🏃‍♂️🏃‍♀️ Aerofit Case Study – Customer Segmentation & Marketing Insights 🏃‍♂️🏃‍♀️

Company Background

Aerofit is a leading fitness solutions provider in India with strong roots in the sports industry. The journey began in 1928 with M/s. Sachdev Sports Co., founded in Hyderabad. Over the decades, the company expanded from sports goods to importing fitness equipment under the Aerofit brand. Today, under Nityasach Fitness Pvt. Ltd., the company bridges global fitness technology with the Indian market by offering affordable, high-quality equipment.

Product Portfolio

Aerofit's portfolio includes:

- Treadmills (entry-level, mid-level, and advanced models)
- Exercise bikes
- Gym equipment
- Fitness accessories

Within treadmills, the key models are:

KP281 → Entry-level, USD 1,500

KP481 → Mid-level, USD 1,750

KP781 → Advanced features, USD 2,500

Business Objective

The primary goal of this study is to understand customer characteristics and preferences for different treadmill models. This involves:

Conducting descriptive analytics to create detailed customer profiles.

Performing exploratory data analysis (EDA) to identify key patterns and trends in purchasing behavior.

Using two-way contingency tables to calculate marginal and conditional probabilities.

Applying hypothesis testing to validate observed differences across customer groups.

These analyses will help Aerofit enhance customer targeting, product recommendations, and marketing strategies.

Data Overview

Data was collected over a three-month period from Aerofit store purchases.

Information is stored in a single CSV file containing customer demographics and treadmill purchase details.

The dataset enables segmentation and comparison of customer behavior across the three treadmill models.

Features of the dataset:

• Product: 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.

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

• Income: Annual income (in $)

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

Exploratory Data Analysis

In [2]:
```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
# loading the dataset
df_aerofit = pd.read_csv('/content/aerofit.csv')
```

In [5]:
```
df= df_aerofit.copy()
df
```

Out[5]:

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

180 rows × 9 columns

In [ ]:
```
df.shape
```

Out[ ]:
```
(180, 9)
```

In [ ]:
```
df.head(5)
```

Out[ ]:

	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 [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Product        180 non-null    object
 1   Age            180 non-null    int64 
 2   Gender         180 non-null    object
 3   Education      180 non-null    int64 
 4   MaritalStatus  180 non-null    object
 5   Usage          180 non-null    int64 
 6   Fitness        180 non-null    int64 
 7   Income         180 non-null    int64 
 8   Miles          180 non-null    int64 
dtypes: int64(6), object(3)
memory usage: 12.8+ KB
```

Insights

- From the above analysis, it is clear that, data has total of 9 features with mixed alpha numeric data. Also we can see that there is no missing data in the columns.

- The data type of all the columns are matching with the data present in them. But we will change the datatype of Usage and Fitness into category.

Changing the Datatype of Columns

In [ ]:
```
for col in df.columns:
 if df[col].dtype == 'object':
  df[col] = df[col].astype('category')
```

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 180 entries, 0 to 179
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype   
---  ------         --------------  -----   
 0   Product        180 non-null    category
 1   Age            180 non-null    int64   
 2   Gender         180 non-null    category
 3   Education      180 non-null    int64   
 4   MaritalStatus  180 non-null    category
 5   Usage          180 non-null    int64   
 6   Fitness        180 non-null    int64   
 7   Income         180 non-null    int64   
 8   Miles          180 non-null    int64   
dtypes: category(3), int64(6)
memory usage: 9.5 KB
```

Statistical Summary

In [ ]:
```
df.describe().T
```

Out[ ]:

	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

Insights

- **Age**: Customers fall between 18 and 50 years, with an average age of 29, showing a younger customer base.
- **Education**: Education ranges from 12 to 21 years, averaging 16 years, which aligns with customers having graduate-level education.
- **Usage**: Customers use treadmills 2 to 7 times per week, with an average of 3 times, reflecting moderate but consistent usage.
- **Fitness**: On a 5-point scale, the average fitness rating is 3, indicating a moderate fitness level among customers.
- **Income**: Annual income lies between USD 30,000 and 100,000, with an average of USD 54,000, positioning customers in the middle-to-upper income segment.
- **Miles**: Weekly running goals vary from 21 to 360 miles, with an average of 103 miles, highlighting a serious commitment to fitness goals.

In [ ]:
```
# statisctical summary of category type columns
df.describe(include = 'category').T
```

Out[ ]:

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

Insights

Product: Over the past three months, the **KP281** treadmill recorded the highest sales performance compared to the other two models.

Gender: Around **58%** of the buyers were **Male**, while **42%** were **Female**, showing a slightly male-dominated customer base.

Marital Status: Approximately **60%** of the buyers were **Married**, whereas **40%** were **Single**, indicating a stronger presence of family-oriented customers.

Detecting Duplicates

In [ ]:
```
df.duplicated().value_counts()
```

Out[ ]:

	count
False	180

dtype: int64

Insights

- There are no duplicate entries in the dataset

Checking Missing Values and Unique Values

In [ ]:
```
# checking unique values
for i in df.columns:
 print('Unique Values in',i,'column are :-')
 print(df[i].unique())
 print('-'*70)
```

```
Unique Values in Product column are :-
['KP281', 'KP481', 'KP781']
Categories (3, object): ['KP281', 'KP481', 'KP781']
----------------------------------------------------------------------
Unique Values in Age column are :-
[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]
----------------------------------------------------------------------
Unique Values in Gender column are :-
['Male', 'Female']
Categories (2, object): ['Female', 'Male']
----------------------------------------------------------------------
Unique Values in Education column are :-
[14 15 12 13 16 18 20 21]
----------------------------------------------------------------------
Unique Values in MaritalStatus column are :-
['Single', 'Partnered']
Categories (2, object): ['Partnered', 'Single']
----------------------------------------------------------------------
Unique Values in Usage column are :-
[3 2 4 5 6 7]
----------------------------------------------------------------------
Unique Values in Fitness column are :-
[4 3 2 1 5]
----------------------------------------------------------------------
Unique Values in Income column are :-
[ 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]
----------------------------------------------------------------------
Unique Values in Miles column are :-
[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]
----------------------------------------------------------------------
```

In [ ]:
```
# checking missing values
df.isna().any()
```

Out[ ]:

	0
Product	False
Age	False
Gender	False
Education	False
MaritalStatus	False
Usage	False
Fitness	False
Income	False
Miles	False

dtype: bool

Insights

- The dataset contains 180 rows and 9 columns, with most features being numerical data.
- There are no missing values present.
- There are 3 unique treadmill models – KP281, KP481, and KP781. Among these, KP281 is the most frequently purchased product.
- The dataset does not contain any abnormal values.

Feature Engineering-(Discretization)

- Feature binning was performed by converting continuous variables (Age, Education, Income, and Miles) into categorical classes to simplify analysis and improve visualization.

In [15]:
```
# Categorization of age

# 0-21 -> Teenage
# 22-35 -> Adult
# 36-45 -> Middle Age
# 46-60 -> Elderly

df['age_category'] = df.Age
df['age_category'] = pd.cut(df.age_category,bins=[0,20,35,45,60],labels=['Teen','Adult','Middle Aged','Elderly'])
df.sample(3)
```

Out[15]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	age_category
4	KP281	20	Male	13	Partnered	4	2	35247	47	Teen
20	KP281	23	Male	14	Single	4	3	38658	113	Adult
100	KP481	25	Female	14	Partnered	5	3	47754	106	Adult

In [16]:
```
df.age_category.value_counts()
```

Out[16]:

age_category	count
Adult	142
Middle Aged	22
Teen	10
Elderly	6

dtype: int64

In [17]:
```
# Categorization of miles

# Light Activity - Upto 50 miles
# Moderate Activity - 51 to 100 miles
# Active Lifestyle - 101 to 200 miles
# Fitness Enthusiast - Above 200 miles


df['fitness_level'] = df.Miles
df['fitness_level'] = pd.cut(df.fitness_level,bins=[0,50,100,200,float('inf')],labels = ['Light Activity', 'Moderate

df.sample(3)
```

Out[17]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	age_category	fitness_level
102	KP481	25	Female	14	Single	2	3	43206	64	Adult	Moderate Activity
130	KP481	35	Female	16	Single	3	2	50028	64	Adult	Moderate Activity
160	KP781	27	Male	18	Single	4	3	88396	100	Adult	Moderate Activity

In [18]:
```
# Categorization of education

# Primary Education: upto 12
# Secondary Education: 13 to 15
# Higher Education: 16 and above

bin_range1 = [0,12,15,float('inf')]
bin_labels1 = ['Primary Education', 'Secondary Education', 'Higher Education']
df['education_level'] = pd.cut(df['Education'],bins = bin_range1,labels = bin_labels1)

df.sample(3)
```

Out[18]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	age_category	fitness_level	education_lev
172	KP781	34	Male	16	Single	5	5	92131	150	Adult	Active Lifestyle	High Educati
150	KP781	25	Male	16	Partnered	4	5	49801	120	Adult	Active Lifestyle	High Educati
176	KP781	42	Male	18	Single	5	4	89641	200	Middle Aged	Active Lifestyle	High Educati

In [19]:
```
# Categorization of Income

# Low Income - Upto 40,000
# Moderate Income - 40,000 to 60,000
# High Income - 60,000 to 80,000
# Very High Income - Above 80,000

bin_range2 = [0,40000,60000,80000,float('inf')]
bin_labels2 = ['Low Income','Moderate Income','High Income','V.High Income']
df['income_category'] = pd.cut(df['Income'],bins = bin_range2,labels = bin_labels2)

df.sample(3)
```

Out[19]:

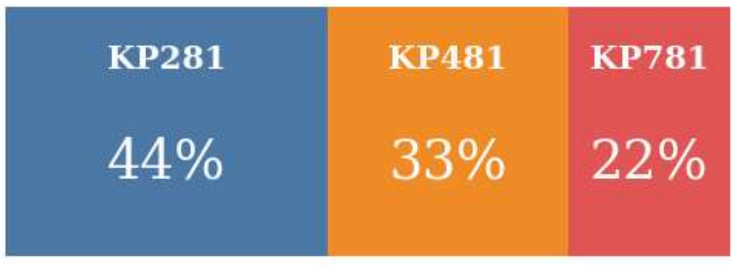
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	age_category	fitness_level	education_lev
24	KP281	24	Male	14	Single	2	3	45480	113	Adult	Active Lifestyle	Seconda Educati
48	KP281	28	Male	14	Single	4	3	54576	113	Adult	Active Lifestyle	Seconda Educati
138	KP481	45	Male	16	Partnered	2	2	54576	42	Middle Aged	Light Activity	High Educati

Univariate & Bivariate Analysis

In [21]:
```
#color_palette
cp  = ['#4E79A7', '#F28E2B', '#E15759', '#76B7B2', '#59A14F']
cp1 = ['#EDC948', '#B07AA1', '#FF9DA7', '#9C755F']
cp2 = ['#76B7B2', '#E15759', '#F28E2B']
cp3 = ['#59A14F', '#F1CE63', '#B6992D']
cp4 = ['#4E79A7', '#F28E2B', '#E15759', '#76B7B2','#EDC948', '#B07AA1', '#59A14F']

plt.figure(figsize=(14,2.3))
plt.subplot(1,2,1)
product_count = df['Product'].value_counts()
product_count['percent'] = ((product_count.values/df.shape[0])* 100).round()
plt.style.use('default')
plt.style.use('seaborn-v0_8-bright')
plt.barh(product_count.index[0],product_count.loc['percent'][0],color = cp[0])
plt.barh(product_count.index[0],product_count.loc['percent'][1],left = product_count.loc['percent'][0],color = cp[1])
plt.barh(product_count.index[0],product_count.loc['percent'][2],
 left = product_count.loc['percent'][0] + product_count.loc['percent'][1], color = cp[2])
plt.xlim=(0,100)
plt.axis('off')
product_count['info_percent'] =[product_count['percent'][0]/2,product_count['percent'][0] + product_count['percent'][
 product_count['percent'][0] + product_count['percent'][1] + product_count['percent'][2]/2]
for i in range(3):
 plt.text(product_count['info_percent'][i],0.23,product_count.index[i],
 va = 'center', ha='center',fontsize=15, fontweight='bold', fontfamily='serif',color='white')
 plt.text(product_count['info_percent'][i],-0.1,f"{product_count['percent'][i]:.0f}%",
 va = 'center', ha='center',fontsize=25, fontweight='light', fontfamily='serif',color='white')
```

```
plt.subplot(1,2,2)
product_portfolio = [['KP281','$1500','$120k'],['KP481','$1750','$105k'],['KP781','$2500','$100k']]
color_2d = [[cp[0],'#FFFFFF','#FFFFFF'],[cp[1],'#FFFFFF','#FFFFFF'],[cp[2],'#FFFFFF','#FFFFFF']]
table = plt.table(cellText = product_portfolio, cellColours=color_2d, cellLoc='center',colLabels =['Product','Price',
 colLoc = 'center',bbox =[0, 0, 1, 1])
plt.axis('off')
plt.show()
```



Product	Price	Sales
KP281	$1500	$120k
KP481	$1750	$105k
KP781	$2500	$100k

Insights

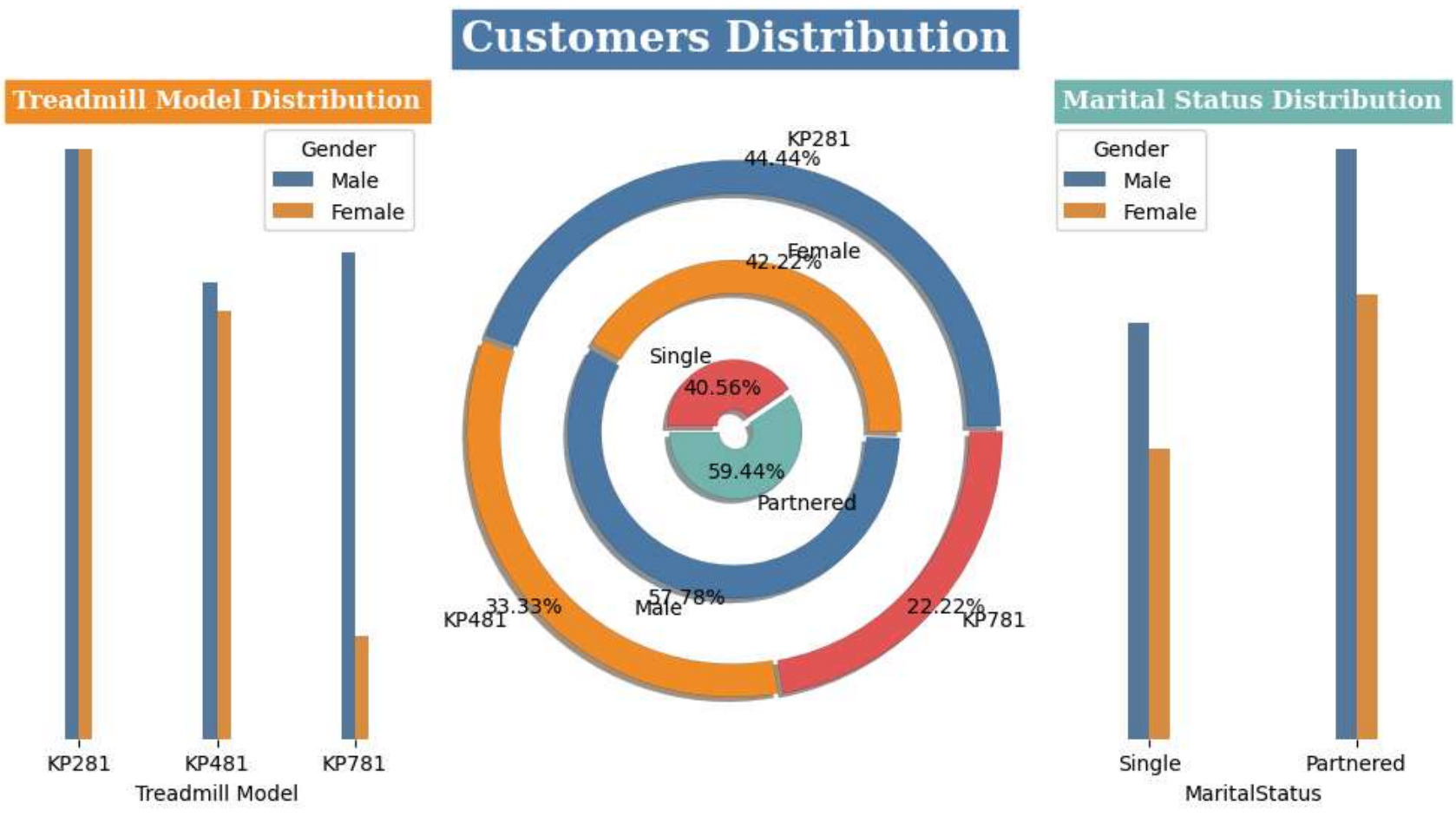
44.44% of customers bought **KP281** product type

33.33% of customers bought **KP481** product type

22.22% of customers bought **KP781** product type

Customer Distribution

In [22]:
```
plt.figure(figsize=(10,5.5))
plt.suptitle('Customers Distribution',fontfamily='serif',fontweight='bold',fontsize=20,
 backgroundcolor=cp[0],color='w')
plt.style.use('default')
plt.style.use('seaborn-v0_8-bright')
plt.subplot(1,3,1)
plt.title('Treadmill Model Distribution',fontfamily='serif',fontweight='bold',fontsize=12,
 backgroundcolor=cp[1],color='w')
a = sns.countplot(data=df,x='Product',hue='Gender',palette=cp,width=0.2)
plt.xlabel('Treadmill Model')
plt.ylabel('')
plt.yticks([])
plt.subplot(1,3,3)
a = sns.countplot(data=df,x='MaritalStatus',hue='Gender',palette=cp,width=0.2)
plt.title('Marital Status Distribution',fontfamily='serif',fontweight='bold',fontsize=12,
 backgroundcolor=cp[3],color='w')
plt.yticks([])
plt.ylabel('')
sns.despine(left=True,bottom=True,trim=True)
plt.subplot(1,3,2)
plt.pie(df.Product.value_counts(), labels=df.Product.value_counts().index,
 counterclock=True , explode=(0.02,0.02,0.02) , autopct='%.2f%%', pctdistance=1.025,
 colors=cp , textprops={'color':'k','fontsize':10} , shadow=True, radius=1.6,
 wedgeprops=dict(edgecolor='k',linewidth=0.1,width=0.2))
plt.pie(df.Gender.value_counts(), labels=df.Gender.value_counts().index,
 startangle=150 , explode=(0.02,0.02) , autopct='%.2f%%',pctdistance=1.035,
 colors=cp , textprops={'color':'k','fontsize':10} , shadow=True,radius=1,
 wedgeprops=dict(edgecolor='k',linewidth=0.1,antialiased=True,width=0.2))
plt.pie(df.MaritalStatus.value_counts(), labels=df.MaritalStatus.value_counts().index,
 startangle=180 , explode=(0.02,0.02) , autopct='%.2f%%',
 colors=cp2 , textprops={'color':'k','fontsize':10} , shadow=True, radius=0.4,
 wedgeprops=dict(edgecolor='k',linewidth=0.1,antialiased=True,width=0.3))
plt.tight_layout()
plt.show()
```



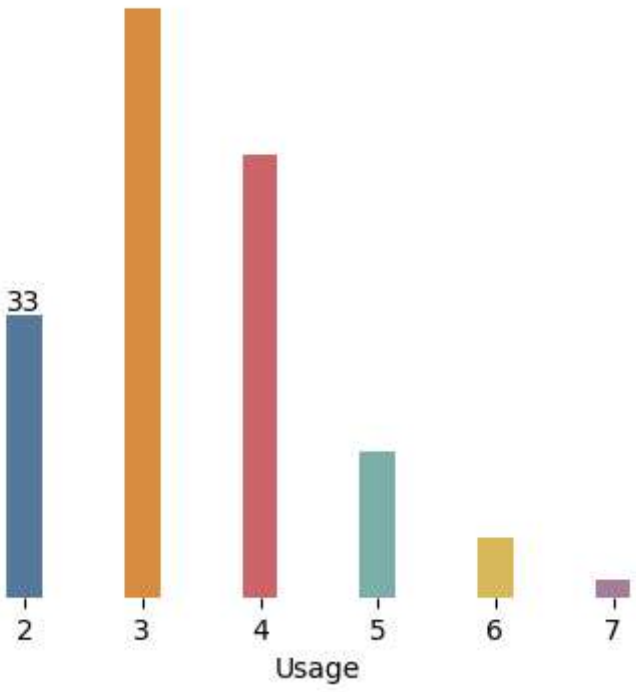
Customer Usage, Fitness

In [23]:

```
plt.figure(figsize=(10,4))
plt.suptitle('Customers usage',fontfamily='serif',fontweight='bold',fontsize=20,
 backgroundcolor=cp2[1],color='w')
plt.style.use('default')
plt.style.use('seaborn-v0_8-bright')
plt.subplot(1,2,1)
usage_info = [['3','38%'],['4','29%'],['2','19%'],['5','9%'],['6','4%'],['7','1%']]
color_2d = [[cp4[0],'#FFFFFF'],[cp4[1],'#FFFFFF'],[cp4[2],'#FFFFFF'],[cp4[3],'#FFFFFF'],[cp4[5],'#FFFFFF'],
 [cp1[0],'#FFFFFF']]
plt.table(cellText = usage_info, cellColours=color_2d, cellLoc='center',colLabels =['Usage Per Week','Percent'],
 colLoc = 'center',bbox =[0, 0, 1, 1])
plt.axis('off')
plt.subplot(1,2,2)
u = df['Usage'].value_counts()
a = sns.barplot(x=u.index,y = u.values,palette=cp4,width=0.3)
a.bar_label(a.containers[0],label_type='edge')
sns.despine(left=True,bottom=True)
#plt.xticks([])
plt.yticks([])
plt.ylabel('')
plt.figure(figsize=(10,4))
plt.suptitle('Customers Fitness',fontfamily='serif',fontweight='bold',fontsize=20,
 backgroundcolor=cp2[2],color='w')
plt.style.use('default')
plt.style.use('seaborn-v0_8-bright')
plt.subplot(1,2,1)
fitness_info = [['3','54%'],['5','17%'],['2','15%'],['4','13%'],['1','1%']]
color_2d = [[cp[0],'#FFFFFF'],[cp[1],'#FFFFFF'],[cp[2],'#FFFFFF'],[cp[3],'#FFFFFF'],[cp[4],'#FFFFFF']]
plt.table(cellText = fitness_info, cellColours=color_2d, cellLoc='center',colLabels =['Fitness','Percent'],
 colLoc = 'center',bbox =[0, 0, 1, 1])
plt.axis('off')
plt.subplot(1,2,2)
f = df['Fitness'].value_counts()
b = sns.barplot(x=f.index,y = f.values,palette=cp4,width=0.3)
b.bar_label(b.containers[0],label_type='edge')
plt.title('Customer count based on Fitness')
sns.despine(left=True,bottom=True)
plt.yticks([])
#plt.xticks([])
plt.ylabel('')
plt.show()
```

Customers usage

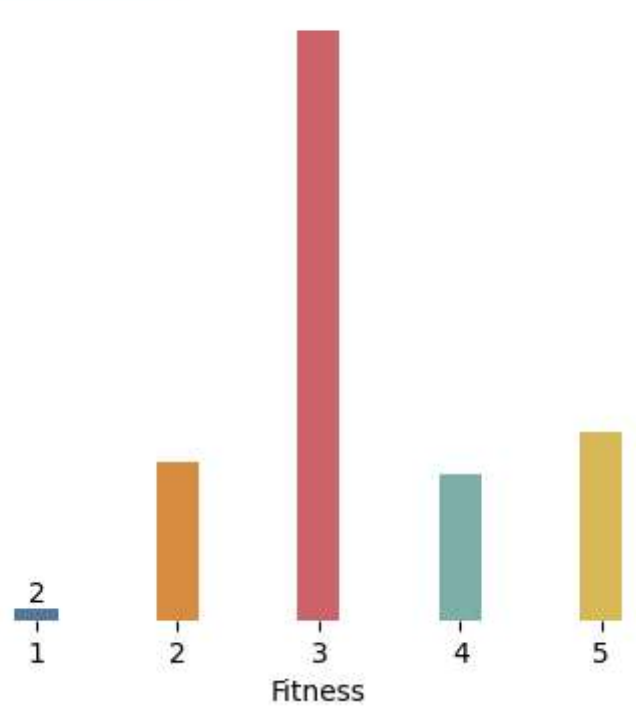
Usage Per Week	Percent
3	38%
4	29%
2	19%
5	9%
6	4%
7	1%



Customers Fitness

er count based on Fitness

Fitness	Percent
3	54%
5	17%
2	15%
4	13%
1	1%

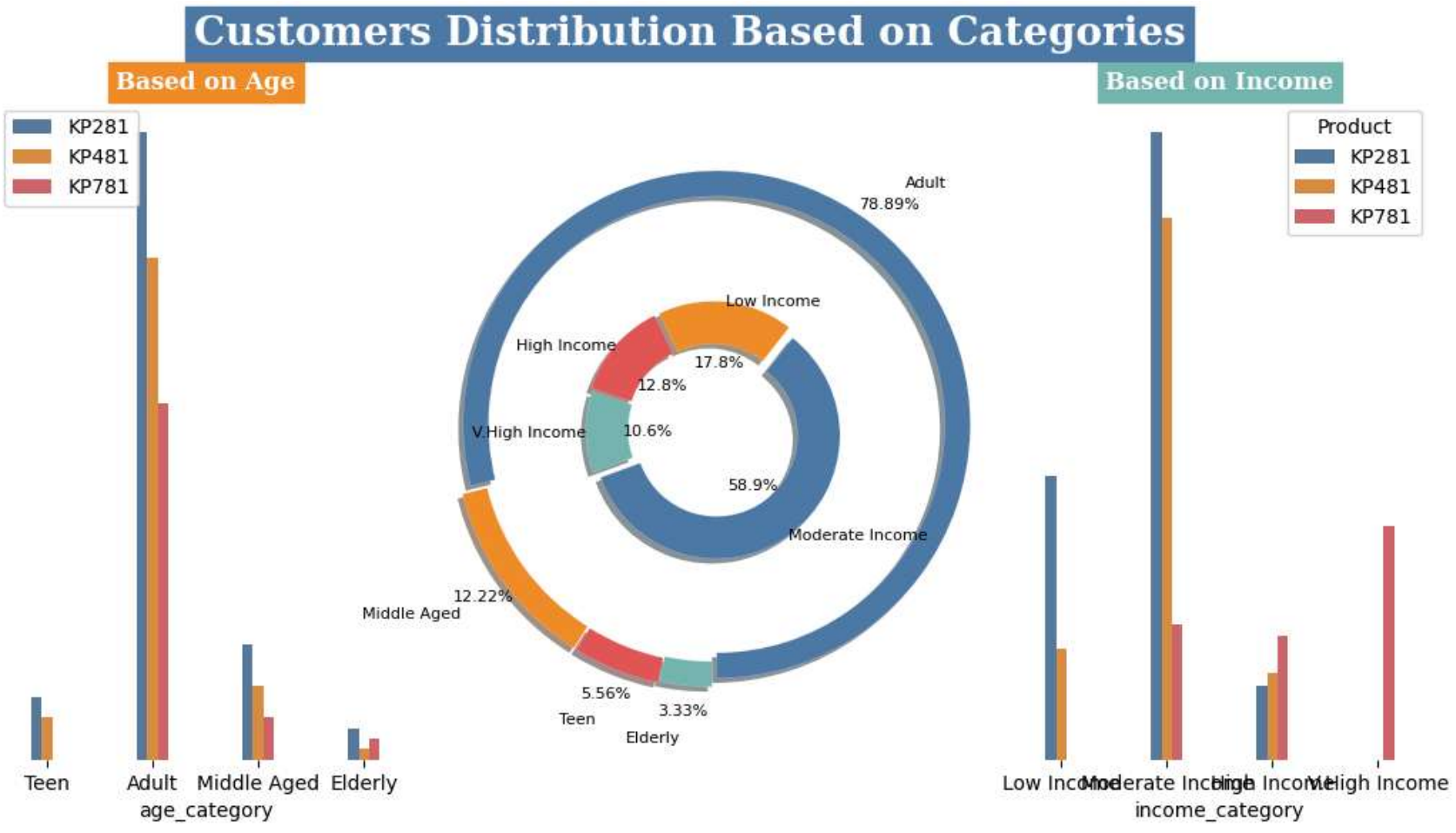


Insights

- Approximately 85% of customers plan to use the treadmill 2 to 4 times per week, while only 15% use it 5 or more times. Additionally, 54% of customers have rated their fitness level as 3 on a 5-point scale.
- Furthermore, a substantial 84% of the total customers have rated themselves at 3 or higher, indicating commendable fitness levels.

Customers Distribution Based on Categories

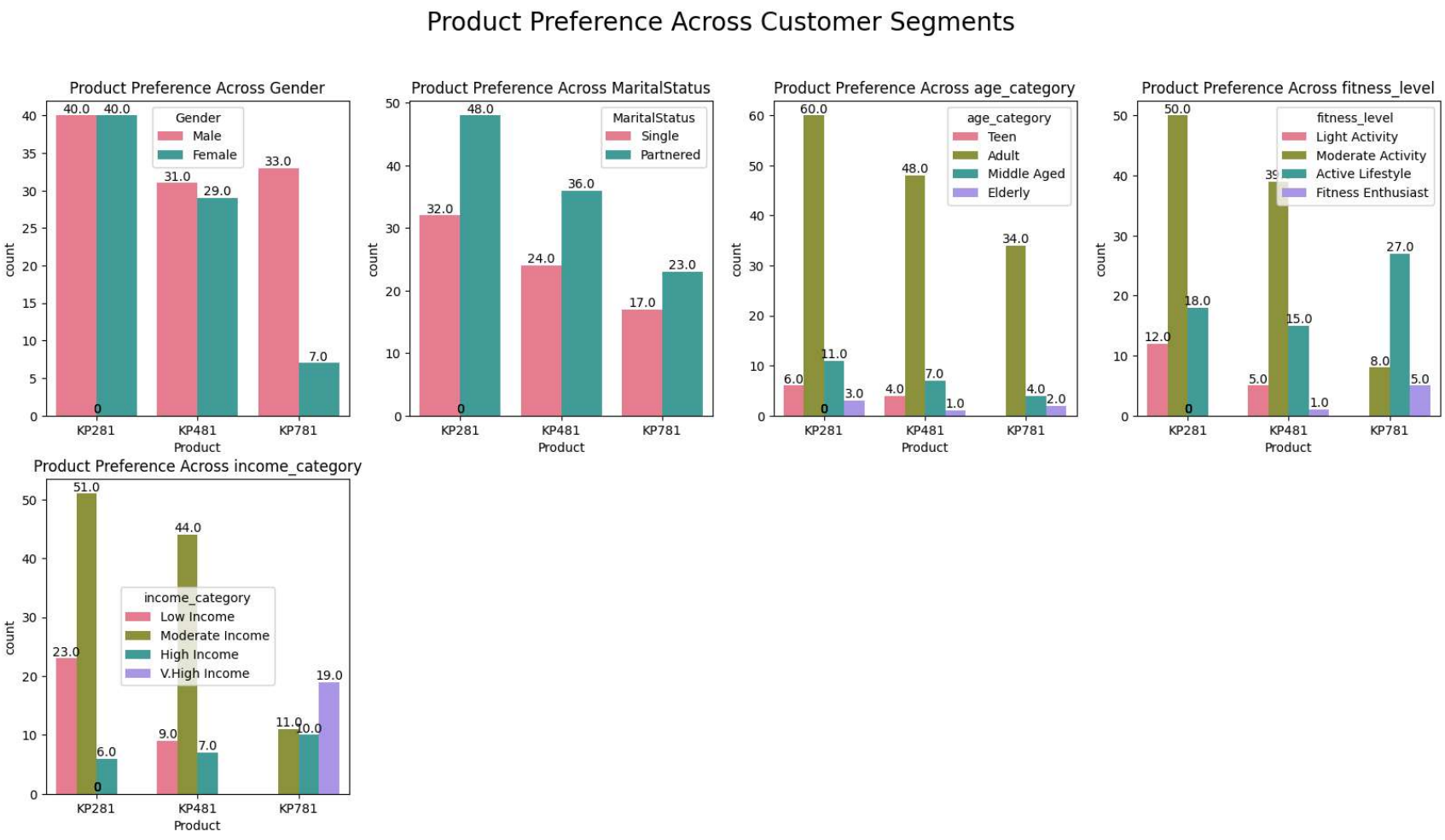
In [24]:
```
plt.figure(figsize=(13,6))
plt.suptitle('Customers Distribution Based on Categories',fontfamily='serif',fontweight='bold',fontsize=20,
 backgroundcolor=cp4[0],color='w')
plt.style.use('default')
plt.style.use('seaborn-v0_8-bright')
plt.subplot(1,3,1)
sns.countplot(df,x='age_category',hue='Product',palette=cp,width=0.3)
plt.title('Based on Age',fontfamily='serif',fontweight='bold',fontsize=12,backgroundcolor=cp4[1],color='w')
plt.yticks([])
plt.legend(loc='upper left')
plt.ylabel('')
plt.subplot(1,3,2)
plt.pie(df.age_category.value_counts(), labels=df.age_category.value_counts().index,
 explode=(0.04,0.03,0.03,0.02) , autopct='%.2f%%', pctdistance=1.1,startangle=270,
 colors=cp , textprops={'color':'k','fontsize':8} , shadow=True, labeldistance=1.21,
 wedgeprops=dict(edgecolor='w',linewidth=0.1,width=0.15), radius=1.5)
plt.pie(df.income_category.value_counts(), labels=df.income_category.value_counts().index, counterclock=True,
 startangle=200 , explode=(0.04,0.03,0.03,0.02) , autopct='%.1f%%', pctdistance=0.5,
 colors=cp , textprops={'color':'k','fontsize':8} , shadow=True, labeldistance=1,
 wedgeprops=dict(edgecolor='w',linewidth=0.1,width=0.25), radius=0.73)
plt.subplot(1,3,3)
sns.countplot(df,x='income_category',hue='Product',palette=cp,width=0.3)
plt.title('Based on Income',fontfamily='serif',fontweight='bold',fontsize=12,backgroundcolor=cp4[3],color='w')
sns.despine(left=True,bottom=True,trim=True)
plt.yticks([])
plt.ylabel('')
plt.show()
```



Product Preferences Across Customer Segments

In [ ]:
```
# Creating a function to give values in chart:
def text_format(fig):
  for bar in ax.patches:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, yval, str(yval), ha = "center", va = "bottom")
```

In [ ]:
```
cat_var = ["Gender", "MaritalStatus", "age_category", "fitness_level","income_category"]
plt.figure(figsize=(20,10))
for i,j in enumerate(cat_var):
 plt.subplot(2,4,i+1)
 sns.countplot(x = "Product", hue = j , data = df, palette="husl")
 ax = plt.gca()
 text_format(fig)
 plt.title(f"Product Preference Across {j}")
 plt.suptitle("Product Preference Across Customer Segments", fontsize = 20)
 plt.tight_layout
```



Insights

- Gender: The KP781 model is more popular among male customers compared to females.
- Marital Status: Married customers show a stronger preference for the KP281.
- Age Category: Adults primarily prefer the KP281, while middle-aged customers lean towards the KP481.
- Fitness Levels: Highly fitness-oriented customers are better matched with the KP781, while those with moderate fitness goals tend to prefer the KP281 or KP481.

- Income Categories: Medium-income customers mainly prefer the KP281, followed by the KP481. High-income customers typically opt for the KP781, while low-income customers lean towards the KP281.
- Sales Distribution: Sales are divided across models as follows — 37% from KP281, 32.3% from KP481, and 30.7% from KP781.

Product Purchase Probability Analysis

From the Given Data

- **Marginal Probabilities:**
  - **44.44**% of customers purchased **KP281**
  - **33.33**% of customers purchased **KP481**
  - **22.22**% of customers purchased **KP781**
- **Analysis of Conditional Probabilities** -The next step involves analyzing how product preferences vary when conditioned on different customer attributes.

Probability: Male + Single

In [ ]:
```
pms = df[(df['Gender']=='Male') & (df['MaritalStatus']=='Single')]
p1=round(len(pms[pms['Product']=='KP281'])/(len(pms))*100,2)
p2=round(len(pms[pms['Product']=='KP481'])/(len(pms))*100,2)
p3=round(len(pms[pms['Product']=='KP781'])/(len(pms))*100,2)
print('Purchase Probability among Male, Single Customers')
print(f'- KP281 is {p1}%\n- KP481 is {p2}%\n- KP781 is {p3}%')
```

```
Purchase Probability among Male, Single Customers
- KP281 is 44.19%
- KP481 is 23.26%
- KP781 is 32.56%
```

Probability: Male + Partnered

In [ ]:
```
pmp = df[(df['Gender']=='Male') & (df['MaritalStatus']=='Partnered')]
p1=round(len(pmp[pmp['Product']=='KP281'])/(len(pmp))*100,2)
p2=round(len(pmp[pmp['Product']=='KP481'])/(len(pmp))*100,2)
p3=round(len(pmp[pmp['Product']=='KP781'])/(len(pmp))*100,2)
print('Purchase Probability among Male, Partnered Customers')
print(f'- KP281 is {p1}%\n- KP481 is {p2}%\n- KP781 is {p3}%')
```

```
Purchase Probability among Male, Partnered Customers
- KP281 is 34.43%
- KP481 is 34.43%
- KP781 is 31.15%
```

Probability: Female + Single

In [ ]:
```
pmp = df[(df['Gender']=='Female') & (df['MaritalStatus']=='Single')]
p1=round(len(pmp[pmp['Product']=='KP281'])/(len(pmp))*100,2)
p2=round(len(pmp[pmp['Product']=='KP481'])/(len(pmp))*100,2)
p3=round(len(pmp[pmp['Product']=='KP781'])/(len(pmp))*100,2)
print('Purchase Probability among Female, Single Customers')
print(f'- KP281 is {p1}%\n- KP481 is {p2}%\n- KP781 is {p3}%')
```

```
Purchase Probability among Female, Single Customers
- KP281 is 43.33%
- KP481 is 46.67%
- KP781 is 10.0%
```

Probability: Female + Partnered

In [ ]:
```
pmp = df[(df['Gender']=='Female') & (df['MaritalStatus']=='Partnered')]
p1=round(len(pmp[pmp['Product']=='KP281'])/(len(pmp))*100,2)
p2=round(len(pmp[pmp['Product']=='KP481'])/(len(pmp))*100,2)
p3=round(len(pmp[pmp['Product']=='KP781'])/(len(pmp))*100,2)
print('Purchase Probability among Female, Partnered Customers')
print(f'- KP281 is {p1}%\n- KP481 is {p2}%\n- KP781 is {p3}%')
```

```
Purchase Probability among Female, Partnered Customers
- KP281 is 58.7%
- KP481 is 32.61%
- KP781 is 8.7%
```

Insights

- **KP281**: Purchase probability rises from 44.44% to 58.7% when the customer is Female and Partnered.
- **KP481:** Purchase probability rises from 33.33% to 46.67% when the customer is Female and Single.
- **KP781:** Purchase probability rises from 22.22% to 32.56% when the customer is Male and Single.

- **KP481 & KP781:** Combined purchase probability increases from 33.33% & 22.22% to 34.43% when the customer is Male and Single.
- **KP781:** Purchase probability decreases from 22.22% to 8.7% when the customer is Female and Partnered.

Probability by Usage

In [ ]:
```
pby = round(pd.crosstab(df.Product,df.Usage,margins=True,normalize=True)*100,2)
pby
```

Out[ ]:

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

Insights

- **Usage = 3 times/week**

  Overall purchase probability: **38%**

  Conditional probabilities: **KP281 – 21% | KP481 – 17% | KP781 – 1%**

- **Usage = 4 times/week**

  Overall purchase probability: **29%**

  Conditional probabilities: **KP281 – 12% | KP481 – 7% | KP781 – 10%**

- **Usage = 2 times/week**

  Overall purchase probability: **18%**

  Conditional probabilities: **KP281 – 11% | KP481 – 8% | KP781 – 0%**

Probability by Age Group

In [ ]:
```
pd.crosstab(df.Product,df.age_category,margins=True).T
```

Out[ ]:

Product	KP281	KP481	KP781	All
age_category				
Teen	6	4	0	10
Adult	60	48	34	142
Middle Aged	11	7	4	22
Elderly	3	1	2	6
All	80	60	40	180

In [ ]:
```
round(pd.crosstab(df.Product,df.age_category,normalize='columns')*100,2).T
```

Out[ ]:

Product	KP281	KP481	KP781
age_category			
Teen	60.00	40.00	0.00
Adult	42.25	33.80	23.94
Middle Aged	50.00	31.82	18.18
Elderly	50.00	16.67	33.33

In [ ]:
```
round(pd.crosstab(df.Product,df.age_category,margins=True,normalize=True)*100,2).T
```

Out[ ]:

age_category	Product	KP281	KP481	KP781	All
Teen		3.33	2.22	0.00	5.56
Adult		33.33	26.67	18.89	78.89
Middle Aged		6.11	3.89	2.22	12.22
Elderly		1.67	0.56	1.11	3.33
All		44.44	33.33	22.22	100.00

Insights

- **Teens (0–20 years)**

  Overall purchase probability: **6%**

  Conditional probabilities: **KP281 – 3% | KP481 – 2% | KP781 – 0%**

- **Adults (21–35 years)**

  Overall purchase probability: **79%**

  Conditional probabilities: **KP281 – 33% | KP481 – 26% | KP781 – 19%**

- **Middle-aged (36–45 years)**

  Overall purchase probability: **12%**

- **Elders (46+ years)**

  Overall purchase probability: **3%**

Probability by Income Group

In [ ]:
```
pd.crosstab(index=df.Product , columns=df.income_category).T
```

Out[ ]:

income_category	Product	KP281	KP481	KP781
Low Income		23	9	0
Moderate Income		51	44	11
High Income		6	7	10
V.High Income		0	0	19

In [ ]:
```
np.round(pd.crosstab(index=df.Product , columns=df.income_category , normalize='columns')*100,2).T
```

Out[ ]:

income_category	Product	KP281	KP481	KP781
Low Income		71.88	28.12	0.00
Moderate Income		48.11	41.51	10.38
High Income		26.09	30.43	43.48
V.High Income		0.00	0.00	100.00

In [ ]:
```
np.round(pd.crosstab(index=df.Product , columns=df.income_category , margins=True , normalize=True)*100,2).T
```

Out[ ]:

income_category	Product	KP281	KP481	KP781	All
Low Income		12.78	5.00	0.00	17.78
Moderate Income		28.33	24.44	6.11	58.89
High Income		3.33	3.89	5.56	12.78
V.High Income		0.00	0.00	10.56	10.56
All		44.44	33.33	22.22	100.00

Insights

- **Low Income (≤ 40,000)**

Overall purchase probability: **18%**

Conditional probabilities: **KP281 – 13% | KP481 – 5% | KP781 – 0%**

- **Moderate Income (40,001–60,000)**

  Overall purchase probability: **59%**

  Conditional probabilities: **KP281 – 28% | KP481 – 24% | KP781 – 6%**

- **High Income (60,001–80,000)**

  Overall purchase probability: **13%**

  Conditional probabilities: **KP281 – 3% | KP481 – 4% | KP781 – 6%**

- **Very High Income (> 80,000)**

  Overall purchase probability: **11%**

  Conditional probabilities: **KP281 – 0% | KP481 – 0% | KP781 – 11%**

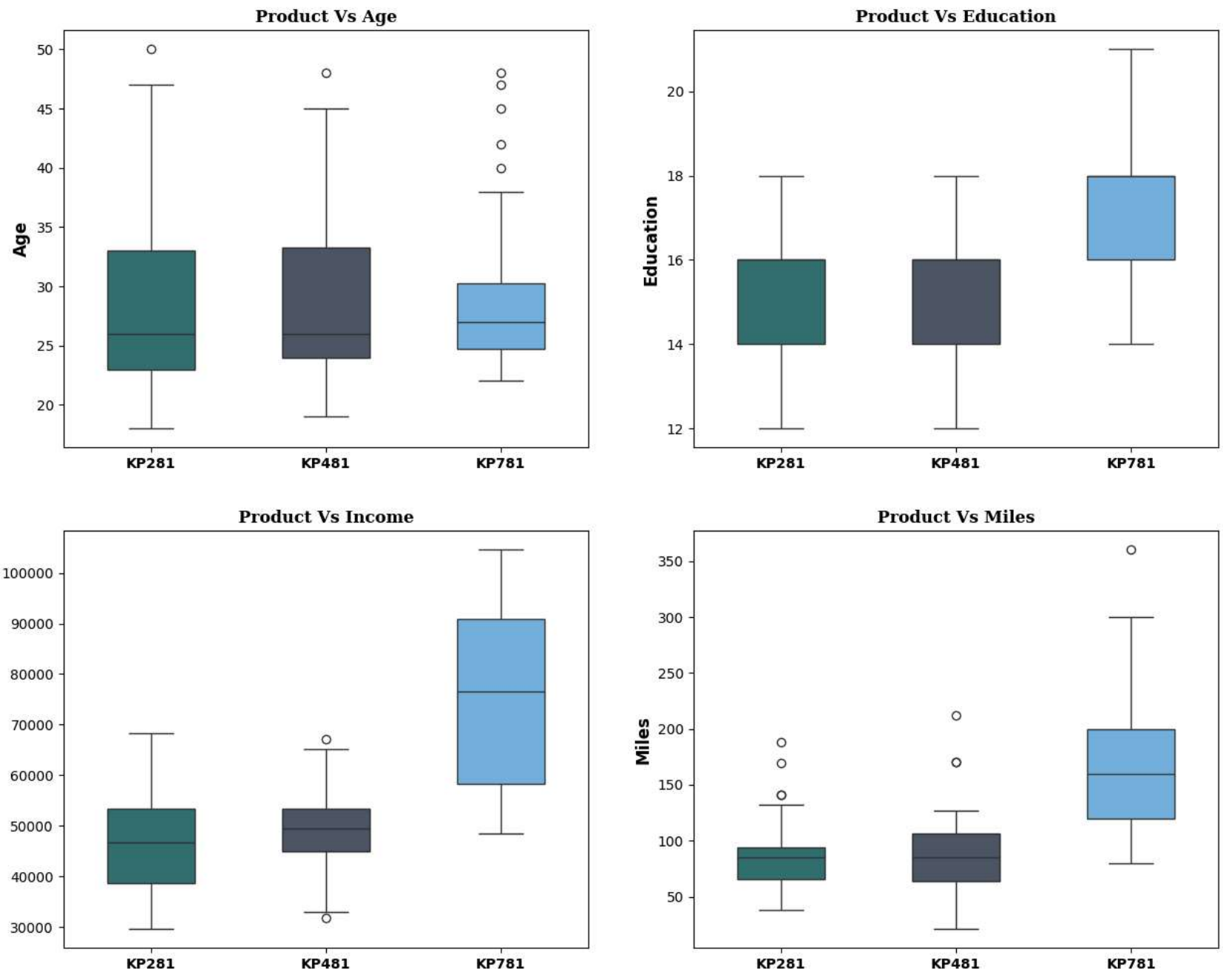
Product Type Analysis

In [ ]:
```
fig = plt.figure(figsize = (15,12))
gs = fig.add_gridspec(2,2)
for i,j,k in [(0,0,'Age'),(0,1,'Education'),(1,0,'Income'),(1,1,'Miles')]:

 #plot position
 ax0 = fig.add_subplot(gs[i,j])
 #plot
 sns.boxplot(data = df, x = 'Product', y = k ,ax = ax0,width = 0.5, palette =["#2C7A7B", "#4A5568" , "#63B3ED"])
 #plot title
 ax0.set_title(f'Product Vs {k}',{'font':'serif', 'size':12,'weight':'bold'})

 #customizing axis
 ax0.set_xticklabels(df['Product'].unique(),fontweight = 'bold')
 ax0.set_ylabel(f'{k}',fontweight = 'bold',fontsize = 12)
 ax0.set_xlabel('')

plt.show()
```



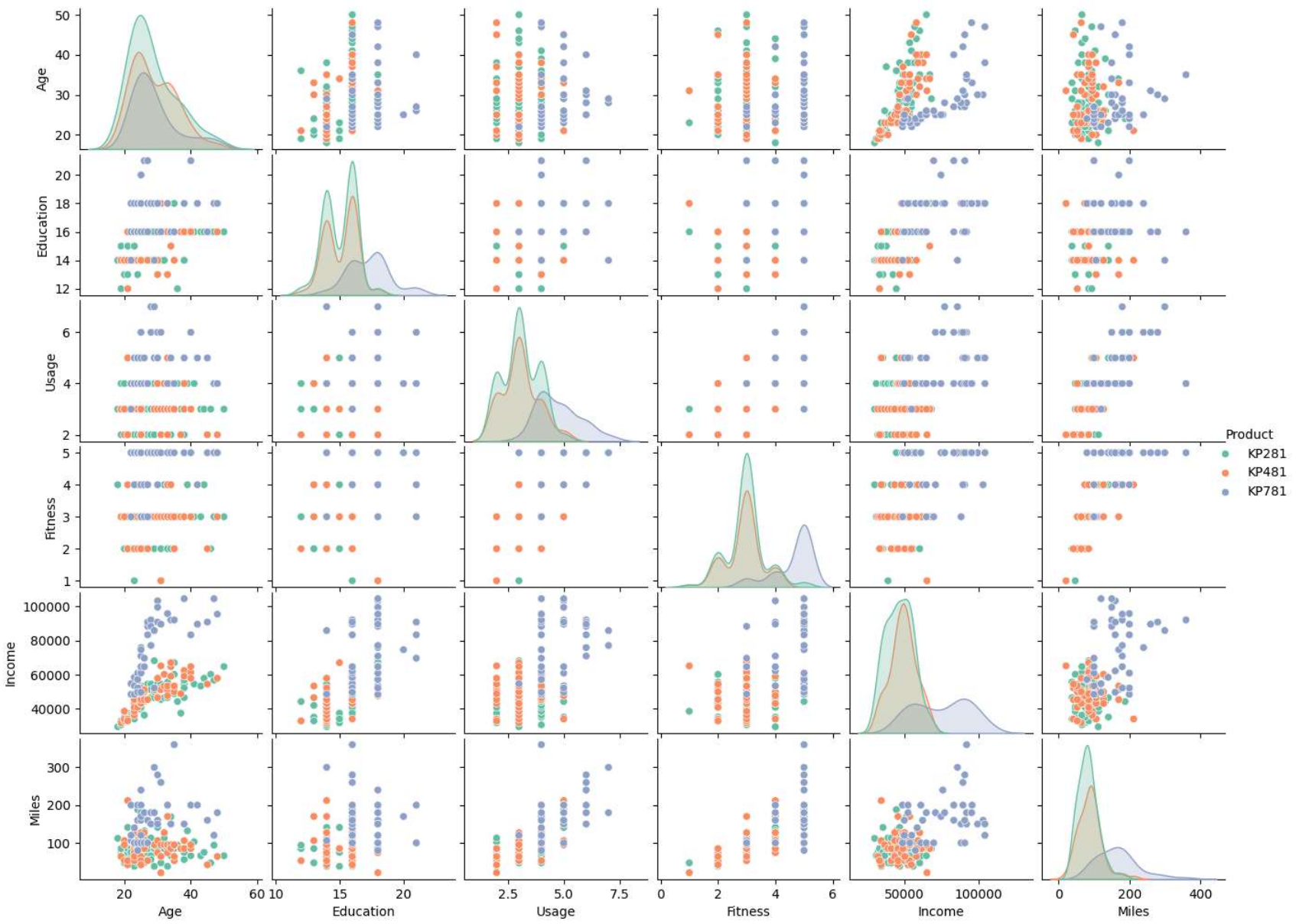
Insights

- The analysis highlights a strong preference for the KP781 treadmill among customers with higher education, higher income, and weekly running goals exceeding 150 miles.

Correlation Analysis

Pairplot

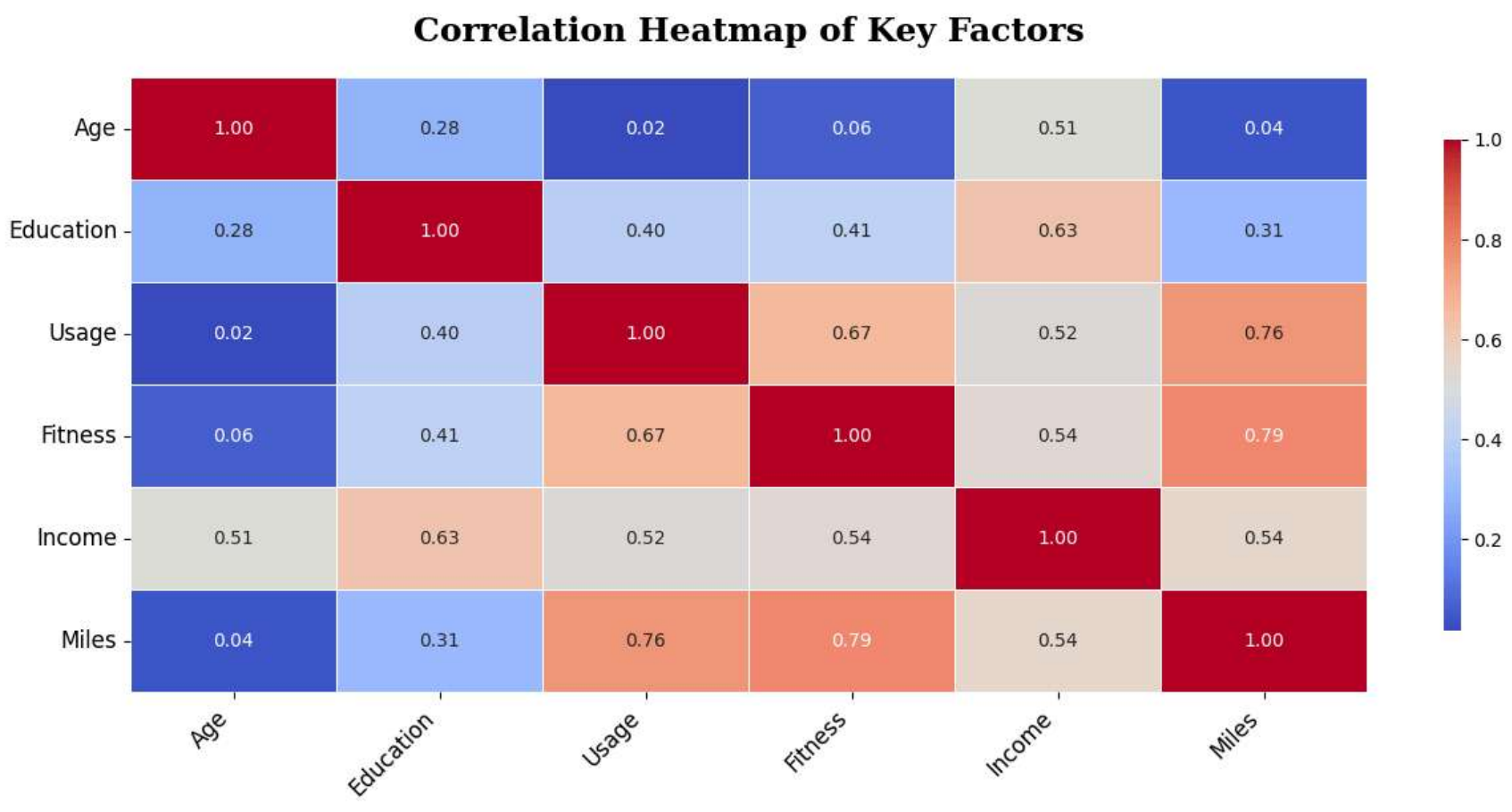
In [ ]:
```
prplt = sns.pairplot(df, hue='Product', palette='Set2', height=3.5,diag_kind='kde')
prplt.fig.set_size_inches(14, 10)
plt.show()
```



Heatmap

In [ ]:
```
corr_df = df[['Age','Education','Usage','Fitness','Income','Miles']].corr()

plt.figure(figsize=(15,6))
ax = sns.heatmap( corr_df, annot=True, fmt='.2f', linewidths=.7, linecolor='white', cmap="coolwarm", cbar_kws={"shrin
plt.title('Correlation Heatmap of Key Factors',fontfamily='serif', fontweight='bold', fontsize=18, pad=20)
plt.xticks(rotation=45, ha='right', fontsize=12)
plt.yticks(rotation=0, fontsize=12)
plt.show()
```



Insights

- From the pair plot, it is evident that Age and Income exhibit a positive correlation, which is further supported by the heatmap indicating a strong relationship between the two.
- Similarly, Education and Income are highly correlated, which is an expected outcome. Education also shows a meaningful correlation with both Fitness rating and Treadmill Usage, suggesting that higher education levels may influence healthier lifestyle habits.
- Additionally, Usage is strongly correlated with both Fitness and Miles, indicating that increased treadmill usage is associated with improved fitness levels and higher mileage covered.

Customer Profiling

Based on the analysis, the overall probability of purchase is as follows:

- KP281: 44%
- KP481: 33%
- KP781: 22%

KP281 – Entry-Level Treadmill

- **Customer Characteristics**
  - Age: Predominantly 18–35 years, with some in the 35–50 range.
  - Education: 13+ years of education.
  - Annual Income: USD 35,000 – 55,000.
  - Weekly Usage: 3–4 times.
  - Fitness Scale: 2–4 (moderate fitness).
  - Running Mileage: 50–100 miles per week.
- **Customer Segments**
  - More popular among Single Females and Partnered Males.
  - Attracts customers seeking an affordable, beginner-friendly treadmill.

KP481 – Intermediate Treadmill

- **Customer Characteristics**
  - Age: Primarily 18–35 years, with some between 35–50.
  - Education: 13+ years of education.
  - Annual Income: USD 40,000 – 80,000.
  - Weekly Usage: 2–4 times.
  - Fitness Scale: 2–4.
  - Running Mileage: 50–200 miles per week.
- **Customer Segments**
  - More popular with Female customers compared to Males.
  - Attracts customers looking for balanced features and value for money.

KP781 – Advanced Treadmill

- **Customer Characteristics**
  - Age: Mostly 18–35 years.
  - Education: 15+ years of education.
  - Annual Income: USD 80,000 and above.
  - Weekly Usage: 4–7 times.
  - Fitness Scale: 3–5 (highly fitness-oriented).
  - Running Mileage: Above 100 miles per week.

- **Customer Segments**
  - Preferred by Partnered Females more than Partnered Males.
  - Chosen by customers where Education and Income show a strong positive correlation.
  - Appeals to high-income, fitness-driven individuals willing to invest in advanced features.

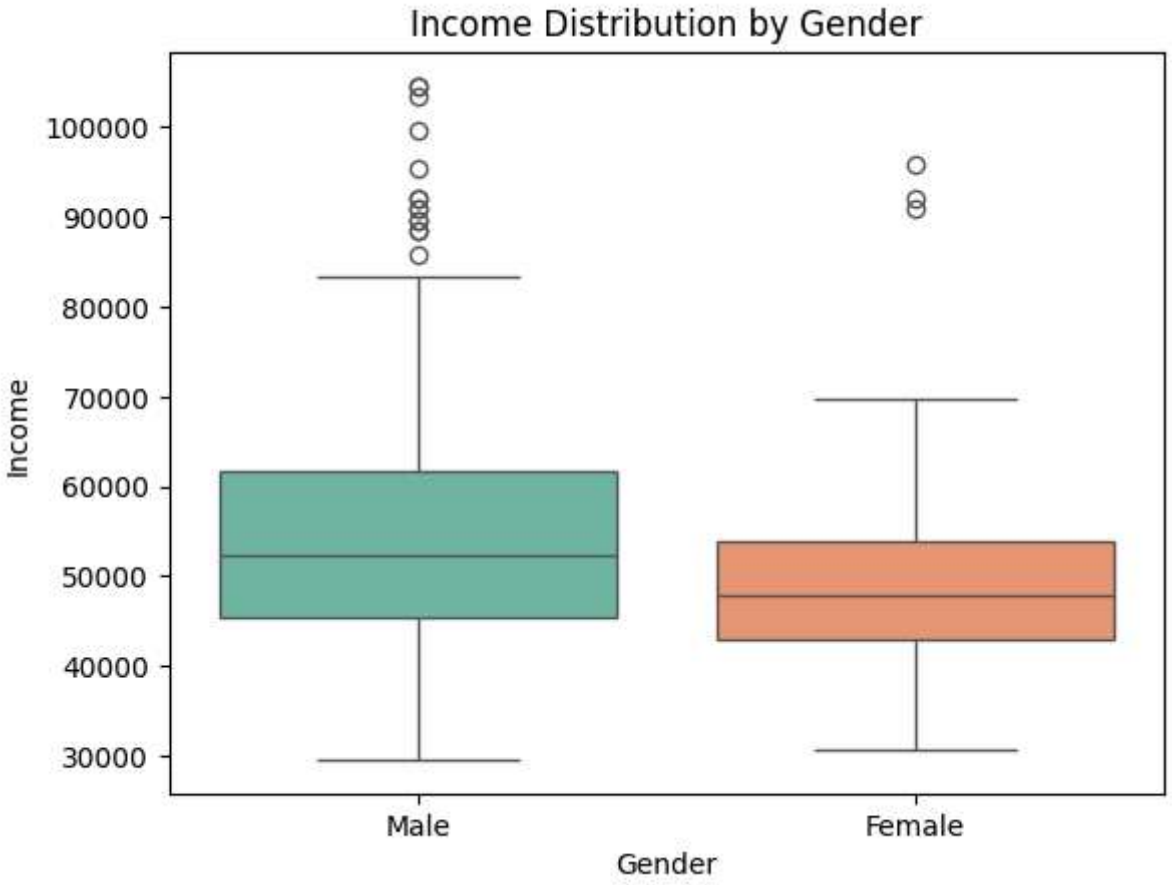
Hypothesis Testing

In [6]:
```
from scipy.stats import norm
from scipy.stats import ttest_ind
from scipy.stats import chi2_contingency
from scipy.stats import f_oneway
```

T-Test Analysis: Comparing Average Income across Genders

Boxplot of Income Distribution by Gender

In [8]:
```
sns.boxplot(x='Gender', y='Income', data=df, palette="Set2")
plt.title("Income Distribution by Gender")
plt.show()
```



From this plot, salaries of men seem more than that of women on average.

Let's test the same using the T-test(Assuming Normal Distribution)

Null and Alternate Hypothesis

- Null Hypothesis (H_0): There is no difference in the average income between men and women.
- Alternative Hypothesis (H_a): Men earn a higher average income than women.

In [10]:
```
male_income = df[df['Gender'] == 'Male']['Income']
female_income = df[df['Gender'] == 'Female']['Income']
```

In [14]:
```
# Performng T-test

t_stat, pvalue = ttest_ind(male_income, female_income, alternative="greater")
pvalue
```

Out[14]:
```
np.float64(0.003263631548607129)
```

In [15]:
```
alpha = 0.05 # 95% confidence
if pvalue < alpha:
  print('Reject H0')
  print('Men earn more than women.')
else:
  print ('Fail to Reject H0')
```

```
Reject H0
Men earn more than women.
```

Insights

- This analysis suggests that, on average, male customers have higher incomes than female customers. This difference is statistically significant when assuming the data follows a normal distribution.

Chi-Square Test Analysis: Assessing the Impact of Gender on Product Preference

Null and Alternate Hypothesis

- Null Hypothesis (H_0): Gender has no effect on treadmill purchase decisions.
- Alternative Hypothesis (H_a): Gender influences treadmill purchase decisions.

In [44]:
```
# Distribution: Chi_square distribution
# Creating a Contingency Table

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

Out[44]:

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

In [30]:
```
# Performing Chi-Square Test

chi_stat, p_value, dof, exp_val = chi2_contingency(gender_product_ct)
print("chi_stat:",chi_stat)
print("p_value:",p_value)
```

```
chi_stat: 12.923836032388664
p_value: 0.0015617972833158714
```

In [20]:
```
alpha = 0.05

if p_value < alpha:
    print("Reject H0")
    print("Gender influences product purchase decision")
else:
    print("Fail to reject H0")
    print("Gender does not effect product purchase decision")
```

```
Reject H0
Gender influences product purchase decision
```

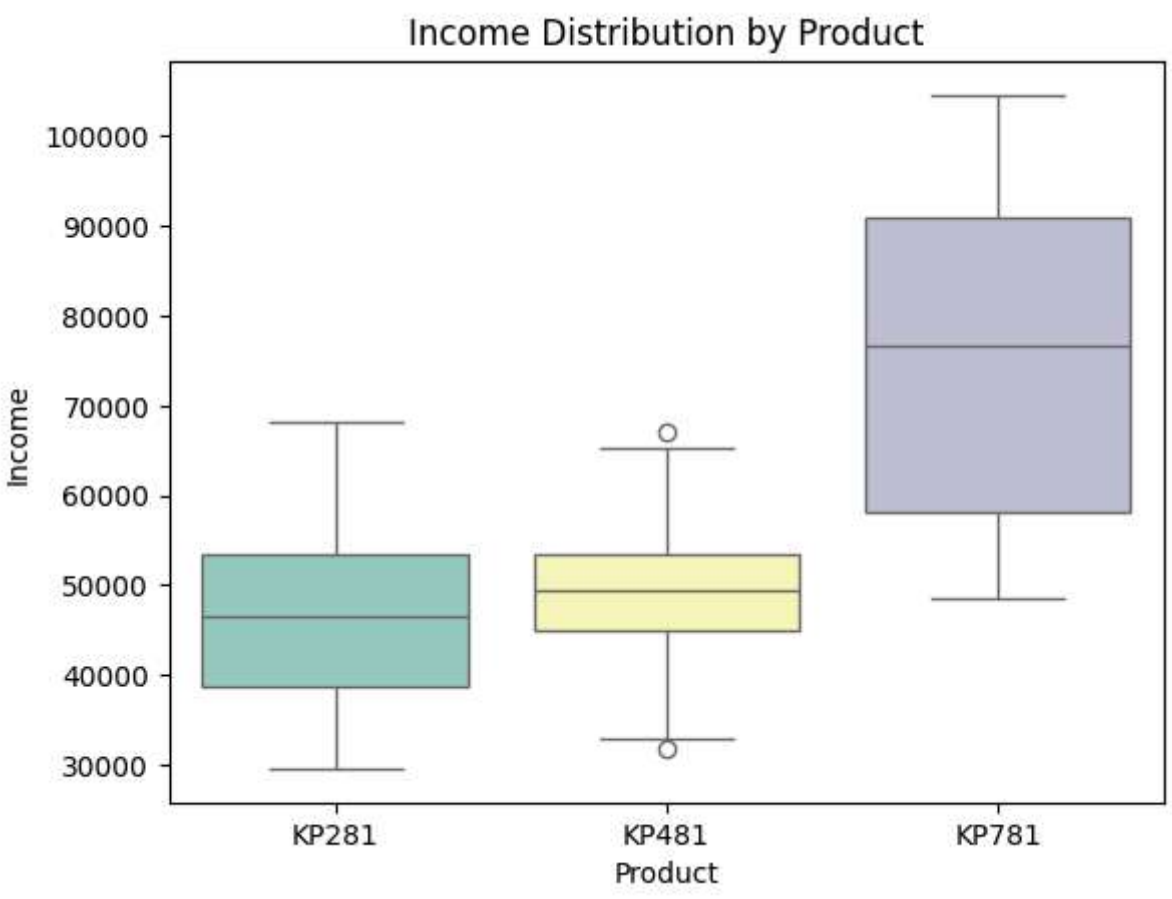
Insights

- Customer gender plays a role in treadmill product choice, with males and females showing distinct preferences.

ANOVA Analysis: Comparing Average Income Across Products

Boxplot of Income Distribution by Product

In [37]:
```
sns.boxplot(x='Product', y='Income', data=df, palette="Set3")
plt.title("Income Distribution by Product")
plt.show()
```



Null and Alternate Hypothesis

- Null Hypothesis (H_0): The average income is the same across all product groups.
- Alternative Hypothesis (H_a): At least one product group has a different average income.

In [38]:
```
# Segmenting Income by Product

income_kp281 = df[df["Product"]=="KP281"]["Income"]
income_kp481 = df[df["Product"]=="KP481"]["Income"]
income_kp781 = df[df["Product"]=="KP781"]["Income"]
```

In [40]:
```
# Distribution: Guassian distribution
# Performing Anova Test

f_stats, p_val = f_oneway(income_kp281, income_kp481, income_kp781)
print("test statistic:",f_stats)
print("p_value:",p_val)
```

```
test statistic: 89.25903546601671
p_value: 1.5644991316342492e-27
```

In [43]:
```
alpha = 0.05
if p_val < alpha:
    print("Reject H0")
    print("At least one product group has a different average income")
else:
    print("Fail to reject H0")
    print("Average income is the same across all product groups.")
```

```
Reject H0
At least one product group has a different average income
```

Insights

- The ANOVA results indicate that at least one treadmill product group has a significantly different average customer income, suggesting that income influences product choice.

Strategic Recommendations

1. Gender-Targeted Marketing for KP781

- **Insight:** Only 18% of KP781 sales come from female customers.
- **Action:** Launch exclusive promotions, trials, and marketing campaigns to increase female engagement and drive sales.

2. Affordable Pricing and Flexible Payment Plans for KP281 & KP481

- **Insight:** Target customers have moderate incomes and varied budgets.
- **Action:** Offer competitive pricing and flexible EMI/payment plans to make these models more accessible and attractive.

3. Enhanced Digital Experience through User-Friendly App Integration

- **Insight:** Customers seek engagement and performance tracking.
- **Action:** Develop an app to track weekly mileage, provide real-time feedback, and offer personalized workout recommendations.

4. Strategic Product Promotions

- **Insight:** KP781 customers are typically high-income males with a preference for premium features.
- **Action:** Tailor marketing messages and campaigns to highlight features aligned with this customer segment.

5. Incentivized KP481 Marketing

- **Insight:** KP481 appeals to mid-level users seeking affordability.
- **Action:** Promote with no-cost EMI options to enhance purchase attractiveness.

6. Targeted Online Marketing

- **Insight:** Customers respond to personalized content.
- **Action:** Use E-commerce and social media platforms to run data-driven targeted ads based on demographics and preferences.

7. Encouraging Female Fitness Engagement

- **Insight:** Female participation in treadmill usage is lower than male.
- **Action:** Create inclusive marketing campaigns emphasizing health benefits and empowerment for female users.

8. Budget-Friendly Positioning for KP281 & KP481

- **Insight:** Entry-level and mid-level models need wider appeal.
- **Action:** Highlight affordability and flexible payment options to attract a broader customer base.

9. Premium Positioning and Endorsements for KP781

- **Insight:** KP781 is a high-end product with features suited for professionals and athletes.
- **Action:** Collaborate with fitness influencers and athletes to enhance credibility and reach.

10. Market Expansion and Older Demographics

- **Insight:** Current products mainly target customers under 50.
- **Action:** Research health considerations and preferences for customers above 50 to expand market reach.

11. Customer Support and Upgrade Pathways

- **Insight:** Customer satisfaction is linked to guidance and upgrades.
- **Action:** Provide robust support and recommend higher-tier treadmill upgrades after consistent usage to boost loyalty.

12. Tailored Recommendations for Female KP781 Customers

- **Insight:** Female users engaging in advanced workouts require tailored features.
- **Action:** Highlight advanced but user-friendly features in marketing materials for female users.

13. Age-Specific Targeting for KP781

- **Insight:** Customers above 40 may prefer KP781 due to health and performance features.
- **Action:** Run campaigns emphasizing benefits that resonate with the 40+ age group.