```
!gdown "https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749" -O aer.csv
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/001/125/original/aerofit_treadmill.csv?1639992749
To: /content/aer.csv
100% 7.28k/7.28k [00:00<00:00, 24.5MB/s]
```

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

```
df=pd.read_csv('aer.csv')
df.head()
```

	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

```
df.describe()
```

	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

From the above description, it can be inferred that the variables Income and Miles might have outliers.

```
color=['#00688B','#48D1CC','#AFEEEE']
```

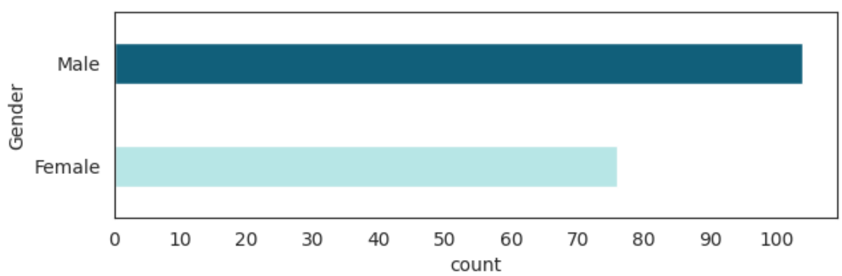
Null Value Detection

```
df.isnull().values.any()
```

```
False
```

There's no null value in the dataset.

```
plt.figure(figsize=(7,2))
sns.set_style("white")
plt.xticks(np.arange(0,110,step=10))
sns.countplot(data=df, y='Gender',hue='Gender',palette=color[0:3:2], width=0.4)
plt.show()
```



Contigency Tables and Probability of buying a product given Marital Status and Gender

```
cont_gen=pd.crosstab(index=df['Gender'],columns=df['Product'],margins=True)
cont_gen
```

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

```
cont_mar=pd.crosstab(index=df['MaritalStatus'],columns=df['Product'],margins=True)
cont_mar
```

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

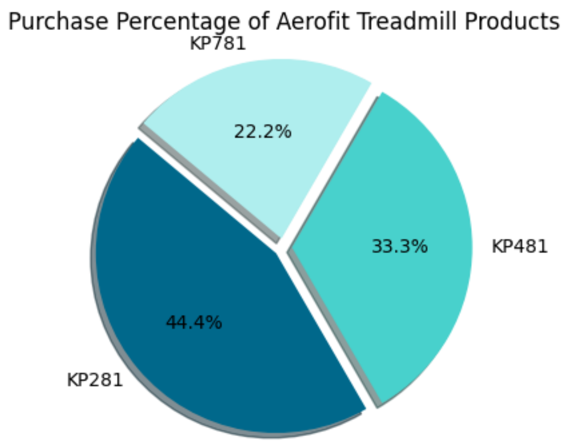
```
cont_mar=pd.crosstab(index=df['MaritalStatus'],columns=df['Product'],margins=True,normalize=True)
cont_mar*100
```

Product MaritalStatus	KP281	KP481	KP781	All
Partnered	26.666667	20.000000	12.777778	59.444444
Single	17.777778	13.333333	9.444444	40.555556
All	44.444444	33.333333	22.222222	100.000000

Purchase Distribution of Products

```
df_new=cont_mar.drop('All',axis=1)
plt.figure(figsize=(4,4))
plt.pie(
    df_new.iloc[len(cont_mar.index)-1],
    labels=df_new.columns,
    autopct='%1.1f%%',
    startangle=140,
    explode=(0.05, 0.05, 0.05),
    shadow=True
)

plt.title('Purchase Percentage of Aerofit Treadmill Products')
plt.axis('equal')
plt.show()
```



Probabilty of one being Partenered or Single given their preference of Product

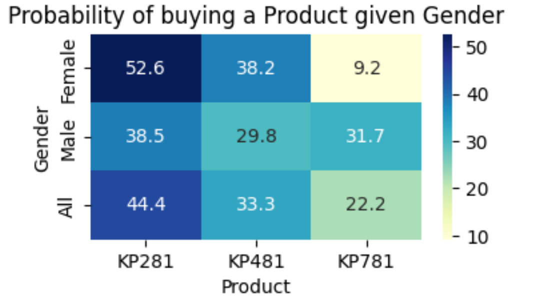
```
#P(Product|MaritalStat)
cont_mar=pd.crosstab(index=df['MaritalStatus'],columns=df['Product'],margins=True,normalize='columns')
cont_mar*100
```

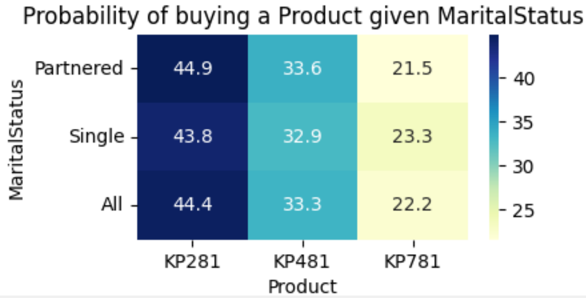
Product	KP281	KP481	KP781	All
MaritalStatus				
Partnered	60.0	60.0	57.5	59.444444
Single	40.0	40.0	42.5	40.555556

```
# CONDITIONAL PROBABILTY(P(Product|Gender))
def calculate_conditional_probability(variable_a, variable_b):
    contingency_table_ab = pd.crosstab(index=df[variable_b], columns=df[variable_a], margins=True)
    p_a_b=pd.crosstab(index=df[variable_b],columns=df[variable_a],margins=True,normalize='index')
    p_b_a=pd.crosstab(index=df[variable_b],columns=df[variable_a],margins=True,normalize='columns')

    return contingency_table_ab, p_a_b*100, p_b_a*100

variable_pairs = [
    ('Product', 'Gender'),
    ('Product', 'MaritalStatus'),
]
for variable_a, variable_b in variable_pairs:
    contingency_tab,p_a_b,p_b_a = calculate_conditional_probability(variable_a, variable_b)
    fig,ax=plt.subplots(figsize=(4,2))
    sns.heatmap(p_a_b, annot=True, cmap="YlGnBu", fmt=".1f")
    plt.title(f"\nProbability of buying a {variable_a} given {variable_b}")
    plt.xlabel(variable_a)
    plt.ylabel(variable_b)
plt.show()
```






```
import warnings
warnings.filterwarnings('ignore')
```

```
plt.figure(figsize=(10,6))
sns.pairplot(df, palette=color)
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```

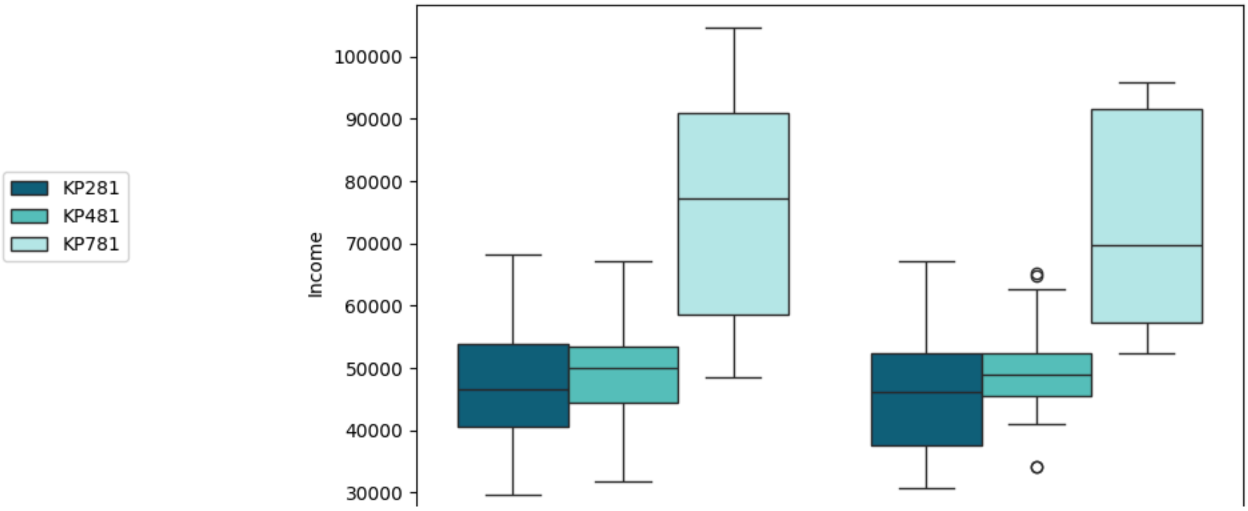
From the above plots, we can infer that,

- There might me direct correlation between Age and Income, and Age and Miles.
- The distribution of Income, Age and Miles are right skewed.
- There are several datapoints which lie significantly away from the main cluster in the scatterplots of Miles vs Income, Age vs Income, and Miles vs Age. Those distant points might be potential outliers.

Age Education Usage Fitness Income Miles

Box Plot for Outlier Detection

```
plt.figure(figsize=(8,5))
sns.set_palette(color)
sns.boxplot(x=df['Gender'], y=df['Income'], hue=df['Product'])
plt.legend(loc=(-0.5,0.5), ncol=1)
plt.show()
```



Displaying The Outlier Rows

```
# OUTLIERS DETECTION

for (gender, product), group in df.groupby(['Gender', 'Product']):
  Q1 = df['Income'].quantile(0.25)
  Q3 = df['Income'].quantile(0.75)
  IQR = Q3 - Q1
  lower_bound = Q1 - 1.5 * IQR
  upper_bound = Q3 + 1.5 * IQR
  outliers = df[(df['Income'] < lower_bound) | (df['Income'] > upper_bound)]
outliers
```

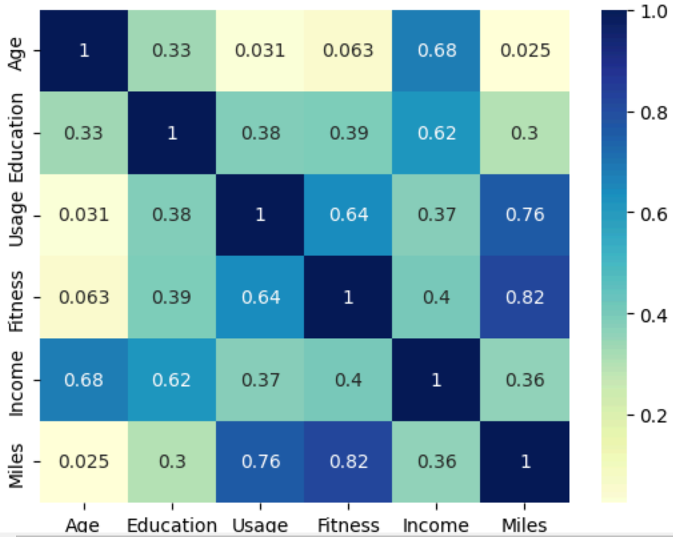
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
159	KP781	27	Male	16	Partnered	4	5	83416	160
160	KP781	27	Male	18	Single	4	3	88396	100
161	KP781	27	Male	21	Partnered	4	4	90886	100
162	KP781	28	Female	18	Partnered	6	5	92131	180
164	KP781	28	Male	18	Single	6	5	88396	150
166	KP781	29	Male	14	Partnered	7	5	85906	300
167	KP781	30	Female	16	Partnered	6	5	90886	280
168	KP781	30	Male	18	Partnered	5	4	103336	160
169	KP781	30	Male	18	Partnered	5	5	99601	150
170	KP781	31	Male	16	Partnered	6	5	89641	260
171	KP781	33	Female	18	Partnered	4	5	95866	200
172	KP781	34	Male	16	Single	5	5	92131	150
173	KP781	35	Male	16	Partnered	4	5	92131	360
174	KP781	38	Male	18	Partnered	5	5	104581	150
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

```
from scipy.stats import spearmanr
```

Correlation Heatmap among Different Variables

```
df_new=df.drop(['Gender','MaritalStatus','Product'],axis=1)
```

```
sns.heatmap(df_new.corr(method='spearman'), annot=True, cmap='YlGnBu')
plt.show()
```



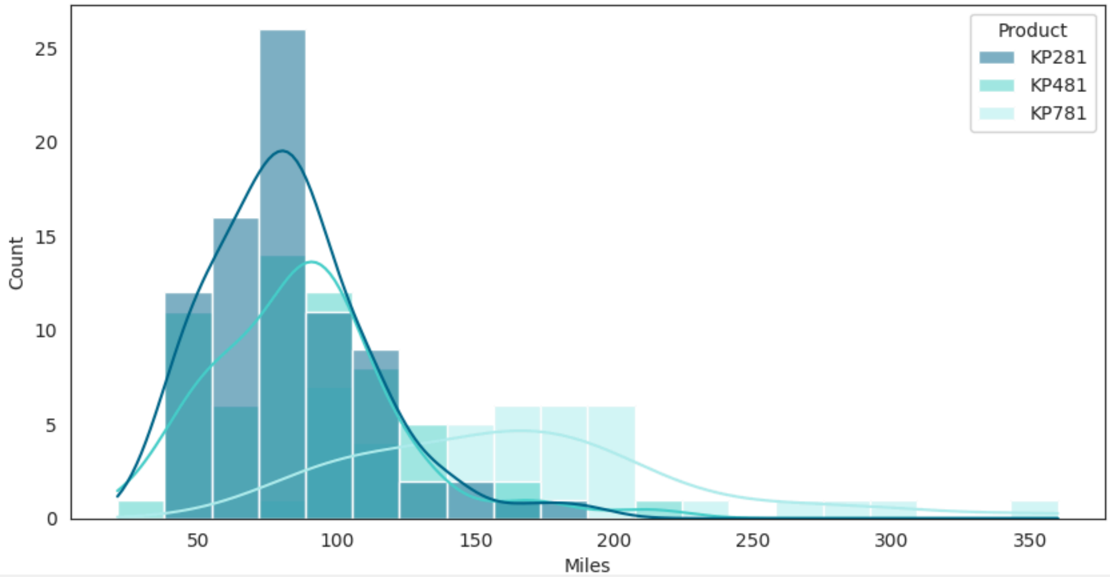
- The heatmap indicates strong positive correlation among the following pairs suggesting that as one variable increases, other tends to increase as well:
  Education and Income, Usage and Fitness, Usage and Miles, and Fitness and Miles.
- The correlation coefficient for the following pairs indicate moderate positive correlations:
  Age and Income, Education and Fitness, Usage and Income, Fitness and Income, Miles and Income.
  These relationships suggest that while there is some association among these variables, further analysis is needed to explore their implications.

Variation of Miles Run per Week Across Different Age Groups

```
bins = [18, 26, 34, 42, 50]
labels = ['18-26', '27-34', '35-42', '43-50']
age_groups = pd.cut(df['Age'], bins=bins, labels=labels, right=True)
miles_by_age_group=df.groupby([age_groups,'Gender'])['Miles'].mean().unstack(fill_value=0)
```

```
<ipython-input-89-3c7c8acd4fd1>:4: FutureWarning: The default of observed=False is deprecated and will be changed to True in a futur
  miles_by_age_group=df.groupby([age_groups,'Gender'])['Miles'].mean().unstack(fill_value=0)
```

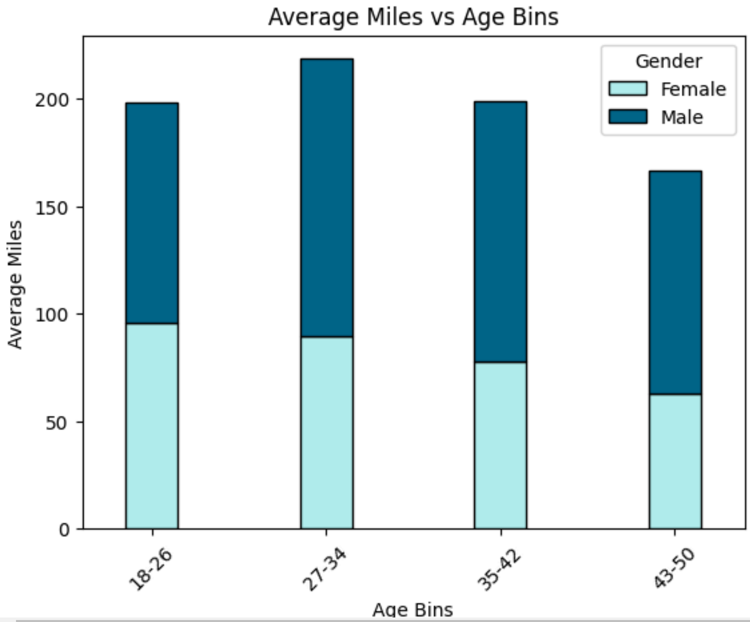
```
plt.figure(figsize=(10,5))
sns.set_palette(color)
sns.histplot(data=df, x='Miles',hue='Product',kde=True)
plt.show()
```



Although sale of KP281 is more than KP781 but most of KP781 users are logging more miles. This could indicate that KP781 is preferred by more serious runners.

```
# Question: How do average miles walked or run per week vary across different age groups?
bins = [18, 26, 34, 42, 50]
labels = ['18-26', '27-34', '35-42', '43-50']
age_groups = pd.cut(df['Age'], bins=bins, labels=labels, right=True)
miles_by_age_group=df.groupby([age_groups,'Gender'])['Miles'].mean().unstack(fill_value=0)
plt.figure(figsize=(8, 5))
sns.set_palette(color[-1:-4:-2])
miles_by_age_group.plot(kind= 'bar', stacked=True, width=0.3, edgecolor='Black')
plt.title('Average Miles vs Age Bins')
plt.xlabel('Age Bins')
plt.ylabel('Average Miles')
plt.xticks(rotation=45)
plt.show()
```

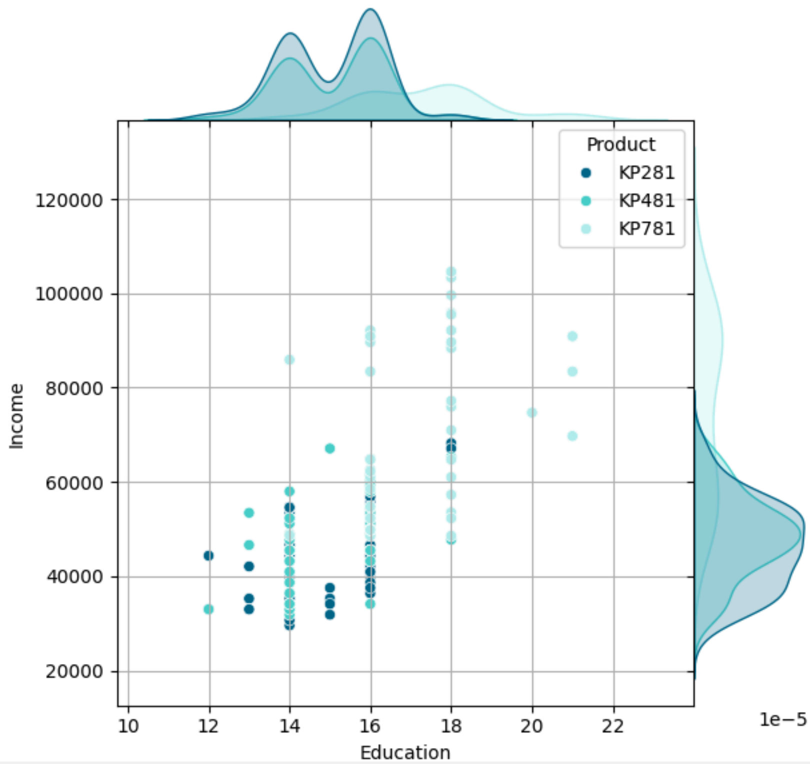
```
<Figure size 800x500 with 0 Axes>
```



Men of the age group 27-34 run the more miles than the other age groups.

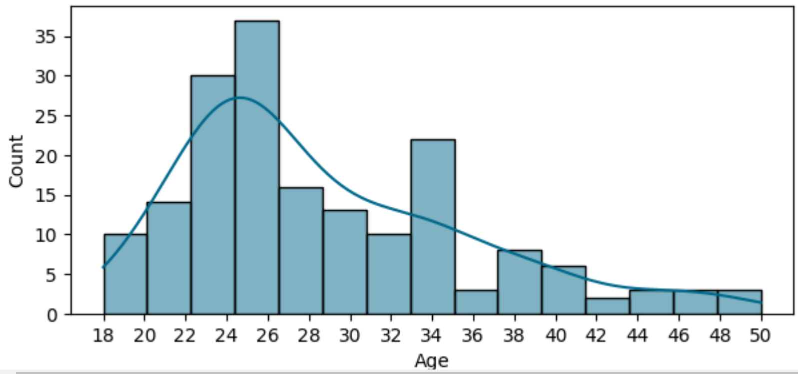
```
plt.figure(figsize=(10,6))
sns.set_palette(color)
sns.jointplot(x='Education', y='Income', hue='Product', data=df,space=0)
plt.grid(True)
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```

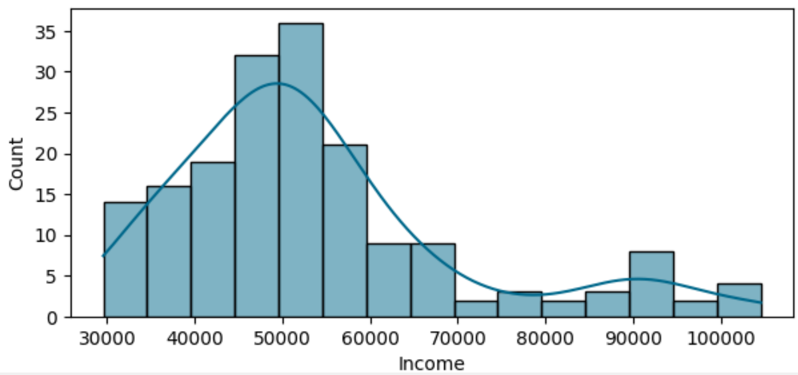


From the above plot we can infer that :
Consumers who are highly educated and have better income are more likely to buy KP781.
Consumers with 18 years of education are the most potential customers of KP781.
Futher investigation is needed to understand the trend of purchasing KP281 and KP481.

```
plt.figure(figsize=(7,3))
sns.set_palette(color)
sns.histplot(x='Age',data=df,kde=True, bins=15)
plt.xticks(np.arange(18,52,2))
plt.show()
```




```
plt.figure(figsize=(7,3))
sns.set_palette(color)
sns.histplot(x='Income',data=df,kde=True)
plt.show()
```



Since the bar graphs indicate a higher frequency of customers in the age group of 20 to 30 and an income range of $\$40,000$ to $\$60,000$, we will conduct a further analysis of this demographic segment before exploring additional consumer groups.

```
bracket=df.loc[(df['Income']>40000)&(df['Income']<60000)&(df['Age']>20)&(df['Age']<30)]
bracket.head()
```

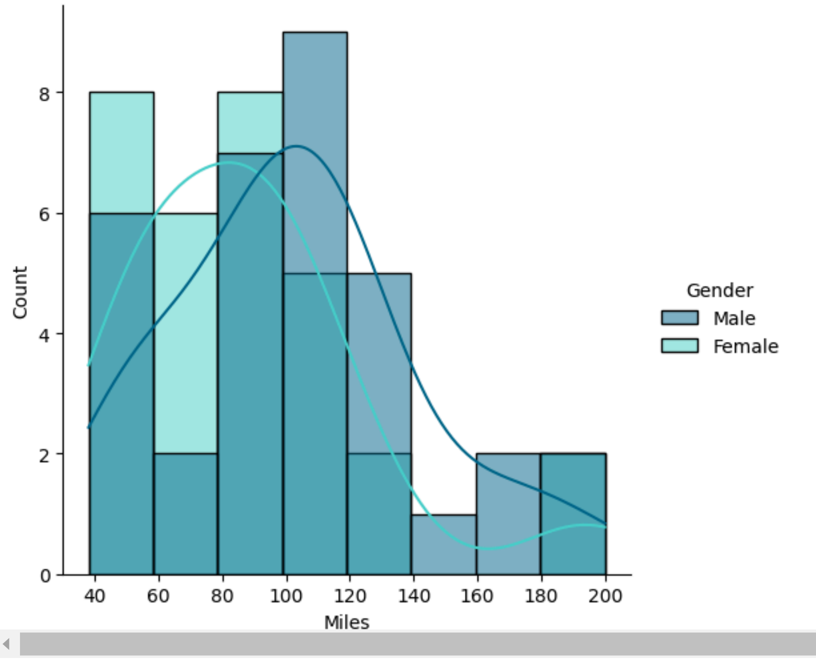
	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
15	KP281	23	Male	16	Partnered	3	3	40932	75
21	KP281	23	Male	16	Single	4	3	40932	94
22	KP281	24	Female	16	Single	4	3	42069	94
23	KP281	24	Female	16	Partnered	5	5	44343	188
24	KP281	24	Male	14	Single	2	3	45480	113

```
arr=[len(bracket),(len(df)-len(bracket))]
print(f"There are {arr[0]} consumers who lie within the age group 20 to 30 and income group $40,000 to $60,000 out of 180 consumers.")
```

```
There are 65 consumers who lie within the age group 20 to 30 and income group $40,000 to $60,000 out of 180 consumers.
```

```
plt.figure(figsize=(4,2))
sns.set_palette(color)
sns.displot(data=bracket, x='Miles',hue='Gender', kde=True)
plt.show()
```

```
<Figure size 400x200 with 0 Axes>
```

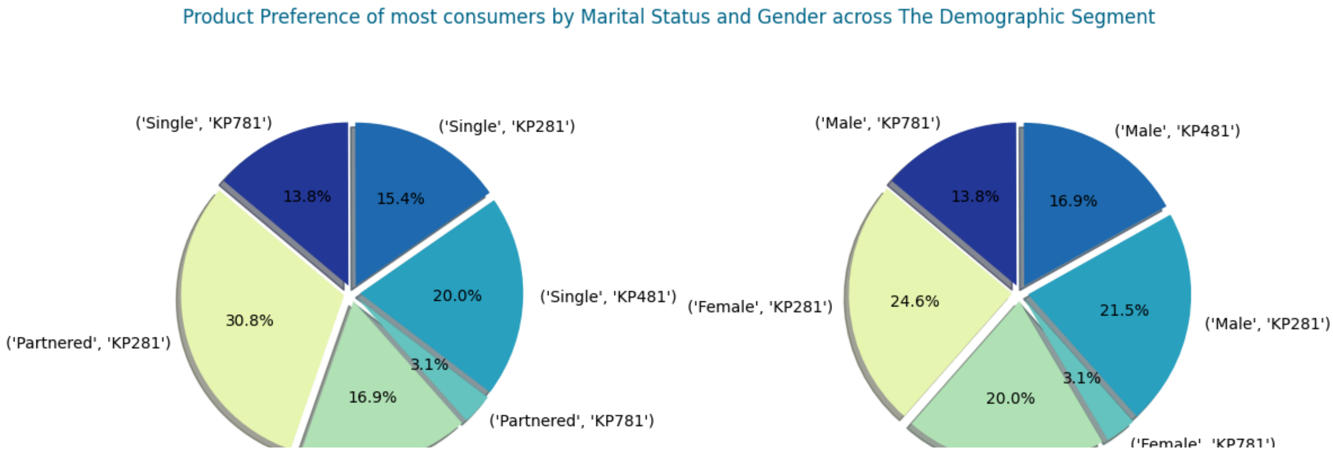


Most of the men in this group run approximately 100 to 120 miles, while most of the women run around 80 miles.

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(11,3.5))
sns.set_palette(color)
ax1.pie(
    bracket['MaritalStatus'].value_counts(),
    labels=bracket['MaritalStatus'].value_counts().index,
    autopct='%1.1f%%',
    startangle=140,
    wedgeprops={'edgecolor':'black','linewidth':1}
)
ax1.axis('equal')
ax2.pie(
    bracket['Gender'].value_counts(),
    labels=bracket['Gender'].value_counts().index,
    autopct='%1.1f%%',
    startangle=140,
    wedgeprops={'edgecolor':'black','linewidth':1}
)
ax2.axis('equal')
fig.suptitle('Marital Status and Gender Distribution acorss The Specific Demographic Segment',color=color[0])
plt.tight_layout()
plt.show()
```


```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12,5))
sns.set_palette('YlGnBu')
ax1.pie(
    bracket.groupby('MaritalStatus')['Product'].value_counts(),
    labels=bracket.groupby('MaritalStatus')['Product'].value_counts().index,
    autopct='%1.1f%%',
    startangle=140,
    explode=(0.05, 0.05, 0.05, 0.05,0.05, 0.05),
    shadow=True
)
ax1.axis('equal')
ax2.pie(
    bracket.groupby('Gender')['Product'].value_counts(),
    labels=bracket.groupby('Gender')['Product'].value_counts().index,
```

```
    autopct='%1.1f%%',
    startangle=140,
    explode=(0.05, 0.05, 0.05, 0.05,0.05, 0.05),
    shadow=True
)
ax2.axis('equal')
fig.suptitle('Product Preference of most consumers by Marital Status and Gender across The Demographic Segment',color=color[0])
plt.tight_layout()
plt.show()
```

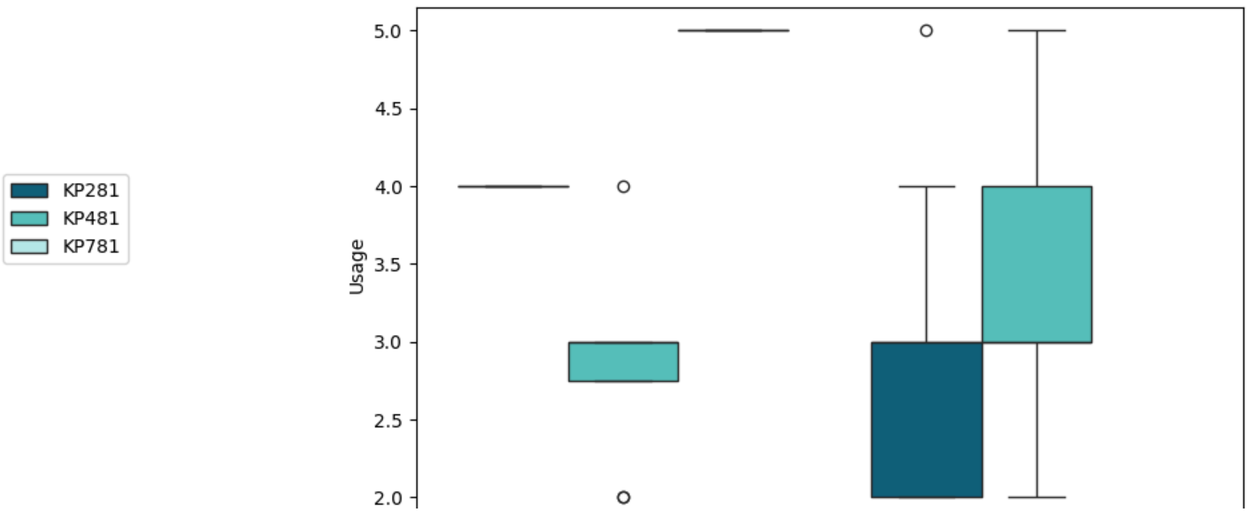


These charts show that majority of married consumer and women within the particular demographic segment prefer the treadmill model 'KP281', and they are less likely to buy 'KP781'. On the contrary, single men buy the most of 'KP781'.

Usage distribution of different products among women and men in this demographic segment

```
b_female=bracket.loc[(bracket['Gender']=='Female')]
```

```
plt.figure(figsize=(8,5))
sns.set_palette(color)
sns.boxplot(x=b_female['MaritalStatus'], y=df['Usage'], hue=df['Product'])
plt.legend(loc=(-0.5,0.5), ncol=1)
plt.show()
```



- KP481 is primarily preferred by single women of this particular bracket with a consistent usage pattern.
- KP281 seems to be a popular choice for both the married and single women. But single women exhibits higher usage index on average even though, there their subset is very small.
- KP781 is purchased by very few single women only with higher usage habits.

```
for (marital, product), group in b_female.groupby(['MaritalStatus', 'Product']):
  b=b_female.loc[(b_female['MaritalStatus']==marital)&(b_female['Product']==product)
  Q1 = b['Usage'].quantile(0.25)
  Q2 = b['Usage'].quantile(0.5)
  Q3 = b['Usage'].quantile(0.75)
  IQR = Q3 - Q1
```

```
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    print(f"Marital Status: {marital}, Product: {product}")
    print(f"Lower Bound: {lower_bound}, Median: {Q2}, Upper Bound: {upper_bound}")
    print("\n")
```

```
Marital Status: Partnered, Product: KP281
Lower Bound: 0.5, Median: 3.0, Upper Bound: 4.5


Marital Status: Partnered, Product: KP481
Lower Bound: 1.5, Median: 3.0, Upper Bound: 5.5


Marital Status: Single, Product: KP281
Lower Bound: 4.0, Median: 4.0, Upper Bound: 4.0


Marital Status: Single, Product: KP481
Lower Bound: 2.375, Median: 3.0, Upper Bound: 3.375


Marital Status: Single, Product: KP781
Lower Bound: 5.0, Median: 5.0, Upper Bound: 5.0
```

```
b_female.loc[((b_female['Product']=='KP781')& (b_female['MaritalStatus']=='Single'))|((b_female['Product']=='KP281')& (b_female['Marita
```

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
22	KP281	24	Female	16	Single	4	3	42069	94
26	KP281	24	Female	16	Single	4	3	46617	75
144	KP781	23	Female	18	Single	5	4	53536	100
148	KP781	24	Female	16	Single	5	5	52291	200

```
b_male=bracket.loc[(bracket['Gender']=='Male')]
```

```
plt.figure(figsize=(8,5))
sns.set_palette(color)
sns.boxplot(x=b_male['MaritalStatus'], y=df['Usage'], hue=df['Product'])
plt.legend(loc=(-0.5,0.5), ncol=1)
plt.show()
```