Aerofit

October 12, 2024

[50]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1 Checking the Number of Rows and Columns in the DataSet

[53]:
```python
# Reading the DataSet
df_Aerofit = pd.read_csv('Aerofit.csv')
df_Aerofit
```

[53]:
```
    Product  Age  Gender  Education MaritalStatus  Usage  Fitness  Income  \
0     KP281   18    Male         14        Single      3        4   29562
1     KP281   19    Male         15        Single      2        3   31836
2     KP281   19  Female         14     Partnered      4        3   30699
3     KP281   19    Male         12        Single      3        3   32973
4     KP281   20    Male         13     Partnered      4        2   35247
..      ...  ...     ...        ...           ...    ...      ...     ...
175   KP781   40    Male         21        Single      6        5   83416
176   KP781   42    Male         18        Single      5        4   89641
177   KP781   45    Male         16        Single      5        5   90886
178   KP781   47    Male         18     Partnered      4        5  104581
179   KP781   48    Male         18     Partnered      4        5   95508

     Miles
0      112
1       75
2       66
3       85
4       47
..     ...
175    200
176    200
177    160
178    120
```

```
179   180
[180 rows x 9 columns]
```

There are 180 rows in the Dataset and 10 columns

2 Checking the Null Values

[5]:
```
# Checking the Null Values
df_Aerofit.isnull().sum()
```

[5]:
```
Product          0
Age              0
Gender           0
Education        0
MaritalStatus    0
Usage            0
Fitness          0
Income           0
Miles            0
dtype: int64
```

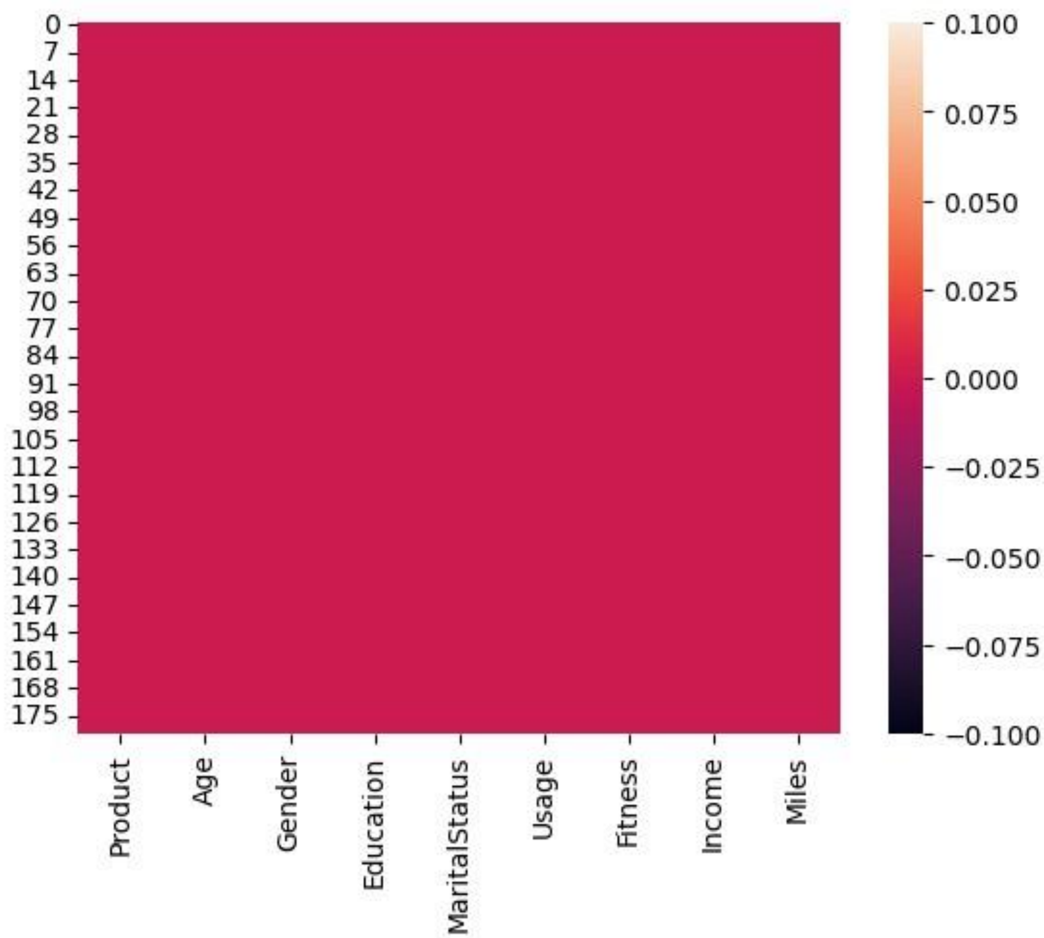
[ ]:
```
There are no missing values  in the data.
```

[6]:
```
# Checking the Number of Rows and Columns in the DataSet

print(f"Number of rows: {df_Aerofit.shape[0]} \nNumber of columns: {df_Aerofit.
↪shape[1]}")
```

```
Number of rows: 180
Number of columns: 9
```

[7]:
```
# Graphical Analysis using heat map
sns.heatmap(df_Aerofit.isnull())
```

[7]:
```
<Axes: >
```



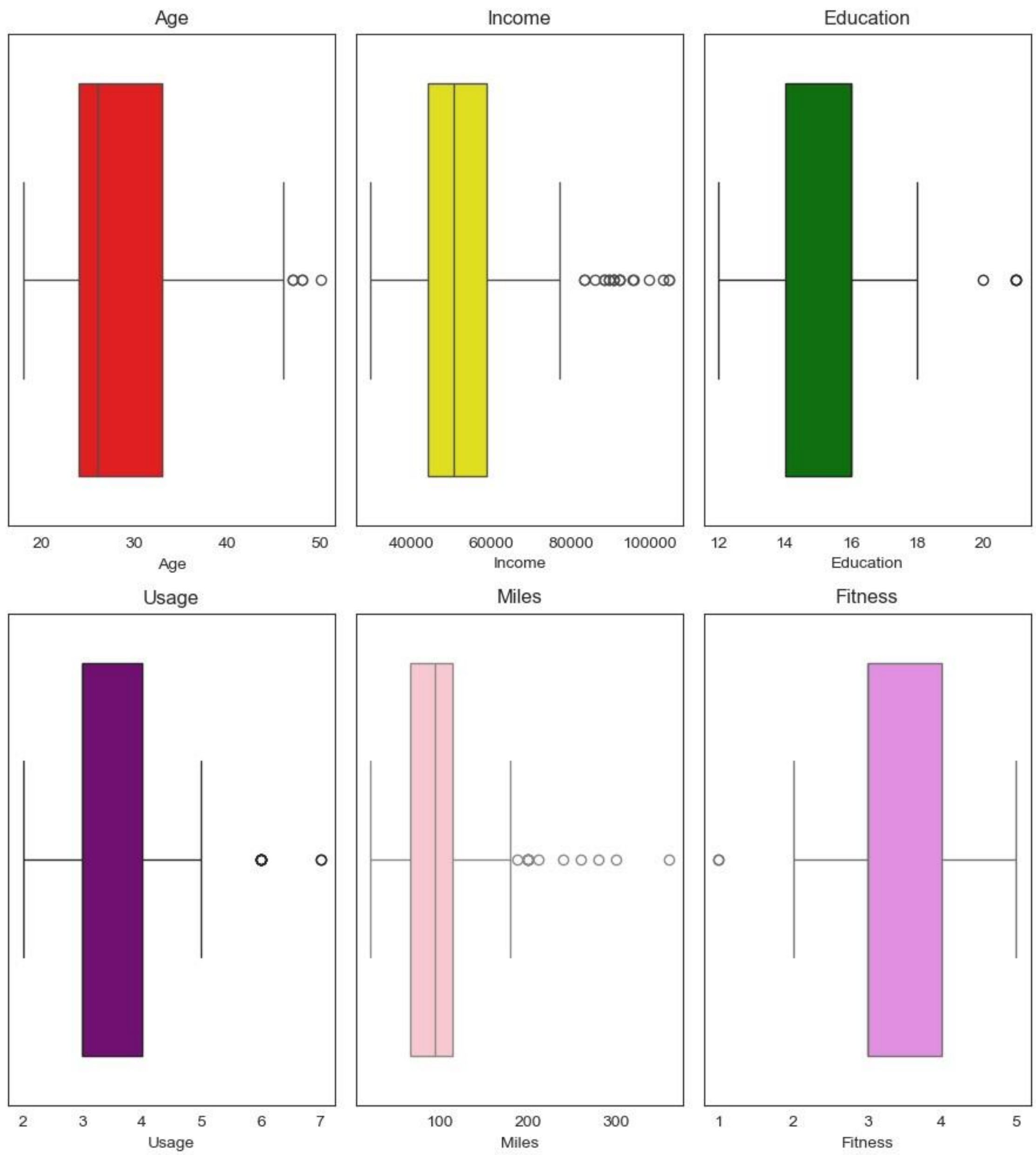
[8]:
```
#Checking the information present in the data

df_Aerofit.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
```

[55]:

```python
# Let us distribute our data evenly to see how data is distributed amongst␣
↪different factors using a boxplot
# Outliers detec on using BoxPlots

plt.figure(figsize = (9,10))
plt.subplot(2,3,1)
sns.boxplot(x = 'Age',data = df_Aerofit,color = 'red', orient='h')
plt.title('Age')

plt.subplot(2,3,3)
sns.boxplot(x = 'Education',data = df_Aerofit,color = 'green', orient='h')
plt.title('Education')

plt.subplot(2,3,4)
sns.boxplot(x = 'Usage',data = df_Aerofit,color = 'purple', orient='h')
plt.title('Usage')

plt.subplot(2,3,6)
sns.boxplot(x = 'Fitness',data = df_Aerofit,color = 'violet', orient='h')
plt.title('Fitness')

plt.subplot(2,3,2)
sns.boxplot(x = 'Income',data = df_Aerofit,color = 'yellow', orient='h')
plt.title('Income')

plt.subplot(2,3,5)
sns.boxplot(x = 'Miles',data = df_Aerofit,color = 'pink', orient='h')
plt.title('Miles')

plt.tight_layout()
```



As per the above code we can know how the data is evenly distributed amongst the data We will accordingly breif understandings in the recommendations section. Observations: Even from the boxplots it is quite clear that: • Age, Education and Usage are having very few outliers. • While Income and Miles are having more outlirs.

[9]:
```
df_Aerofit.describe(include="all")
```

[9]:
```
 Product  Age  Gender  Education  MaritalStatus  Usage  \  count  180
180.000000 180 180.000000 180 180.000000 unique 3 NaN 2 NaN 2 NaN
top    KP281        NaN  Male       NaN   Partnered       NaN
```

```
freq       80        NaN   104        NaN         107        NaN
mean      NaN  28.788889   NaN  15.572222         NaN   3.455556
std       NaN   6.943498   NaN   1.617055         NaN   1.084797
min       NaN  18.000000   NaN  12.000000         NaN   2.000000
25%       NaN  24.000000   NaN  14.000000         NaN   3.000000
50%       NaN  26.000000   NaN  16.000000         NaN   3.000000
75%       NaN  33.000000   NaN  16.000000         NaN   4.000000
max       NaN  50.000000   NaN  21.000000         NaN   7.000000

           Fitness         Income      Miles
count   180.000000   180.000000 180.000000

unique         NaN            NaN        NaN

top            NaN            NaN        NaN

freq           NaN            NaN        NaN

mean      3.311111  53719.577778 103.194444

std       0.958869 16506.684226  51.863605

min       1.000000 29562.000000  21.000000

25%       3.000000 44058.750000  66.000000

50%       3.000000 50596.500000  94.000000

75%       4.000000  58668.000000 114.750000

max       5.000000  104581.000000 360.000000
```

Observations:• There are no missing values in the data. • There are 3 unique products in the dataset. • KP281 is the most frequent product. • Minimum & Maximum age of the person is 18 & 50, mean is 28.79 and 75% of persons have age less than or equal to 33. • Most of the people are having 16 years of education i.e., 75% of persons are having education <= 16 years. • Out of 180 data points, 104's gender is Male and rest are the female. • Standard deviation for Income & Miles is very high. These variables might have the outliers

[10]:
```python
# Seeing the first 5 rows

df_Aerofit.head()
```

[10]:

	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

[11]:
```python
# Checking how many count values are there in our column
df_Aerofit.count()
```

[11]:
```
Product        180
```

```
Age              180
Gender           180
Education        180
MaritalStatus    180
Usage            180
Fitness          180
Income           180
Miles            180
dtype:
int64
```

[12]:
```
#Shows the columns in our data set
df_Aerofit.columns
```

[12]:
```
Index(['Product', 'Age', 'Gender', 'Education', 'MaritalStatus',
'Usage',
       'Fitness', 'Income', 'Miles'],
      dtype='object')
```

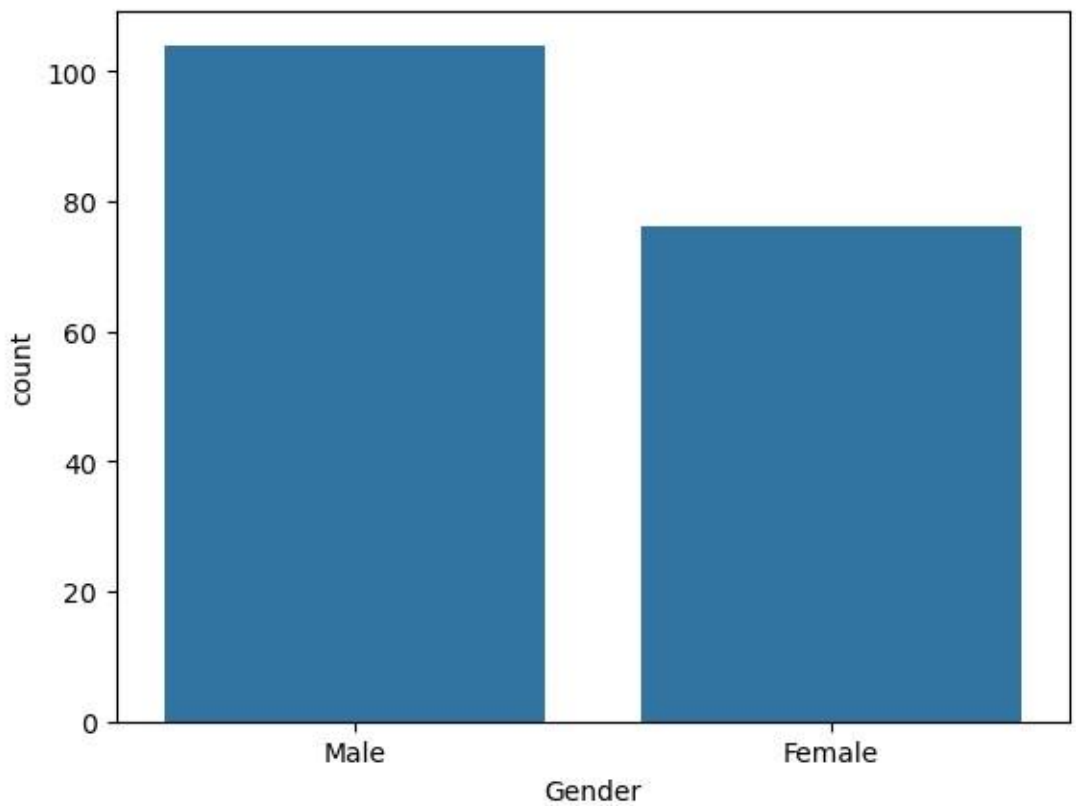
[13]:
```
# Showing the number of unique values
df_Aerofit.nunique()
```

[13]:
```
Product           3
Age              32
Gender            2
Education         8
MaritalStatus     2
Usage             6
Fitness           5
Income           62
Miles            37
dtype: int64
```

This shows the number of unique values of the product mentioned.

[14]:
```
# Let us understand what product is recommended for the male and the

female sns.countplot(x = 'Gender',data = df_Aerofit)
```

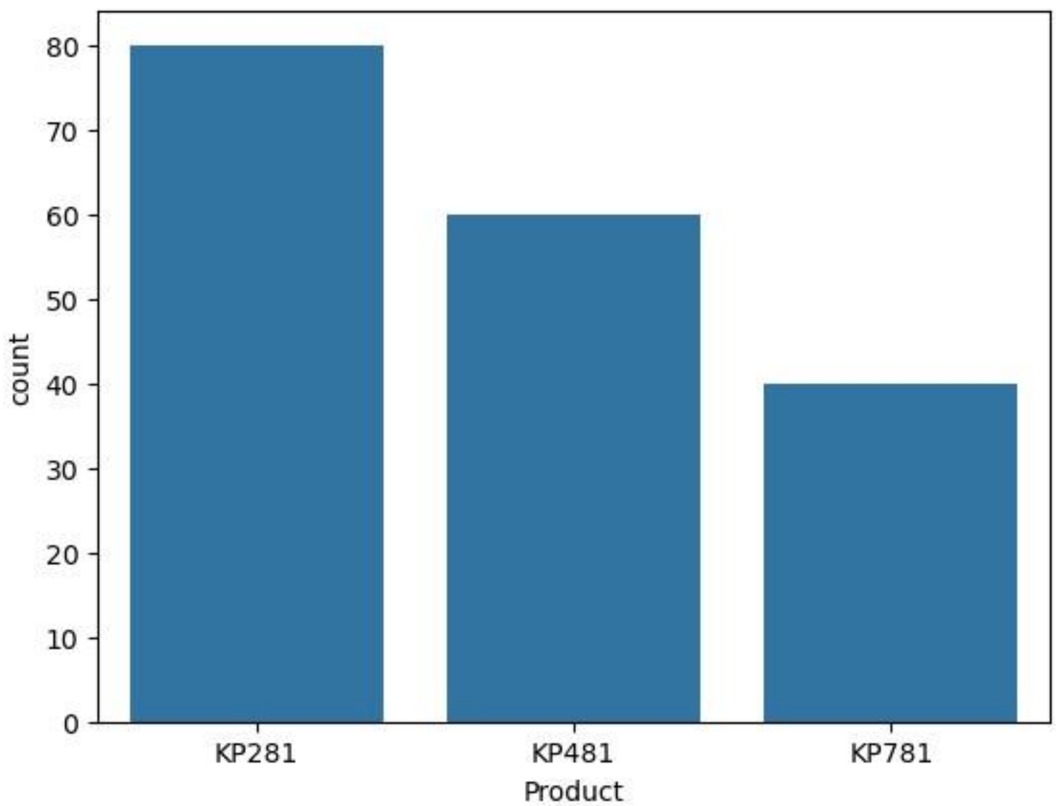
[14]: <Axes: xlabel='Gender', ylabel='count'>



We conclude that the males are the one who are buying out more products then females

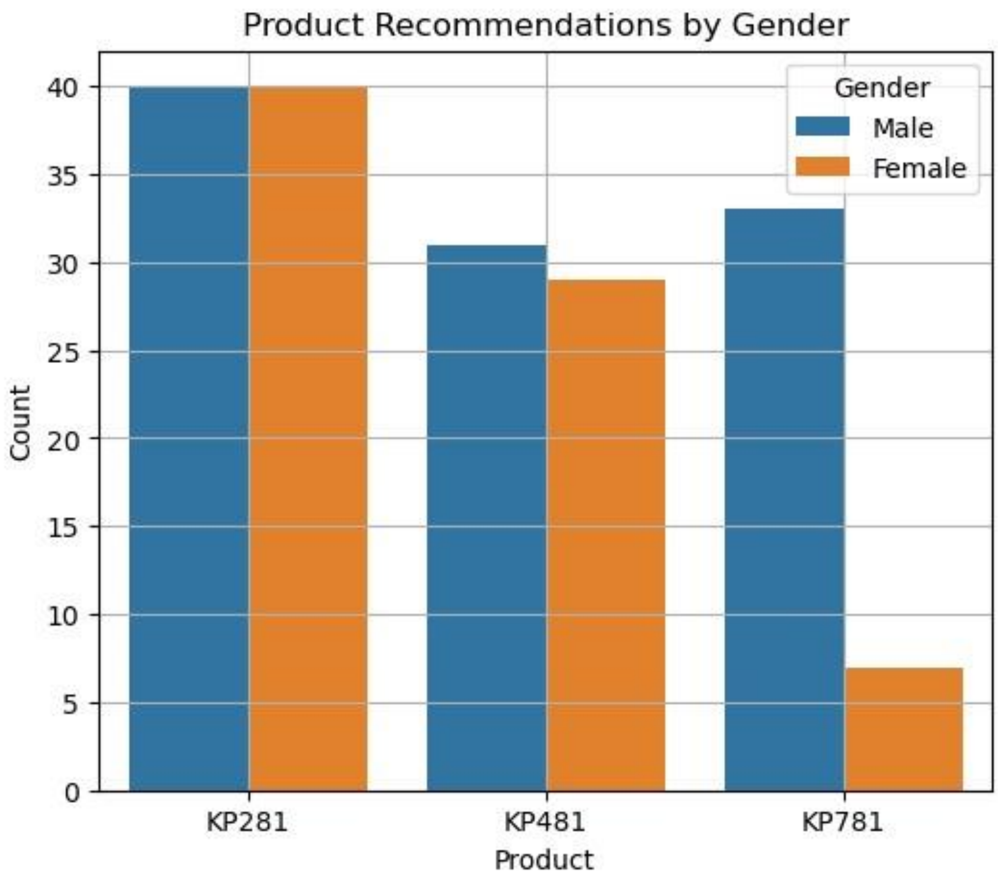
[15]:
```
sns.countplot(x = 'Product',data = df_Aerofit)
```

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




```
[ ]: Thus we can conclude that amongst male  and female the most recommended product
     ↪is KP281
```

```
[16]: # We can further more combine these two graphs for more valuable insights

plt.figure(figsize=(6, 5))
sns.countplot(x='Product', hue='Gender', data=df_Aerofit)

plt.title('Product Recommendations by Gender ')
plt.xlabel('Product')
plt.ylabel('Count')
plt.legend(title='Gender')
plt.grid()
plt.show()
```

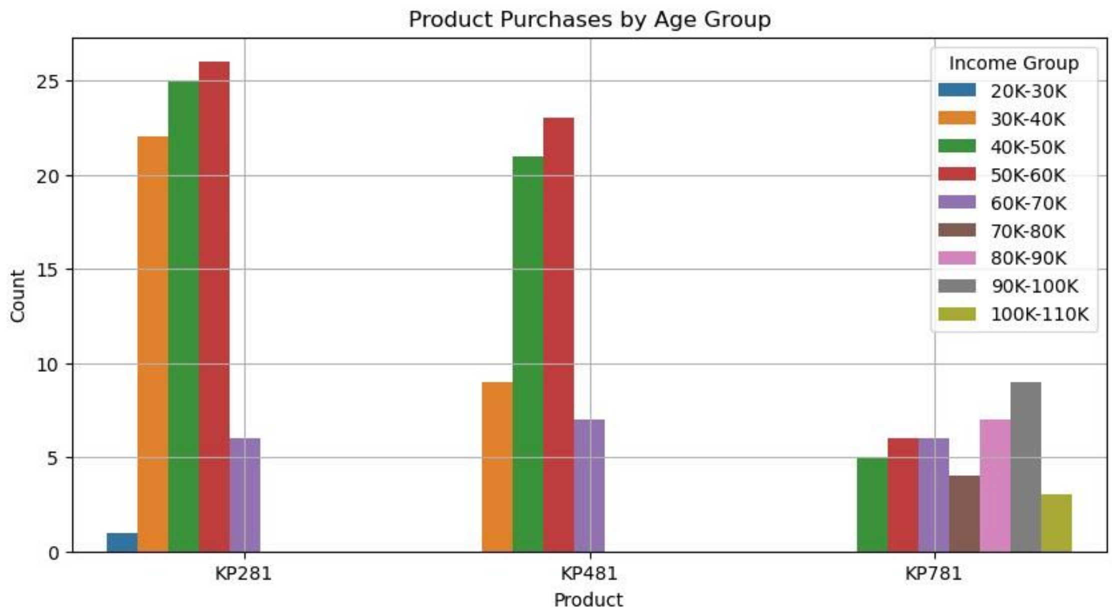


From the above graphs we conclude that KP281 are bought by equal number of men and women. But we should also appreciate the high end product KP781 is also famous amongst most of the males The performance of KP481 is also favourable amongst most men and womn

```
[17]: # # Let us further navigate what income groups are buying the most
products
```

```
df_Aerofit['IncomeGroup'] = pd.cut(df_Aerofit['Income'],
                        bins=[20000, 30000, 40000, 50000, 60000,
                        70000,␣
 ↪80000, 90000, 100000, 110000],
                        labels=['20K-30K', '30K-40K', '40K-50K',
'50K-60K',␣ ↪'60K-70K', '70K-80K', '80K-90K', '90K-100K', '100K-
110K']) plt.figure(figsize = (10,5)) sns.countplot(x='Product',
hue='IncomeGroup', data=df_Aerofit)
plt.title('Product Purchases by Age Group')
plt.xlabel('Product')
plt.ylabel('Count')
plt.legend(title='Income
```

```
Group') plt.grid()
plt.show()
```



Key factors to consider here income between 40K - 50k & 50K - 60k are the income groups to buy KP281 However the people who have income more then 80 are directly opting for KP781 Infact most of the people whose income is 80k to 90k are buying KP81

3 Univariant Analysis

4 Understanding the distribution of the data for the quantive atributes:

5 1. Age

6 2. Education

7 3. Usage

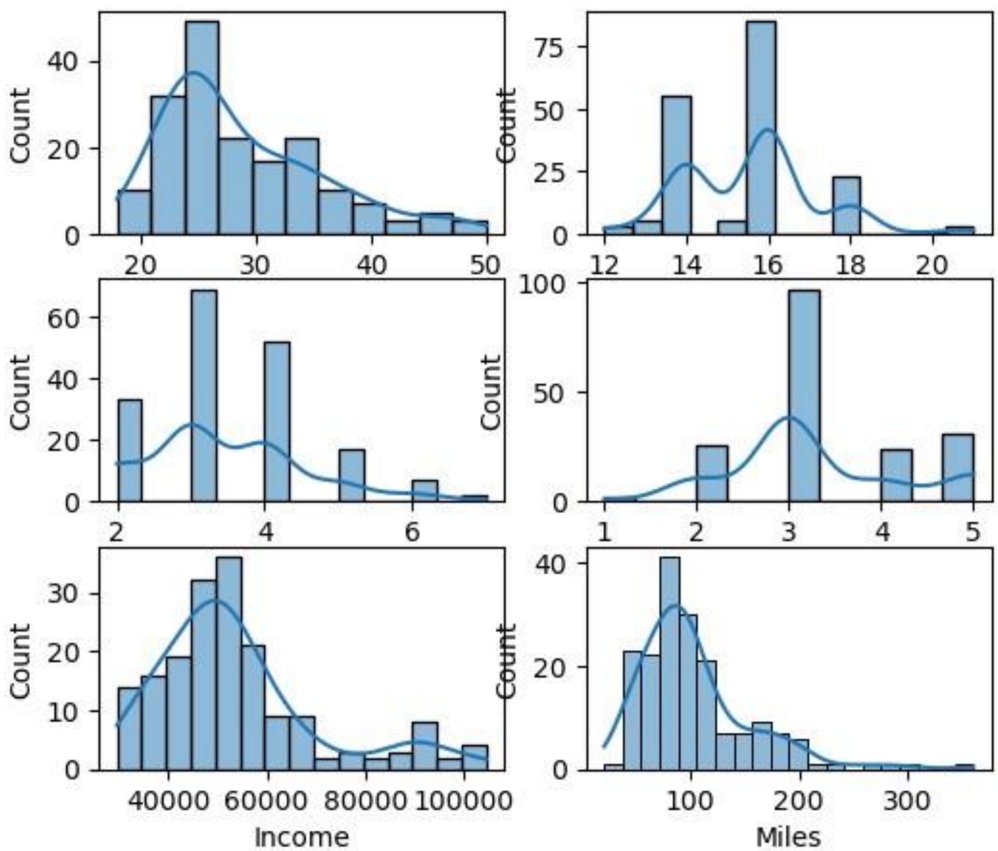
8 4. Fitness

9 5. Income

10 6. Miles

[18]:
```
fig, axis = plt.subplots(nrows=3, ncols=2, figsize=(6, 5))

fig.subplots_adjust(top=0.9) # Adjust the figure layout

sns.histplot(data=df_Aerofit, x="Age", kde=True, ax=axis[0, 0])
```

```
sns.histplot(data=df_Aerofit, x="Education", kde=True, ax=axis[0, 1])
sns.histplot(data=df_Aerofit, x="Usage", kde=True, ax=axis[1, 0])
sns.histplot(data=df_Aerofit, x="Fitness", kde=True, ax=axis[1, 1])
sns.histplot(data=df_Aerofit, x="Income", kde=True, ax=axis[2, 0])
sns.histplot(data=df_Aerofit, x="Miles", kde=True, ax=axis[2, 1])

plt.show()
```



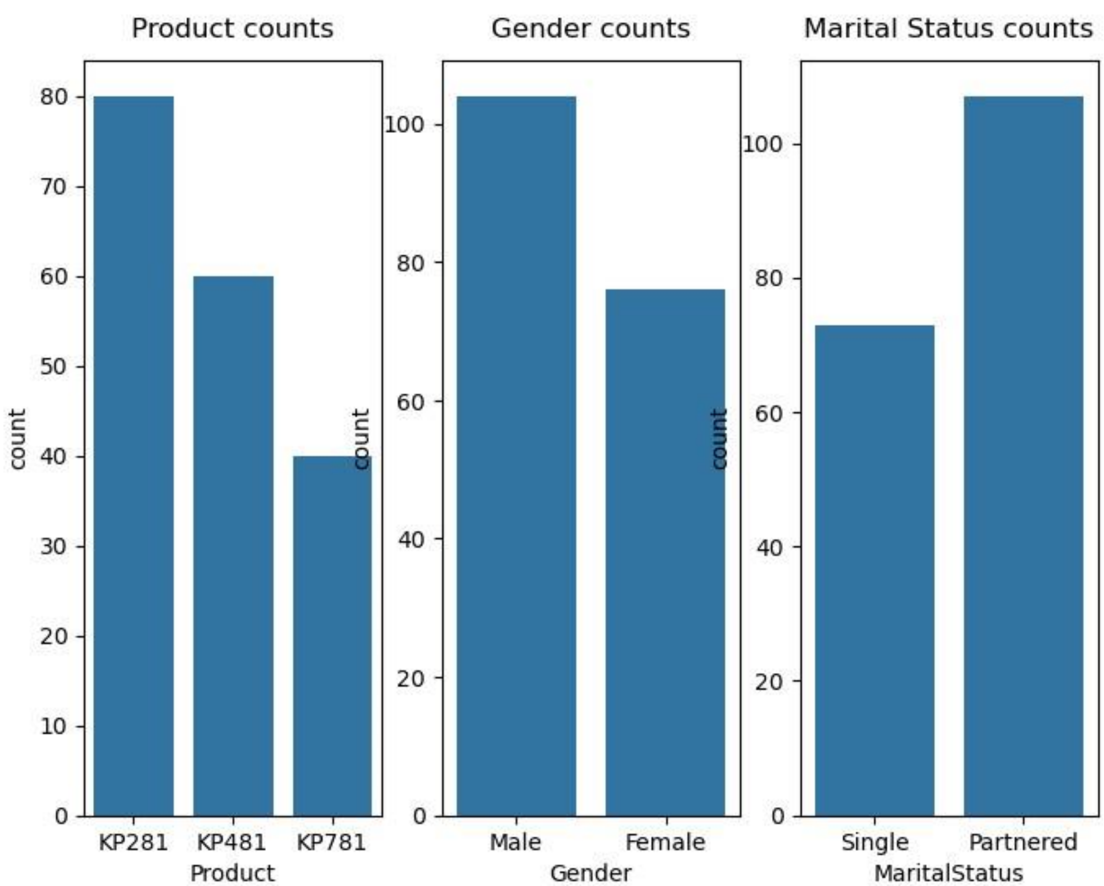
11 Understanding the distribution of the data for the qualitative attributes:

12 1. Product

13 2. Gender

14 3. MaritalStatus

[19]:
```
fig, axs = plt.subplots (nrows=1, ncols=3, figsize=(8,6))


sns.countplot (data=df_Aerofit, x='Product', ax=axs[0])
sns.countplot(data=df_Aerofit, x='Gender', ax=axs[1])
sns.countplot(data=df_Aerofit, x='MaritalStatus', ax=axs [2])

axs[0].set_title("Product counts", pad=10, fontsize=12)
axs [1].set_title("Gender counts", pad=10, fontsize=12)
axs [2].set_title("Marital Status counts", pad=10, fontsize=12)
plt.show()
```



Observations for the above • KP281 is the most frequent product. • There are more Males in the data than Females. • More Partnered persons are there in the daa.

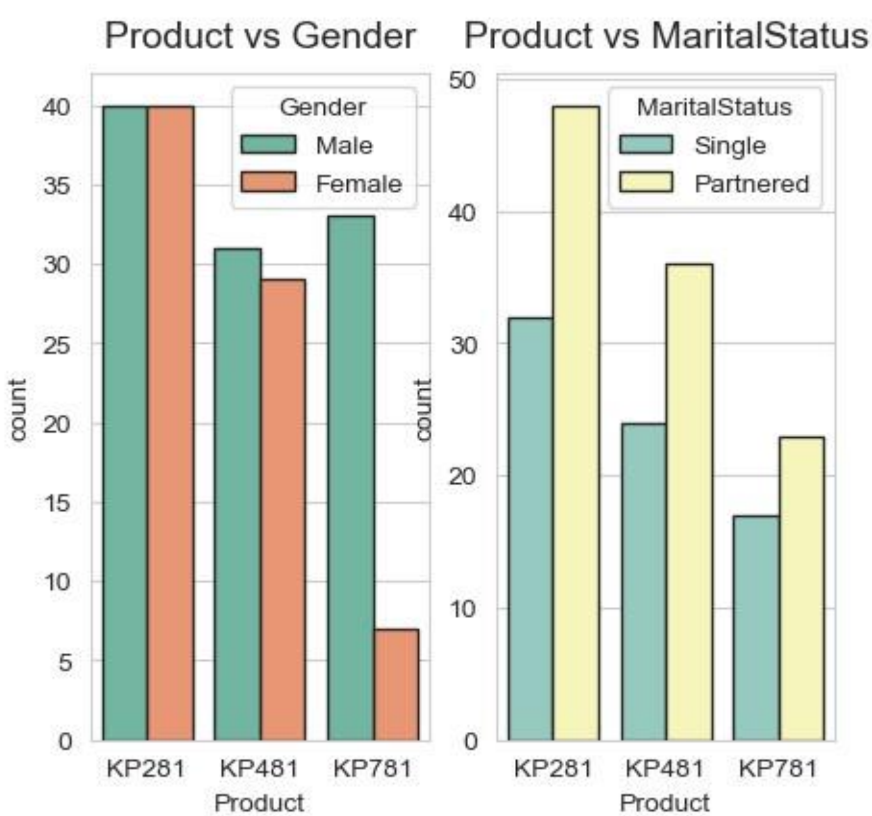
[20]:
```
# To be precise - normalized count for each variable is shown below:
df1_Aerofit = df_Aerofit[['Product', 'Gender',
'MaritalStatus']].melt() df1_Aerofit.groupby(['variable',
'value'])[['value']].count() / len(df_Aerofit)
```

[20]:
```
value variable   value
Gender        Female    0.422222
              Male      0.577778
```

```
MaritalStatus Partnered 0.594444
              Single    0.405556

 Product      KP281     0.444444
              KP481     0.333333

              KP781     0.222222
```

Observations for the above • Product 44.44% of the customers have purchased KP2821 product. 33.33% of the customers have purchased KP481 product. 22.22% of the customers have purchased KP781 product. • Gender 57.78% of the customers are Male. • MaritalStatus 59.44% of the customers are Patnered.

[21]:
```
# Bivariant Analysis

sns.set_style(style='whitegrid')
fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(5, 4.5))
sns.countplot(data=df_Aerofit, x='Product', hue='Gender',
edgecolor="0.15", palette='Set2', ax=axs[0])
sns.countplot(data=df_Aerofit, x='Product', hue='MaritalStatus',
edgecolor="0.15", palette='Set3', ax=axs[1])
axs[0].set_title("Product vs Gender", pad=10, fontsize=14)
axs[1].set_title("Product vs MaritalStatus", pad=10,
fontsize=14) plt.show()
```

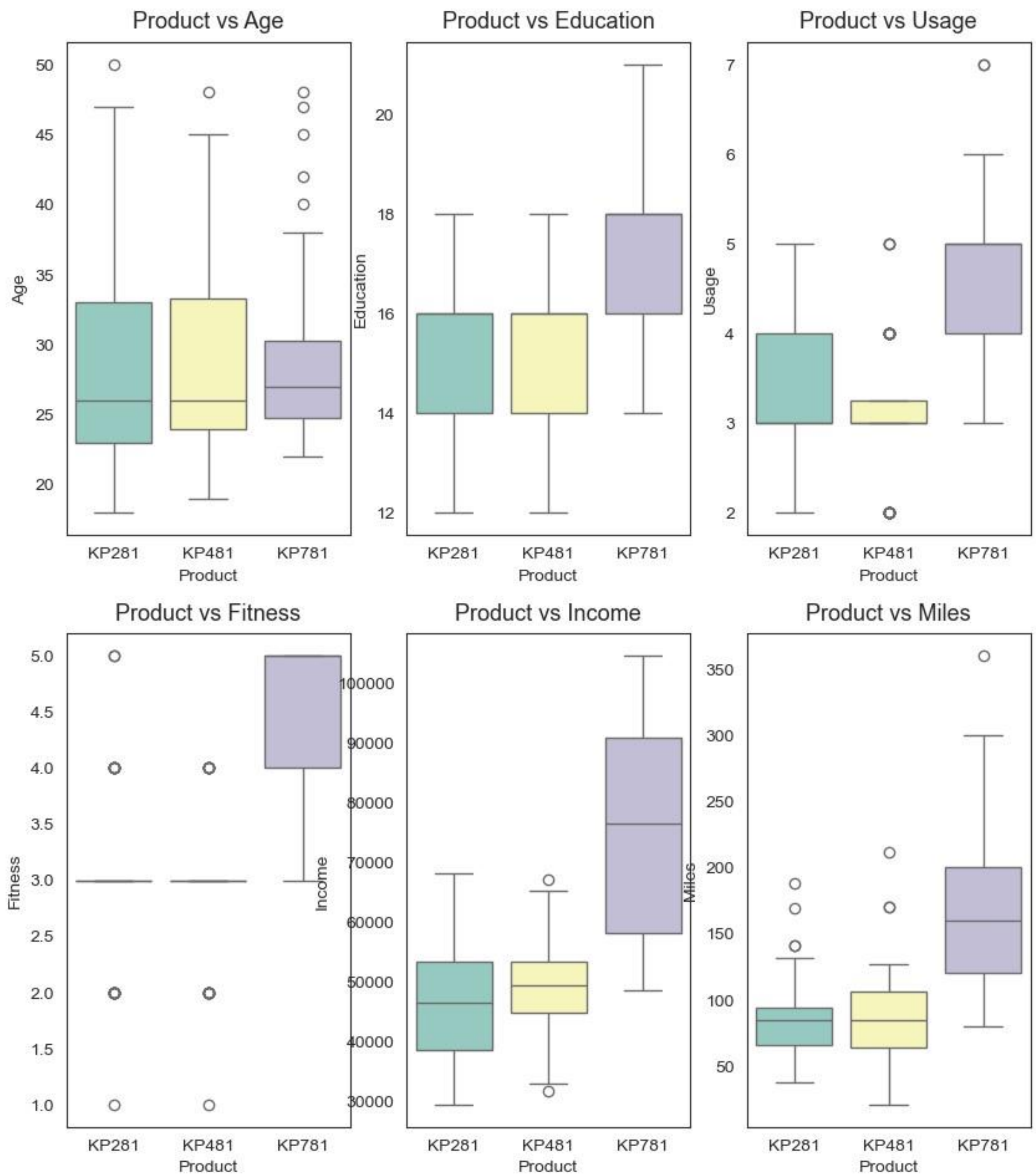


Observations • Product vs Gender 1) Equal number of males and females have purchased KP281 product and Almost same for the product KP481 2) Most of the Male customers have purchased the KP781 product. • Product vs MaritalStatus 1) Customer who is Partnered, is more likely to purchase the product.

[22]:
```
#Checking if following features have any effect on the product purchased:
# 1. Age
# 2. Education
# 3. Usage
# 4. Fitness
# 5. Income
# 6. Miles

attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']
sns.set_style("white")
fig, axs  = plt.subplots(nrows=2, ncols=3, figsize=(10,8))
fig.subplots_adjust(top=1.2)
count = 0


for i in range(2):
    for j in range(3):
        sns.boxplot(data=df_Aerofit, x='Product',  =attrs[count],
        ax=axs[i,j], palette='Set3')

        axs[i,j].set_title(f"Product vs  {attrs[count]}",
        pad=8, fontsize=13)
        count += 1
```

```
C:\Users\yjoth\AppData\Local\Temp\ipykernel_27420\4126019289.py:18:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=df_Aerofit, x='Product', y=attrs[count],
C:\Users\yjoth\AppData\Local\Temp\ipykernel_27420\4126019289.py:18:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=df_Aerofit, x='Product', y=attrs[count],
C:\Users\yjoth\AppData\Local\Temp\ipykernel_27420\4126019289.py:18:
FutureWarning:
```

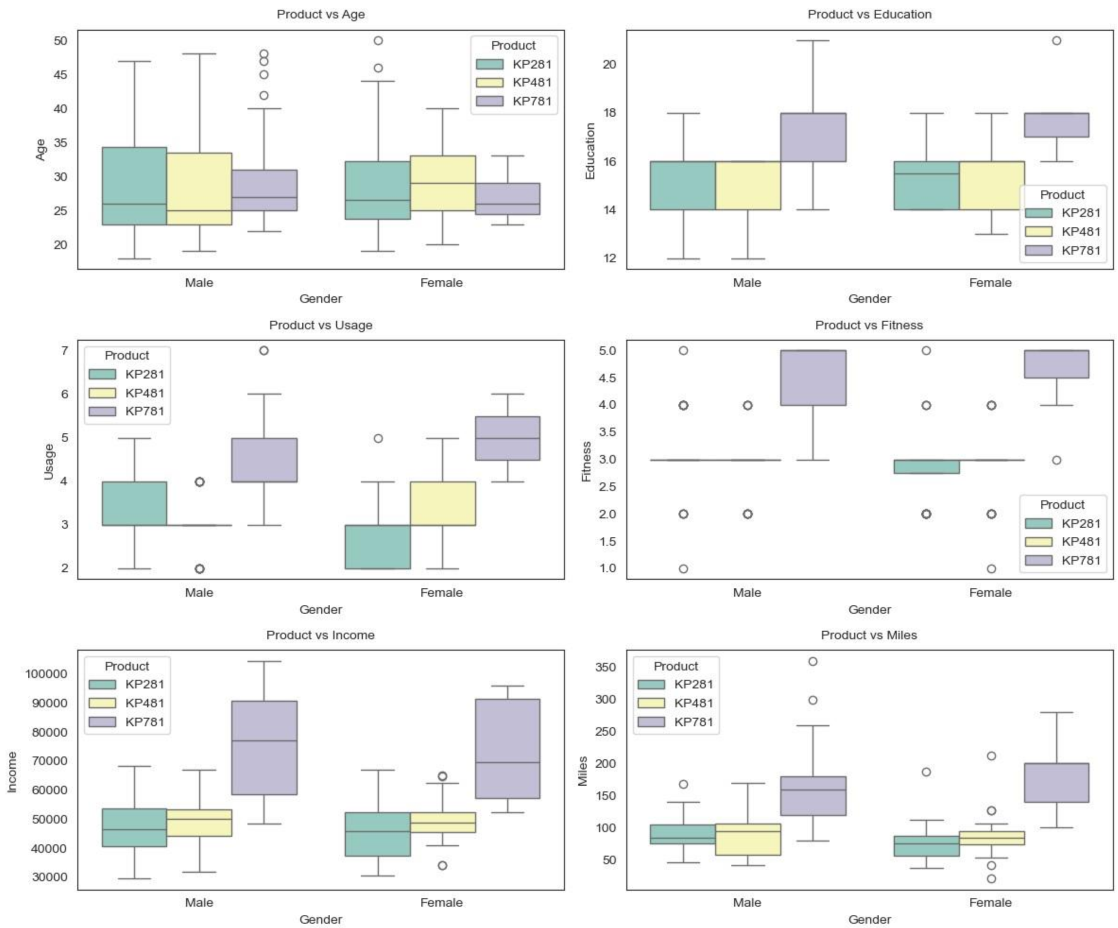
```
Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=df_Aerofit, x='Product', y=attrs[count],
C:\Users\yjoth\AppData\Local\Temp\ipykernel_27420\4126019289.py:18:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=df_Aerofit, x='Product', y=attrs[count],
C:\Users\yjoth\AppData\Local\Temp\ipykernel_27420\4126019289.py:18:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=df_Aerofit, x='Product', y=attrs[count],
  C:\Users\yjoth\AppData\Local\Temp\ipykernel_27420\4126019289.py:18:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=df_Aerofit, x='Product', y=attrs[count],
```



Observations: 1. Product vs Age • Customers purchasing products KP281 & KP481 are having same Age median value. • Customers whose age lies between 25-30, are more likely to buy KP781 product 1. Product vs Education • Customers whose Education is greater than 16, have more chances to purchase the KP781 product. • While the customers with Education less than 16 have equal chances of purchasing KP281 or KP481. 1. Product vs Usage • Customers who are planning to use the treadmill greater than 4 times a week, are more likely to purchase the KP781 product. • While the other customers are likely to purchasing KP281 or KP481. 1. Product vs Fitness •

The more the customer is fit (fitness >= 3), higher the chances of the customer to purchase the KP781 product. 1. Product vs Income • Higher the Income of the customer (Income >= 60000), higher the chances of the customer to purchase the KP781 product. 1. Product vs Miles • If the customer expects to walk/run greater than 120 Miles per week, it is more likely that the customer will buy KP781 product.

[23]:
```
# MultiVariate Analysis
# Attributes for boxplot comparison
attrs = ['Age', 'Education', 'Usage', 'Fitness', 'Income', 'Miles']

# Set style for the plot
sns.set_style("white")
# Create subplots, adjust the size for better
fitting fig, axs = plt.subplots(nrows=3, ncols=2,
figsize=(12, 10))

# Adjust the spacing between subplots to avoid overlap
plt.subplots_adjust(hspace=0.4, wspace=0.3)

# Initialize counter for iterating through
attributes count = 0 for i in range(3):
    for j in range(2):
        # Create a boxplot for each attribute and plot it in the
        appropriate␣
 ↪subplot sns.boxplot(data=df_Aerofit, x='Gender', y=attrs[count],
        hue='Product',␣
 ↪ax=axs[i,j], palette='Set3') axs[i,j].set_title(f"Product vs
        {attrs[count]}", pad=8, fontsize=10) count += 1
# Apply tight layout to make the plot fit well in the Jupyter notebook
plt.tight_layout()

# Save the plot as an image file for better resolution if you want to
take a␣
 ↪screenshot
plt.savefig('boxplots_output.png',
dpi=300)
# Show the plot
plt.show()
```



Observations • Females planning to use treadmill 3-4 times a week, are more likely to buy KP481 product

[24]:
```python
# Computing Marginal & Conditional Probabilities:
# Marginal Probability

df_Aerofit['Product'].value_counts(normalize=True)
```

[24]:
```
Product
KP281   0.444444
KP481   0.333333
KP781   0.222222
Name: proportion, dtype: float64
```

[25]:
```python
def p_prod_given_gender(gender, print_marginal=False):
    # Use '==' for string comparison if
    gender != "Female" and gender !=
    "Male":
        return "Invalid gender value."
```

```
    # Create the crosstab for products vs gender
    df1_Aerofit =
    pd.crosstab(index=df_Aerofit['Gender'],␣
↪columns=df_Aerofit['Product'])
    # Calculate probabilities for each product given
    the gender total_for_gender =
    df1_Aerofit.loc[gender].sum() p_781 =
    df1_Aerofit['KP781'][gender] / total_for_gender
    p_481 = df1_Aerofit['KP481'][gender] /
    total_for_gender p_281 =
    df1_Aerofit['KP281'][gender] / total_for_gender
    # Print marginal probabilities if
    requested if print_marginal:
        total_records = len(df_Aerofit) p_male =
    df1_Aerofit.loc['Male'].sum() / total_records
    p_female = df1_Aerofit.loc['Female'].sum() /
    total_records print(f"P(Male): {p_male:.2f}")
    print(f"P(Female): {p_female:.2f}\n") # Print
    conditional probabilities print(f"P(KP781/{gender}):
    {p_781:.2f}") print(f"P(KP481/{gender}):
    {p_481:.2f}") print(f"P(KP281/{gender}):
    {p_281:.2f}\n")
# Call the function
p_prod_given_gender('Male', True)
p_prod_given_gender('Female')
```

```
P(Male): 0.58
P(Female): 0.42

P(KP781/Male): 0.32
P(KP481/Male): 0.30
P(KP281/Male): 0.38

P(KP781/Female): 0.09
P(KP481/Female): 0.38
P(KP281/Female): 0.53
```

15 Actionable insights and Recommendations

Key Insights and Strategic Recommendations: Product Recommendations by Gender:

The KP281 treadmill has emerged as the most popular product, followed closely by the KP481 and KP781 models. Both male and female customers prefer the KP281 due to its cost-effectiveness compared to the other two models. This suggests that future marketing efforts should emphasize the value proposition of the KP281 while also promoting the KP481 and KP781 as premium alternatives.

Product Purchase by Age Group:

The majority of KP281 purchases are made by customers aged 20 to 30. This indicates a strong preference among younger adults for this model, likely due to its affordability and features. Customers aged 10 to 20 are less likely to purchase fitness equipment, as they are typically focused on school and college. However, this demographic represents a potential future market as they age and enter the workforce. Offering discounts and targeted advertisements could increase brand loyalty and future sales within this group. For customers aged 30 to 40, all product varieties can be marketed effectively. This group may prioritize product specifications over price, making them a good target for upselling higher-end models. The 40 to 50 age group shows a strong preference for the KP781 and KP281 models. However, it is important to note that customers over 50 tend to avoid Aerofit products. To capture this market, introducing new designs tailored to the needs of older adults could increase interest and sales.

Product Purchase by Marital Status:

Non-single customers are more likely to purchase fitness equipment. To increase sales among single customers, consider offering attractive discount coupons or easy EMI (Equated Monthly Installment) options, making it easier for them to purchase high-quality products. The 40 to 45 age group is more likely to run longer distances compared to the 25 to 30 age group. This trend may be attributed to weight gain in the older group, while younger individuals are more focused on building muscle mass rather than weight loss.

Education vs. Product Preferences:

Customers with 20 years of experience are more likely to purchase the KP781, while those with less experience tend to choose the KP281 and KP481 models. This suggests that more experienced individuals might be inclined towards premium products, whereas less experienced customers prefer more affordable options.

Purchase Probabilities:

The probability of a customer being male is 58%. Given that a customer is male, the probability of purchasing a KP781 treadmill increases to 32%.