FITNESS WRT (PRODUCT , GENDER , AGE , MARITAL STATUS)|
- USAGE : MILES WRT (PRODUCT , GENDER , AGE , MARITAL STATUS)

In [475]:

```
# df
```

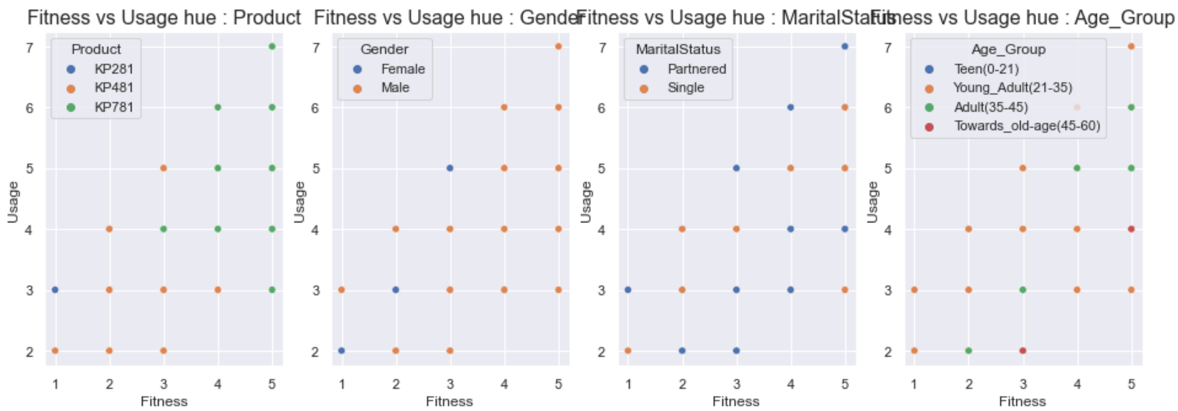
In [476]:

```
# df.info()
```

USAGE : FITNESS WRT (PRODUCT , GENDER , AGE , MARITAL STATUS)

In [434]:

```
fig, ax = plt.subplots(1 ,4 , figsize=(16, 5))
ax = ax.flatten()
i=0
for category in df.select_dtypes('category').columns:
    sns.scatterplot(data=df, x='Fitness', y='Usage', hue=category, ax=ax[i])
    ax[i].set_title(f'Fitness vs Usage hue : {category}', size=16)
    i = i+1
plt.show()
```



Observation

- Most high performance is from KP 781
- Most frequent users are males
- Most high performance users are single
- Adults are the most fit age group

In [430]:

```
# df.info()
```

In [429]:

```
# fig1, axes1 =plt.subplots(3,2,figsize=(14, 19))
# 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_c

```

MILES : FITNESS WRT (PRODUCT , GENDER , AGE , MARITAL STATUS)|

In [171]:

```
fig, ax = plt.subplots(1 ,4 , figsize=(16, 5))
ax = ax.flatten()
i=0
for category in df.select_dtypes('category').columns:
    sns.scatterplot(data=df, x='Fitness', y='Miles', hue=category, ax=ax[i])
    ax[i].set_title(f'Fitness vs Miles hue : {category}', size=16)
    i = i+1
plt.show()
```

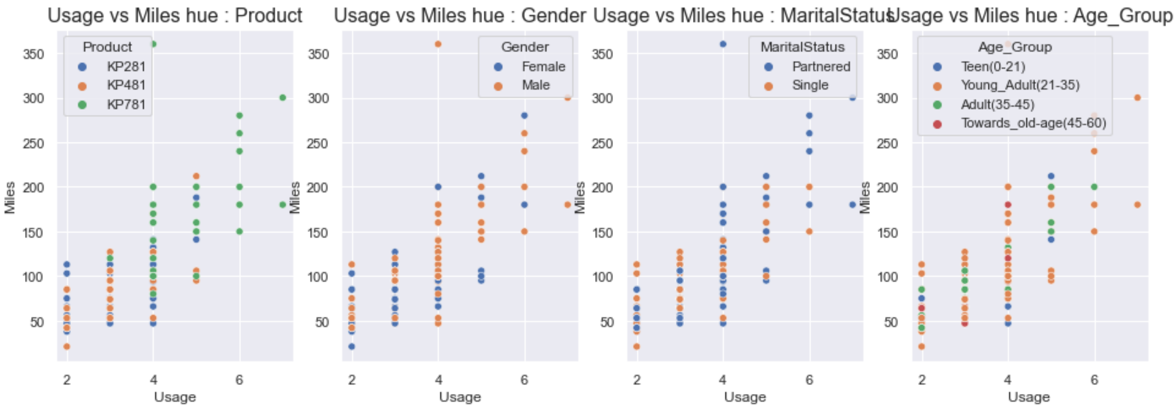
Observations

- Most miles run are with the KP 781 Product
- Most Well shaped and Healthy catagory are dominated by males
- Most miles run are by the Young adults

USAGE : MILES WRT (PRODUCT , GENDER , AGE , MARITAL STATUS)

In [173]:

```
fig, ax = plt.subplots(1 ,4 , figsize=(16, 5))
ax = ax.flatten()
i=0
for category in df.select_dtypes('category').columns:
    sns.scatterplot(data=df, x='Usage', y='Miles', hue=category, ax=ax[i])
    ax[i].set_title(f'Usage vs Miles hue : {category}', size=16)
    i = i+1
plt.show()
```



Observation

- The usage '4'is densly populated showing that the most fittest people used the KP 781 product
- Catagory "4" good fitness people are mostly males
- Catagory "4" good fitness people are mostly Partnered
- Catagory "4" good fitness people are mostly in age group Young Adults (21 - 35)

```
    - In general the Good fitness people are - Male , Married , Use KP 781 and in
  Age 21 - 35
```

Probability and Conditional Probability

In [237]:

```
df.groupby(['MaritalStatus','Product']).Usage.value_counts()
```

Out[237]:

```
MaritalStatus  Product  Usage
Partnered      KP281    3        23
                        2        12
                        4        12
                        5         1
               KP481    3        17
                        2        10
                        4         6
                        5         3
               KP781    4        11
                        5         5
                        6         5
                        7         2
Single         KP281    3        14
                        4        10
                        2         7
                        5         1
               KP481    3        14
                        4         6
                        2         4
               KP781    4         7
                        5         7
                        6         2
                        3         1
Name: Usage, dtype: int64
```

Observation

- Partnered are more likely to buy KP 281 if they have usage rating between 2 to 4
- Single person is more likely to buy KP281 if the usage rating is 3 or 4.

In [332]:

```
df.groupby(['MaritalStatus' , 'Gender','Product'])['Usage'].count()
```

Out[332]:

```
MaritalStatus  Gender  Product
Partnered      Female  KP281      27
                       KP481      15
                       KP781       4
               Male    KP281      21
                       KP481      21
                       KP781      19
Single         Female  KP281      13
                       KP481      14
                       KP781       3
               Male    KP281      19
                       KP481      10
                       KP781      14
Name: Usage, dtype: int64
```

Observations

- KP 481 is bought more by Single females
- KP 281 remains good choice for Partnered females
- A partnered male is equally likely to buy KP 281 and KP 481
- Single male is more likely to buy either KP 281 or KP 481

probability of product Gender

Defining the Probability for Product and Gender :

`

- P(KP): Probability of any given product

```
- (KP 281 , KP 481 , KP 781)
```

- P(G): Probability of particular gender

```
 - ( Male , Female)
```

- P(KP ∩ G): Probability of male / female using any of the given three products

```
 - [P(KP 281 ∩ M) P(KP 281 ∩ F) P(KP 481 ∩ M , P(KP 481 ∩ F), P(KP 781 ∩ M) ,
P(KP 781 ∩ F))]
```

- P(KP|G): Probability of USING A PARTICULAR PRODUCT GIVEN SPECIFIC GENDER

```
- P(KP 781 | M) = P(KP 781 ∩ M) / P(M)
```

In [363]:

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

Out[363]:

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

In [366]:

```
pd.crosstab([df["Product"]],df["Gender"],margins=True,normalize="columns") #p(P/G)
```

Out[366]:

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

In [376]:

```
from IPython.display import display_html
from itertools import chain,cycle

def display_side_by_side(*args,titles=cycle([''])):
    html_str=''
    for df,title in zip(args, chain(titles,cycle(['</br>'])) ):
        html_str+='<th style="text-align:center"><td style="vertical-align:top">'
        html_str+=f'<h2>{title}</h2>'
        html_str+=df.to_html().replace('table','table style="display:inline"')
        html_str+='</td></th>'
    display_html(html_str,raw=True)


df1 = pd.crosstab([df["Product"]],df["Gender"],margins=True)
df2 = pd.crosstab([df["Product"]],df["Gender"],normalize="columns")


display_side_by_side(df1,df2 , titles = ['Product wise count' , 'P(product|Gender)'])
```

Productwise count

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

P(product|Gender)

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

Observations

- the probability of a female buying the KP281 is the highest among all the products
- for male the product probability is close between KP 281 and KP 781

probability of product given marital status

Defining the Probability for Product and Gender :

- P(KP): Probability of any given product

```
- (KP 281 , KP 481 , KP 781)
```

- P(M): Probability of particular Marital Status

```
- ( Partnered , single)
```

- P(KP ∩ M): Probability of male / female using any of the given three products

```
- [P(KP 281 ∩ P) P(KP 281 ∩ S) P(KP 481 ∩ P , P(KP 481 ∩ S), P(KP 781 ∩ P) ,
P(KP 781 ∩ S))]
```

- P(KP|M): Probability of USING A PARTICULAR PRODUCT GIVEN MARITAL STATUS

```
- P(KP 781 | M) = P(KP 781 ∩ P) / P(P)
```

In [367]:

```
pd.crosstab([df["Product"]],df["MaritalStatus"],margins=True)
```

Out[367]:

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

In [370]:

```
pd.crosstab([df["Product"]],df["MaritalStatus"],normalize="columns") #p(p/single) or p(
```

Out[370]:

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

In [378]:

```
from IPython.display import display_html
from itertools import chain,cycle

def display_side_by_side(*args,titles=cycle([''])):
    html_str=''
    for df,title in zip(args, chain(titles,cycle(['</br>'])) ):
        html_str+='<th style="text-align:center"><td style="vertical-align:top">'
        html_str+=f'<h2>{title}</h2>'
        html_str+=df.to_html().replace('table','table style="display:inline"')
        html_str+='</td></th>'
    display_html(html_str,raw=True)


df3 = pd.crosstab([df["Product"]],df["MaritalStatus"],margins=True)
df4 = pd.crosstab([df["Product"]],df["MaritalStatus"],normalize="columns")


display_side_by_side(df3,df4 , titles = ['Product wise count' , 'P(Product|MaritalStatu

```

Product wise count

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

P(Product|MaritalStatus)

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

Observation

- Most partnered as well as the single people prefers KP281. Makes sense as KP281 is the cheapest and most sold product

In [474]:

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

In [473]:

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

```

Marginal Probability

MARGINAL PROBABILITIES of the customers who usages (from twice a week to 7 times a week) buying

any of the three products:

In [459]:

```
marg_prob1 = round(pd.crosstab(index=df['Usage'],columns=df['Product'],margins=True,nor
marg_prob1
# marg_prob1.loc[marg_prob1[2]]
```

Out[459]:

Product	KP281	KP481	KP781	All
Usage				
2	10.56	7.78	0.00	18.33
3	20.56	17.22	0.56	38.33
4	12.22	6.67	10.00	28.89
5	1.11	1.67	6.67	9.44
6	0.00	0.00	3.89	3.89
7	0.00	0.00	1.11	1.11
All	44.44	33.33	22.22	100.00

MARGINAL PROBABILITIES of the customers who are in the age groups(15-64) buying any of the three products:

In [448]:

```
marg_prob2 = round(pd.crosstab(index=df['Education'],columns=df['Product'],margins=True
marg_prob2
```

Out[448]:

Product	KP281	KP481	KP781	All
Education				
12	1.11	0.56	0.00	1.67
13	1.67	1.11	0.00	2.78
14	16.67	12.78	1.11	30.56
15	2.22	0.56	0.00	2.78
16	21.67	17.22	8.33	47.22
18	1.11	1.11	10.56	12.78
20	0.00	0.00	0.56	0.56
21	0.00	0.00	1.67	1.67
All	44.44	33.33	22.22	100.00

MARGINAL PROBABILITIES of the customers who are either married or single and buying any of the three products:

In [449]:

```
marg_prob3 = round(pd.crosstab(index=df['MaritalStatus'],columns=df['Product'],margins=
marg_prob3
```

Out[449]:

Product MaritalStatus	KP281	KP481	KP781	All
Partnered	26.67	20.00	12.78	59.44
Single	17.78	13.33	9.44	40.56
All	44.44	33.33	22.22	100.00

Observation

- High Price/Best featured KP 781 product 'usage' is more among people who are buying it. So the company should focus more on this product
- (MALES who are MARRIED and have higher income) and (who uses the product more than or equal to 4 times in a week(usage)) and (who have education more than or equal to 16 years))

customer profile

KP 281

- This model has same level of popularity in Male customers as well as Female customers as it has same numbers of Male and Female customers.
- Average age of customer who purchases KP 281 is 28.5.
- This model is popular among Bachelors as average years of education of customers for this product is 15.
- Users expect to use this treadmill 3-4 times a week.
- Self rate fitness level of customer is average.
- It is the most popular model (in all genders) because of its falshy price and affordability with 33.3% of sales.
- Customers of this treadmill are on the process if getting into better shape and thus the price is the major attracting factor in this product.

KP 481

- Customers with lower income purchase KP 4 or KP 281 model may be because of lower cost of the Treadmill.
- Average age of customer who purchases KP 481 is 29.
- Customers expecting KP 481 model to use less frequently but to run more miles per week on this.

KP 781

- This is the least sold product(22.2% sales) Treadmill, may be because of it heafty price range making it Company's Premium product.
- Average age of customer who purchases TM798 is 29.
- Treadmill may have some advanced features as people with high income are ready to spend money to buy this model
- Customers expected usage on this model is 4-5 day a week with moderate Miles to run.
- Male customers who are more serious about fitness or Professionals buy this mode (self fitness rating 3-5).
- Customers of this treadmill are on high fitness and elite product with the High fitness major attraction of the product.

Recommendations

- Recommend the KP 781 to users who have high usage rating
- Recommending the KP 781 to the People with higher income as it has more features
- KP 781 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.
- Recommend KP 281 to the Single people with usage rating less than 4 to 5
- KP 481 can be recommended to the Single females
- Aerofit should conduct market research to determine if it can attract customers with income under USD 1750 to expand its customer base.

In [472]:

```
# from pandas_profiling import ProfileReport
# profile = ProfileReport(df)
# profile
```

In [ ]:

```

```