Business Case: Aerofit - Descriptive Statistics & Probability

Exploratory Data Analysis

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

In [ ]:
```
# importing data
!wget "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/a
```

```
--2024-05-03 04:51:11--  https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/
001/125/original/aerofit_treadmill.csv?1639992749
Resolving d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)... 99.84.178.22
6, 99.84.178.93, 99.84.178.172, ...
Connecting to d2beiqkhq929f0.cloudfront.net (d2beiqkhq929f0.cloudfront.net)|99.84.178.22
6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7279 (7.1K) [text/plain]
Saving to: ‘aerofit_treadmill.csv?1639992749’

aerofit_treadmill.c 100%[===================>]   7.11K  --.-KB/s    in 0s

2024-05-03 04:51:11 (1.99 GB/s) - ‘aerofit_treadmill.csv?1639992749’ saved [7279/7279]
```

In [ ]:
```
# reading csv file
df = pd.read_csv('/content/aerofit_treadmill.csv?1639992749')
df.head()
```

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

finding the number of rows and columns given in the dataset

In [ ]:
```
# checking no of rows and col
df.shape
```

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

insights:

No of rows = 180

No of column = 9

The data type of all columns in the “customers” table.

In [ ]:
```
# finding the datatype, name, total entries in each column
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:

- Product, Gender and Marital Status are object(string)
- Age, Education, Usage, Fitness, Income and Miles are in int64(integer)

Check for the missing values and find the number of missing values in each column

In [ ]:
```
# finding missing values in each column
df.isnull().sum()
```

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

Insights:

Dataset doesn't contain any missing values.

In [ ]:
```
# checking duplicate
df.duplicated().value_counts()
```

Out[ ]:
```
False    180
```

```
Name: count, dtype: int64
```

Insights:

In our dataset doesn't contain duplicates value.

Analysing basic metrics

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

Out[ ]:

	Age	Education	Usage	Fitness	Income	Miles
count	180.000000	180.000000	180.000000	180.000000	180.000000	180.000000
mean	28.788889	15.572222	3.455556	3.311111	53719.577778	103.194444
std	6.943498	1.617055	1.084797	0.958869	16506.684226	51.863605
min	18.000000	12.000000	2.000000	1.000000	29562.000000	21.000000
25%	24.000000	14.000000	3.000000	3.000000	44058.750000	66.000000
50%	26.000000	16.000000	3.000000	3.000000	50596.500000	94.000000
75%	33.000000	16.000000	4.000000	4.000000	58668.000000	114.750000
max	50.000000	21.000000	7.000000	5.000000	104581.000000	360.000000

insights:

- Total count of all columns is 180
- Age: Mean age of the customer is 28 years, half of the customer's mean age is 26.
- Education: Mean Education is 15 with maximum as 21 and minimum as 12.
- Usage: Mean Usage per week is 3.4, with maximum as 7 and minimum as 2.
- Fitness: Average rating is 3.3 on a scale of 1 to 5.
- Miles: Average number of miles the customer walks is 103 with maximum distance travelled by most people is almost 115 and minimum is 21.
- Income (in $): Most customer earns around 58K annually, with maximum of 104K and minimum almost 30K

In [ ]:
```
df['Gender'].value_counts()
```

Out[ ]:
```
Gender
Male      104
Female     76
Name: count, dtype: int64
```

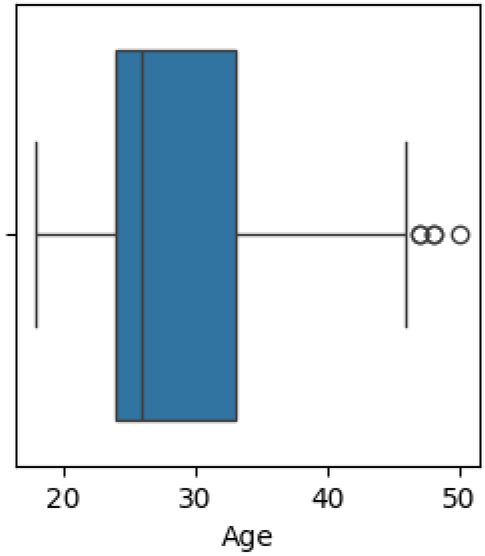
- In dataset we have 104 male and 76v female

---

2. **Detect Outliers**

Finding the outliers for every continuous variable in the dataset

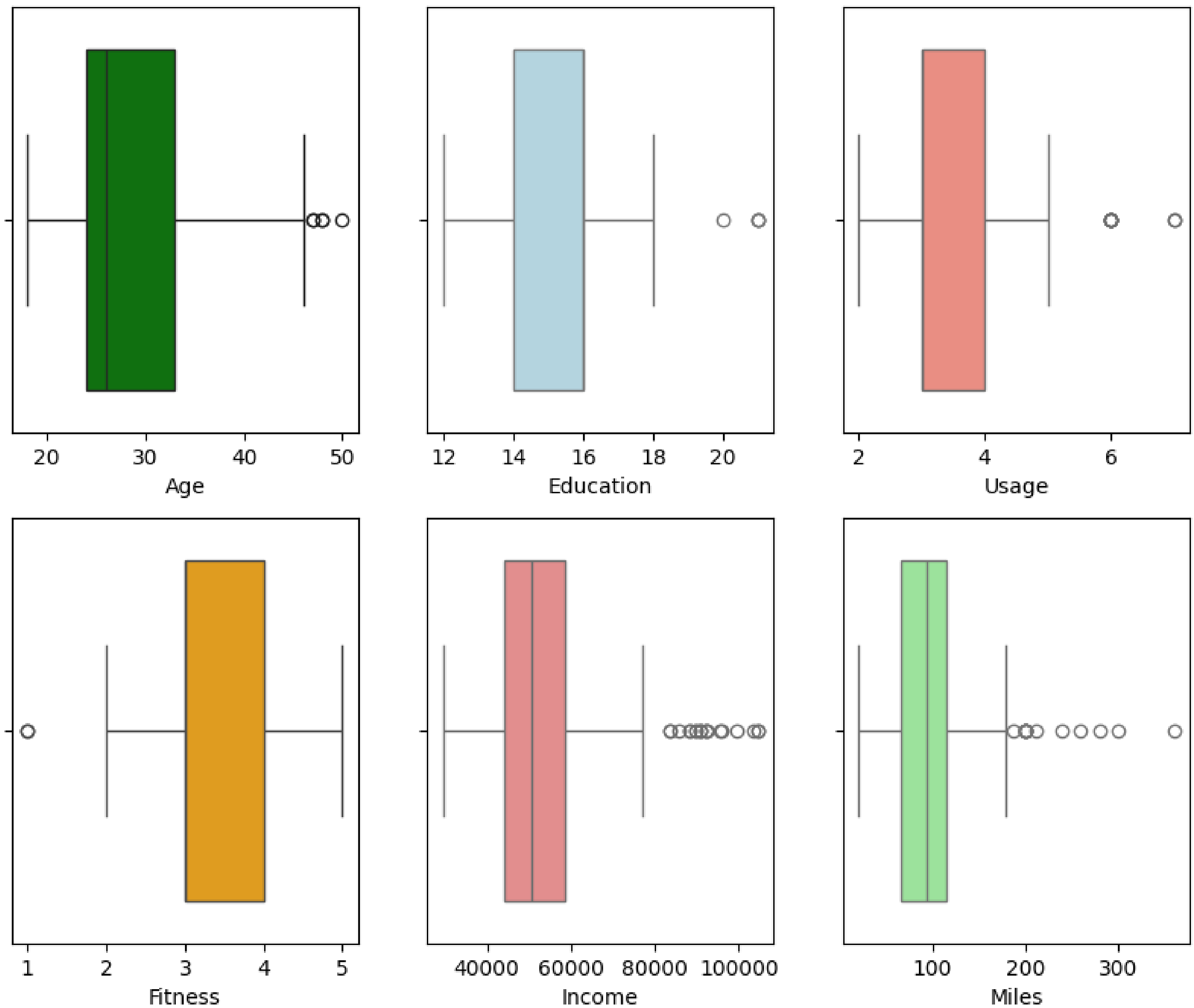
In [ ]:
```
plt.figure(figsize=(3,3))
sns.boxplot(data=df,x='Age')
plt.show()
```



By using subplots

In [ ]:
```
fig,ax = plt.subplots(2,3, figsize=(10,8))
sns.boxplot(data=df, x='Age',color='g',ax=ax[0,0])
sns.boxplot(data=df, x='Education', color='lightblue', ax=ax[0,1])
sns.boxplot(data=df, x='Usage',  color='salmon', ax=ax[0,2])
sns.boxplot(data=df, x='Fitness', color='orange', ax=ax[1,0])
sns.boxplot(data=df, x='Income', color='lightcoral', ax=ax[1,1])
sns.boxplot(data=df, x='Miles', color='lightgreen', ax=ax[1,2])
fig.suptitle('Outliers')
plt.show()
```

Outliers



insights:

Other than **Income** and **Miles** variables have relatively lower presence of outliers.

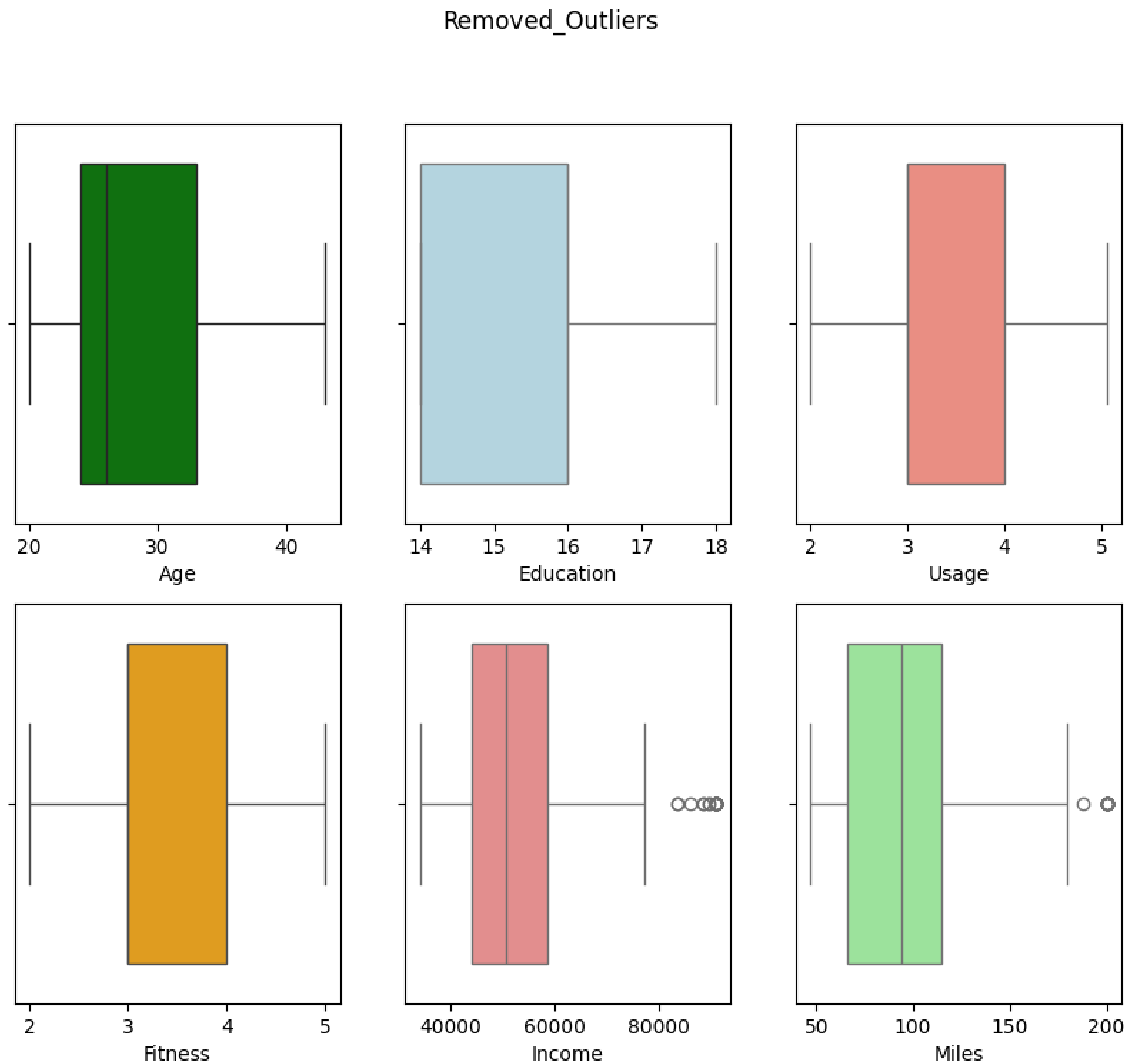
Remove/clip the data between the 5 percentile and 95 percentile

In [ ]:
```
remove_Age = np.clip(df['Age'], np.percentile(df['Age'],5), np.percentile(df['Age'],95))
remove_Education= np.clip(df['Education'], np.percentile(df['Education'],5), np.percenti
remove_Usage= np.clip(df['Usage'], np.percentile(df['Usage'],5), np.percentile(df['Usage
remove_Fitness= np.clip(df['Fitness'], np.percentile(df['Fitness'],5), np.percentile(df[
remove_Income= np.clip(df['Income'], np.percentile(df['Income'],5), np.percentile(df['In
remove_Miles= np.clip(df['Miles'], np.percentile(df['Miles'],5), np.percentile(df['Miles
```

In [ ]:
```
# Printing the result by using subplots

fig,ax = plt.subplots(2,3, figsize=(10,8))
sns.boxplot(data=df, x=remove_Age,color='g',ax=ax[0,0])
sns.boxplot(data=df, x=remove_Education, color='lightblue', ax=ax[0,1])
```

```
sns.boxplot(data=df, x=remove_Usage,  color='salmon', ax=ax[0,2])
sns.boxplot(data=df, x=remove_Fitness, color='orange', ax=ax[1,0])
sns.boxplot(data=df, x=remove_Income, color='lightcoral', ax=ax[1,1])
sns.boxplot(data=df, x=remove_Miles, color='lightgreen', ax=ax[1,2])
fig.suptitle('Removed_Outliers')
plt.show()
```



insights:

Clearly we can see that data has been removed between the 5 percentile and 95 percentile .

---

3. Check if features like marital status, Gender, and age have any effect on the product purchased

Find if there is any relationship between the categorical variables and the output variable in the data.

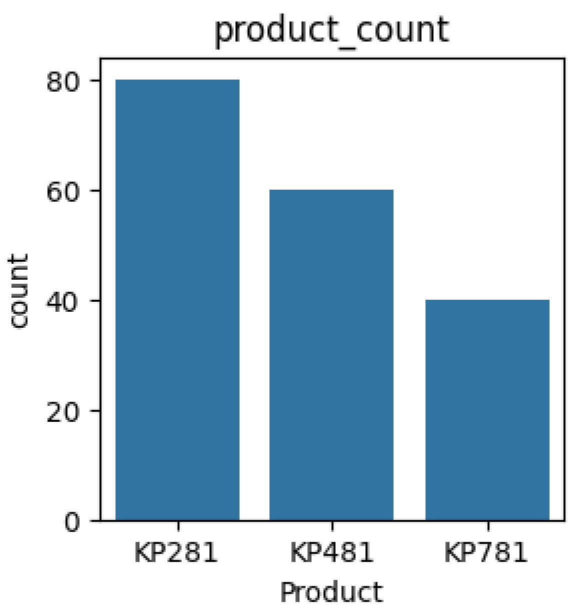
In [ ]:
```python
df.head()
```

Out[ ]:

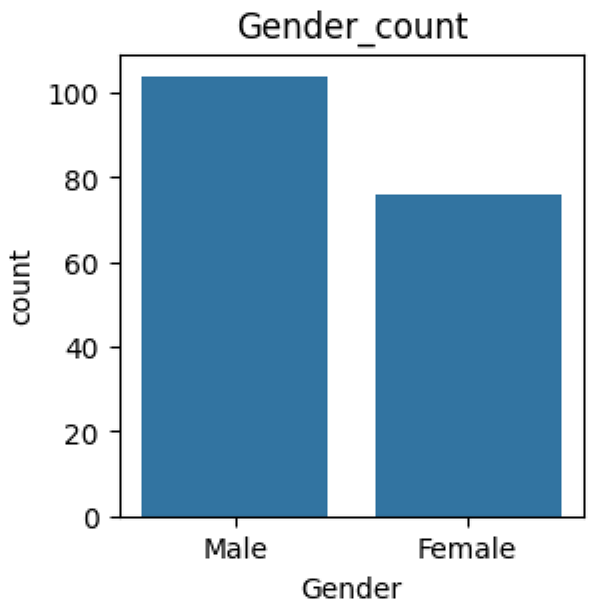
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

Univariate Analysis

In [ ]:
```python
plt.figure(figsize=(3,3))
sns.countplot(data=df,x='Product')
plt.title('product_count')
plt.show()
```

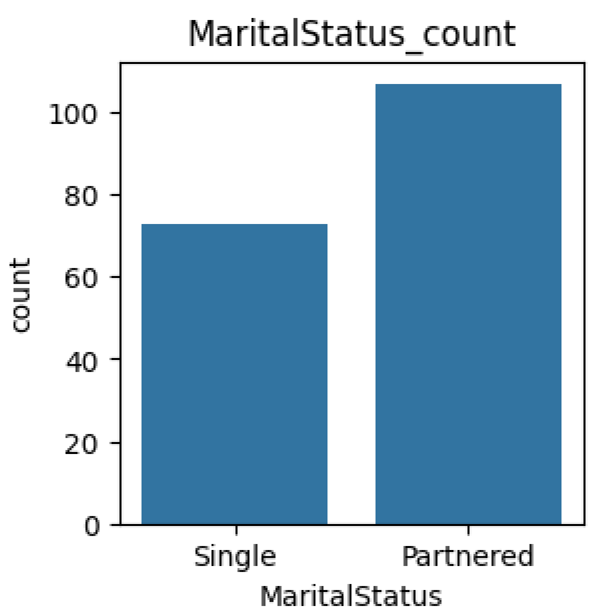


In [ ]:
```python
plt.figure(figsize=(3,3))
sns.countplot(data=df,x='Gender')
plt.title('Gender_count')
plt.show()
```



In [ ]:

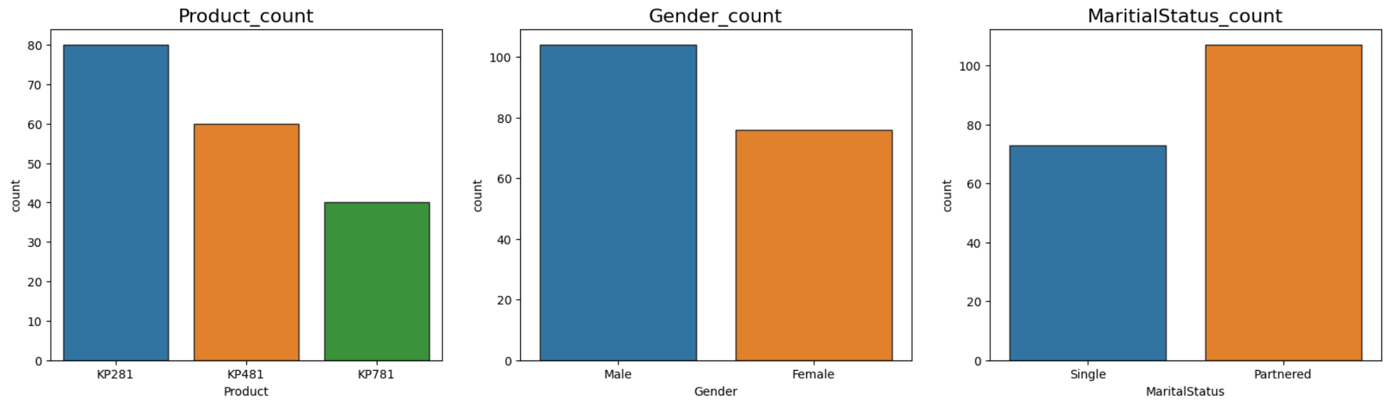
```
plt.figure(figsize=(3,3))
sns.countplot(data=df,x='MaritalStatus')
plt.title('MaritalStatus_count')
plt.show()
```



plotting the all graphs by using subplots

In [ ]:

```
fig,ax = plt.subplots(1,3, figsize=(20,5))
sns.countplot(data=df,x='Product',ax=ax[0],hue='Product',edgecolor="0.15")
sns.countplot(data=df,x='Gender',ax=ax[1],hue='Gender',edgecolor="0.15")
sns.countplot(data=df,x='MaritalStatus',ax=ax[2],hue='MaritalStatus',edgecolor="0.15")
ax[0].set_title('Product_count',fontsize=16)
ax[1].set_title('Gender_count',fontsize=16)
ax[2].set_title('MaritialStatus_count',fontsize=16)
plt.show()
```

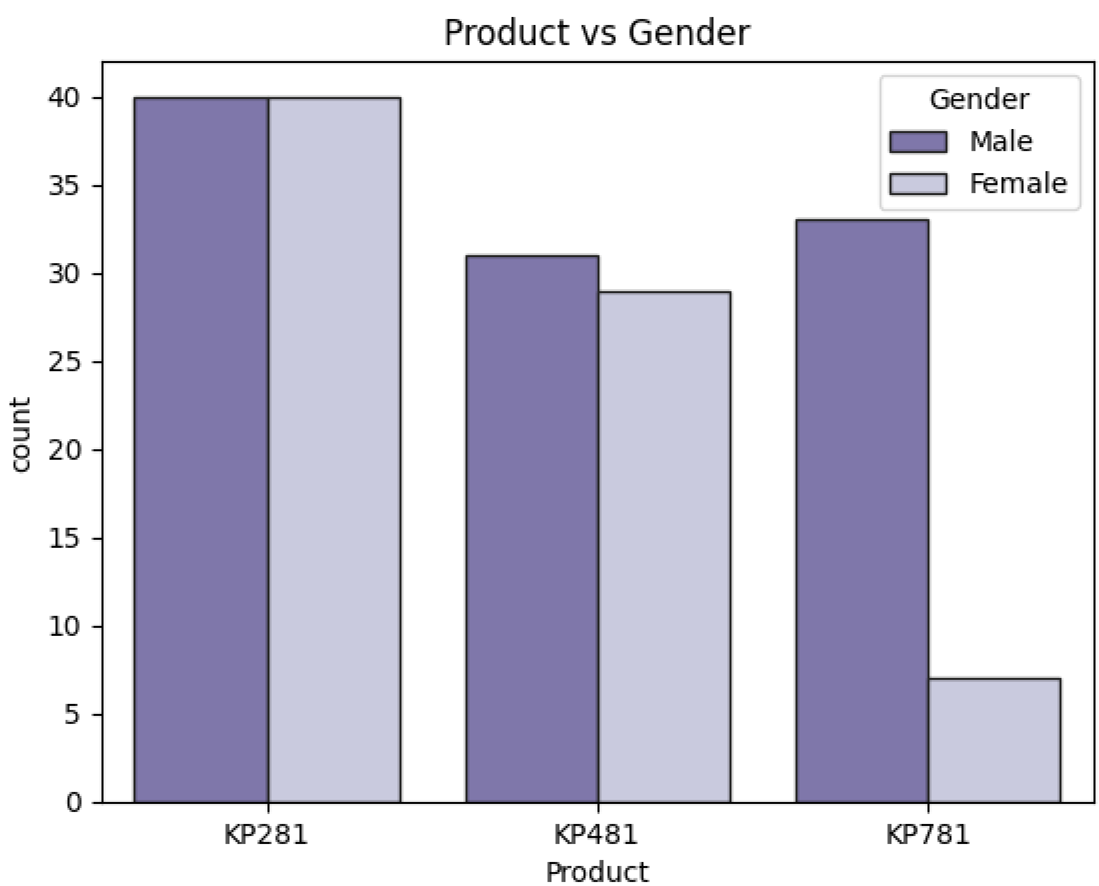


insights:

1. Most frequent Purchased product is KP281.
2. No of male is higher than female .
3. Partnered persons are more .

Bivariate Analysis

In [ ]:
```
sns.countplot(data=df,x='Product',hue='Gender',palette='Purples_r',edgecolor="0.15")
plt.title('Product vs Gender')
plt.show()
```

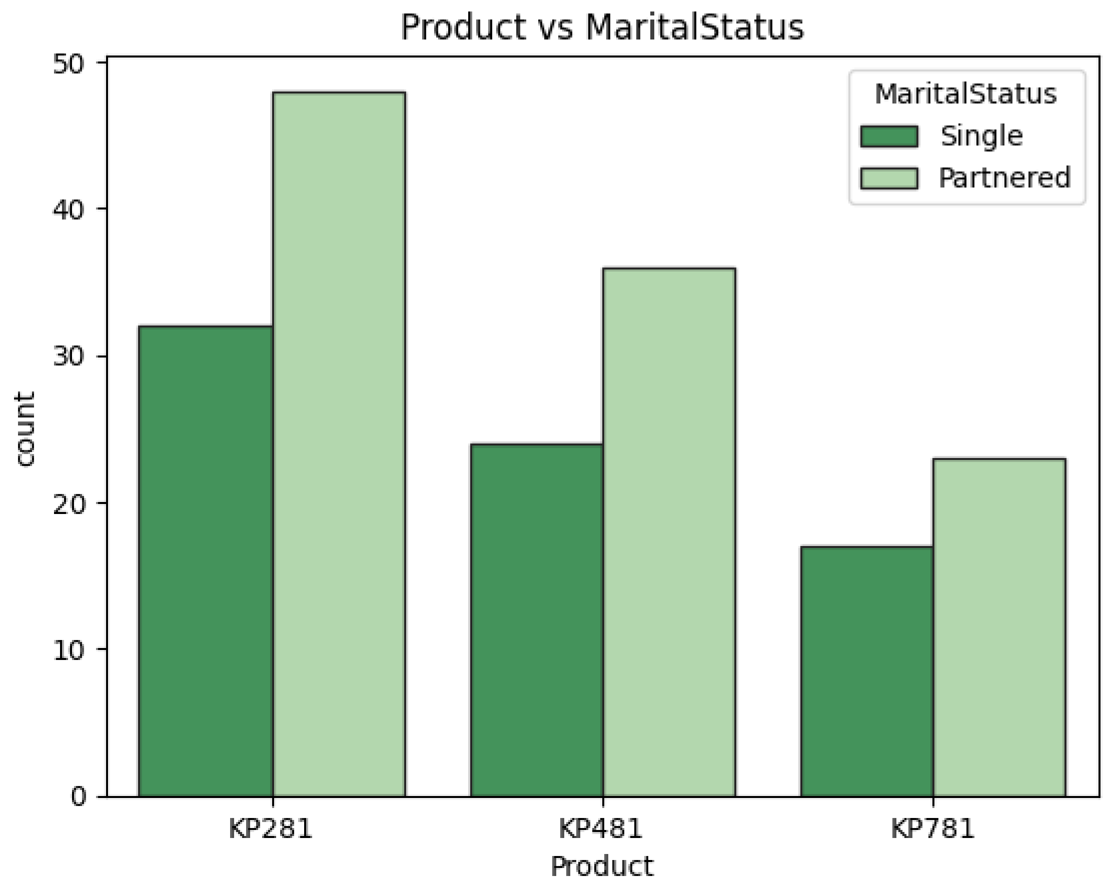


insights: (Product vs Gender)

1. The product have purchased by same number of male and female.
2. Most of the male custmors have purchased KP781 product.

In [ ]:

```
sns.countplot(data=df,x='Product',hue='MaritalStatus',palette='Greens_r',edgecolor="0.15
plt.title('Product vs MaritalStatus')
plt.show()
```



insights:

(Product vs MaritalStatus)

1. All three products have purchased by partnered customer.

Find if there is any relationship between the continuous variables and the output variable in the data.

In [ ]:

```
df.head()
```

Out[ ]:

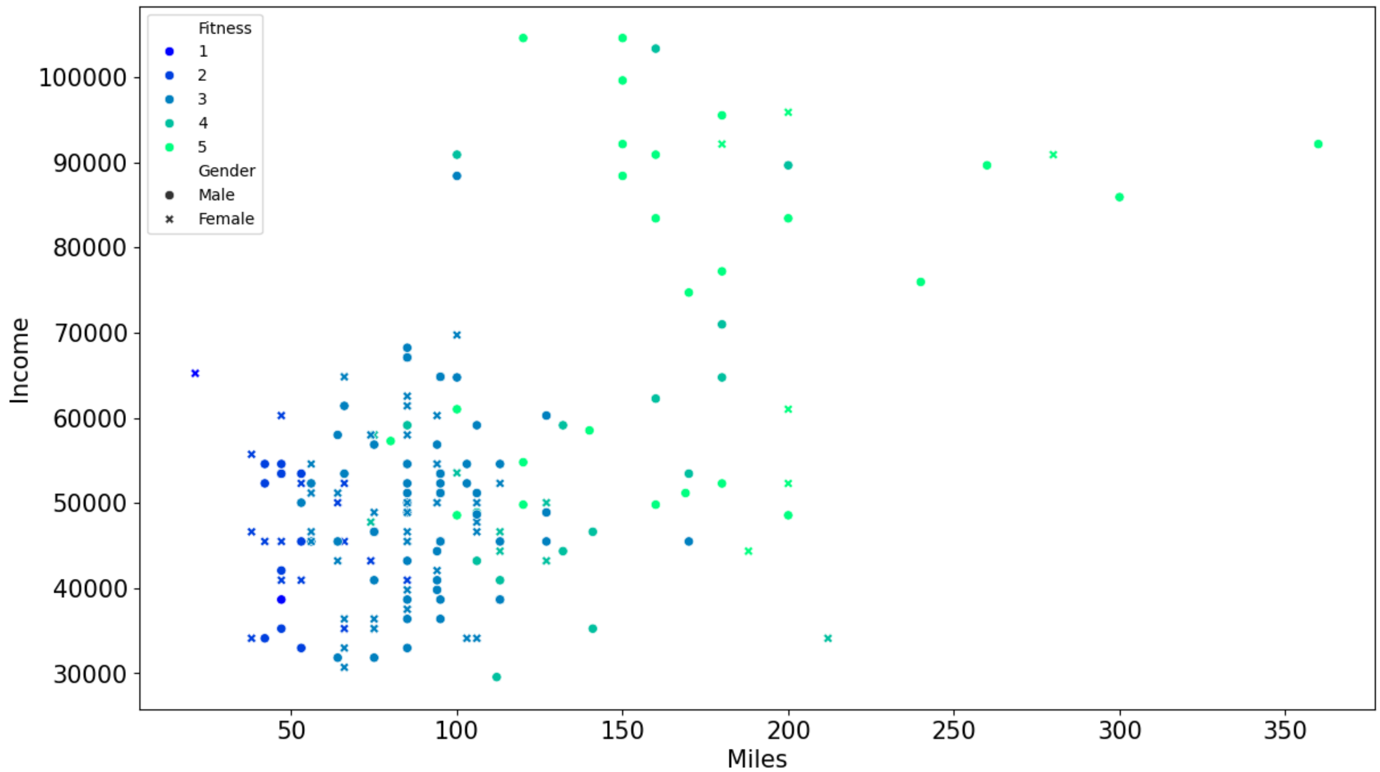
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
0	KP281	18	Male	14	Single	3	4	29562	112
1	KP281	19	Male	15	Single	2	3	31836	75
2	KP281	19	Female	14	Partnered	4	3	30699	66
3	KP281	19	Male	12	Single	3	3	32973	85
4	KP281	20	Male	13	Partnered	4	2	35247	47

In [ ]:

```
plt.figure(figsize=(14,8))
sns.scatterplot(x='Miles',y='Income',data=df,hue='Fitness',style='Gender',palette='winte
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
```

```
plt.xlabel('Miles',fontsize=15)
plt.ylabel('Income',fontsize=15)
```

Out[ ]: Text(0, 0.5, 'Income')



insights:

1. Most of the customers fitness level is 3 and 4 .
2. some customrs have fitness level 5 and which shows that they cover maximum miles.

---

4. Representing the Probability

Find the marginal probability (what percent of customers have purchased KP281, KP481, or KP781)

Categorizing all continuous variable into related categories

In [ ]:
```
Age_bin = [17,25,35,45,60]
Age_labels=['Young Adults', 'Adults', 'Middle-Aged Adults', 'Elder']
df['Age_group']=pd.cut( df['Age'], bins=Age_bin, labels=Age_labels )

Edu_bin = [0,12,15,22]
Edu_labels=['Primary','Secondary','Higher']
df['Edu_group']=pd.cut(df['Education'],bins=Edu_bin,labels=Edu_labels)

Income_bin=[0,40000,60000,80000,200000]
Income_labels=['Low','Modrate','High','Very high']
df['Income_group']=pd.cut(df['Income'], bins=Income_bin, labels=Income_labels)

Miles_bins = [0, 50, 100, 200, 400]
Miles_labels = ['Light Activity', 'Moderate Activity', 'Active Lifestyle', 'Fitness Enth
df['Miles_group'] = pd.cut(df['Miles'], bins=Miles_bins, labels=Miles_labels)
```

```python
df.head()
```

Out[ ]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles	Age_group	Edu_group	Ir
0	KP281	18	Male	14	Single	3	4	29562	112	Young Adults	Secondary	
1	KP281	19	Male	15	Single	2	3	31836	75	Young Adults	Secondary	
2	KP281	19	Female	14	Partnered	4	3	30699	66	Young Adults	Secondary	
3	KP281	19	Male	12	Single	3	3	32973	85	Young Adults	Primary	
4	KP281	20	Male	13	Partnered	4	2	35247	47	Young Adults	Secondary	

In [ ]:

```python
# Probability of product purchase with respect to Gender
pd.crosstab(index = df['Product'], columns = df['Gender'], margins = True, normalize = T
```

Out[ ]:

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

insights:

- For product KP281, 22% of purchases are made by females, 22% by males, and in total, it represents 44% of all purchases.
- Similarly, for product KP481, 16% of purchases are made by females, 17% by males, and in total, it represents 33% of all purchases.
- And for product KP781, 4% of purchases are made by females, 18% by males, and in total, it represents 22% of all purchases.
- The last row and column provide the overall distribution of purchases among genders.

In [ ]:

```python
# Probability of product purchase with respect to Age_group
pd.crosstab(index=df['Product'],columns=df['Age_group'],margins=True,normalize = True).r
```

Out[ ]:

Age_group Product	Young Adults	Adults	Middle-Aged Adults	Elder	All
KP281	0.19	0.18	0.06	0.02	0.44
KP481	0.16	0.13	0.04	0.01	0.33
KP781	0.09	0.09	0.02	0.01	0.22
All	0.44	0.41	0.12	0.03	1.00

insights:

- For product KP281

- 19% of purchases are made by Young Adults,
- 18% by Adults,
- 6% by Middle-Aged adults, and 2% by Elders, totaling to 44% of all purchases.
- for product KP481,
  - 16% of purchases are made by Young Adults,
  - 13% by Adults,
  - 4% by Middle-Aged adults, and 1% by Elders, totaling to 33% of all purchases.
- for product KP781,
  - 9% of purchases are made by Young Adults,
  - 9% by adults, 2% by Middle-Aged adults, and 1% by Elders, totaling to 22% of all purchases.

The last row and column provide the overall distribution of purchases among different age groups.

In [ ]:
```
# Probability of product purchase with respect to education
pd.crosstab(index = df['Product'], columns = df['Edu_group'], margins = True, normalize
```

Out[ ]:

Edu_group	Primary	Secondary	Higher	All
Product				
KP281	0.01	0.21	0.23	0.44
KP481	0.01	0.14	0.18	0.33
KP781	0.00	0.01	0.21	0.22
All	0.02	0.36	0.62	1.00

insights:

1. Customers with Higher Education (Above 15 Years) have a 62% probability of purchasing a treadmill. The conditional probabilities for each treadmill model given Higher Education are:
   - KP281: 23%
   - KP481: 18%
   - KP781: 21%
2. Customers with Secondary Education (13-15 yrs) show a 36% probability of purchasing a treadmill. The conditional probabilities for each treadmill model given Secondary Education are:
   - KP281: 21%
   - KP481: 14%
   - KP781: 1%

In [ ]:
```
# Probability of product purchase with respect to  income
pd.crosstab(index = df['Product'], columns = df['Income_group'], margins = True, normali
```

Out[ ]:

Income_group	Low	Modrate	High	Very high	All
Product					
KP281	0.13	0.28	0.03	0.00	0.44
KP481	0.05	0.24	0.04	0.00	0.33
KP781	0.00	0.06	0.06	0.11	0.22
All	0.18	0.59	0.13	0.11	1.00

insights:

- Low income (<40k)
  - probability of purchasing KP281 is 13%
  - probability of purchasing K481 is 5%
  - probability of purchasing KP781 is 0%
- modrate income(40k-60k)
  - probability of purchasing KP281 is 29%
  - probability of purchasing KP481 is 25%
  - probability of purchasing KP781 is 60%
- High income(60k-80k)
  - p(KP281):3%
  - p(KP481):4%
  - p(KP781):6%
- very high(80k-1l)
  - p(KP281):0%
  - p(KP481):0%
  - p(KP781):11%

In [ ]:

```
# Probability of product purchase with respect to miles
pd.crosstab(index = df['Product'], columns = df['Miles_group'], margins = True, normaliz
```

Out[ ]:

Miles_group Product	Light Activity	Moderate Activity	Active Lifestyle	Fitness Enthusiast	All
KP281	0.07	0.28	0.10	0.00	0.44
KP481	0.03	0.22	0.08	0.01	0.33
KP781	0.00	0.04	0.15	0.03	0.22
All	0.09	0.54	0.33	0.03	1.00

insights:

- For customers with a Light Activity lifestyle (0 to 50 miles/week), the probability of purchasing a treadmill is 9%. Among these customers:
  - p(KP281):7%
  - p(KP481):3%
  - p(KP781):0%
- Customers with a Moderate Activity lifestyle (51 to 100 miles/week) have a 54% probability of purchasing a treadmill. Within this group:
  - p(KP281):28%
  - p(KP481):22%
  - p(KP781):4%
- For customers with an Active Lifestyle (100 to 200 miles/week), the probability of purchasing a treadmill is 33%. Among these customers:
  - p(KP281):10%
  - p(KP481):8%
  - p(KP781):15%

In [ ]:
```
# Probability of product purchase with respect to maritalstatus
pd.crosstab(index = df['Product'], columns = df['MaritalStatus'], margins = True, normal
```

Out[ ]:

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

insights:

- Married customers are more likely to purchase a treadmill, with a probability of 59%. When considering married customers:
  - KP281 is 27%
  - KP481 is 20%
  - KP781 is 13%.
- Unmarried customers have a probability of 41% of purchasing a treadmill. When considering unmarried customers:
  - KP281 is 18%
  - KP481 is 13%
  - KP781 is 9%

In [ ]:
```
# Probability of product purchase with respect to usage
pd.crosstab(index = df['Product'], columns = df['Usage'], margins = True, normalize = Tr
```

Out[ ]:

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

insights:

- Customers using the treadmill 2 times per week have a purchasing probability of 18%
- For customers with a usage of 3 times per week, the probability of purchasing a treadmill is 38%
- When customers use the treadmill 4 times per week, the probability of a purchase is 29%

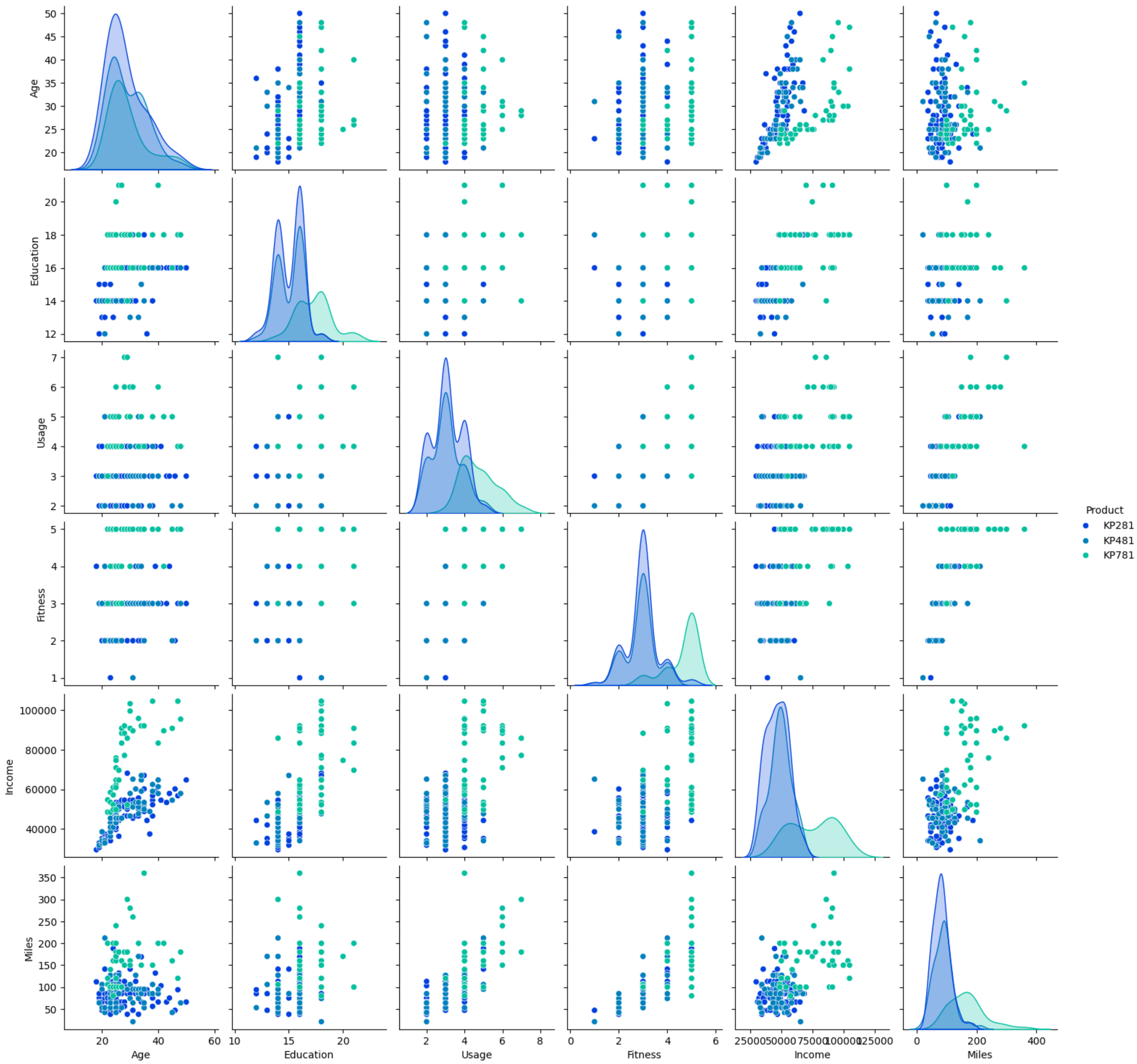
---

5. Check the correlation among different factors

Find the correlation between the given features in the table.

for correlation : pairplot and heatmap

In [ ]:
```
sns.pairplot(data=df,hue='Product',palette='winter')
plt.show()
```



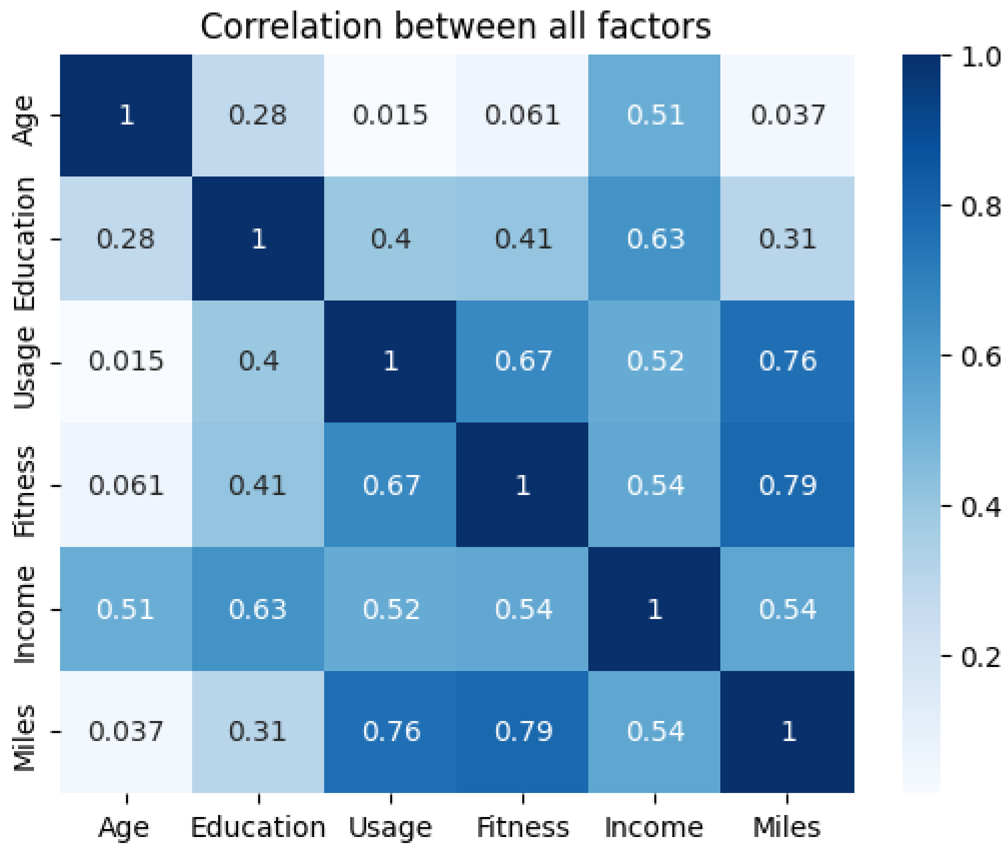
insights:

- we observe a positive correlation between Age and Income.
- Usage displays a strong correlation with Fitness and Miles, indicating that higher treadmill usage

In [ ]:
```
sns.pairplot(data=df)
```

Out[ ]:
```
<seaborn.axisgrid.PairGrid at 0x79ba02aa71f0>
```

In [ ]:
```
sns.heatmap(df.corr(numeric_only=True),  cmap= "Blues", annot=True)
plt.title('Correlation between all factors')
plt.show()
```



insights

- Correlation between Age and Miles is 0.03
- Correlation between Education and Income is 0.63
- Correlation between Usage and Fitness is 0.67
- Correlation between Fitness and Age is 0.061
- Correlation between Income and Usage is 0.52
- Correlation between Miles and Age is 0.037

A heat map plots rectangular data as a color-encoded matrix.

Stronger the colour, stronger the correlation b/w the variables

---

6. Customer profiling and recommendation

6.1.1 Overview:

- Probability of purchasing KP281: 44%
- Probability of purchasing KP481: 33%
- Probability of purchasing KP781: 22%

6.1.2 Customer Profile for KP281 Treadmill:

- Age: 18 to 35 years, with some aged 35 to 50
- Education: 13 years and above
- Income: Below
- USD 60,000 annually
- Usage: 2 to 4 times weekly

- Fitness: Scale of 2 to 4
- Miles: 50 to 100 miles per week

6.1.3 Customer Profile for KP481 Treadmill:

- Age: Mainly 18 to 35 years, with some aged 35 to 50
- Education: 13 years and above
- Income: Between USD 40,000 to USD 80,000 annually
- Usage: 2 to 4 times weekly
- Fitness: Scale of 2 to 4
- Miles: 50 to 200 miles per week

6.1.4 Customer Profile for KP781 Treadmill:

- Gender: Male
- Age: Primarily 18 to 35 years
- Education: 15 years and above
- Income: USD 80,000 and above annually
- Fitness: Scale of 3 to 5
- Miles: 100 miles and above per week

6.2. Recommendations

- KP281 and KP481 also brings in significant amount of revenue and is prefered mostly by youth , added features and specialized discounts could help boost sales.
- Target the Age group above 40 years to recommend Product KP781.

- Introduce entry-level pricing for KP281, mid-range pricing for KP481, and premium pricing for KP781
- Offer package deals to add value and justify higher price points.

- Host online sessions focusing on fitness topics tailored to different education levels
- Showcase how treadmill models support various fitness goals.

- Offer package deals to add value and justify higher price points.
- Target females and lower-income customers with campaigns emphasizing affordability and moderate exercise suitability.

- we should run a marketing campaign on to encourage women to exercise more

- Provide customer support and recommend users to upgrade from lower versions to next level versions after consistent usages.
