AerofitBusinessCasestudy:

The market research team at AeroFit wants to identify the characteristics of the target audience for each type of treadmill offered by the company.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

Data Loading and analysing basic metrics

```
aerofit = pd.read_csv("/content/aerofit_treadmill.csv")

aerofit.columns

aerofit.shape

aerofit.info()

aerofit.sample(5)

aerofit.isnull().sum()

plt.figure(figsize=(24,5))

sns.heatmap(aerofit.isnull(),cmap='Blues')
plt.title('Visual Check of Nulls',fontsize=20)
plt.show()
```

1. Descriptive Statistics Summary

```
aerofit.describe()
```

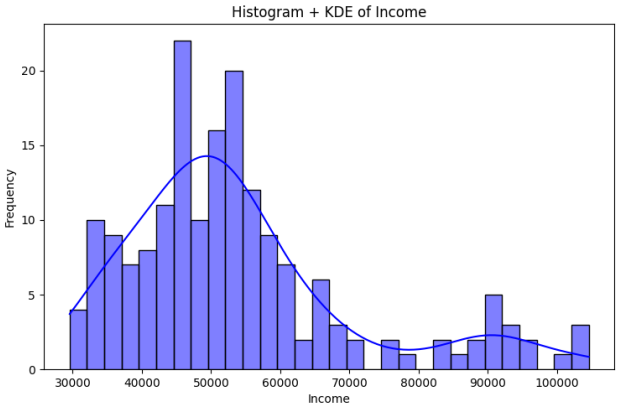

```
{"summary":"{\n  \"name\": \"aerofit\",\n  \"rows\": 8,\n  \"fields\":
[\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 55.58832332198464,\n
\"min\": 6.943498135399795,\n        \"max\": 180.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
28.788888888888888,\n          26.0,\n          180.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Education\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
59.04362112875324,\n        \"min\": 1.6170548978065553,\n
\"max\": 180.0,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          180.0,\n          15.572222222222223,\n
16.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

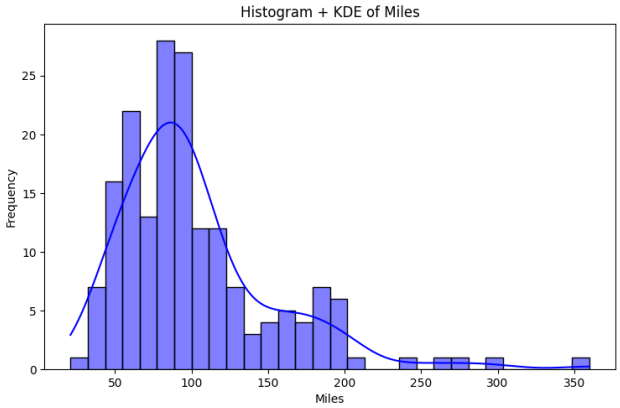
```
\"Usage\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 62.474604277313155,\n        \"min\": 1.0847970343962445,\n
\"max\": 180.0,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          180.0,\n          3.4555555555555557,\n
4.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Fitness\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 62.63086276036247,\n        \"min\":
0.958868565619312,\n        \"max\": 180.0,\n
\"num_unique_values\": 7,\n        \"samples\": [\n          180.0,\n
3.3111111111111111,\n          4.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Income\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
31403.855763201762,\n        \"min\": 180.0,\n        \"max\":
104581.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
53719.57777777778,\n          50596.5,\n          180.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Miles\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 106.52090041797726,\
n        \"min\": 21.0,\n        \"max\": 360.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
103.19444444444444,\n          94.0,\n          180.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

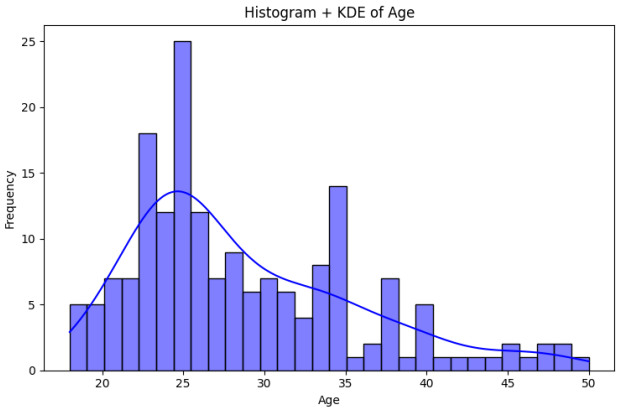

Distribution Analysis:

```
import math
num_col=['Income','Miles','Age']
# for col in list(aerofit.columns):
#   if aerofit[col].dtype == 'int64' or aerofit[col].dtype
=='float64':
#     num_col.append(col)
n=len(num_col)

cols = 2 if n > 1 else 1
rows = math.ceil(n / cols)


fig, axes = plt.subplots(rows, cols, figsize=(15, 5 * rows))
axes = axes.flatten() if n > 1 else [axes]

for i, col in enumerate(num_col):
    sns.histplot(aerofit[col], kde=True, ax=axes[i], bins=30,
color='blue', edgecolor='black')
    axes[i].set_title(f'Histogram + KDE of {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Frequency')
```

```
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()
```



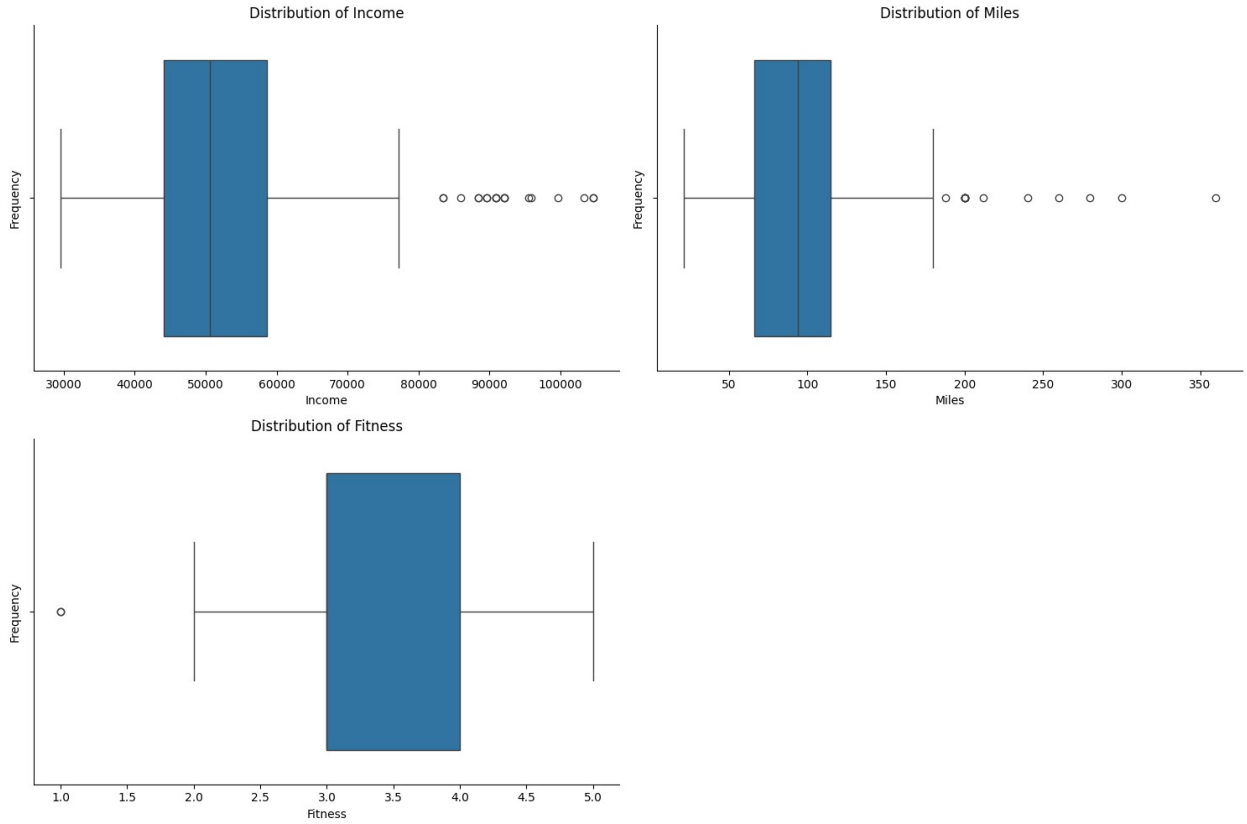




Checking for outliers

```
num_col=['Income','Miles','Fitness']
# for col in list(aerofit.columns):
#   if aerofit[col].dtype == 'int64' or aerofit[col].dtype
=='float64':
#     num_col.append(col)
n=len(num_col)

cols = 2 if n > 1 else 1
rows = math.ceil(n / cols)


fig, axes = plt.subplots(rows, cols, figsize=(15, 5 * rows))
axes = axes.flatten() if n > 1 else [axes]

for i, col in enumerate(num_col):
```

```
    sns.boxplot(x=aerofit[col],data=aerofit, ax=axes[i])
    axes[i].set_title(f'Distribution of {col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Frequency')


for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

sns.despine()
plt.tight_layout()
plt.show()
```



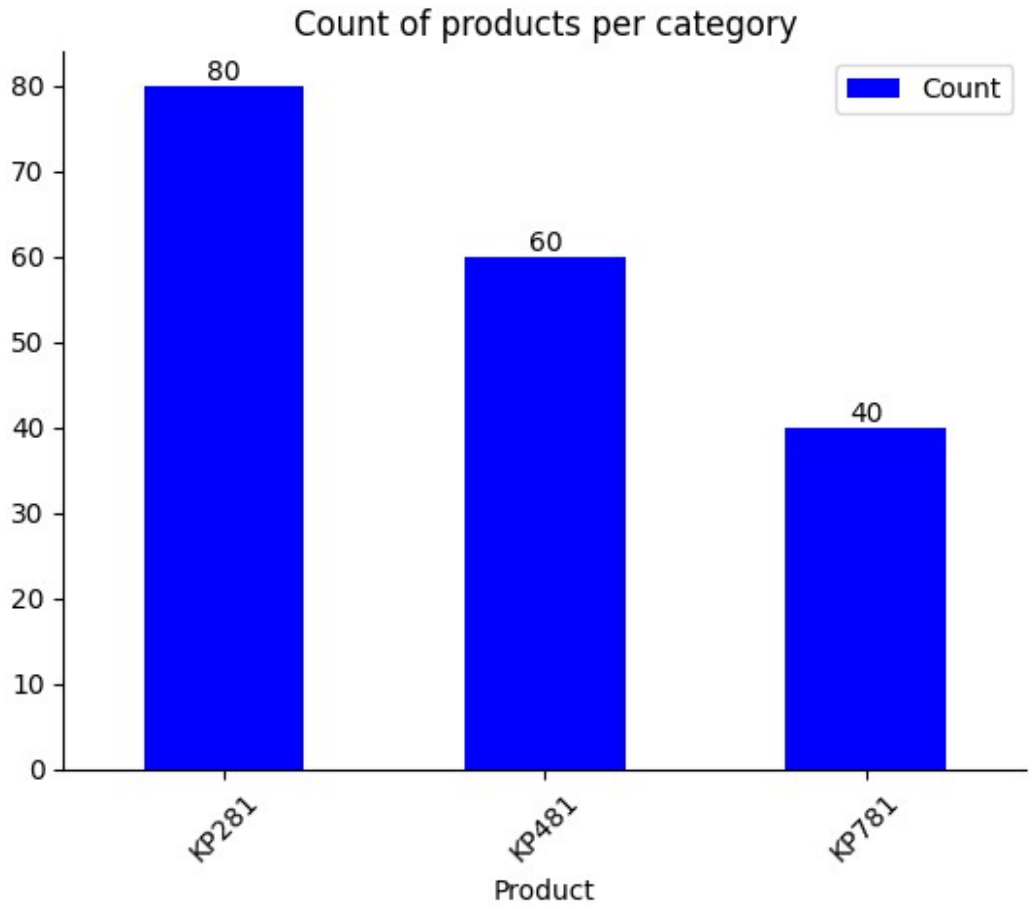
Univariant Analysis:

1. Product: Countplot of each treadmill type

```
plt.figure(figsize=(10,8))
product_counts = aerofit['Product'].value_counts().reset_index()
product_counts.columns = ['Product', 'Count']
b=product_counts.plot(kind='bar', x='Product', y='Count',color='blue')

for container in b.containers:
  b.bar_label(container,fmt='%d')
plt.title('Count of products per category')
```

```
sns.despine()
plt.xticks(rotation=45)
plt.show()
```

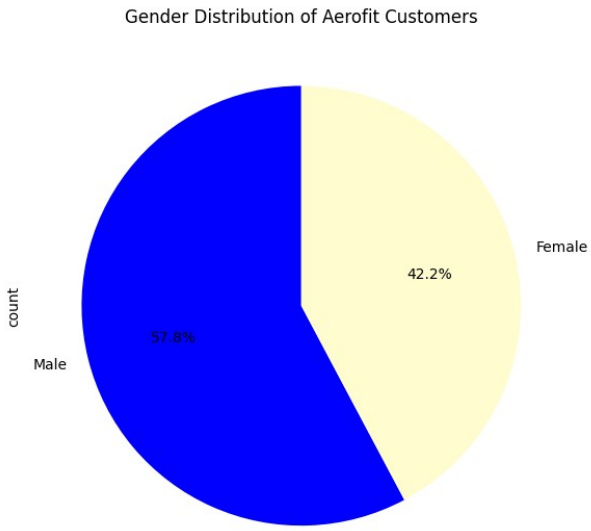
```
<Figure size 1000x800 with 0 Axes>
```

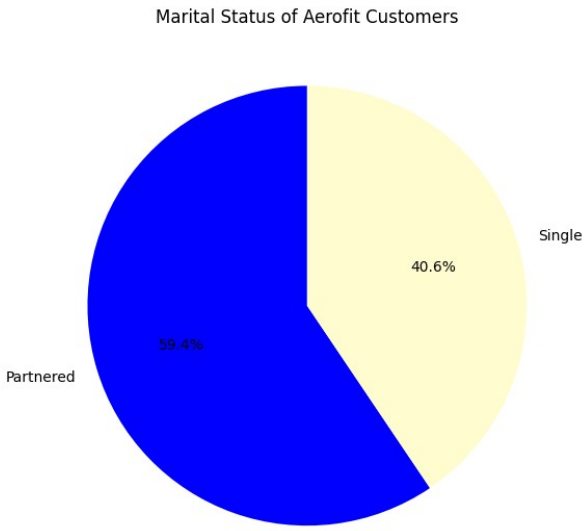


1. Gender Ratio and Marital Status of Aerofit Customers

```
gender_counts = aerofit['Gender'].value_counts()
MaritalStatus = aerofit['MaritalStatus'].value_counts()
plt.figure(figsize=(15,8))

plt.subplot(1,2,1)
gender_counts.plot(kind='pie', autopct='%1.1f%%', startangle=90,
colors=['blue', '#fffdd0'])
plt.title('Gender Distribution of Aerofit Customers')

plt.subplot(1,2,2)
MaritalStatus.plot(kind='pie', autopct='%1.1f%%', startangle=90,
colors=['blue', '#fffdd0'])
plt.title('Marital Status of Aerofit Customers')
```

```
plt.ylabel('')
plt.show()
```



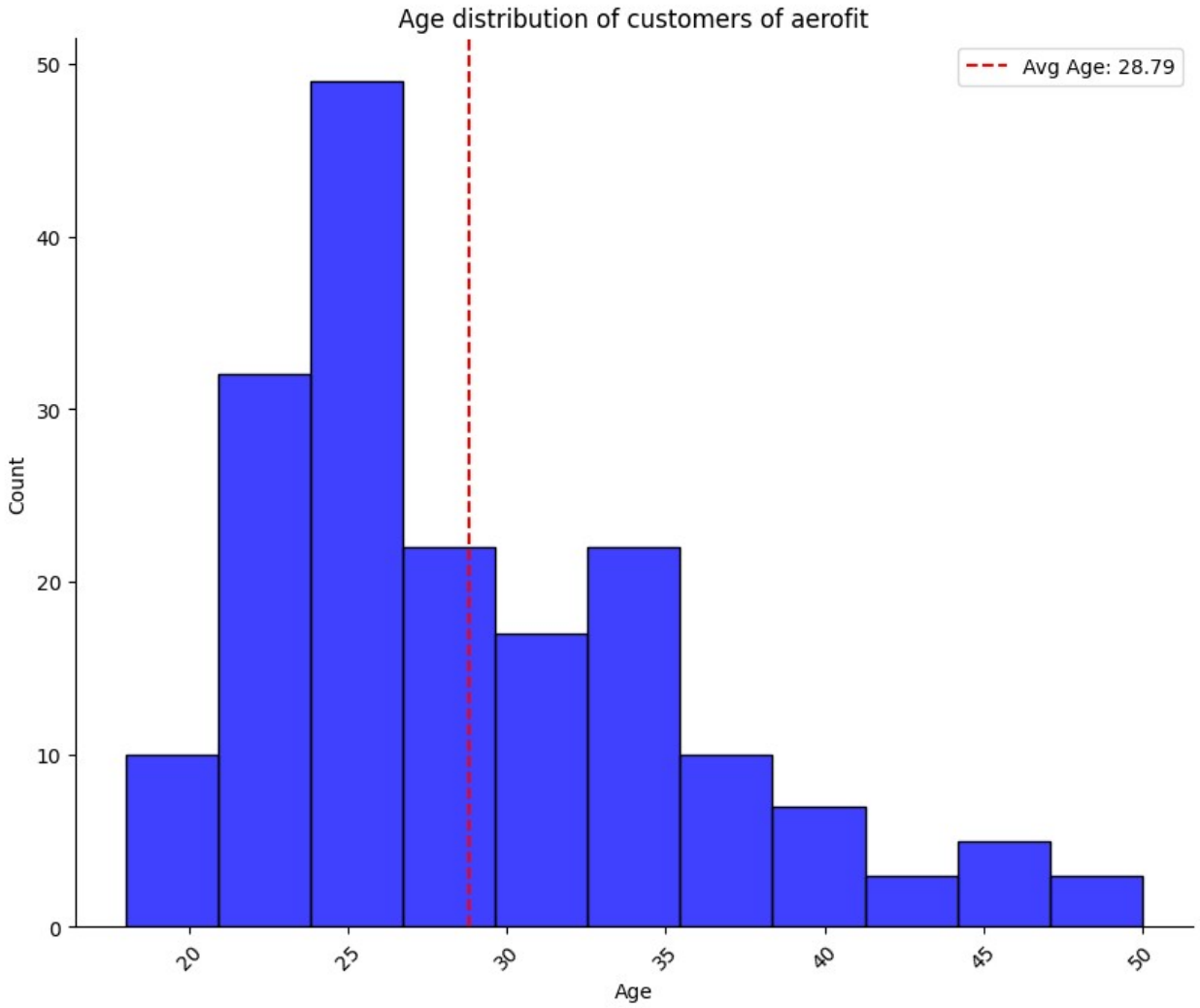



```
mean_age = aerofit.Age.mean()
mean_age
```

```
np.float64(28.788888888888888)
```

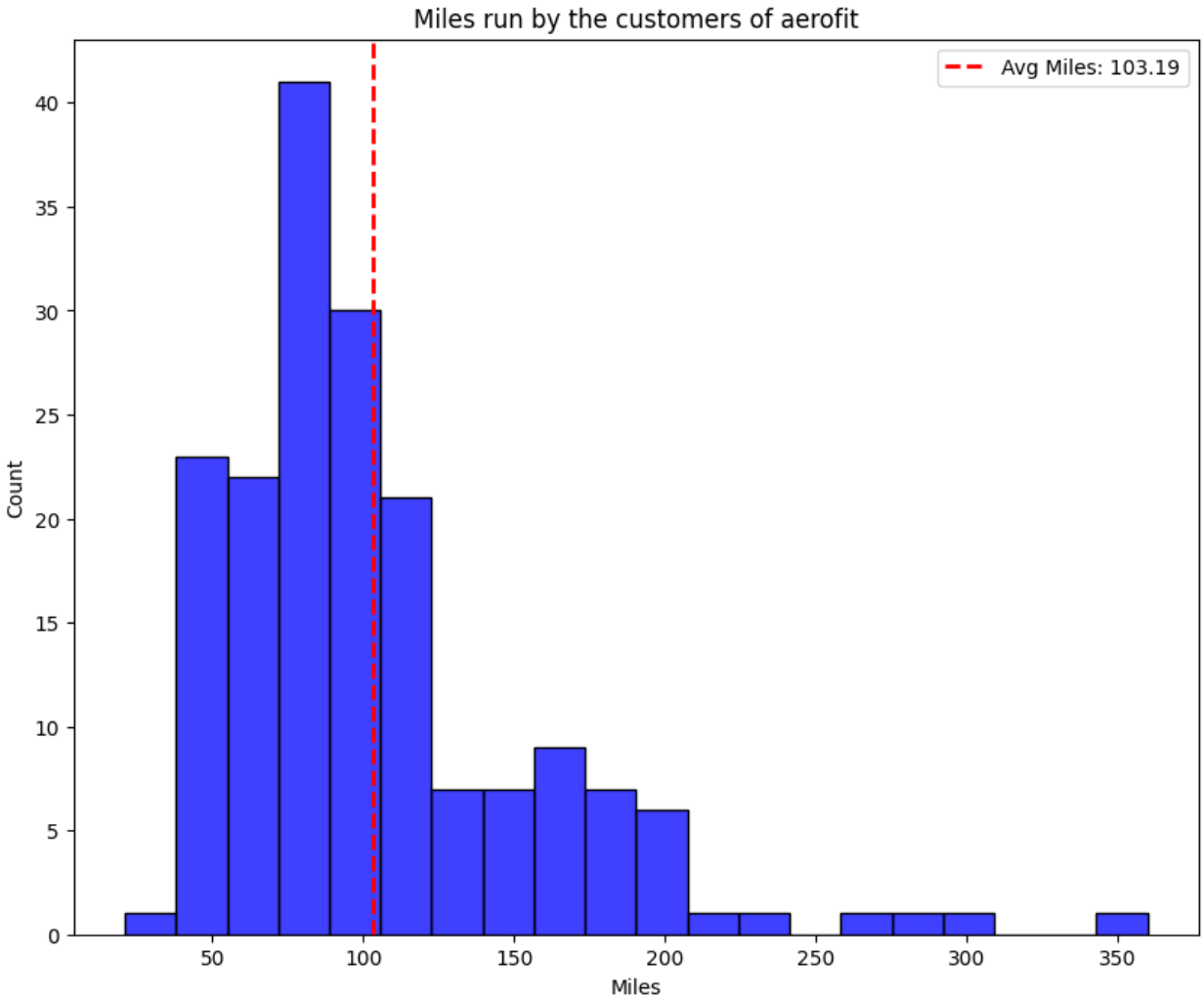
1. Age distribution of customers of aerofit

```
plt.figure(figsize=(10,8))
sns.histplot(data= aerofit, x= "Age",color ='blue')

plt.axvline(mean_age, color = "red",ls = "--",label=f'Avg Age:
{mean_age:.2f}')
plt.xticks(rotation=45)
plt.title("Age distribution of customers of aerofit")
sns.despine()
plt.legend()
plt.show()
```



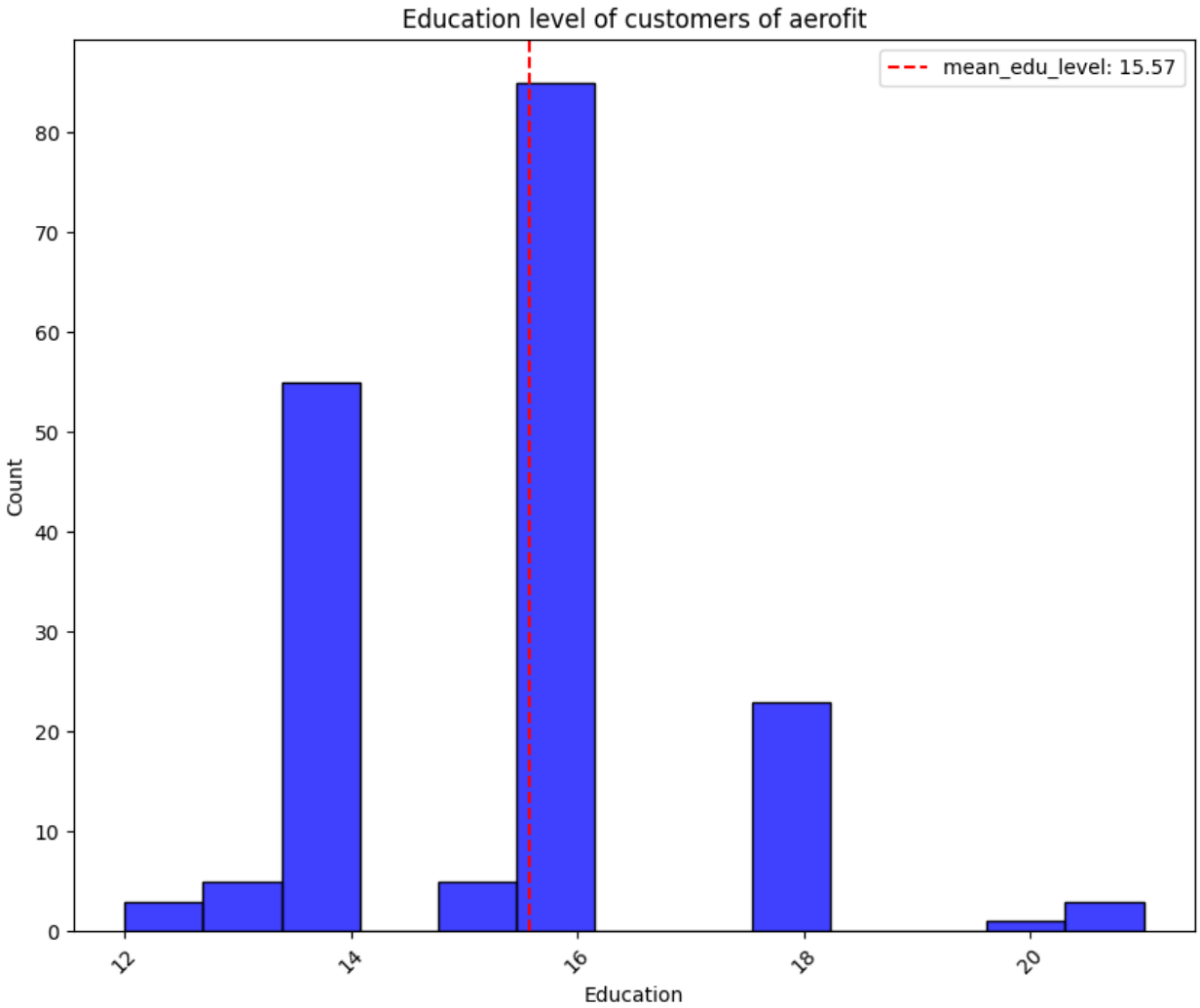
4.Miles run by the customers of aerofit

```
avg_miles= aerofit.Miles.mean()

plt.figure(figsize=(10,8))
sns.histplot(data= aerofit,x = 'Miles',bins='auto',color='blue')
plt.axvline(avg_miles, color='red', linestyle='--', linewidth=2,
label=f'Avg Miles: {avg_miles:.2f}')

plt.title("Miles run by the customers of aerofit")
plt.legend()
plt.show()
```



1. Education level of Aerofit Customers

```
mean_edu_level= aerofit.Education.mean()

plt.figure(figsize=(10,8))

sns.histplot(data= aerofit, x=
"Education",color='blue',edgecolor='black')
# plt.grid(axis= "y", ls = "--")
plt.axvline(mean_edu_level, color = "red",ls =
"--",label=f'mean_edu_level: {mean_edu_level:.2f}')
plt.xticks(rotation=45)
plt.title("Education level of customers of aerofit")
plt.legend()
plt.show()
```



Observations:

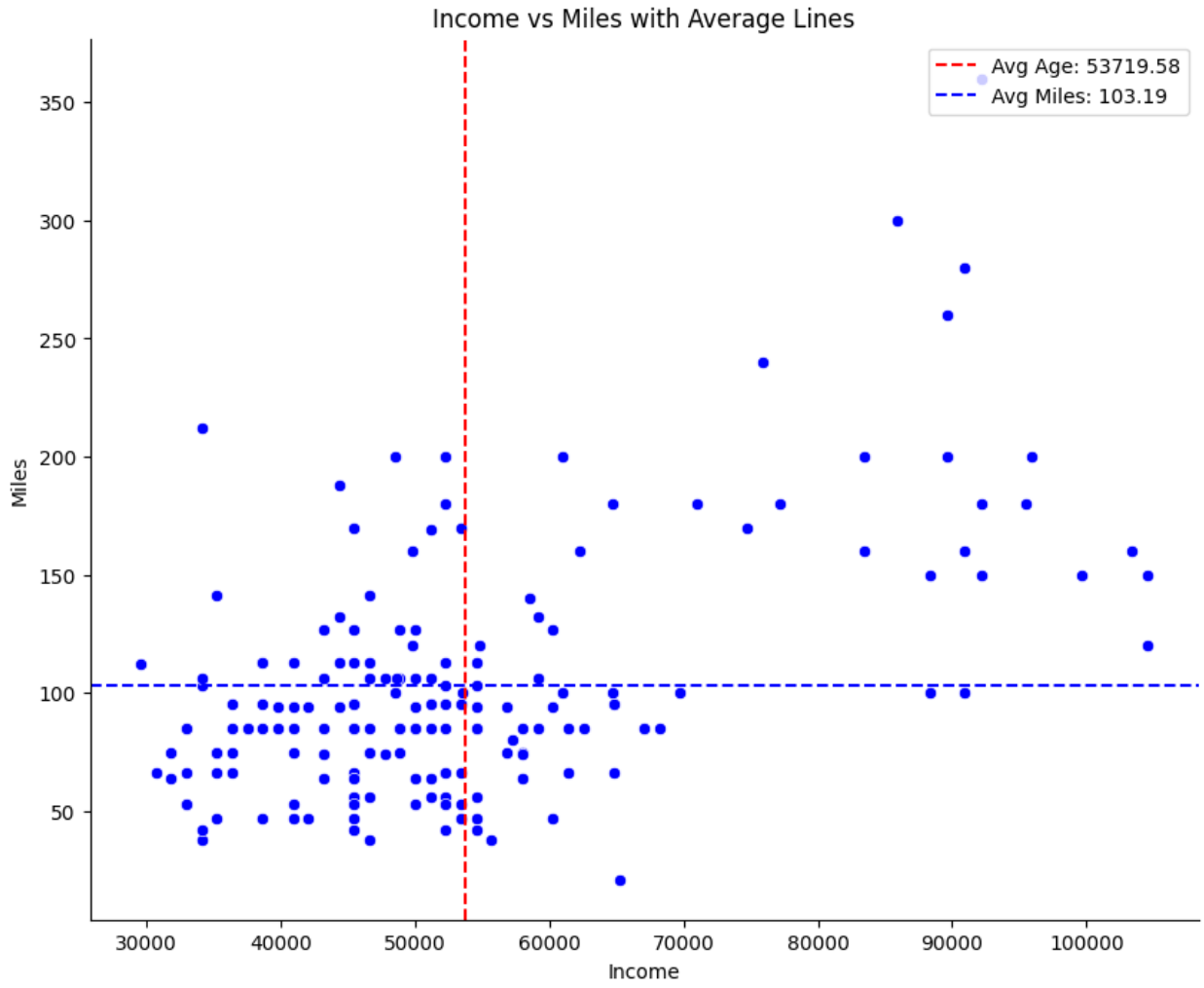
- The most sold product of Aerofit is **KP281** after that KP481,KP781.
- Gender Ratio among the customers is **male: 58% and female: 42%**.
- Marital Status of customers is **married: 60% and unmarried: 40%**.
- Average age of customers is less than **29 years**.
- Average miles dones by customers of aerofit is **103.19 miles**.
- The education level of customers of aerofit is **15.57 years**.

$$Bi-VariantAnalysis:$$

1. Income vs Miles: Do higher income customers use treadmills more?

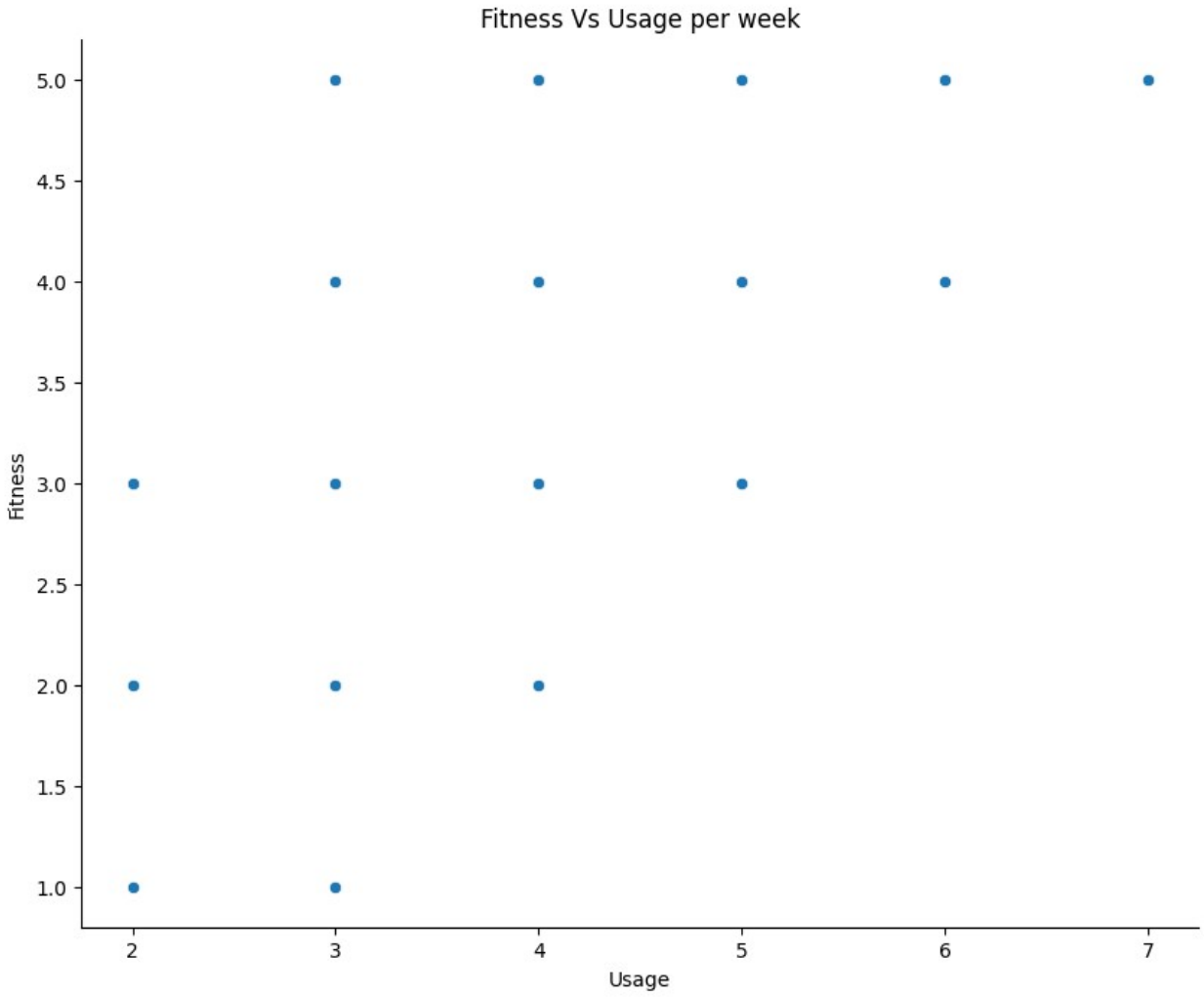
```
avg_income = aerofit.Income.mean()
avg_miles = aerofit.Miles.mean()

plt.figure(figsize=(10,8))
sns.scatterplot(data=aerofit,x='Income',y='Miles',color='blue')
plt.axvline(avg_income, color = "red",ls = "--",label=f'Avg Age:
```

```
{avg_income:.2f}')
plt.axhline(avg_miles, color="blue", linestyle="--", label=f'Avg
Miles: {avg_miles:.2f}')
plt.title("Income vs Miles with Average Lines")
plt.legend(loc='upper right')
sns.despine()
plt.show()
```



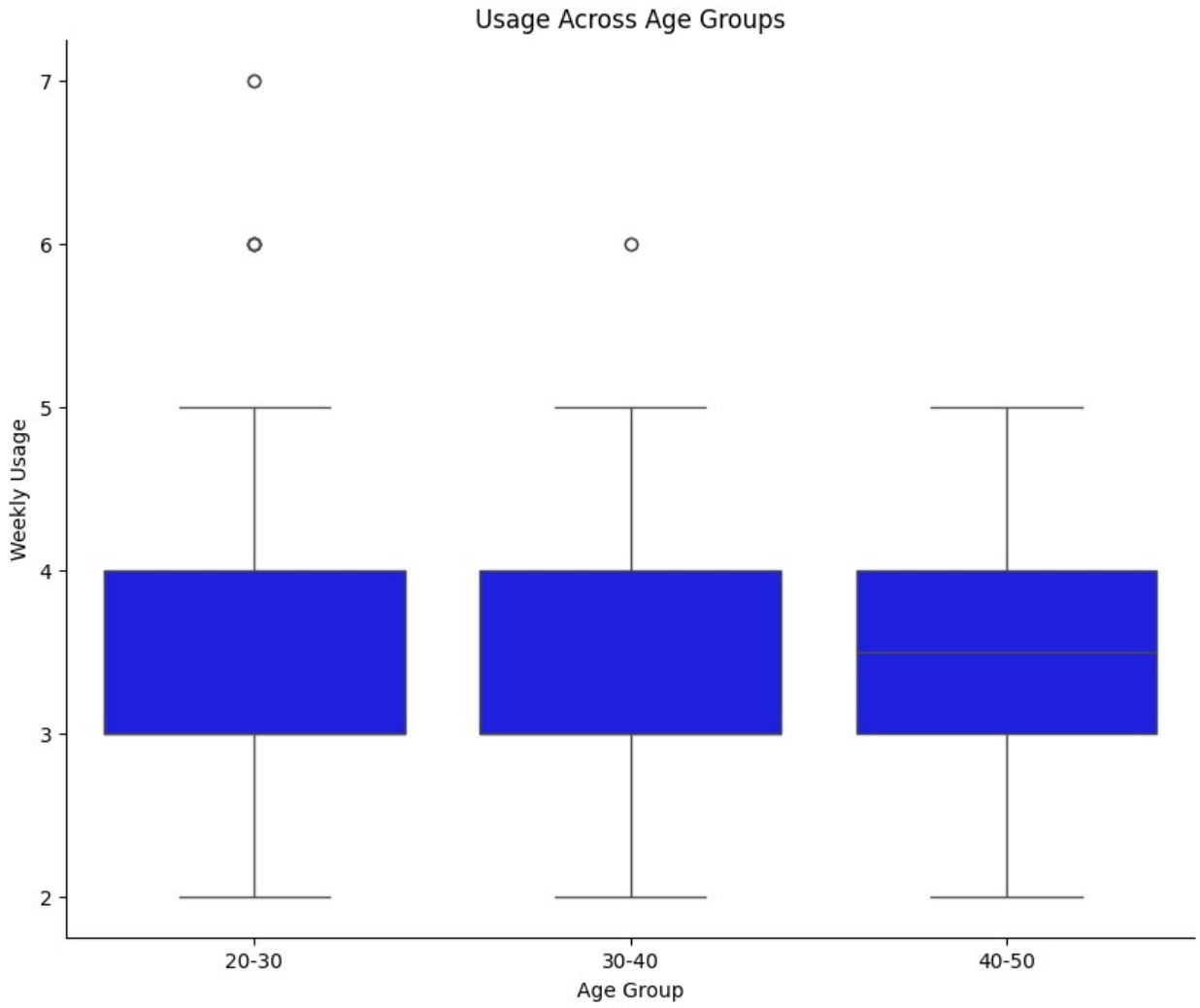
1. Fitness vs Usage: Are fitter people using the product more?

```
plt.figure(figsize=(10,8))
plt.title("Fitness Vs Usage per week")
sns.scatterplot(data=aerofit, x='Usage' ,y='Fitness')
sns.despine()
plt.show()
```



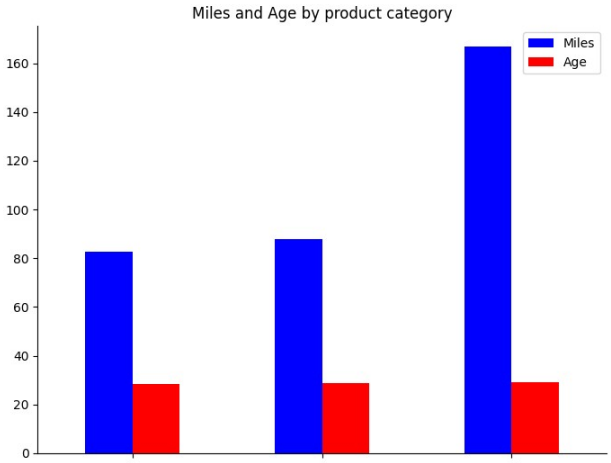
1. Age vs Usage: Are older customers using it less?

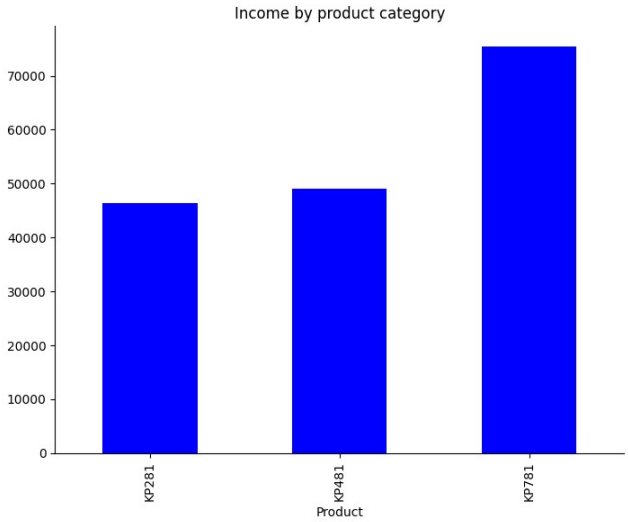
```
plt.figure(figsize=(10,8))
aerofit['Age_Group'] = pd.cut(aerofit['Age'], bins=[20, 30, 40, 50],
                              labels=['20-30', '30-40', '40-50'])

sns.boxplot(data=aerofit, x='Age_Group', y='Usage',color='blue')
plt.title('Usage Across Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Weekly Usage')
sns.despine()
plt.show()
```



1. Group by Product and compare average Miles, Income, Age

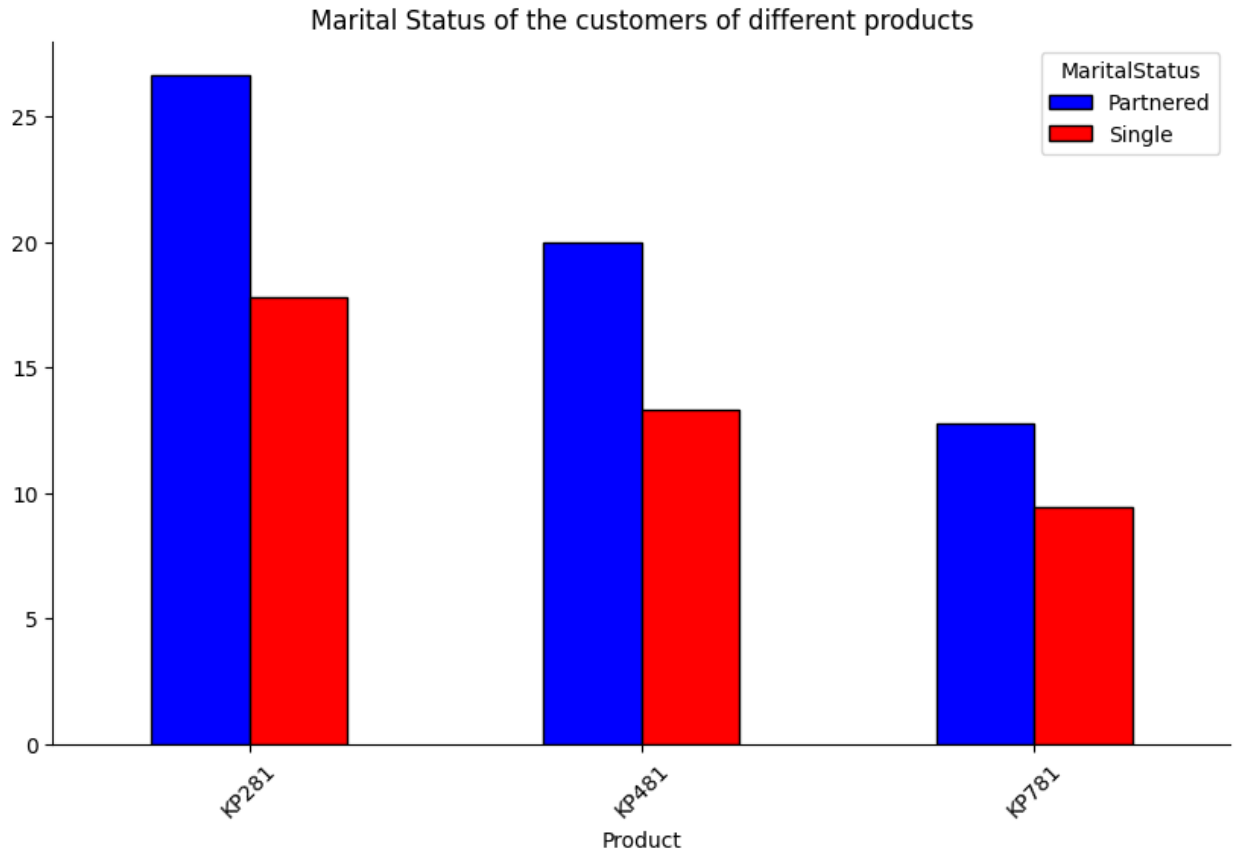
```
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

plt.subplot(1,2,1)
aerofit.groupby('Product')
[['Miles','Age']].mean().reset_index().plot(kind='bar',ax=axes[0],colo
r=['Blue','Red'])
axes[0].set_title('Miles and Age by product category')

plt.subplot(1,2,2)
aerofit.groupby('Product')
['Income'].mean().plot(kind='bar',ax=axes[1],color='Blue')
axes[1].set_title('Income by product category')

sns.despine()
plt.tight_layout()
plt.show()
```





1. Marital Status of the customers of different products

```
df=pd.crosstab(aerofit['Product'],aerofit['MaritalStatus'],normalize=
True)*100
df.head()
df.plot(kind='bar', figsize=(10, 6), color=['blue', 'red'],
edgecolor='black')
plt.title("Marital Status of the customers of different products")
plt.xticks(rotation=45)
sns.despine()
plt.show()
```

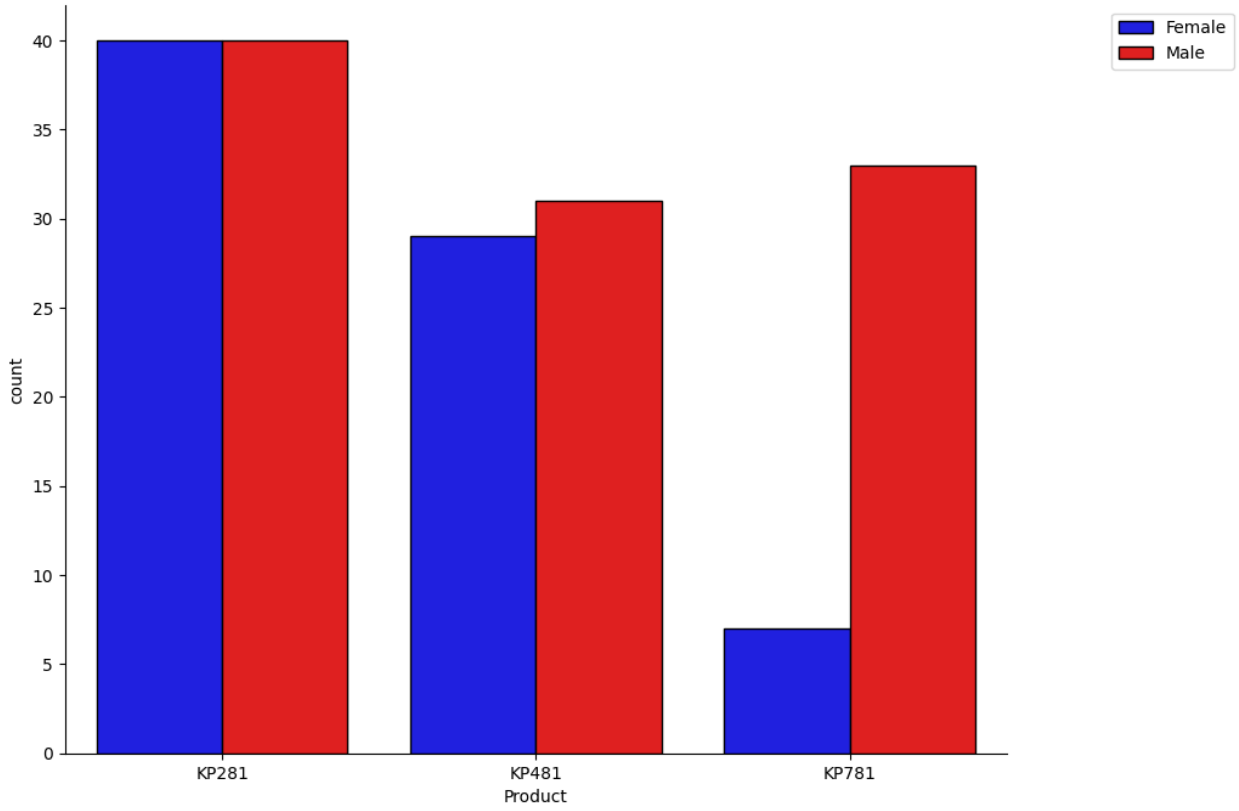


1. Product analysis with Gender of customers

```
df1=aerofit.groupby(['Product','Gender']).size().reset_index(name='cou
nt')
df1
```

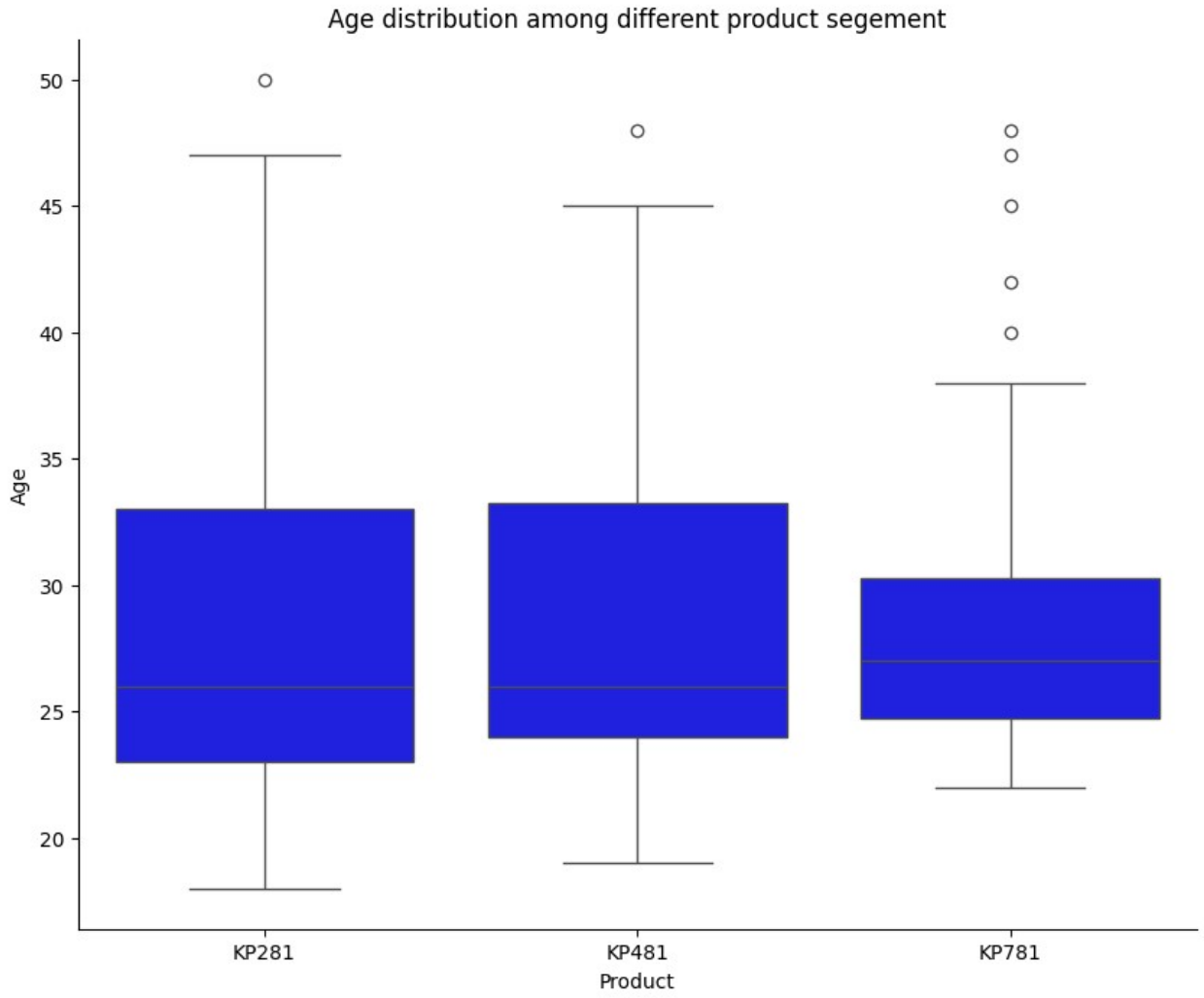

```
{"summary":"{\n  \"name\": \"df1\",\n  \"rows\": 6,\n  \"fields\": [\n
{\n      \"column\": \"Product\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"KP281\",\n          \"KP481\",\n
\"KP781\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Male\",\n          \"Female\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"count\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 12,\n
\"min\": 7,\n        \"max\": 40,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          29,\n          33\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df1"}
```

```
plt.figure(figsize=(10,8))
sns.barplot(x='Product',y= 'count',data=df1,hue='Gender',edgecolor =
'black', palette=['blue', 'red'])
plt.legend(bbox_to_anchor=(1.25,1),loc='upper right')
sns.despine()
plt.show()
```



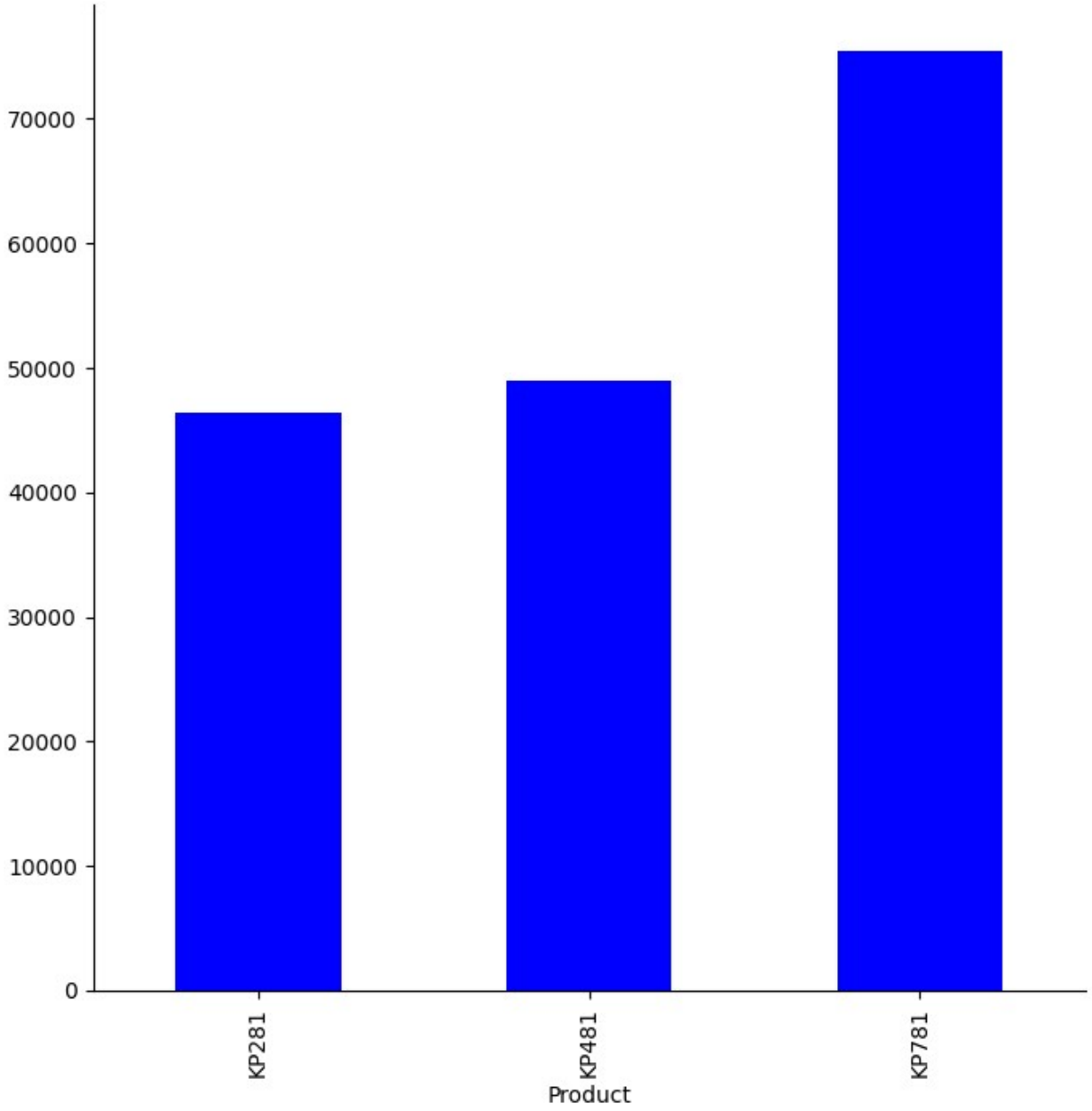
1. Product analysis with respect to age of the customers

```
plt.figure(figsize=(10,8))
sns.boxplot(x= 'Product',y='Age',data=aerofit,color='blue')
plt.title('Age distribution among different product segement')
sns.despine()
plt.show()
```

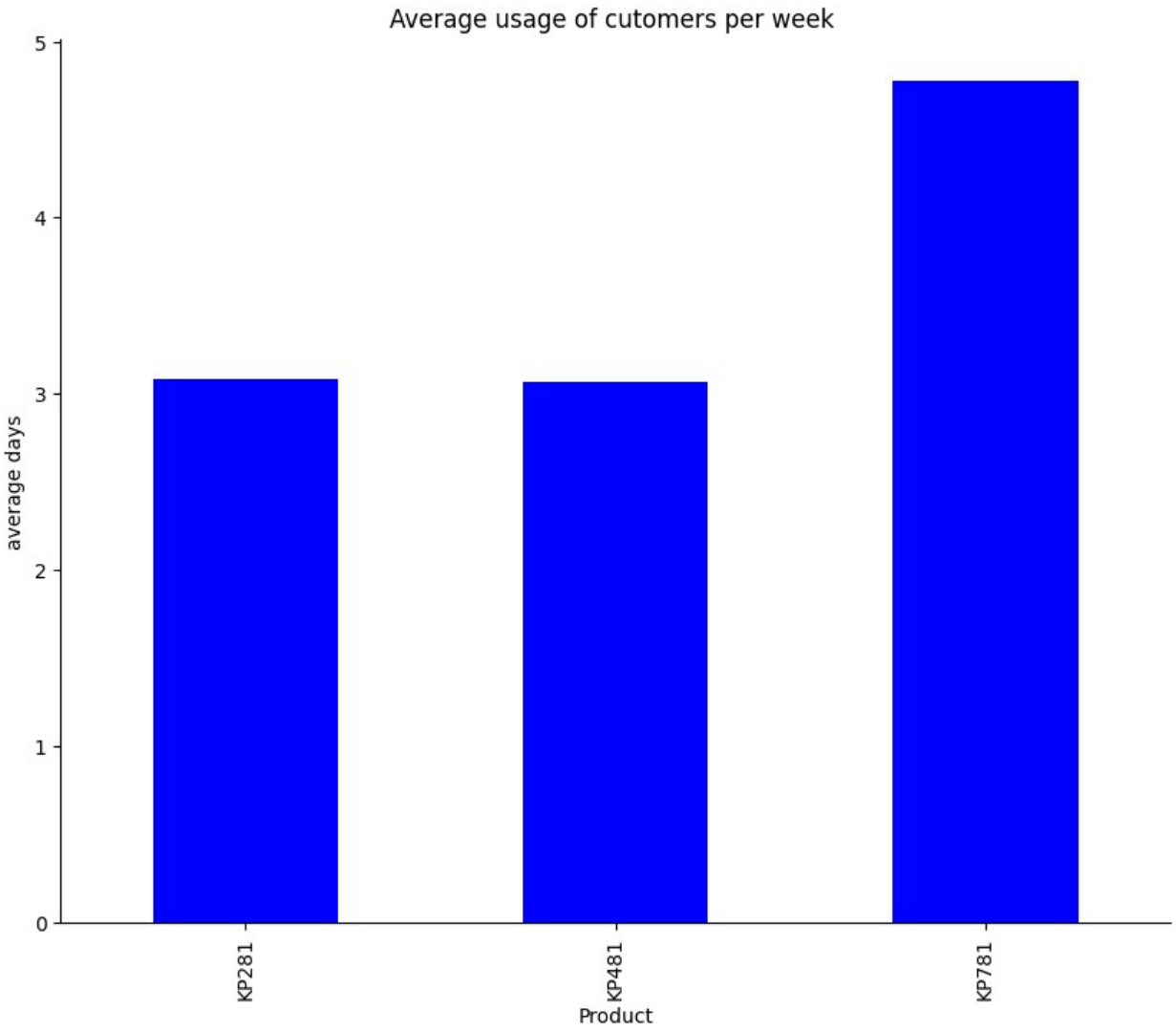


1. Average Income of people buying different models of aerofit treadmills

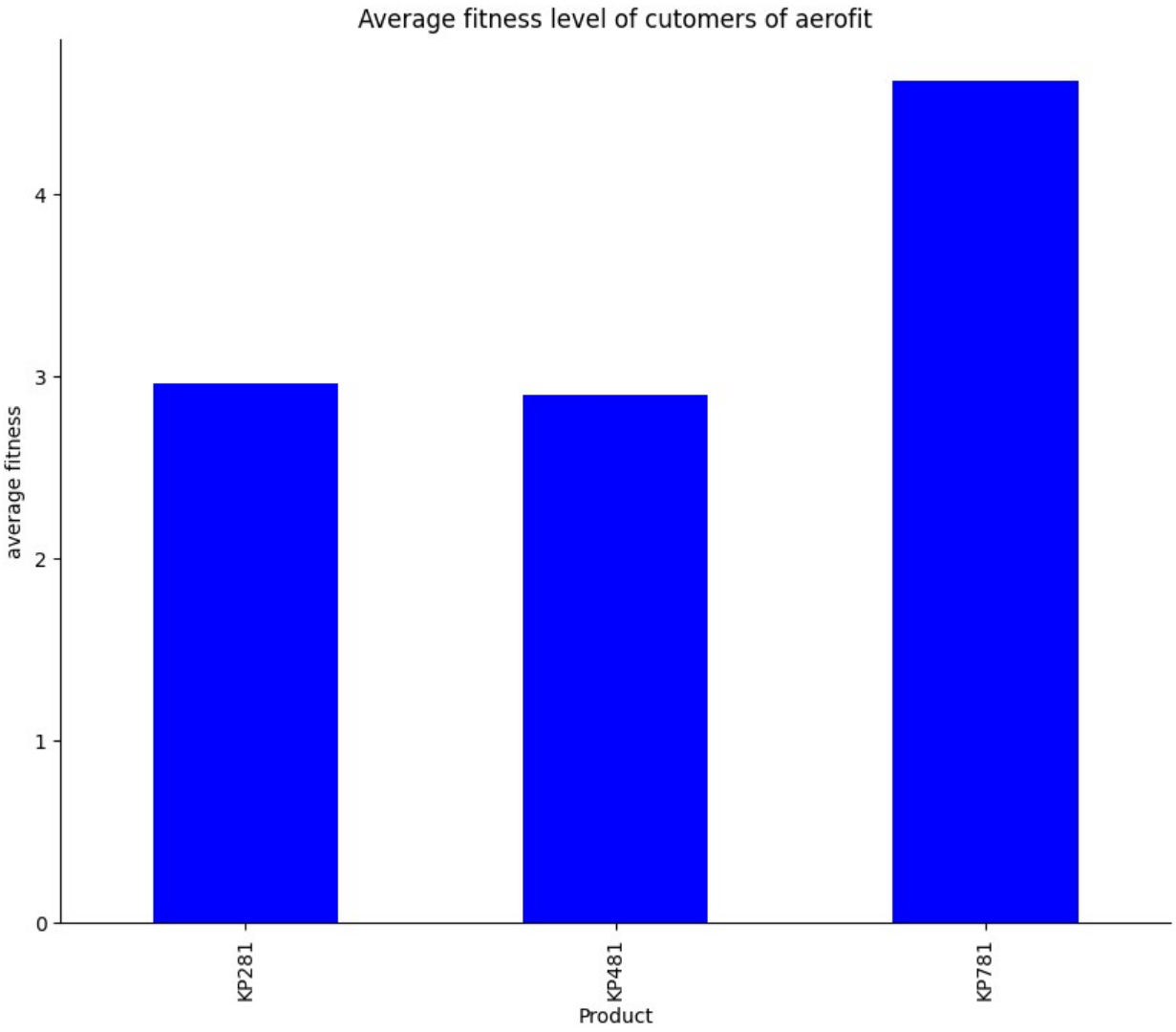
```
plt.figure(figsize=(10,8))
avg_income=aerofit.groupby('Product')['Income'].mean()
avg_income.plot(kind='bar',figsize=(8,8),color='blue')
sns.despine()
plt.show()
```



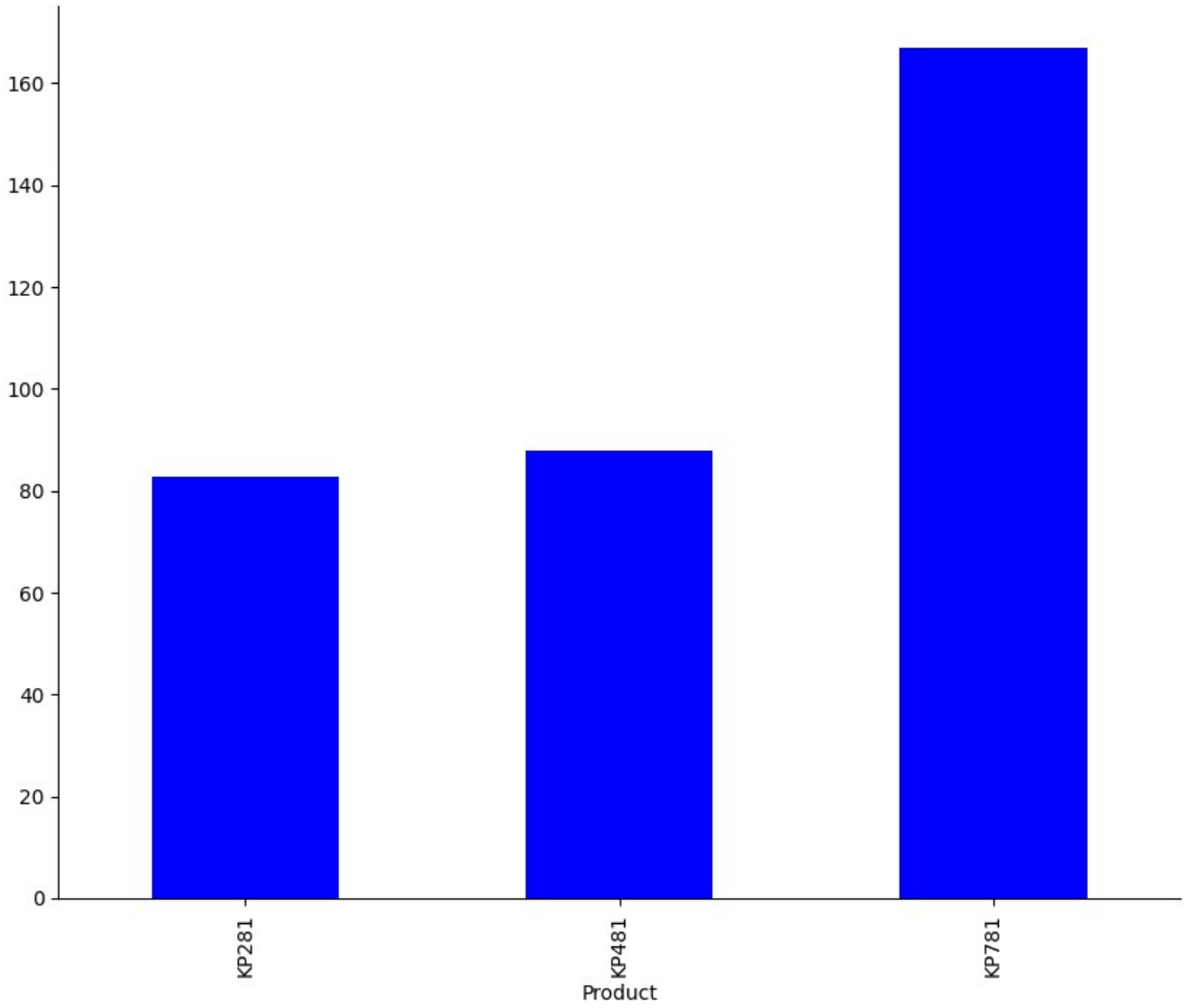

```
avg_usage= aerofit.groupby('Product')['Usage'].mean()

avg_usage.plot(kind= 'bar',figsize=(10,8),color='blue')
plt.ylabel("average days")
plt.title("Average usage of cutomers per week ")
sns.despine()
plt.show()
```




```
avg_fitness= aerofit.groupby('Product')['Fitness'].mean()

avg_fitness.plot(kind= 'bar',figsize=(10,8),color='blue')
plt.ylabel("average fitness")
plt.title("Average fitness level of cutomers of aerofit")
sns.despine()
plt.show()
```




```
avg_miles = aerofit.groupby('Product')['Miles'].mean()
avg_miles.plot(kind= 'bar',figsize=(10,8),color='blue')
sns.despine()
plt.show()
```



Observations:

- People with high fitness level use the aerofit products more than **5 times** a week
- People with high income levels are buying KP781 and these people tends to do more miles
- For KP281 we have equal number of customers for both genders(Male,Female),for KP481 we have male customers slightly more than female and for KP781 the number of male customers are far more than female customers.
- In the two product segements like KP281,KP481 they have almost the same gender ratio of customers but for KP781 there is a vast difference between female and male ratio. Mostly men are buying KP781.
- The age distribution for customers of KP281,KP481 is around the same level, most of the buyers for both the products are of the same age, but for KP781 we can see a lot of people are of higher age.
- Most of the people buying KP281,KP481 have average income less than 50000, whereas the people buying KP781 have average income more than 70000

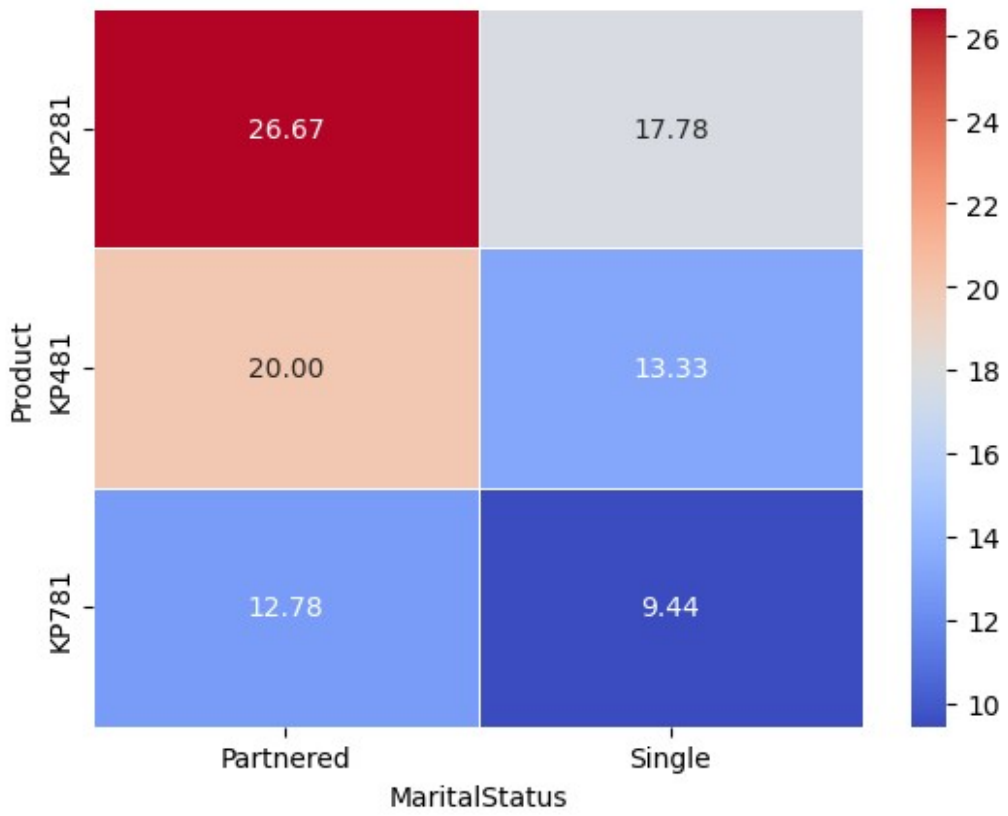
- The avearge usage of KP281,KP481 is less than 3 days a week, but those buying KP781 have avearge usage of 5 days a week.
- Average fitness level of customers of KP281,KP481 is around 3, whereas the customers buying KP781 have average fitness level above 4

```
p,s=aerofit['MaritalStatus'].value_counts()

product_marital_status=pd.crosstab(index = aerofit['Product'],columns=
aerofit['MaritalStatus'],normalize=True)*100
product_marital_status
```


```
{"summary":"{\n  \"name\": \"product_marital_status\",\n  \"rows\":
3,\n  \"fields\": [\n    {\n      \"column\": \"Product\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 3,\n        \"samples\": [\n
\"KP281\",\n          \"KP481\",\n          \"KP781\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Partnered\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
6.946296049448539,\n        \"min\": 12.777777777777777,\n
\"max\": 26.666666666666668,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          26.666666666666668,\n          20.0,\n
12.777777777777777\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Single\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 4.169751944147298,\n        \"min\": 9.444444444444445,\n
\"max\": 17.77777777777778,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          17.77777777777778,\n
13.333333333333334,\n          9.444444444444445\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\
n}","type":"dataframe","variable_name":"product_marital_status"}
```


```
sns.heatmap(product_marital_status,annot = True,cmap=
'coolwarm',fmt=".2f", linewidths=0.5)
```

```
<Axes: xlabel='MaritalStatus', ylabel='Product'>
```




```
total_single =aerofit['MaritalStatus'].value_counts()['Single']
total_partnered = aerofit['MaritalStatus'].value_counts()['Partnered']

print(f"total single customers - {total_single}")
print(f"total partnered customers - {total_partnered}")
```

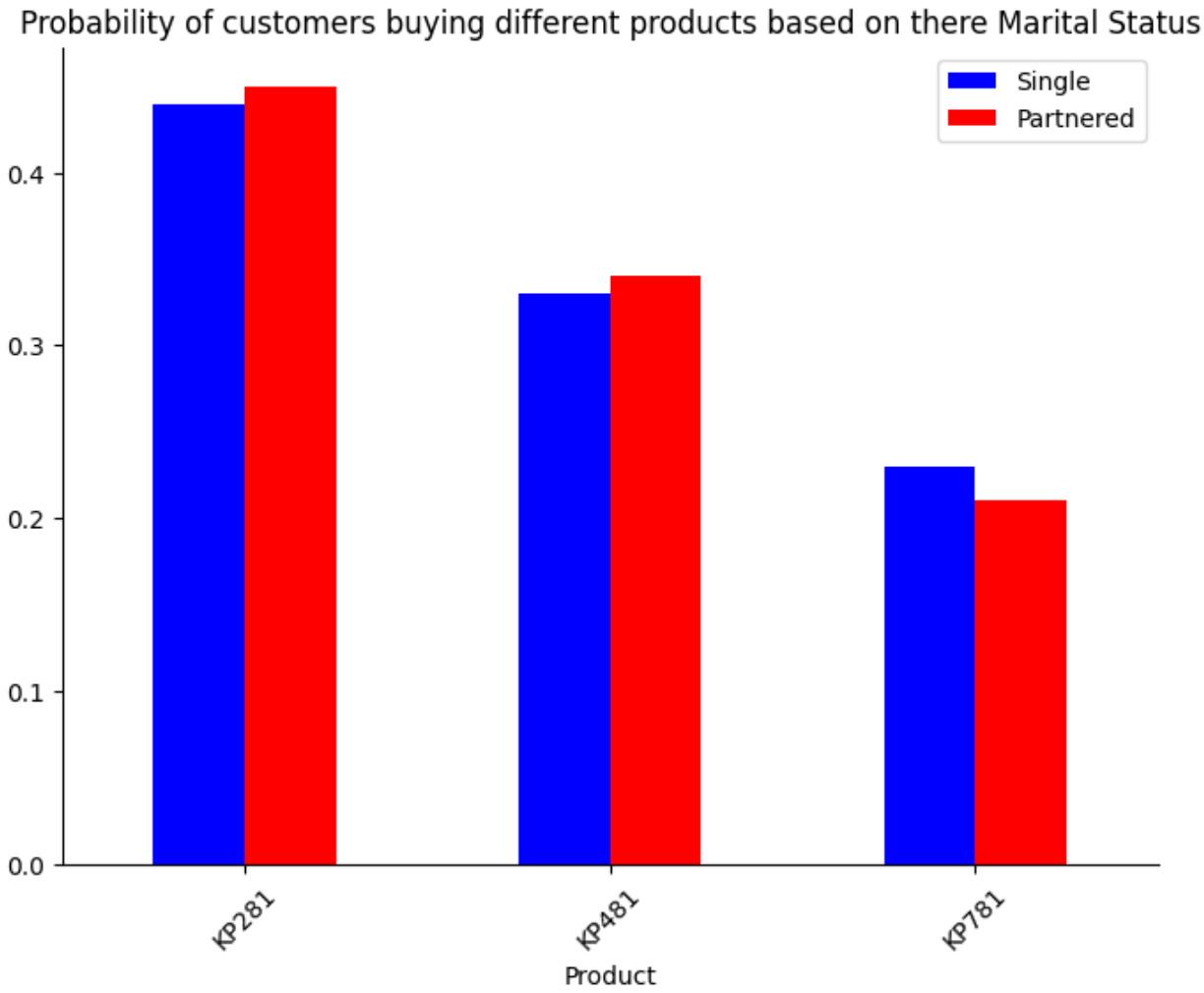
```
total single customers - 73
total partnered customers - 107
```

```
KP281_single= np.round(32/total_single,2)
KP481_single = np.round(24/total_single,2)
KP781_single = np.round(17/total_single,2)

KP281_partnered= np.round(48/total_partnered,2)
KP481_partnered = np.round(36/total_partnered,2)
KP781_partnered = np.round(23/total_partnered,2)

probability_df= pd.DataFrame({'Product': ['KP281','KP481','KP781'],
                              'Single':
[KP281_single,KP481_single,KP781_single],
                               'Partnered':
[KP281_partnered,KP481_partnered,KP781_partnered]})
probability_df.set_index('Product', inplace=True)
probability_df
```

{"summary":"{\n \"name\": \"probability_df\",\n \"rows\": 3,\n
\"fields\": [\n   {\n   \"column\": \"Product\",\n
\"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 3,\n   \"samples\": [\n
\"KP281\",\n   \"KP481\",\n   \"KP781\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Single\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
0.10503967504392486,\n   \"min\": 0.23,\n   \"max\": 0.44,\n
\"num_unique_values\": 3,\n   \"samples\": [\n   0.44,\n
0.33,\n   0.23\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Partnered\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.12013880860626734,\n   \"min\":
0.21,\n   \"max\": 0.45,\n   \"num_unique_values\": 3,\n
\"samples\": [\n   0.45,\n   0.34,\n   0.21\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   }\n]\n}","type":"dataframe","variable_name":"probability_df"}

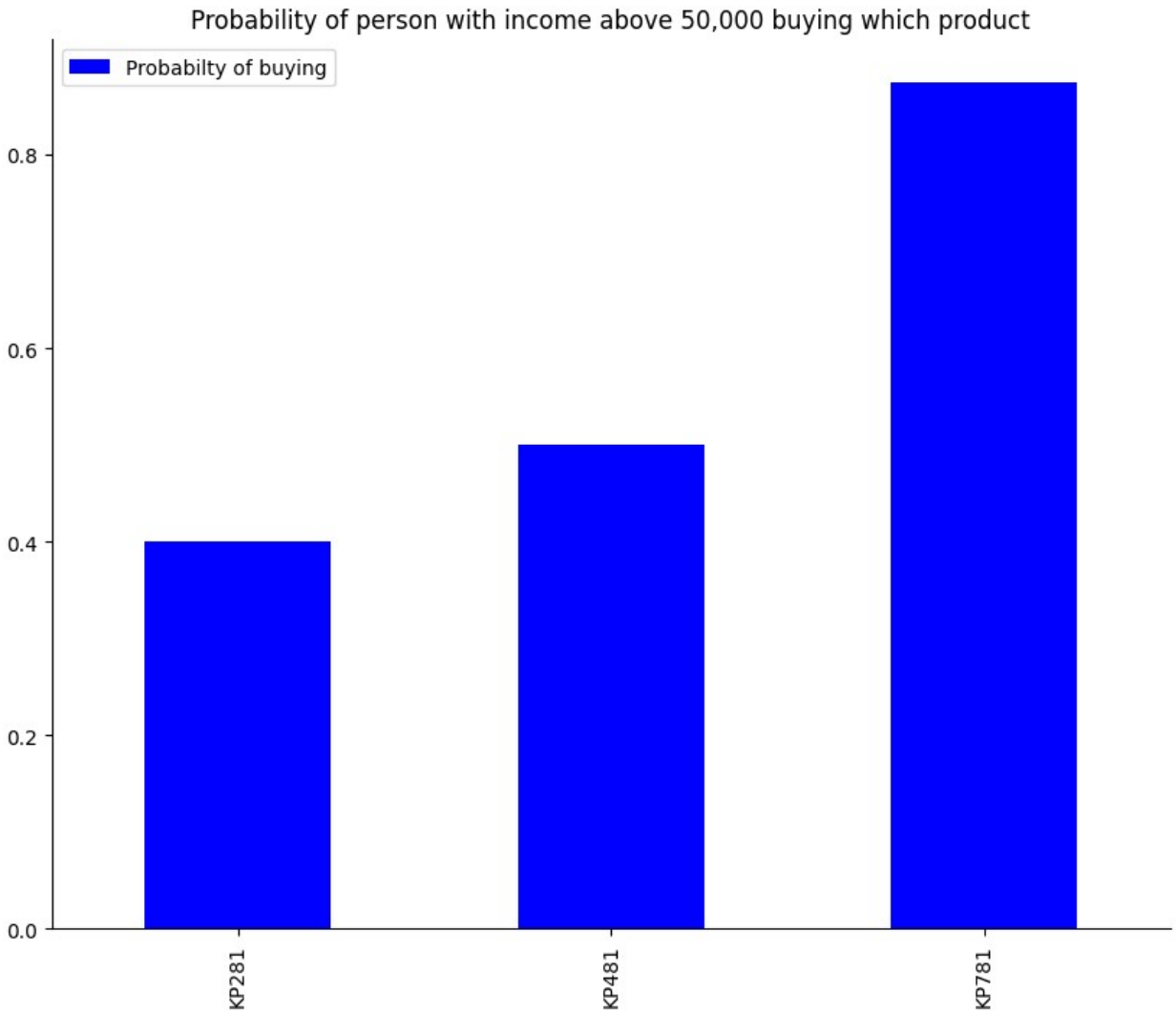
```
plt.figure(figsize=(10,8))
probability_df.plot(kind='bar',figsize= (8,6),color=['blue','red'])
plt.xticks(rotation = 45)
plt.title('Probability of customers buying different products based on
there Marital Status')
sns.despine()
plt.show()
```

```
<Figure size 1000x800 with 0 Axes>
```




```
income_df=aerofit[aerofit['Income'] > 50000]
income_df.groupby('Product')['Income'].count()

Product
KP281    32
KP481    30
KP781    35
Name: Income, dtype: int64

aerofit['Product'].value_counts()

Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64

KP281_above_50 = 32/80
```

```
KP481_above_50 = 30/60

KP781_above_50 = 35/40

income_above_50 = pd.DataFrame({'Product' :['KP281','KP481','KP781'],
                    'Probabilty of buying' :
[KP281_above_50,KP481_above_50,KP781_above_50]})

x = np.arange(len(income_above_50))

income_above_50.plot(kind = 'bar',color='blue',figsize=(10,8))

plt.xticks(x, income_above_50['Product'])
plt.title("Probability of person with income above 50,000 buying which
product")
sns.despine()
plt.show()
```



```
income_df_1=aerofit[aerofit['Income'] < 50000]
income_df_1.groupby('Product')['Income'].count()
```

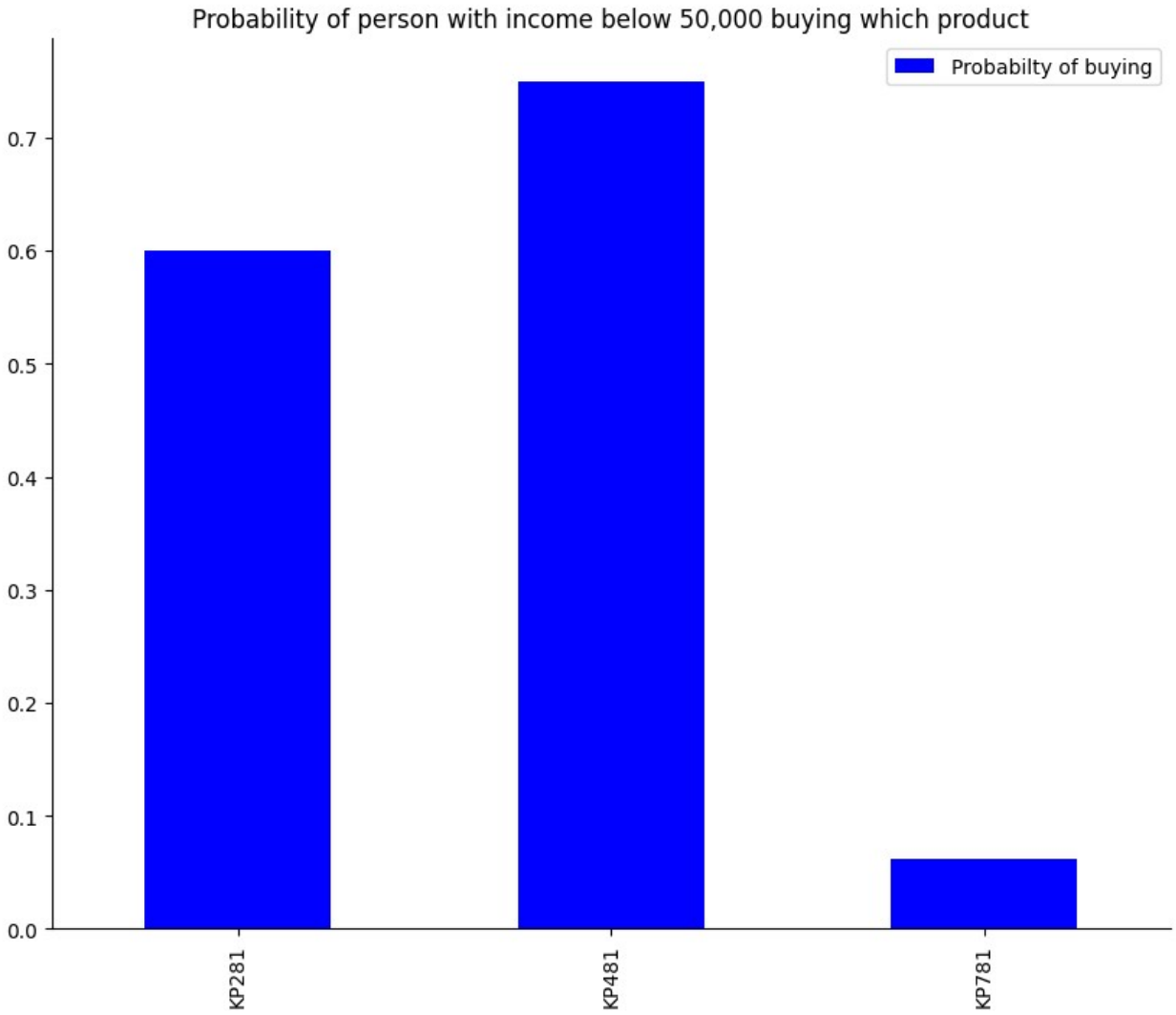
```
Product
KP281    48
KP481    30
KP781     5
Name: Income, dtype: int64
```

```
aerofit['Product'].value_counts()
```

```
Product
KP281    80
KP481    60
KP781    40
Name: count, dtype: int64
```

```
KP281_below_50 = 48/80
KP481_below_50 = 60/80
KP781_below_50 = 5/80

income_below_50 = pd.DataFrame({'Product' :['KP281','KP481','KP781'],
                    'Probabilty of buying' :
[KP281_below_50,KP481_below_50,KP781_below_50]})

x = np.arange(len(income_above_50))
plt.figure(figsize= (8,6))

income_below_50.plot(kind = 'bar',figsize=(10,8),color='blue')

plt.xticks(x, income_above_50['Product'])
plt.title("Probability of person with income below 50,000 buying which
product")
sns.despine()
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```

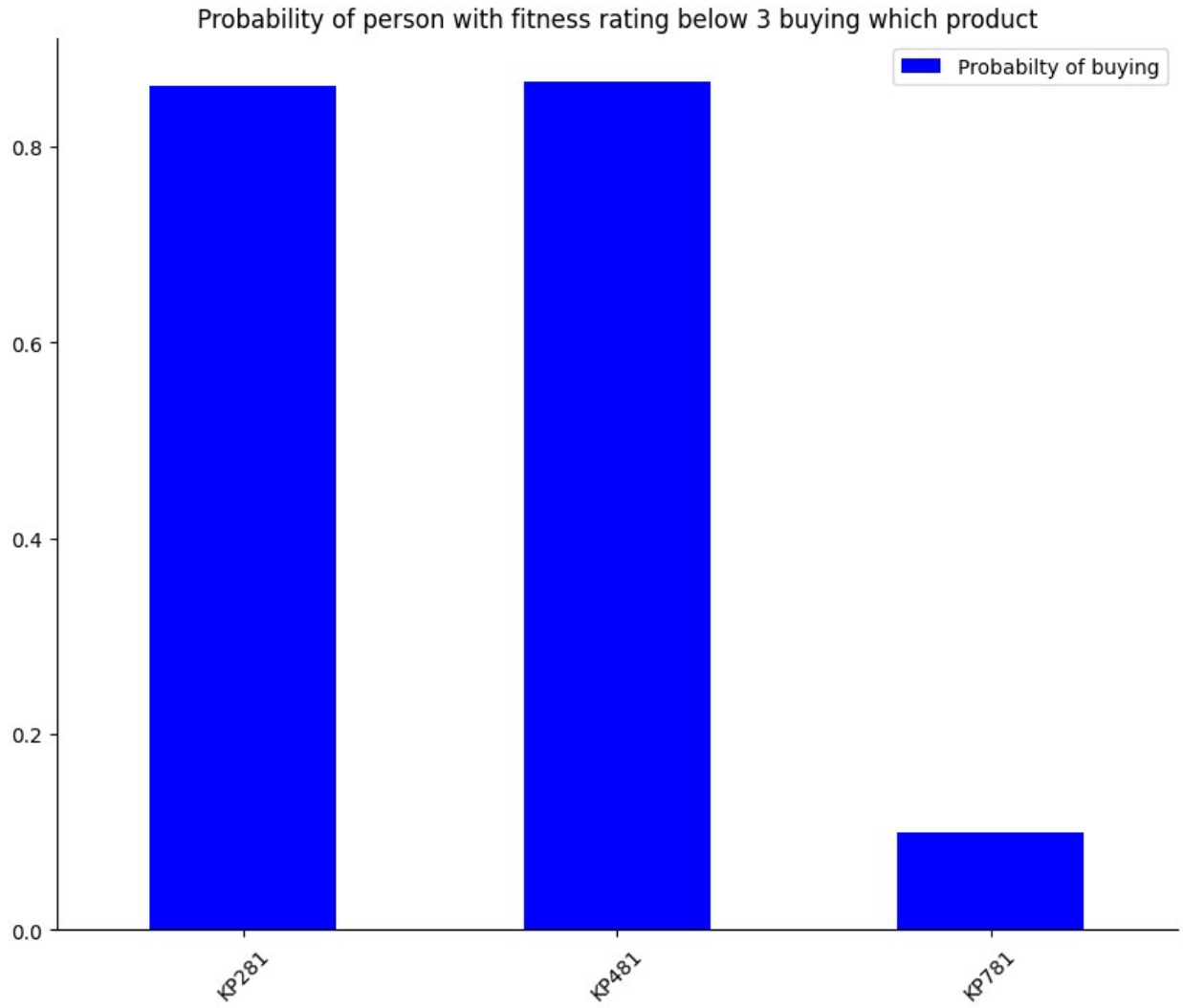



```
fitness_1= aerofit[aerofit['Fitness']<=3]
fitness_1.groupby('Product')['Fitness'].count()
```

```
Product
KP281    69
KP481    52
KP781     4
Name: Fitness, dtype: int64
```

```
KP281_below_3= 69/80
KP481_below_3 = 52/60
KP781_below_3= 4/40

rating_below_3 = pd.DataFrame({'Product' :['KP281','KP481','KP781'],
                    'Probabilty of buying' :
[KP281_below_3,KP481_below_3,KP781_below_3]})
```

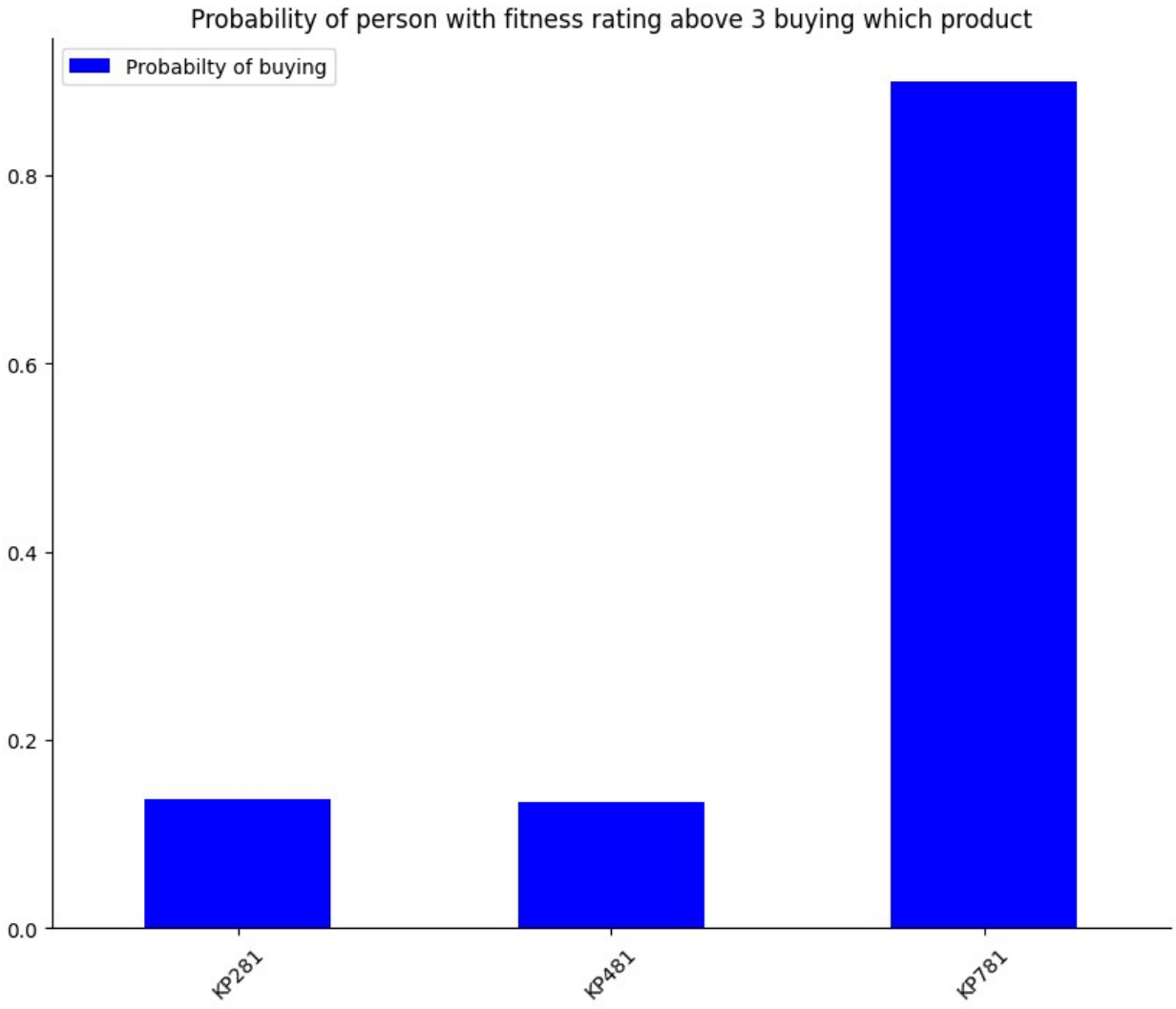
```
x = np.arange(len(rating_below_3))
rating_below_3.plot(kind = 'bar',figsize=(10,8),color='blue')

plt.xticks(x, income_above_50['Product'])
plt.title("Probability of person with fitness rating below 3 buying
which product")
plt.xticks(rotation = 45)
sns.despine()
plt.show()
```




```
fitness_1= aerofit[aerofit['Fitness'] > 3]
fitness_1.groupby('Product')['Fitness'].count()

Product
KP281    11
KP481     8
```

```
KP781    36
Name: Fitness, dtype: int64
```

```
KP281_above_3= 11/80
KP481_above_3 = 8/60
KP781_above_3 = 36/40

rating_above_3 = pd.DataFrame({'Product' :['KP281','KP481','KP781'],
                     'Probabilty of buying' :
[KP281_above_3,KP481_above_3,KP781_above_3]})

x = np.arange(len(rating_above_3))


rating_above_3.plot(kind = 'bar',figsize=(10,8),color='blue')

plt.xticks(x, income_above_50['Product'])
plt.title("Probability of person with fitness rating above 3 buying
which product")
plt.xticks(rotation = 45)
sns.despine()
plt.show()
```



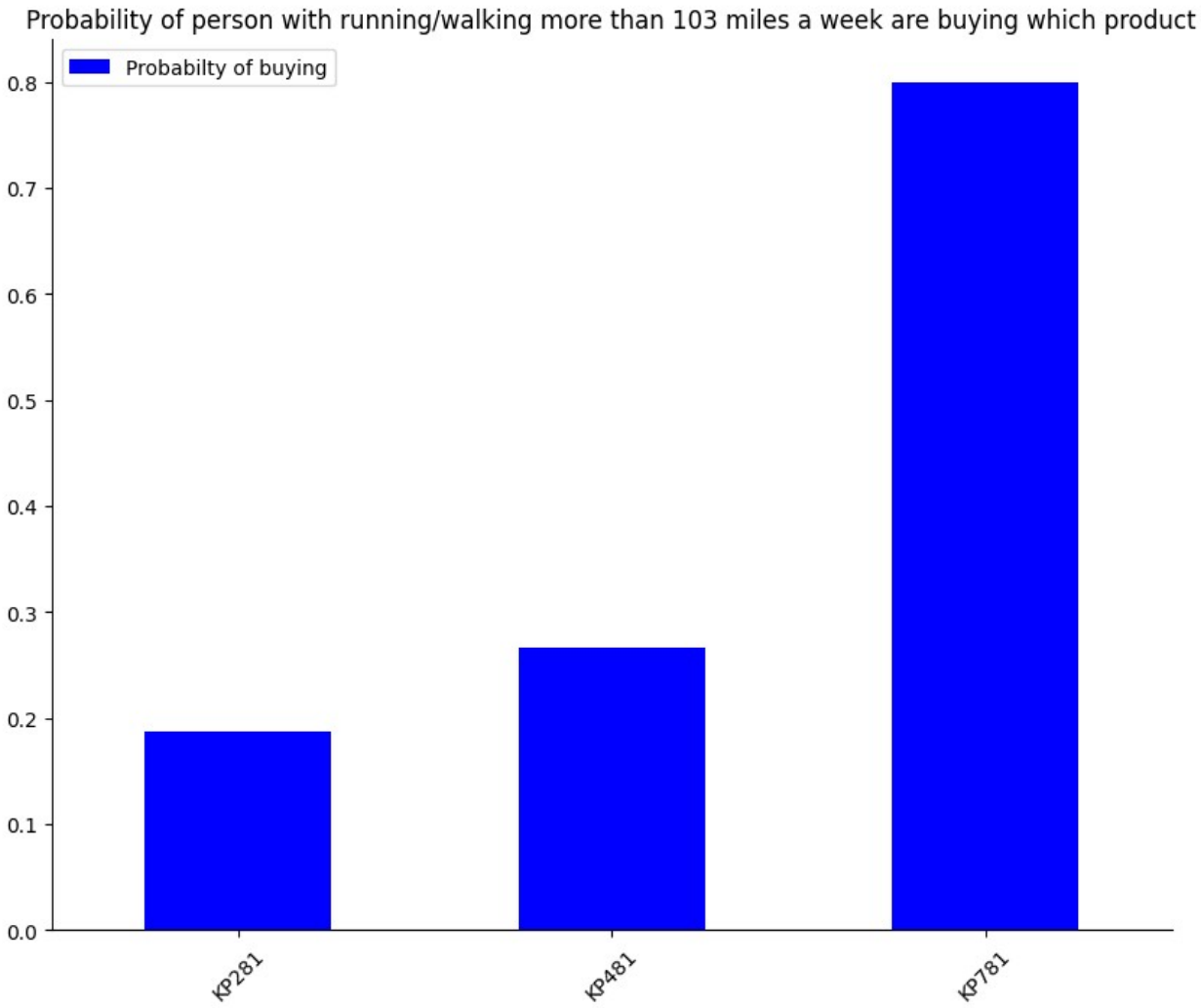

```
aerofit[aerofit['Miles']>103].groupby('Product')['Miles'].count()
```

```
Product
KP281    15
KP481    16
KP781    32
Name: Miles, dtype: int64
```

```
KP281_miles = 15/80
KP481_miles = 16/60
KP781_miles = 32/40

miles_above_avg = pd.DataFrame({'Product' :['KP281','KP481','KP781'],
                     'Probabilty of buying' :
[KP281_miles,KP481_miles,KP781_miles]})

x = np.arange(len(miles_above_avg))
plt.figure(figsize= (8,6))
```

```
miles_above_avg.plot(kind = 'bar',figsize=(10,8),color='blue')

plt.xticks(x, income_above_50['Product'])
plt.title("Probability of person with running/walking more than 103
miles a week are buying which product")
plt.xticks(rotation = 45)
sns.despine()
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```



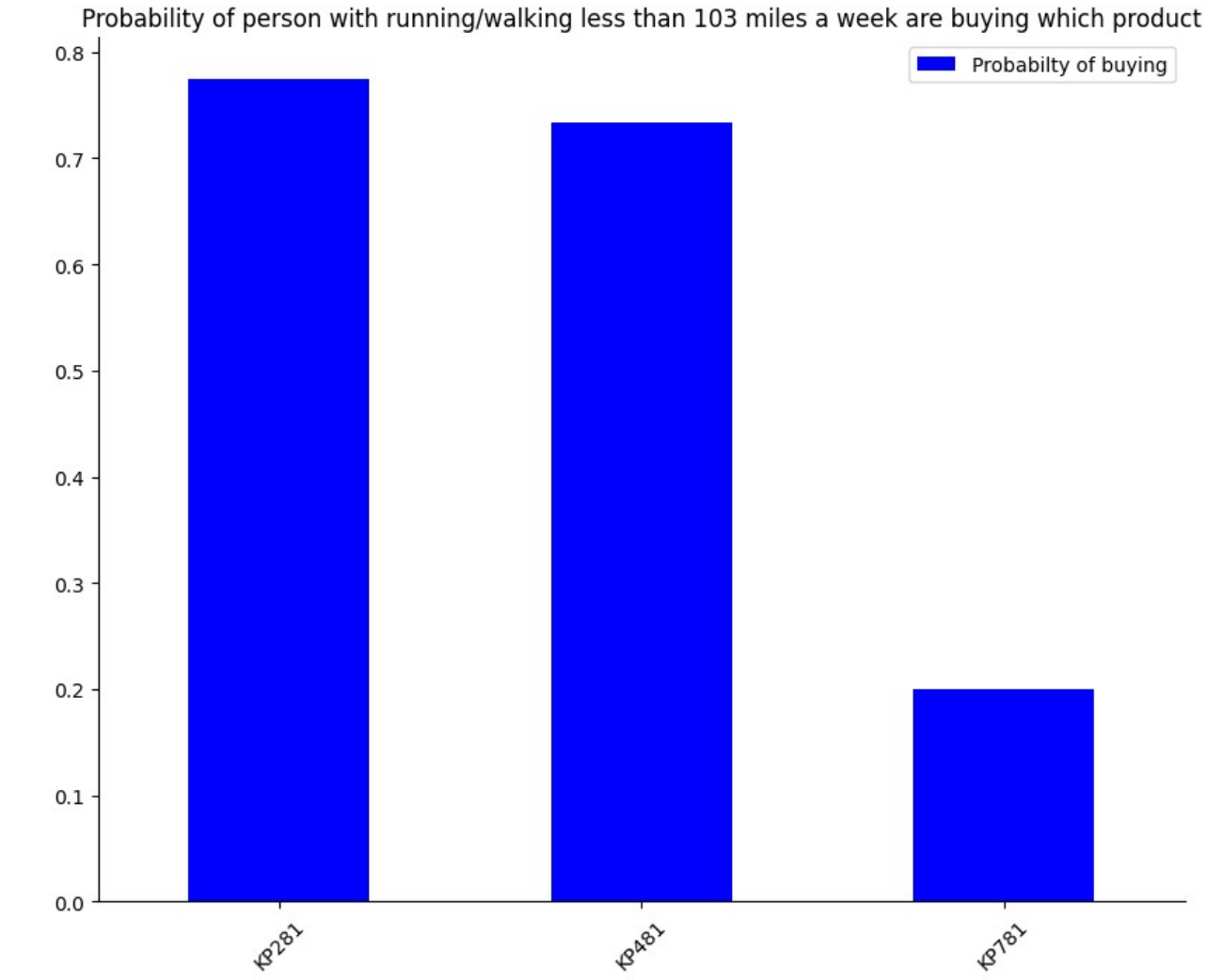

```
aerofit[aerofit['Miles']<103].groupby('Product')['Miles'].count()
```

```
Product
KP281    62
KP481    44
```

```
KP781     8
Name: Miles, dtype: int64
```

```
KP281_low_miles = 62/80
KP481_low_miles = 44/60
KP781_low_miles = 8/40

miles_below_avg = pd.DataFrame({'Product' :['KP281','KP481','KP781'],
                     'Probabilty of buying' :
[KP281_low_miles,KP481_low_miles,KP781_low_miles]})

x = np.arange(len(miles_below_avg))
plt.figure(figsize= (8,6))

miles_below_avg.plot(kind = 'bar',figsize=(10,8),color='blue')

plt.xticks(x, income_above_50['Product'])
plt.title("Probability of person with running/walking less than 103
miles a week are buying which product")
plt.xticks(rotation = 45)
sns.despine()
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```


Probability of person with running/walking less than 103 miles a week are buying which product
Probabilty of buying
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
KP281
KP481
KP781