Aerofit Treadmill Analysis□

About Aerofit

Aerofit is an Indian fitness equipment brand operating under **Nityasach Fitness Pvt Ltd.** Established with the goal of making fitness accessible and affordable for the Indian market, Aerofit imports and distributes a wide range of fitness machinery for both personal home use and commercial gym facilities nationwide.

The brand's extensive product line encompasses all primary areas of physical fitness. Key offerings include a variety of cardiovascular machines such as manual and motorized **treadmills, elliptical trainers, and exercise bikes** (both upright and recumbent models). For strength training, they offer multi-station home gyms (like the popular Aerofit AF 4600 model), benches, dumbbells, weight plates, and resistance accessories. The equipment often features digital displays with pre-set workout programs and safety features.

Aerofit has built a strong reputation for offering durable, reliable equipment at competitive price points. Users and reviews generally commend the brand for its cost-effectiveness, sturdy construction, and dedicated after-sales service and customer support teams across different regions of India. This focus on affordability and quality has positioned Aerofit as a prominent and trusted name in the Indian fitness equipment industry.

Business Problem

The market research team at AeroFit wants **to identify the characteristics of the target audience for each type of treadmill** offered by the company, to provide a better recommendation of the treadmills to the new customers. The team decides to investigate whether there are differences across the product with respect to customer characteristics.

Aim of Project

Perform descriptive analytics **to create a customer profile for each AeroFit treadmill product** by developing appropriate tables and charts. For each AeroFit treadmill product, construct **two-way contingency tables** and compute all **conditional and marginal probabilities** along with their insights/impact on the business.

Dataset

The company collected the data on individuals who purchased a treadmill from the AeroFit stores during the prior three months. The dataset has the following features:

- **Product Purchased:** 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.

- **Income:** Annual income (in $)

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

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

---

Product Portfolio:

1)The `KP281` is an entry-level treadmill that sells for `$1,500` .

---

2)The `KP481` is for mid-level runners that sell for `$1,750` .

---

3)The `KP781` treadmill is having advanced features that sell for `$2,500`.

In [ ]:
```
# Dataset -> 'https://docs.google.com/spreadsheets/d/1OCpMzOL54Yt8mMjCyl0evxIGWGqUkUYUpHo5lf_rViQ/edit?usp=sharing'
# python file -> 'https://colab.research.google.com/drive/1OR_Ec_m5_WpUD-9ctPqw5LNVVPYUv_SS?usp=sharing'
```

Importing Libraries and Loading Dataset

In [ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```
In [ ]:
```
aerofit_treadmill = pd.read_csv('Aerofit treadmill CSV.csv')
aerofit_treadmill
```
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
...	...	...	...	...	...	...	...	...	...
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

Exploration of Data

In [ ]:
```
# making copy of the dataset
df = aerofit_treadmill.copy()
df
```

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
...	...	...	...	...	...	...	...	...	...
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 [ ]:
```
# First 5 rows of the data
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 [ ]:
```
# last 5 rows of the data
df.tail()
```
Out[ ]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
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

In [ ]:
```
df.shape
```
Out[ ]:
```
(180, 9)
```
In [ ]:
```
print(f"Number of Rows' : {df.shape[0]}\n'Number of Columns' : {df.shape[1]}")
```
```
'Number of Rows' : 180
'Number of Columns' : 9
```
In the given dataset, we have 9 columns and 180 rows.

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
```

In [ ]:

df.dtypes

Out[ ]:

	0
Product	object
Age	int64
Gender	object
Education	int64
MaritalStatus	object
Usage	int64
Fitness	int64
Income	int64
Miles	int64

dtype: object

Displaying types of each columns :-

- Only three columns Product, Gender, and MaritialStatus are in `object` data type.
- The remainning six columns are in the `int64` .

In [ ]:

df.duplicated().sum()

Out[ ]:

np.int64(0)

There are no duplicate values.

In [ ]:

df.isnull().sum()

Out[ ]:

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

dtype: int64
There are no null values present in the dataset.

Statistical Summary

In [ ]:

```
df.describe(include = 'all')
```

Out[ ]:

	Product	Age	Gender	Education	MaritalStatus	Usage	Fitness	Income	Miles
count	180	180.000000	180	180.000000	180	180.000000	180.000000	180.000000	180.000000
unique	3	NaN	2	NaN	2	NaN	NaN	NaN	NaN
top	KP281	NaN	Male	NaN	Partnered	NaN	NaN	NaN	NaN
freq	80	NaN	104	NaN	107	NaN	NaN	NaN	NaN
mean	NaN	28.788889	NaN	15.572222	NaN	3.455556	3.311111	53719.577778	103.194444
std	NaN	6.943498	NaN	1.617055	NaN	1.084797	0.958869	16506.684226	51.863605
min	NaN	18.000000	NaN	12.000000	NaN	2.000000	1.000000	29562.000000	21.000000
25%	NaN	24.000000	NaN	14.000000	NaN	3.000000	3.000000	44058.750000	66.000000
50%	NaN	26.000000	NaN	16.000000	NaN	3.000000	3.000000	50596.500000	94.000000
75%	NaN	33.000000	NaN	16.000000	NaN	4.000000	4.000000	58668.000000	114.750000
max	NaN	50.000000	NaN	21.000000	NaN	7.000000	5.000000	104581.000000	360.000000

- **Product KP281** is the most popular, with 80 out of 180 customers (44%) owning it. This suggests it's likely the entry-level or standard model.
- **Age:** The average age is ~29 years, with a range from 18 to 50. The middle 50% of customers are between 24 and 33 years old (the IQR), indicating the primary target market is young professionals and adults.
- **Gender:** The customer base is skewed towards males, who make up 104 of the 180 observations (58%).
- **Marital Status:** A slight majority of the customers are Partnered (107 out of 180, or 59%).
- **Education:** The average education level is ~16 years.
- **Usage:** Customers use their treadmill, on average, ~3.5 times per week. The most common usage frequency (mode) is 3 times a week. However, there are highly dedicated users, with a maximum usage of 7 days a week.
- **Fitness Level:** The average self-rated fitness is 3.3 out of 5. The 25th to 75th percentile is 3 to 4, meaning most customers consider themselves to be in "average" to "good" shape.
- **Miles Run:** Customers run an average of ~103 miles per week. The distribution is right-skewed, as the maximum (360 miles) is far from the 75th percentile (115 miles). This indicates the presence of a small number of extreme "super-users" or marathon trainers.
- **Income:** The average annual income is `$53,720 . Incomes vary widely (std dev ~ $16,507), ranging fro $29,562 to $104,581` . This wide spread suggests the product line may cater to different income segements.

In [ ]:

Data Cleaning

In [ ]:

```
# Converting categorical columns to category dtype
cat_cols = ['Product', 'Gender', 'MaritalStatus']
df[cat_cols] = df[cat_cols].astype('category')

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
```

Explanation

- **Product, Gender, MaritalStatus** are categorical because they represent labels or groups — not numeric quantities.
- Converting them to category optimizes memory and makes analysis cleaner (e.g., for groupby, summary, encoding later).

Non-Graphical Analysis:

In [ ]:

```
df.nunique()
```

Out[ ]:

	0
Product	3
Age	32
Gender	2
Education	8
MaritalStatus	2
Usage	6
Fitness	5
Income	62
Miles	37

dtype: int64

Observations

- Product: 3 unique models (KP281, KP481, KP781)
- Gender: 2 unique categories (Male, Female)
- MaritalStatus: 2 categories (Single, Partnered)
- Usage: 6 unique categories (2–7 times/week)
- Fitness: 5 unique categories (1–5 rating scale)

This shows the dataset is relatively clean with well-defined categories.

A. Product Distribution

In [ ]:

```
prod_counts = df['Product'].value_counts()
prod_percent = df['Product'].value_counts(normalize=True) * 100
pd.concat([prod_counts, prod_percent], axis=1, keys=['Count','Percentage'])
```

Out[ ]:

	Count	Percentage
Product		
KP281	80	44.444444
KP481	60	33.333333
KP781	40	22.222222

Insight

- KP281 - 44.4% ~ Most popular entry-level product
- KP481 - 33.3% ~ Mid-tier product
- KP781 - 22.2% ~ Premium model

Business insight:

```
- KP281 dominates sales, indicating price-sensitive customers.
- KP781 appeals to high-income, advanced users (validated later in analysis).
```

B. Gender Distribution

In [ ]:

```
gender_counts = df['Gender'].value_counts()
gender_percent = df['Gender'].value_counts(normalize=True) * 100
pd.concat([gender_counts, gender_percent], axis=1, keys=['Count','Percentage'])
```

Out[ ]:

	Count	Percentage
Gender		
Male	104	57.777778
Female	76	42.222222

Insight

- Male ~ 58%
- Female ~ 42%

More men buy treadmills than women. So, companies should aim most of their ads at men, but they shouldn't forget about women, because many women buy them too.

C. Marital Status Distribution

In [ ]:

```
marital_counts = df['MaritalStatus'].value_counts()
marital_percent = df['MaritalStatus'].value_counts(normalize=True) * 100
pd.concat([marital_counts, marital_percent], axis=1, keys=['Count','Percentage'])
```

Out[ ]:

	Count	Percentage
MaritalStatus		
Partnered	107	59.444444
Single	73	40.555556

Insight

- Partnered ~ 59%
- Single ~ 41%

More partnered individuals purchase treadmills – possibly influenced by shared household income.

D. Usage (Times per Week)

In [ ]:

```
usage_counts = df['Usage'].value_counts().sort_index()
usage_percent = df['Usage'].value_counts(normalize=True).sort_index()*100
pd.concat([usage_counts, usage_percent], axis=1, keys=['Count','Percentage'])
```

Out[ ]:

	Count	Percentage
Usage		
2	33	18.333333
3	69	38.333333
4	52	28.888889
5	17	9.444444
6	7	3.888889
7	2	1.111111

Insight

- Most buyers use the treadmill 3–4 times/week
- Very few extreme users (6–7 times) ~ active fitness enthusiasts

E. Fitness Level Distribution

In [ ]:

```
fitness_counts = df['Fitness'].value_counts().sort_index()
fitness_percent = df['Fitness'].value_counts(normalize=True).sort_index()*100
pd.concat([fitness_counts, fitness_percent], axis=1, keys=['Count','Percentage'])
```

Out[ ]:

	Count	Percentage
Fitness		
1	2	1.111111
2	26	14.444444
3	97	53.888889
4	24	13.333333
5	31	17.222222

Insight

- Majority rate themselves 3 or 4 (average to good fitness)
- Very few rate themselves as 1 or 5

Summary of Key Non-Graphical Insights

- Product KP281 is the dominant model (44.4%), indicating strong demand for the entry-level segment.
- Male customers form a slight majority (58%), but female participation remains significant.
- Partnered customers (59%) buy more treadmills, possibly due to shared financial resources or joint fitness goals.
- Most customers plan moderate weekly usage (3–4 days), implying the target audience is not professional athletes but general fitness seekers.
- Fitness levels cluster around 3–4, showing that average-fit consumers are the core buyers.

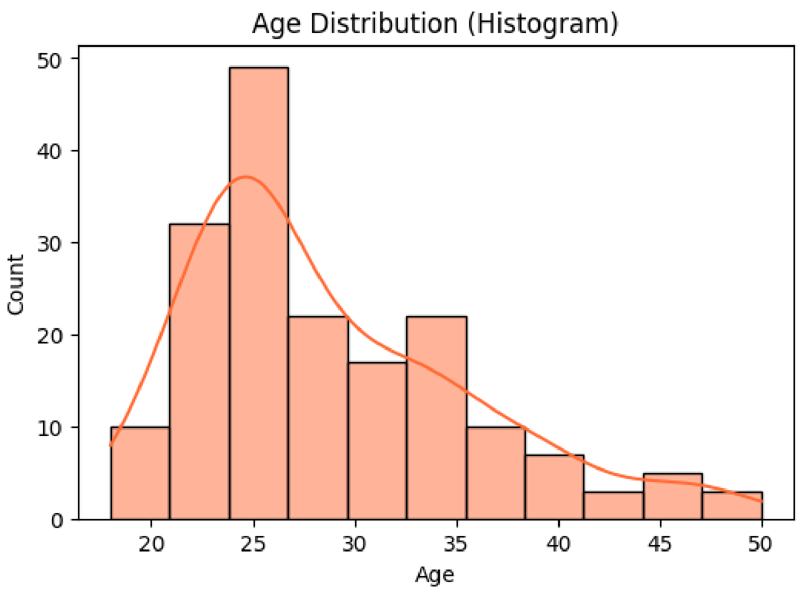
Visual Analysis

Univariate Analysis

1. Age Distribution

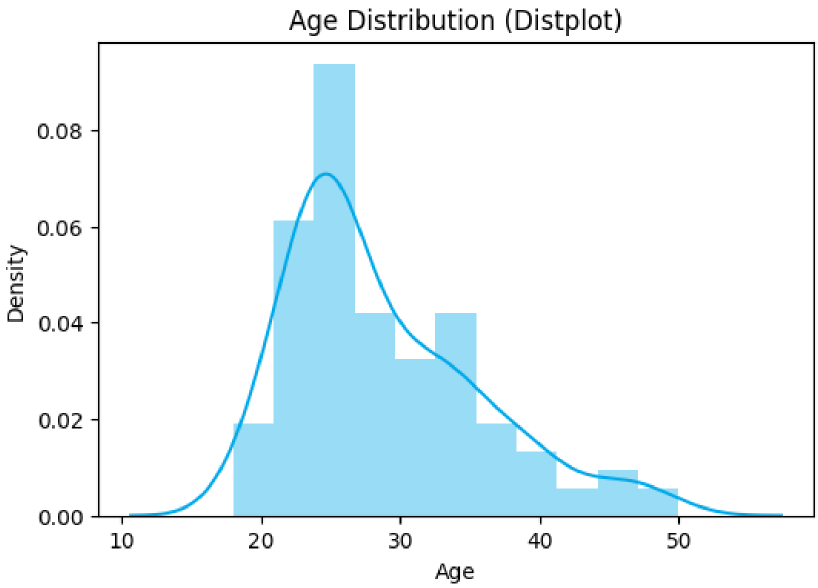
In [ ]:

```
plt.figure(figsize=(6,4))
sns.histplot(df['Age'], kde=True,color='#FF6B35')
plt.title("Age Distribution (Histogram)")
plt.show()
```



In [ ]:

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

plt.figure(figsize=(6,4))
sns.distplot(df['Age'], kde=True,color='#00A8E8')
plt.title("Age Distribution (Distplot)")
plt.show()
```



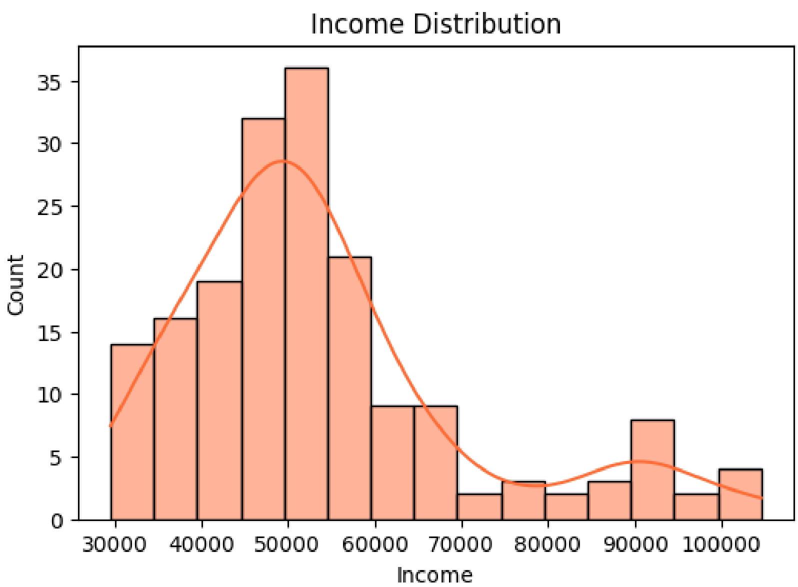
Insight:

- Age is concentrated between 22–35 years, showing younger adults are the primary buyers.
- Distribution is slightly right-skewed → few older buyers (40+).
- This indicates marketing should focus on young working professionals.

2. Income Distribution

In [ ]:

```
plt.figure(figsize=(6,4))
sns.histplot(df['Income'], kde=True, color='#FF6B35')
plt.title("Income Distribution")
plt.show()
```



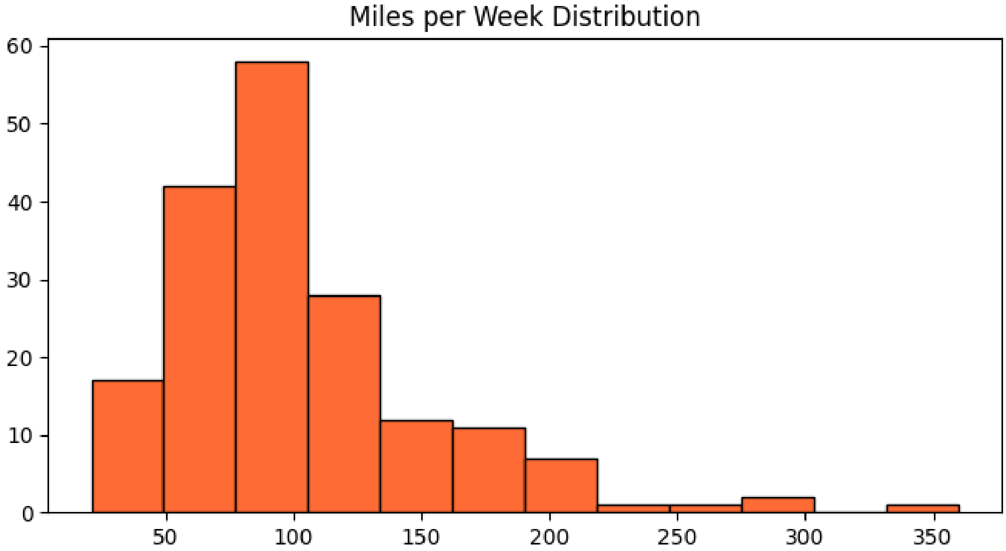
Insight:

- Income ranges widely from thirty thousand dollars to One hundred four thousand dollars.
- Distribution shows a right tail, meaning fewer high-income customers.
- Different treadmill models likely correspond to income levels.

3. Miles Distribution

In [ ]:

```
plt.figure(figsize=(8,4))
plt.hist(df['Miles'], bins=12, color='#FF6B35',edgecolor='black')
plt.title("Miles per Week Distribution")
plt.show()
```



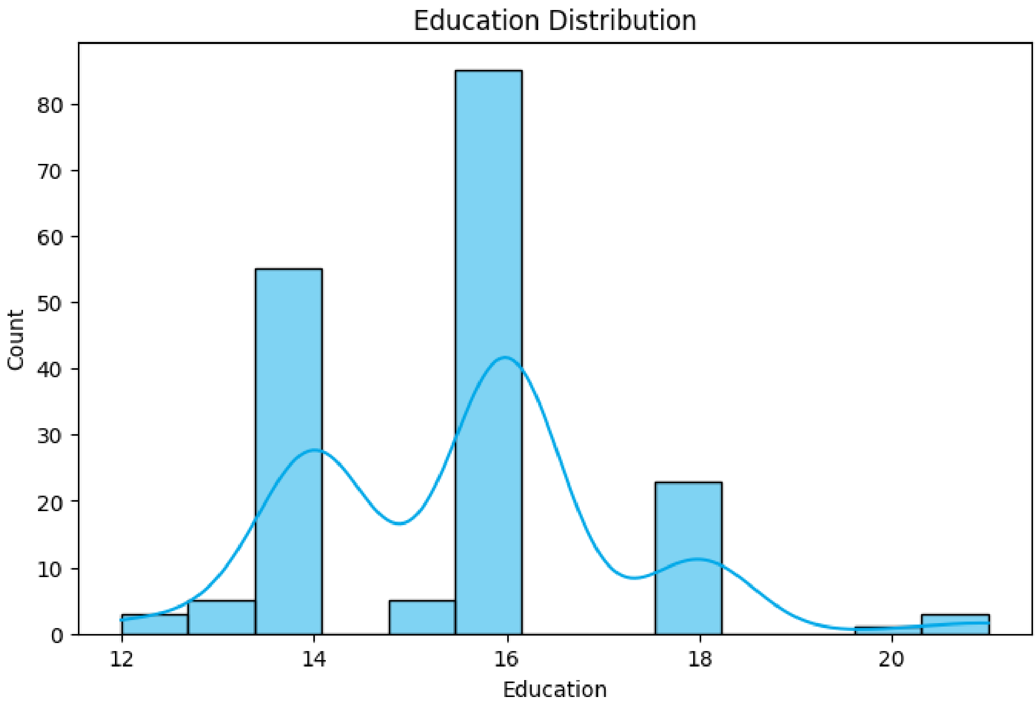
Insight:

- Most customers run 50–150 miles per week.
- Very high values (200+ miles) indicate a small group of advanced fitness users.

4. Education Distribution

In [ ]:

```
plt.figure(figsize=(8,5))
sns.histplot(df['Education'], kde=True,color='#00A8E8')
plt.title("Education Distribution")
plt.show()
```



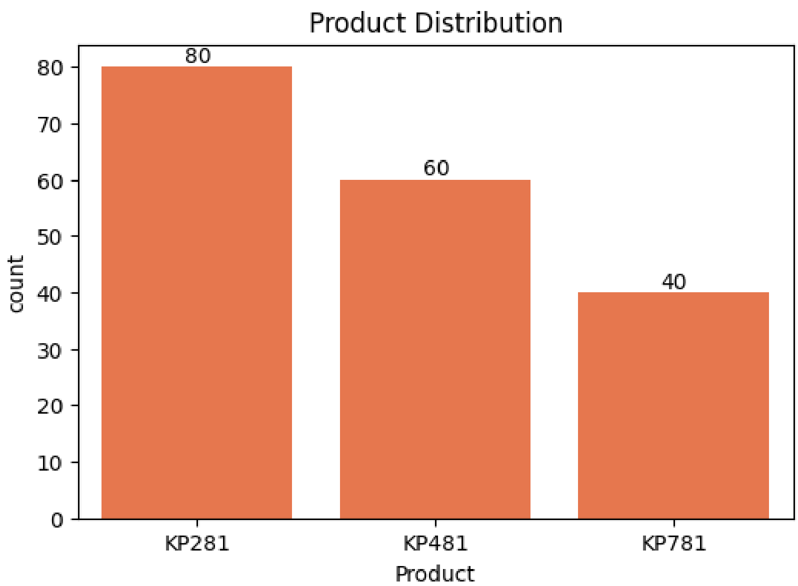
Insight:

- Education clusters around 14–16 years, meaning most buyers are graduates or postgraduates.
- Higher education levels often correlate with health-awareness and disposable income.

5. Product Count

In [ ]:

```
plt.figure(figsize=(6,4))
ax = sns.countplot(x='Product', data=df, color='#FF6B35')
plt.title("Product Distribution")

ax.bar_label(ax.containers[0])

plt.show()
```



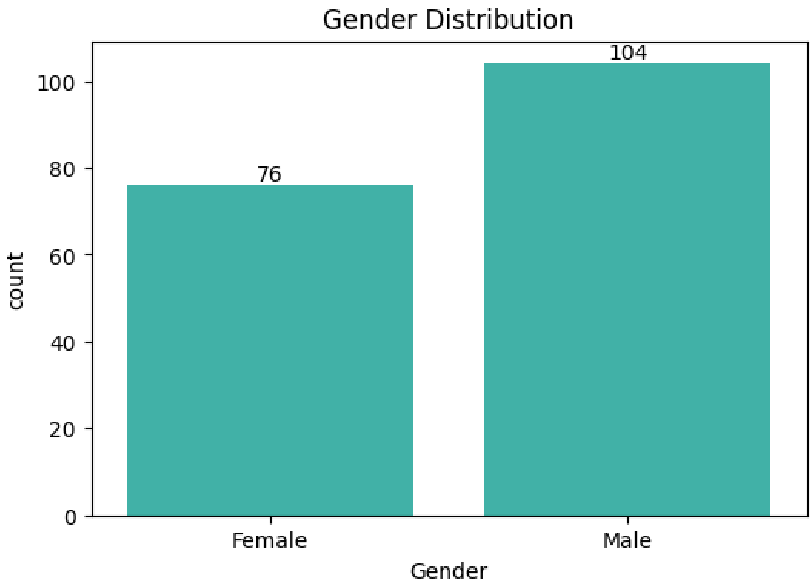
Insights :

- The product 'KP281' is most purchased product i.e. 80.

6. Gender Count

In [ ]:

```
plt.figure(figsize=(6,4))
ax = sns.countplot(x='Gender', data=df, color='#2EC4B6')
plt.title("Gender Distribution")
ax.bar_label(ax.containers[0])
plt.show()
```



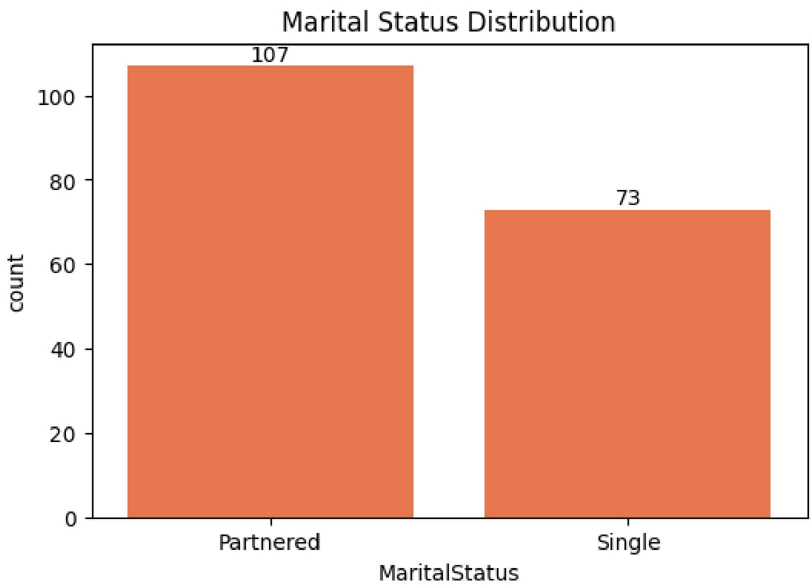
Insight

- Male customers slightly dominate purchases.

7. Marital Status

In [ ]:

```
plt.figure(figsize=(6,4))
ax = sns.countplot(x='MaritalStatus', data=df, color='#FF6B35')
plt.title("Marital Status Distribution")
ax.bar_label(ax.containers[0])
plt.show()
```



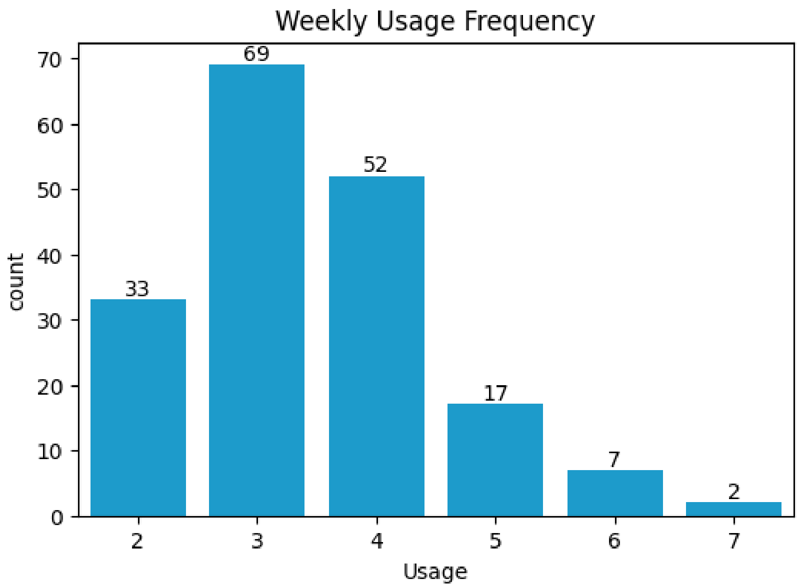
Insight

- Partnered customers are the majority.

8. Usage (Times per Week)

In [ ]:

```
plt.figure(figsize=(6,4))
ax = sns.countplot(x=df['Usage'], color='#00A8E8')
plt.title("Weekly Usage Frequency")
ax.bar_label(ax.containers[0])
plt.show()
```



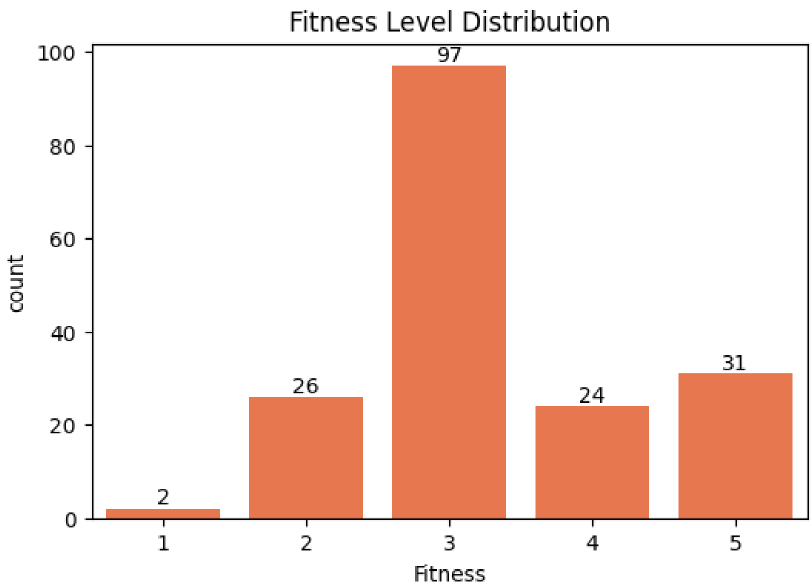
Insight

- Most people use the treadmill 3–4 times per week.

9. Fitness Level

In [ ]:

```
plt.figure(figsize=(6,4))
ax = sns.countplot(x=df['Fitness'], color='#FF6B35')
plt.title("Fitness Level Distribution")
ax.bar_label(ax.containers[0])
plt.show()
```



Insight

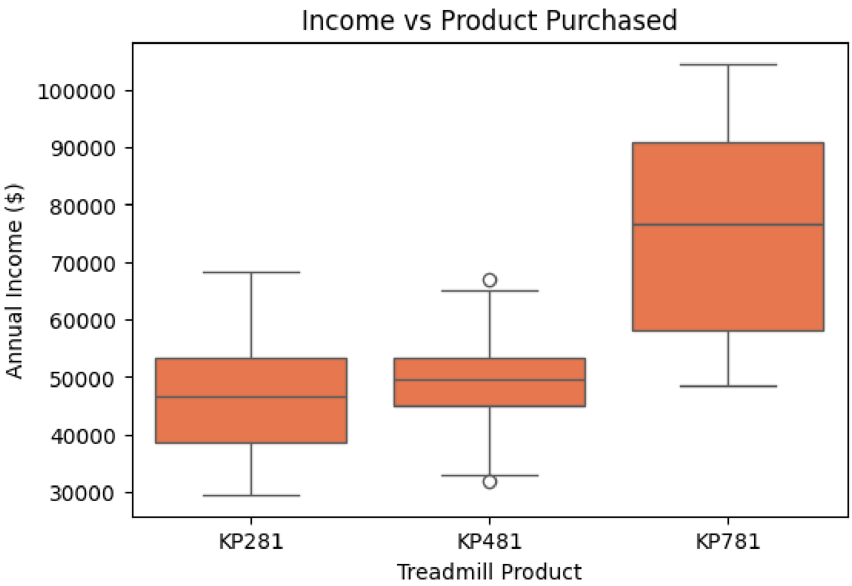
- Fitness rating is mostly 3–4, representing average fitness.

Bivariate Analysis

1. Income vs Product

In [ ]:

```
plt.figure(figsize=(6,4))
sns.boxplot(x='Product', y='Income', data=df, color='#FF6B35')
plt.title("Income vs Product Purchased")
plt.xlabel("Treadmill Product")
plt.ylabel("Annual Income ($)")
plt.show()
```



Insight:

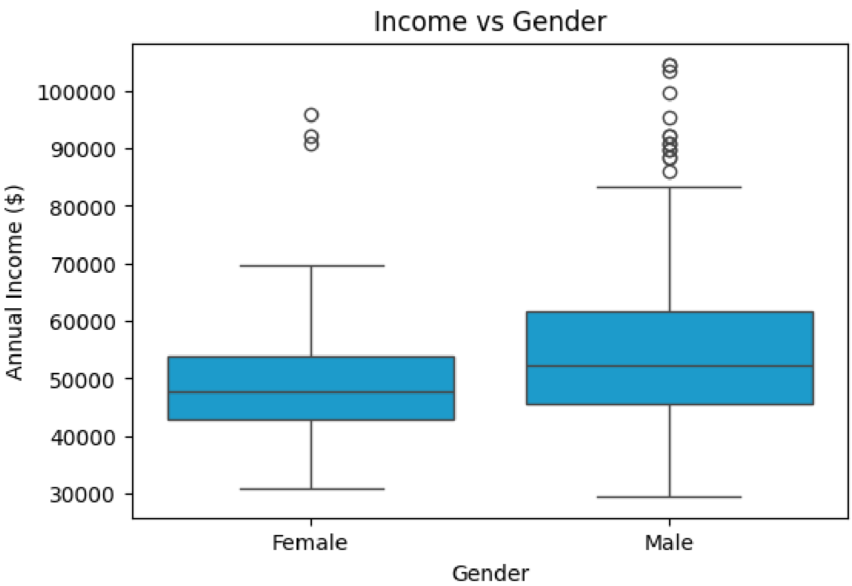
- KP781 buyers have the highest income -> premium segment.
- KP281 buyers belong to a lower income segment -> price-sensitive group.
- KP481 buyers fall in the middle income range.

This confirms pricing strategy is aligned with income segmentation.

2.Income vs Gender

In [ ]:

```
plt.figure(figsize=(6,4))
sns.boxplot(x='Gender', y='Income', data=df, color='#00A8E8')
plt.title("Income vs Gender")
plt.xlabel("Gender")
plt.ylabel("Annual Income ($)")
plt.show()
```



Insight:

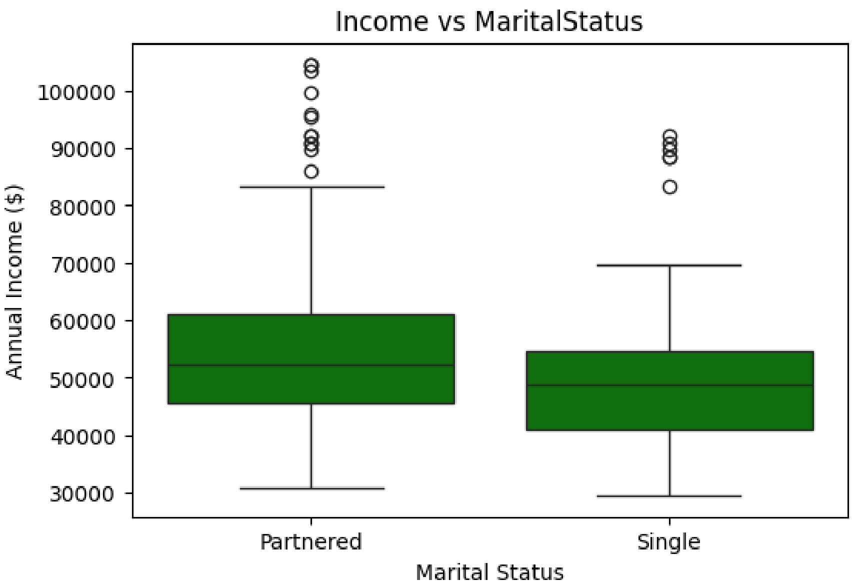
- Male customers show higher average income levels -> key target for premium treadmill models.
- Female customers have relatively lower income distribution -> more price-sensitive segment

This indicates gender-based income segmentation should inform product positioning and financing options.

3. Income vs Marital Status

In [ ]:

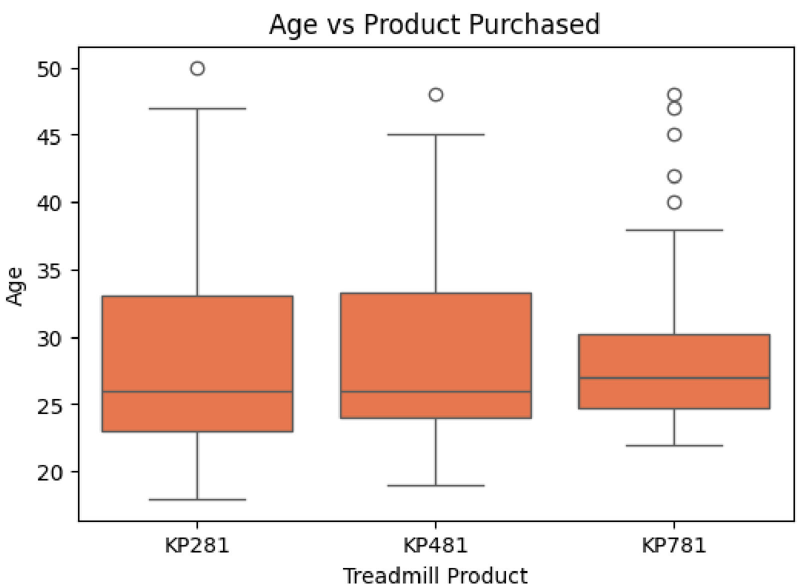
```
plt.figure(figsize=(6,4))
sns.boxplot(x='MaritalStatus', y='Income', data=df, color='green')
plt.title("Income vs MaritalStatus")
plt.xlabel("Marital Status")
plt.ylabel("Annual Income ($)")
plt.show()
```



4. Age vs Product

In [ ]:

```
plt.figure(figsize=(6,4))
sns.boxplot(x='Product', y='Age', data=df, color='#FF6B35')
plt.title("Age vs Product Purchased")
plt.xlabel("Treadmill Product")
plt.ylabel("Age")
plt.show()
```



Insight

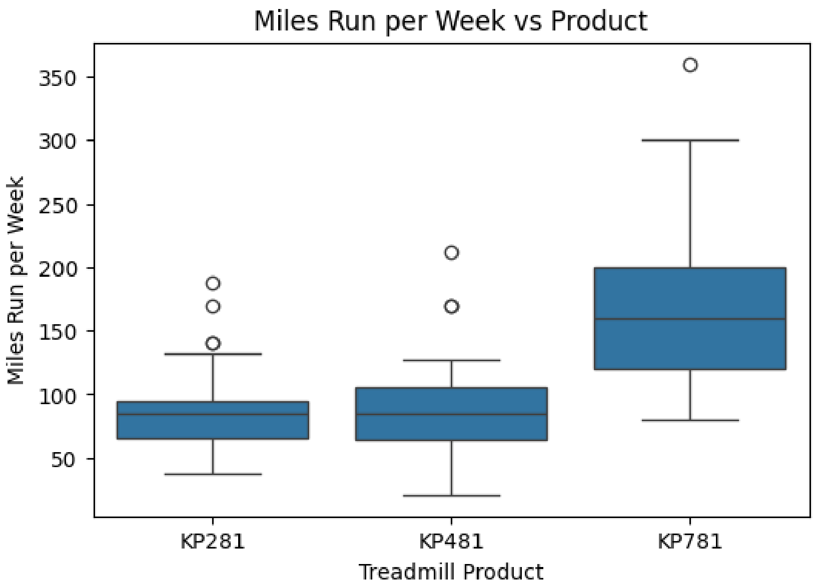
- KP281 customers are youngest.
- KP781 customers are mostly older (30–45).

Younger people prefer cheaper models; older customers invest in advanced ones.

5. Miles vs Product

In [ ]:

```
plt.figure(figsize=(6,4))
sns.boxplot(x='Product', y='Miles', data=df)
plt.title("Miles Run per Week vs Product")
plt.xlabel("Treadmill Product")
plt.ylabel("Miles Run per Week")
plt.show()
```



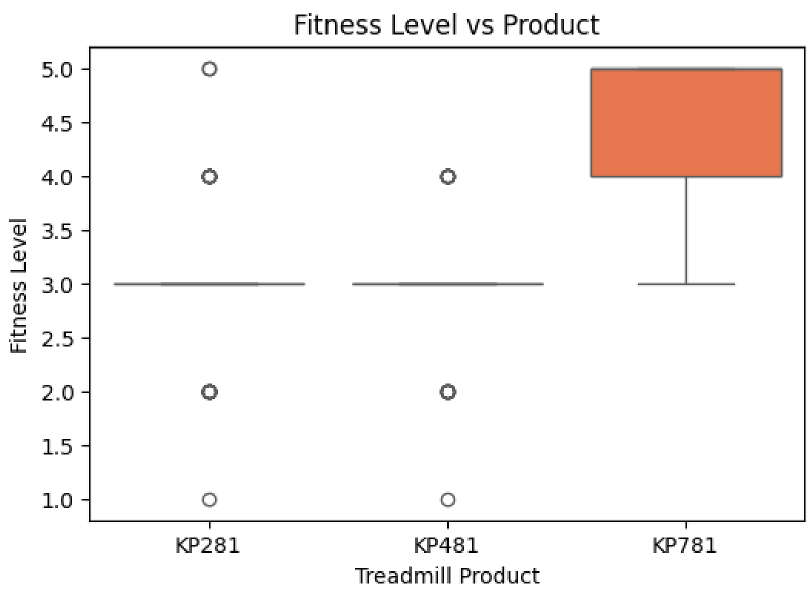
Insight:

- KP781 customers run more miles -> advanced / high-usage users.
- KP281 customers run least miles -> casual users.

6. Fitness vs Product

In [ ]:

```
plt.figure(figsize=(6,4))
sns.boxplot(x='Product', y=df['Fitness'].astype(int), data=df,color='#FF6B35')
plt.title("Fitness Level vs Product")
plt.xlabel("Treadmill Product")
plt.ylabel("Fitness Level")
plt.show()
```



Insight:

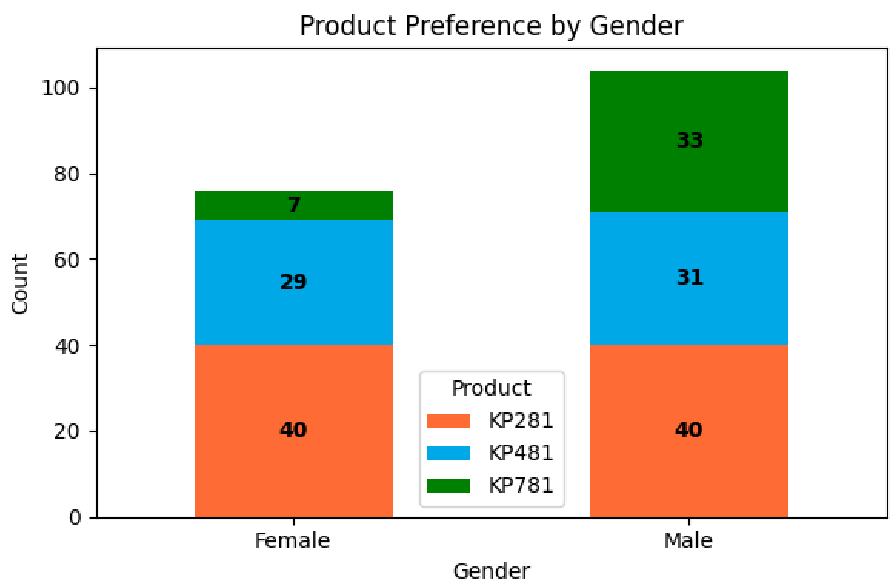
- KP781 buyers have higher fitness scores.
- KP281 buyers have lower fitness levels.

Different product tiers successfully target different fitness levels.

7. Product vs Gender

In [ ]:

```
product_gender = pd.crosstab(df['Gender'], df['Product'])

ax = product_gender.plot(kind='bar', stacked=True,
                    color=['#FF6B35', '#00A8E8', 'green'],
                    figsize=(6,4))

plt.title("Product Preference by Gender")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.legend(title='Product')
plt.xticks(rotation=0)

for container in ax.containers:
    ax.bar_label(container, label_type='center', fontsize=10, fontweight='bold')

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



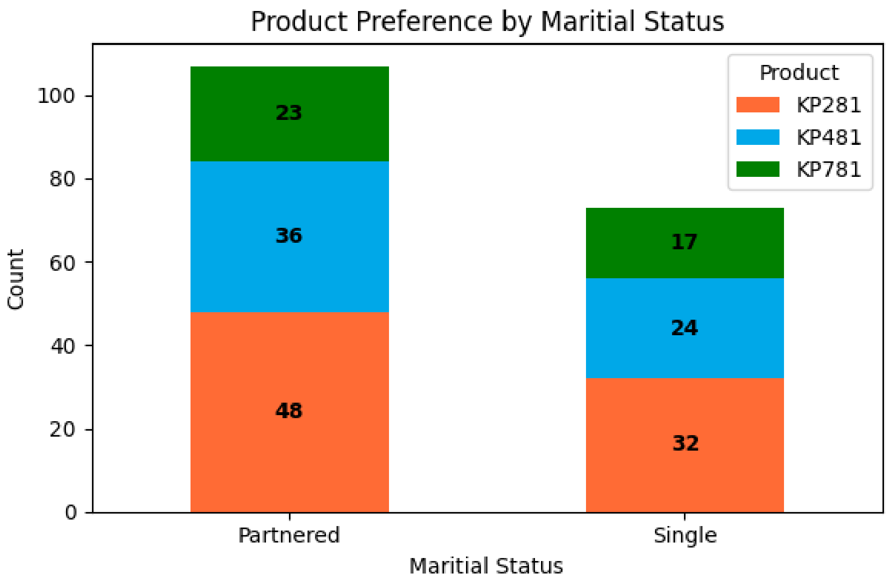
Insight

- The gender-based preferences show clear differences for high-end products.
- Entry-level KP281 is equally preferred by males and females, indicating mass appeal.
- Mid-range KP481 shows nearly equal adoption across genders.
- Premium KP781, however, has a significantly higher purchase rate among male customers (33 males vs 7 females).

8. Product vs Marital Status

In [ ]:

```
product_gender = pd.crosstab(df['MaritalStatus'], df['Product'])

ax = product_gender.plot(kind='bar', stacked=True,
                         color=['#FF6B35', '#00A8E8', 'green'],
                         figsize=(6,4))

plt.title("Product Preference by Maritial Status")
plt.xlabel("Maritial Status")
plt.ylabel("Count")
plt.legend(title='Product')
plt.xticks(rotation=0)

for container in ax.containers:
    ax.bar_label(container, label_type='center', fontsize=10, fontweight='bold')

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



Insights

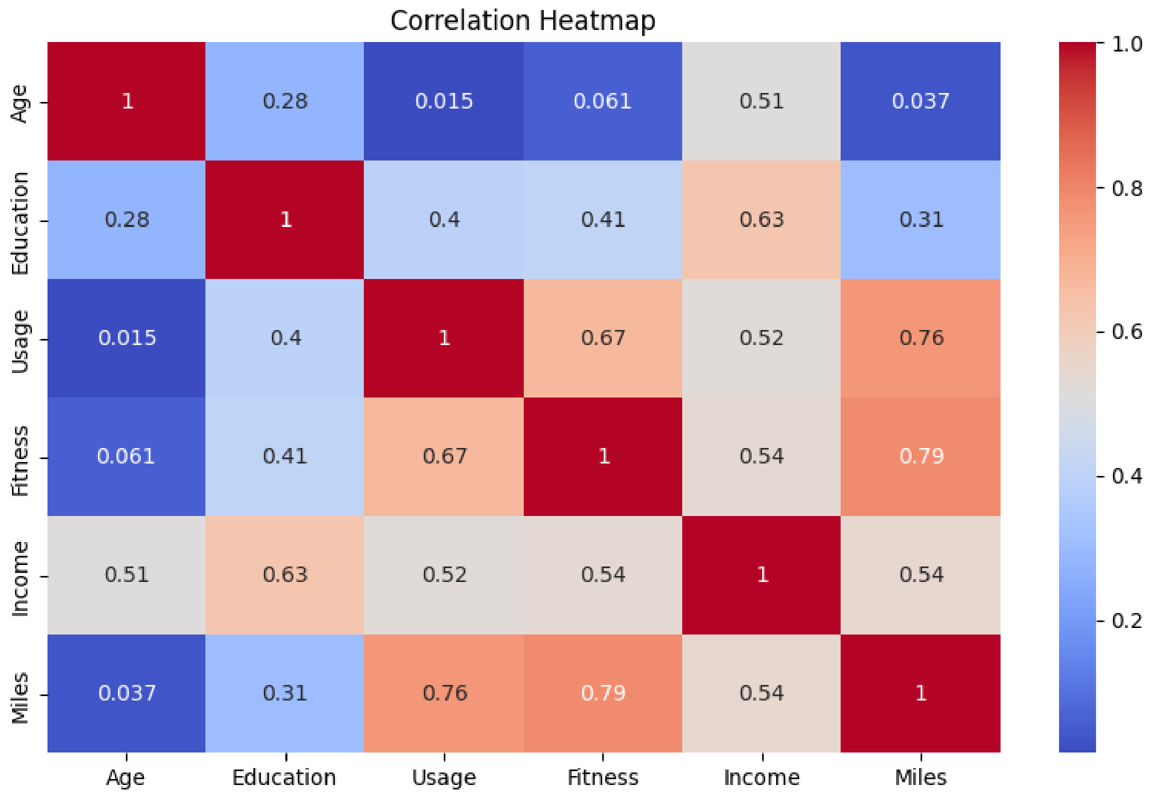
Partnered customers consistently purchase more treadmills than single customers across all product categories. However, both groups follow the same preference pattern, favoring the entry-level KP281 the most and the premium KP781 the least. This suggests that marital status impacts purchasing volume but not product preference. Aerofit can target partnered households with bundled or couple-oriented fitness promotions, while offering value-driven campaigns for single customers.

Correlation — Heatmap & Pairplot

1. Correlation Heatmap

In [ ]:

```
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```



Insights:

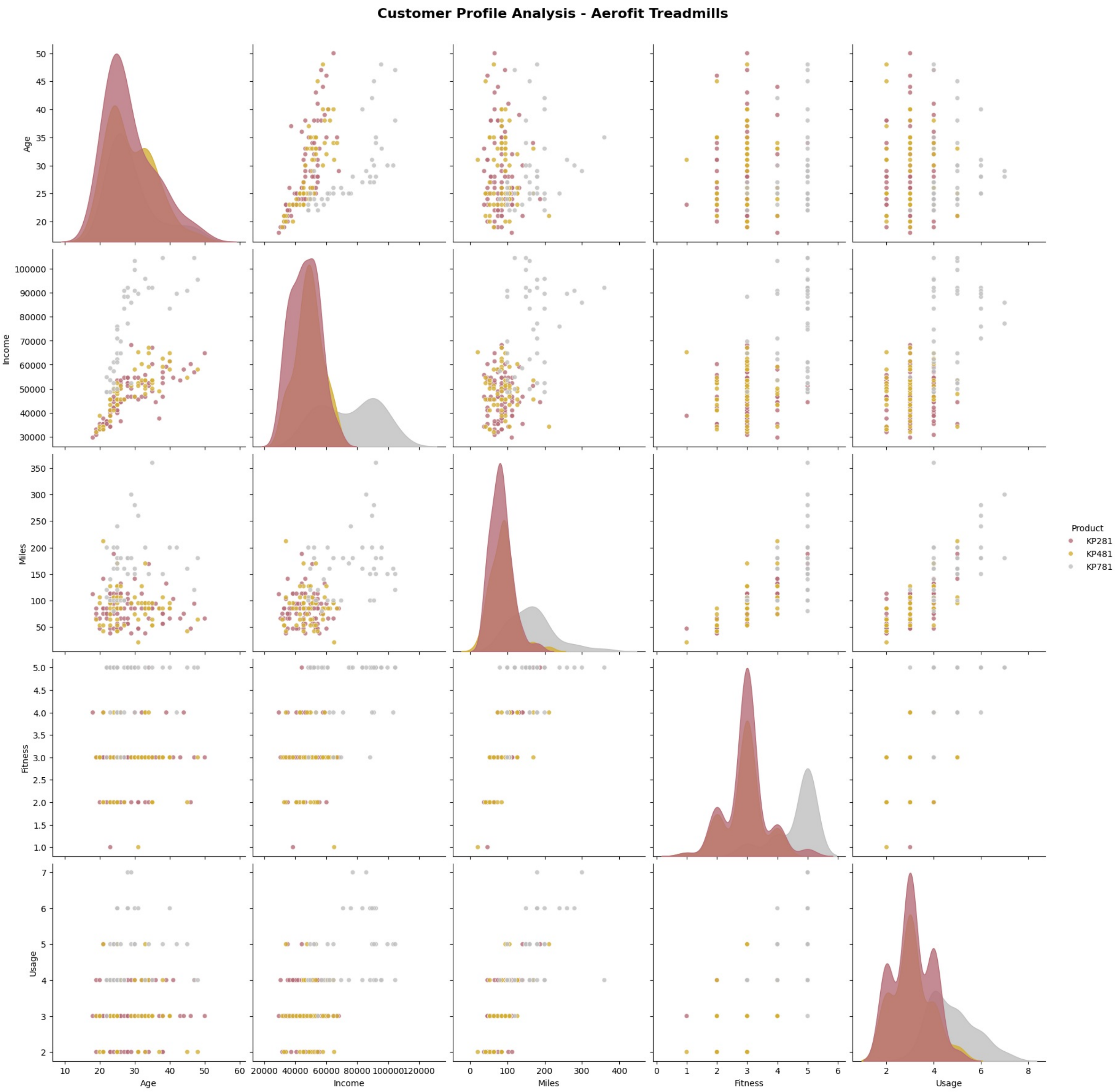
- Miles and Usage show strong positive correlation -> frequent users run more miles.
- Income is moderately correlated with Miles and Fitness -> higher-income customers are more fitness-oriented.
- Age has weak correlation with most attributes.

This helps understand which variables drive treadmill usage.

2. Pairplot

In [ ]:

```
sns.pairplot(df[['Age','Income','Miles','Fitness','Usage','Product']],
        hue='Product',
        palette=['#B76E79', '#D4AF37', '#C0C0C0'],
        diag_kind='kde',
        height=3.5,
        plot_kws={'alpha': 0.8, 's': 30, 'edgecolor': 'white', 'linewidth': 0.5},
        diag_kws={'alpha': 0.8})

plt.suptitle("Customer Profile Analysis - Aerofit Treadmills",
        y=1.02, fontsize=16, fontweight='bold')
plt.show()
```



Insights

The pairplot analysis reveals clear segmentation among Aerofit treadmill customers based on demographic and behavioral attributes. Higher-tier product KP781 is consistently preferred by older, higher-income customers who demonstrate stronger fitness levels, higher weekly usage, and greater running mileage. In contrast, KP281 attracts younger, lower-income buyers with lighter usage patterns and lower fitness scores, while KP481 occupies a balanced middle ground across all variables. Strong clustering patterns show income, fitness, and mileage as key drivers of product choice, with notable positive relationships such as higher usage corresponding to greater miles run. Overall, the visual analysis confirms distinct customer personas for each product tier, providing valuable insights for targeted marketing and product positioning.

Missing Value & Outlier Detection

In [ ]:

```
# Checking for missing values
df.isnull().sum()
```

Out[ ]:

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

dtype: int64

Insight:

There are no missing values in the dataset. Data is clean and ready for analysis.

A) Outlier Detection: AGE

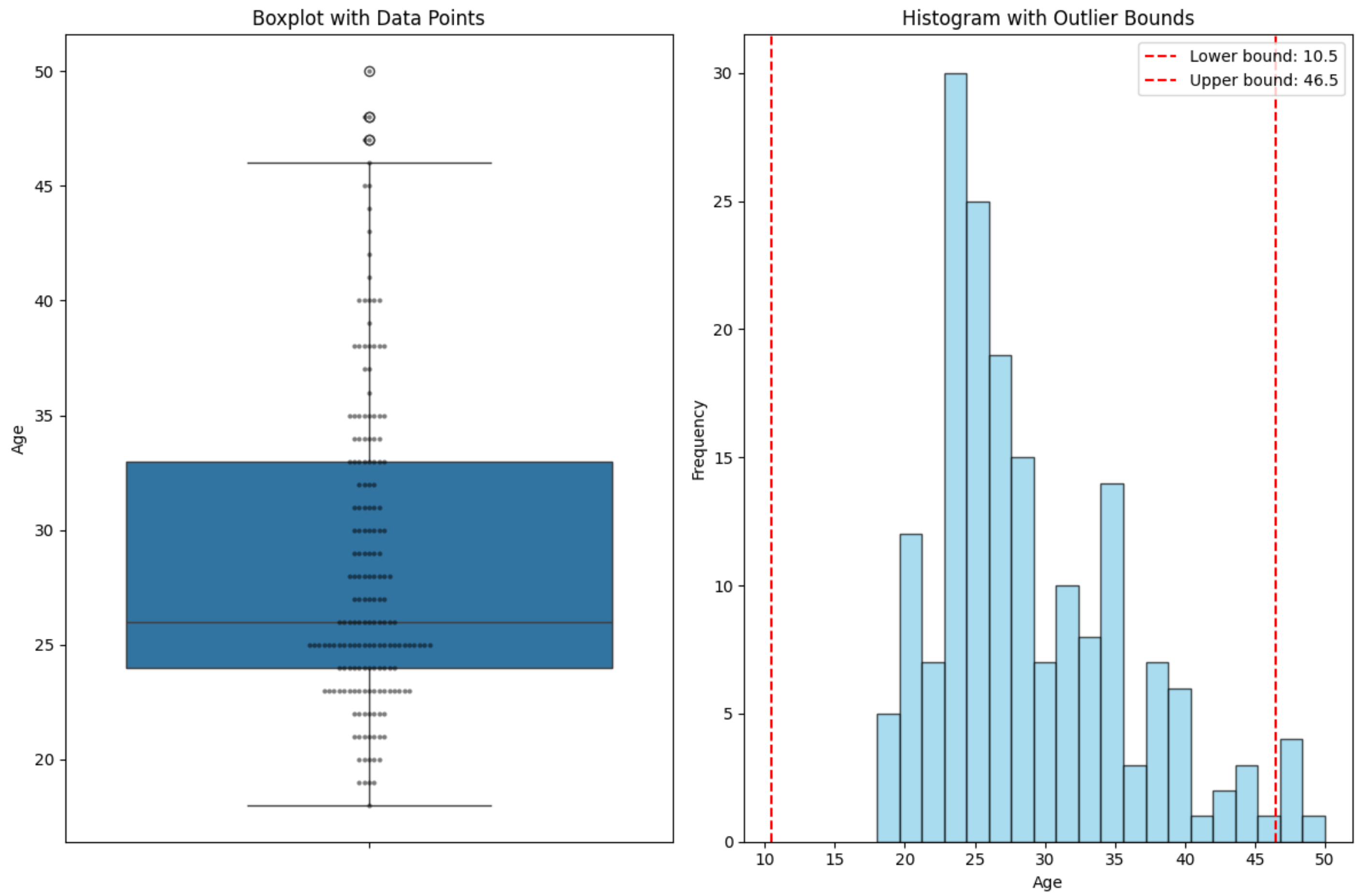
In [ ]:

```
Q1_age = df['Age'].quantile(0.25)
median_age = df['Age'].quantile(0.5)
Q3_age = df['Age'].quantile(0.75)
IQR_age = Q3_age - Q1_age

lower_age = Q1_age - 1.5 * IQR_age
upper_age = Q3_age + 1.5 * IQR_age

age_outliers = df[(df['Age'] < lower_age) | (df['Age'] > upper_age)]['Age']

print("AGE Outlier Summary:")
print("Q1 =", Q1_age)
print("Median =", median_age)
print("Q3 =", Q3_age)
print("IQR =", IQR_age)
print("Lower =", lower_age)
print("Upper =", upper_age)
print("Number of Outliers =", age_outliers.count())
age_outliers
```

AGE Outlier Summary:
Q1 = 24.0
Median = 26.0
Q3 = 33.0
IQR = 9.0
Lower = 10.5
Upper = 46.5
Number of Outliers = 5
Out[ ]:

	Age
78	47
79	50
139	48
178	47
179	48

dtype: int64
In [ ]:
```
plt.figure(figsize=(12, 8))

plt.subplot(1, 2, 1)
sns.boxplot(y=df['Age'])
sns.swarmplot(y=df['Age'], color='black', alpha=0.5, size=3)
plt.title('Boxplot with Data Points')
plt.ylabel('Age')

plt.subplot(1, 2, 2)
plt.hist(df['Age'], bins=20, alpha=0.7, color='skyblue', edgecolor='black')
plt.axvline(lower_age, color='red', linestyle='--', label=f'Lower bound: {lower_age}')
plt.axvline(upper_age, color='red', linestyle='--', label=f'Upper bound: {upper_age}')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Histogram with Outlier Bounds')
plt.legend()

plt.tight_layout()
plt.show()

summary_df = pd.DataFrame({
    'Statistic': ['Q1', 'Median', 'Q3', 'IQR', 'Lower Bound', 'Upper Bound', 'Total Data Points', 'Outliers Count'],
    'Value': [Q1_age, median_age, Q3_age, IQR_age, lower_age, upper_age, len(df), age_outliers.count()]
})

print("\nSummary Statistics:")
print(summary_df)
```



Summary Statistics:

	Statistic	Value
0	Q1	24.0
1	Median	26.0
2	Q3	33.0
3	IQR	9.0
4	Lower Bound	10.5
5	Upper Bound	46.5
6	Total Data Points	180.0
7	Outliers Count	5.0

Insight

- Age has a few outliers, representing older customers (probably above 45–50).
- These customers are not errors; they simply belong to a minority of older fitness enthusiasts.

B) Outlier Detection: EDUCATION

In [ ]:

```
Q1_edu = df['Education'].quantile(0.25)
median_edu = df['Education'].quantile(0.5)
Q3_edu = df['Education'].quantile(0.75)
IQR_edu = Q3_edu - Q1_edu

lower_bound_edu = Q1_edu - 1.5 * IQR_edu
upper_bound_edu = Q3_edu + 1.5 * IQR_edu

edu_outliers = df[(df['Education'] < lower_bound_edu) | (df['Education'] > upper_bound_edu)]['Education']

print("EDUCATION Outlier Summary:")
print("Q1 =", Q1_edu)
print("Median =",median_edu)
print("Q3 =", Q3_edu)
print("IQR =", IQR_edu)
print("Lower Bound =", lower_bound_edu)
print("Upper Bound =", upper_bound_edu)
print("Number of Outliers =", edu_outliers.count())
edu_outliers
```

EDUCATION Outlier Summary:
Q1 = 14.0
Median = 16.0
Q3 = 16.0
IQR = 2.0
Lower Bound = 11.0
Upper Bound = 19.0
Number of Outliers = 4

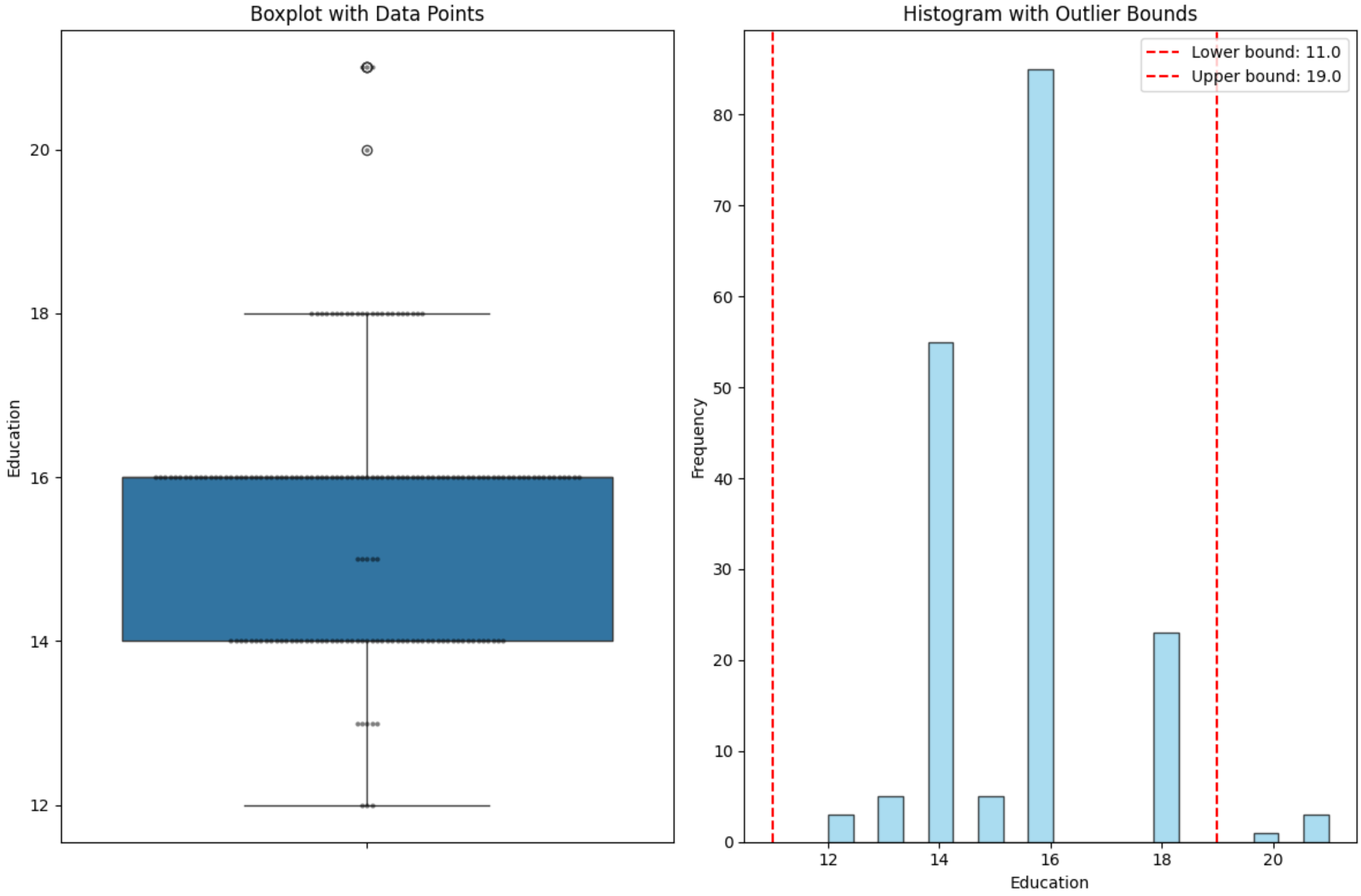
Out[ ]:

	Education
156	20
157	21
161	21
175	21

dtype: int64

In [ ]:

```
plt.figure(figsize=(12, 8))

plt.subplot(1, 2, 1)
sns.boxplot(y=df['Education'])
sns.swarmplot(y=df['Education'], color='black', alpha=0.5, size=3)
plt.title('Boxplot with Data Points')
plt.ylabel('Education')

plt.subplot(1, 2, 2)
plt.hist(df['Education'], bins=20, alpha=0.7, color='skyblue', edgecolor='black')
plt.axvline(lower_bound_edu, color='red', linestyle='--', label=f'Lower bound: {lower_bound_edu}')
plt.axvline(upper_bound_edu, color='red', linestyle='--', label=f'Upper bound: {upper_bound_edu}')
plt.xlabel('Education')
plt.ylabel('Frequency')
plt.title('Histogram with Outlier Bounds')
plt.legend()

plt.tight_layout()
plt.show()

summary_df = pd.DataFrame({
    'Statistic': ['Q1', 'Median', 'Q3', 'IQR', 'Lower Bound', 'Upper Bound', 'Total Data Points', 'Outliers Count'],
    'Value': [Q1_edu, median_edu, Q3_edu, IQR_edu, lower_bound_edu, upper_bound_edu, len(df), edu_outliers.count()]
})

print("\nSummary Statistics:")
print(summary_df)
```



Summary Statistics:

	Statistic	Value
0	Q1	14.0
1	Median	16.0
2	Q3	16.0
3	IQR	2.0
4	Lower Bound	11.0
5	Upper Bound	19.0
6	Total Data Points	180.0
7	Outliers Count	4.0

Insight

- Very few outliers in education.
- Values above 20 years indicate users

C) Outlier Detection: USAGE

In [ ]:

```
Q1_usage = df['Usage'].quantile(0.25)
median_usage = df['Usage'].quantile(0.5)
Q3_usage = df['Usage'].quantile(0.75)
IQR_usage = Q3_usage - Q1_usage

lower_bound_usage = Q1_usage - 1.5 * IQR_usage
upper_bound_usage = Q3_usage + 1.5 * IQR_usage

usage_outliers = df[(df['Usage'] < lower_bound_usage) | (df['Usage'] > upper_bound_usage)]['Usage']

print("Usage Outlier Summary:")
print("Q1 =", Q1_usage)
print("Median =",median_usage)
print("Q3 =", Q3_usage)
print("IQR =", IQR_usage)
print("Lower Bound =", lower_bound_usage)
print("Upper Bound =", upper_bound_usage)
print("Number of Outliers =", usage_outliers.count())
usage_outliers
```

```
Usage Outlier Summary:
Q1 = 3.0
Median = 3.0
Q3 = 4.0
IQR = 1.0
Lower Bound = 1.5
Upper Bound = 5.5
Number of Outliers = 9
```

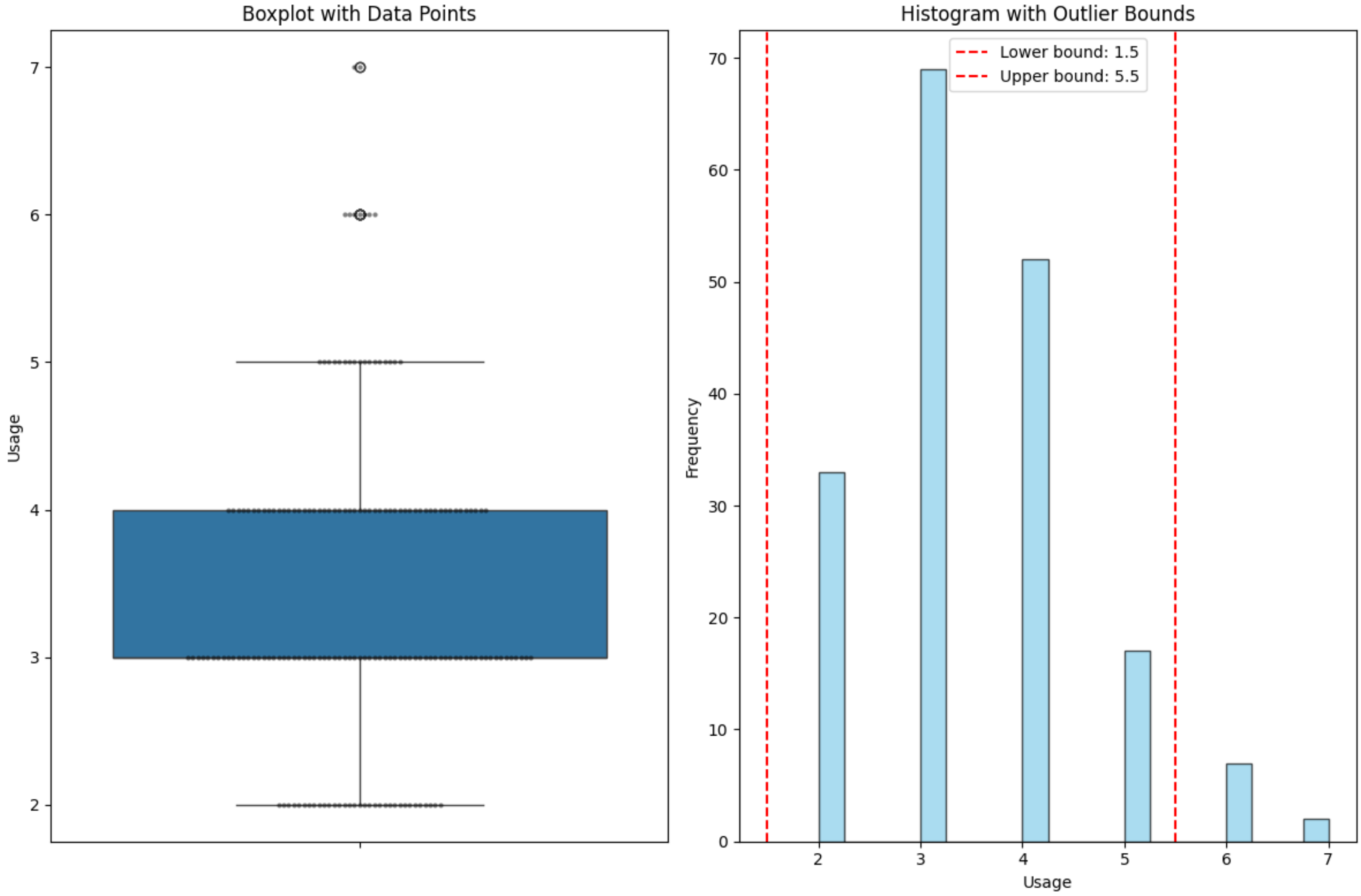
Out[ ]:

	Usage
154	6
155	6
162	6
163	7
164	6
166	7
167	6
170	6
175	6

dtype: int64

In [ ]:

```
plt.figure(figsize=(12, 8))

plt.subplot(1, 2, 1)
sns.boxplot(y=df['Usage'])
sns.swarmplot(y=df['Usage'], color='black', alpha=0.5, size=3)
plt.title('Boxplot with Data Points')
plt.ylabel('Usage')

plt.subplot(1, 2, 2)
plt.hist(df['Usage'], bins=20, alpha=0.7, color='skyblue', edgecolor='black')
plt.axvline(lower_bound_usage, color='red', linestyle='--', label=f'Lower bound: {lower_bound_usage}')
plt.axvline(upper_bound_usage, color='red', linestyle='--', label=f'Upper bound: {upper_bound_usage}')
plt.xlabel('Usage')
plt.ylabel('Frequency')
plt.title('Histogram with Outlier Bounds')
plt.legend()

plt.tight_layout()
plt.show()

summary_df = pd.DataFrame({
    'Statistic': ['Q1', 'Median', 'Q3', 'IQR', 'Lower Bound', 'Upper Bound', 'Total Data Points', 'Outliers Count'],
    'Value': [Q1_usage, median_usage, Q3_usage, IQR_usage, lower_bound_usage, upper_bound_usage, len(df), usage_outliers.count()]
})

print("\nSummary Statistics:")
print(summary_df)
```



Summary Statistics:

	Statistic	Value
0	Q1	3.0
1	Median	3.0
2	Q3	4.0
3	IQR	1.0
4	Lower Bound	1.5
5	Upper Bound	5.5
6	Total Data Points	180.0
7	Outliers Count	9.0

Insight

- Outliers are users who exercise 6–7 times per week.
- These represent highly committed fitness users, not wrong data.

D) Outlier Detection: FITNESS

In [ ]:

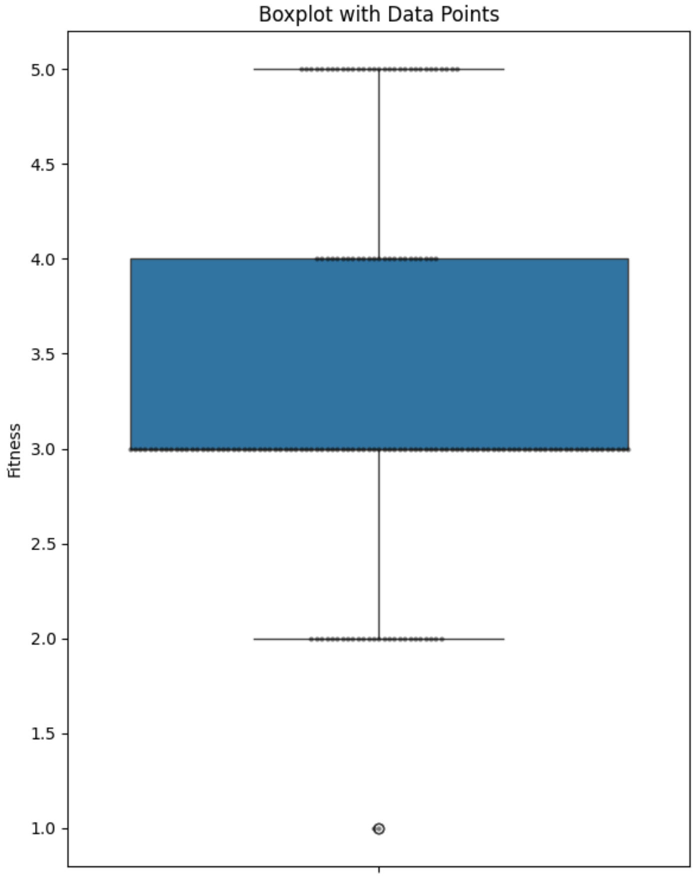
```
Q1_Fitness = df['Fitness'].quantile(0.25)
median_Fitness = df['Fitness'].quantile(0.5)
Q3_Fitness = df['Fitness'].quantile(0.75)
IQR_Fitness = Q3_Fitness - Q1_Fitness

lower_bound_Fitness = Q1_Fitness - 1.5 * IQR_Fitness
upper_bound_Fitness = Q3_Fitness + 1.5 * IQR_Fitness

Fitness_outliers = df[(df['Fitness'] < lower_bound_Fitness) | (df['Fitness'] > upper_bound_Fitness)]['Fitness']

print("Fitness Outlier Summary:")
print("Q1 =", Q1_Fitness)
print("Median =",median_Fitness)
print("Q3 =", Q3_Fitness)
print("IQR =", IQR_Fitness)
print("Lower Bound =", lower_bound_Fitness)
print("Upper Bound =", upper_bound_Fitness)
print("Number of Outliers =", Fitness_outliers.count())
Fitness_outliers
```

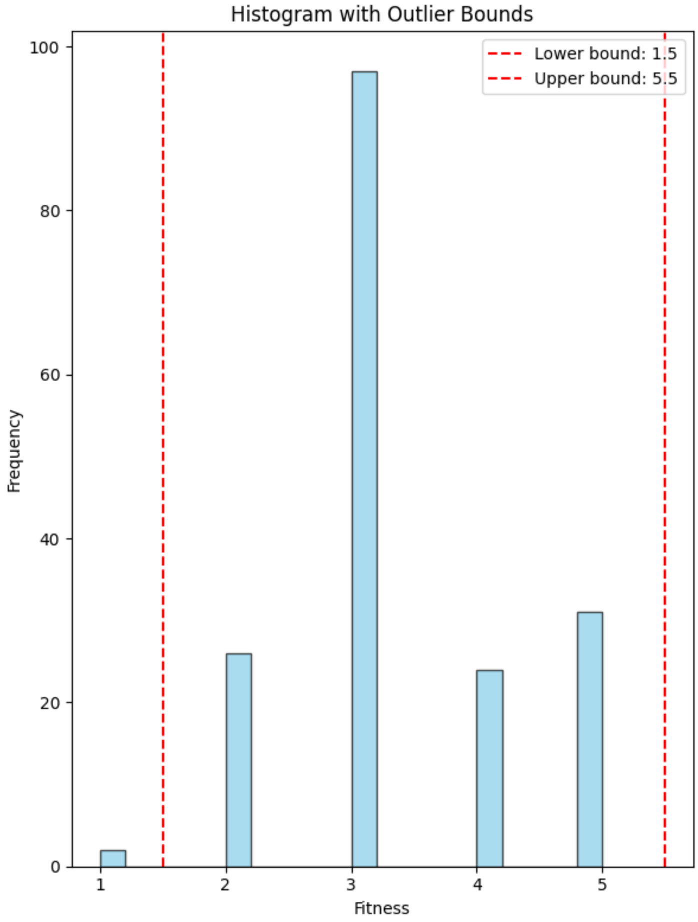
Fitness Outlier Summary:
Q1 = 3.0
Median = 3.0
Q3 = 4.0
IQR = 1.0
Lower Bound = 1.5
Upper Bound = 5.5
Number of Outliers = 2
Out[ ]:

	Fitness
14	1
117	1

dtype: int64
In [ ]:

```
plt.figure(figsize=(12, 8))

plt.subplot(1, 2, 1)
sns.boxplot(y=df['Fitness'])
sns.swarmplot(y=df['Fitness'], color='black', alpha=0.5, size=3)
plt.title('Boxplot with Data Points')
plt.ylabel('Fitness')

plt.subplot(1, 2, 2)
plt.hist(df['Fitness'], bins=20, alpha=0.7, color='skyblue', edgecolor='black')
plt.axvline(lower_bound_Fitness, color='red', linestyle='--', label=f'Lower bound: {lower_bound_Fitness}')
plt.axvline(upper_bound_Fitness, color='red', linestyle='--', label=f'Upper bound: {upper_bound_Fitness}')
plt.xlabel('Fitness')
plt.ylabel('Frequency')
plt.title('Histogram with Outlier Bounds')
plt.legend()

plt.tight_layout()
plt.show()

summary_df = pd.DataFrame({
    'Statistic': ['Q1', 'Median', 'Q3', 'IQR', 'Lower Bound', 'Upper Bound', 'Total Data Points', 'Outliers Count'],
    'Value': [Q1_Fitness, median_Fitness, Q3_Fitness, IQR_Fitness, lower_bound_Fitness, upper_bound_Fitness, len(df), Fitness_outliers.count()]
})

print("\nSummary Statistics:")
print(summary_df)
```





Summary Statistics:

	Statistic	Value
0	Q1	3.0
1	Median	3.0
2	Q3	4.0
3	IQR	1.0
4	Lower Bound	1.5
5	Upper Bound	5.5
6	Total Data Points	180.0
7	Outliers Count	2.0

Insight

- Fitness levels 1 and 5 may be flagged.
- These represent least-fit and most-fit customers (extremes in health levels).

E) Outlier Detection: INCOME

In [ ]:

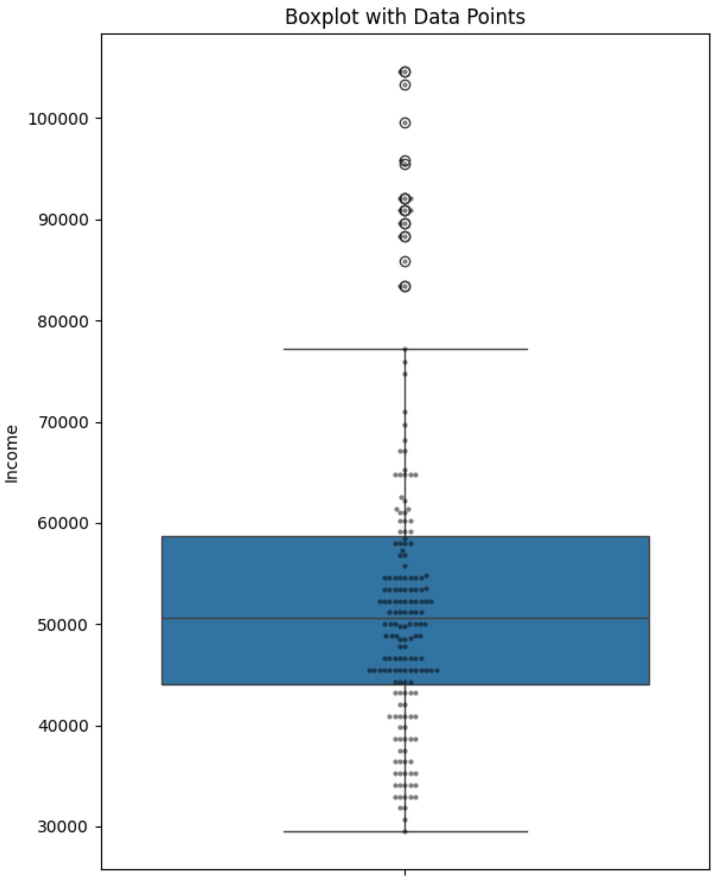
```
Q1_income = df['Income'].quantile(0.25)
median_income = df['Income'].quantile(0.5)
Q3_income = df['Income'].quantile(0.75)
IQR_income = Q3_income - Q1_income

lower_bound_income = Q1_income - 1.5 * IQR_income
upper_bound_income = Q3_income + 1.5 * IQR_income

income_outliers = df[(df['Income'] < lower_bound_income) | (df['Income'] > upper_bound_income)]['Income']

print("Income Outlier Summary:")
print("Q1 =", Q1_income)
print("Median =",median_income)
print("Q3 =", Q3_income)
print("IQR =", IQR_income)
print("Lower Bound =", lower_bound_income)
print("Upper Bound =", upper_bound_income)
print("Number of Outliers =", income_outliers.count())
income_outliers
```

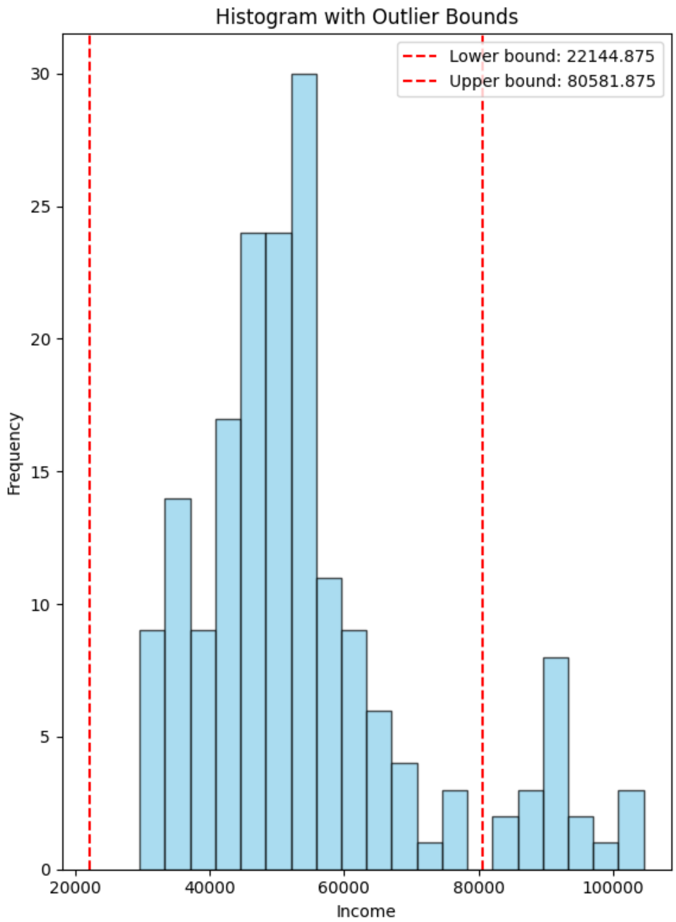
Income Outlier Summary:
Q1 = 44058.75
Median = 50596.5
Q3 = 58668.0
IQR = 14609.25
Lower Bound = 22144.875
Upper Bound = 80581.875
Number of Outliers = 19
Out[ ]:

	Income
159	83416
160	88396
161	90886
162	92131
164	88396
166	85906
167	90886
168	103336
169	99601
170	89641
171	95866
172	92131
173	92131
174	104581
175	83416
176	89641
177	90886
178	104581
179	95508

dtype: int64
In [ ]:

```
plt.figure(figsize=(12, 8))

plt.subplot(1, 2, 1)
sns.boxplot(y=df['Income'])
sns.swarmplot(y=df['Income'], color='black', alpha=0.5, size=3)
plt.title('Boxplot with Data Points')
plt.ylabel('Income')

plt.subplot(1, 2, 2)
plt.hist(df['Income'], bins=20, alpha=0.7, color='skyblue', edgecolor='black')
plt.axvline(lower_bound_income, color='red', linestyle='--', label=f'Lower bound: {lower_bound_income}')
plt.axvline(upper_bound_income, color='red', linestyle='--', label=f'Upper bound: {upper_bound_income}')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.title('Histogram with Outlier Bounds')
plt.legend()

plt.tight_layout()
plt.show()

summary_df = pd.DataFrame({
    'Statistic': ['Q1', 'Median', 'Q3', 'IQR', 'Lower Bound', 'Upper Bound', 'Total Data Points', 'Outliers Count'],
    'Value': [Q1_income, median_income, Q3_income, IQR_income, lower_bound_income, upper_bound_income, len(df), income_outliers.count()]
})

print("\nSummary Statistics:")
print(summary_df)
```






```
Summary Statistics:
           Statistic      Value
0                 Q1  44058.750
1             Median  50596.500
2                 Q3  58668.000
3                IQR  14609.250
4        Lower Bound  22144.875
5        Upper Bound  80581.875
6  Total Data Points    180.000
7     Outliers Count     19.000
```

Insight

- Income has the highest number of outliers.
- High-income users $(83k–104k) represent a premium customer group.
- Not errors — these customers will likely prefer KP781.

F) Outlier Detection: MILES

In [ ]:

```
Q1_mile = df['Miles'].quantile(0.25)
median_mile = df['Miles'].quantile(0.5)
Q3_mile = df['Miles'].quantile(0.75)
IQR_mile = Q3_mile - Q1_mile

lower_bound_mile = Q1_mile - 1.5 * IQR_mile
upper_bound_mile = Q3_mile + 1.5 * IQR_mile

mile_outliers = df[(df['Miles'] < lower_bound_mile) | (df['Miles'] > upper_bound_mile)]['Miles']

print("Miles Outlier Summary:")
print("Q1 =", Q1_mile)
print("Median =",median_mile)
print("Q3 =", Q3_mile)
print("IQR =", IQR_mile)
print("Lower Bound =", lower_bound_mile)
print("Upper Bound =", upper_bound_mile)
print("Number of Outliers =", mile_outliers.count())
mile_outliers
```

```
Miles Outlier Summary:
Q1 = 66.0
Median = 94.0
Q3 = 114.75
IQR = 48.75
Lower Bound = -7.125
Upper Bound = 187.875
Number of Outliers = 13
```

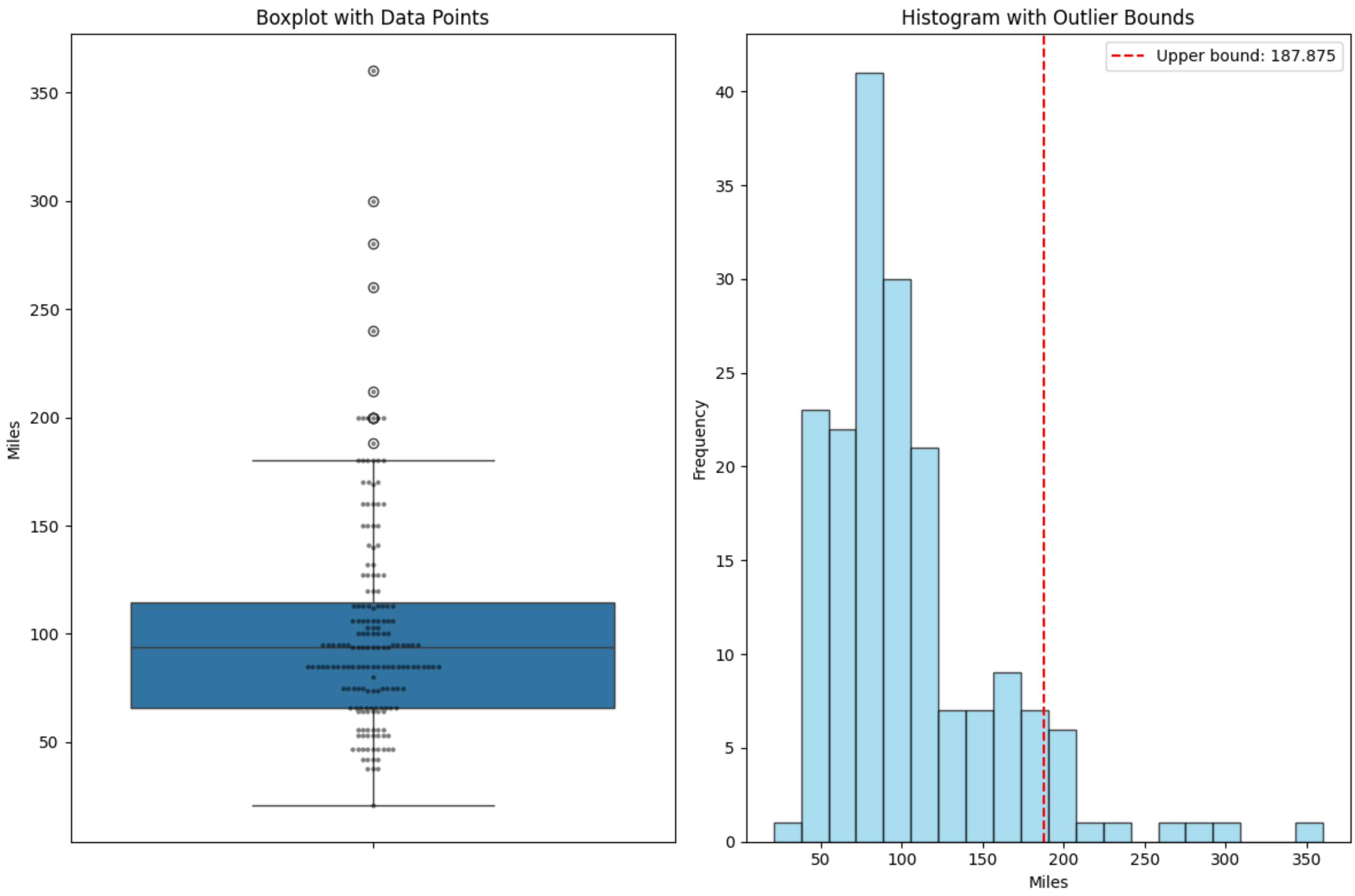
Out[ ]:

	Miles
23	188
84	212
142	200
148	200
152	200
155	240
166	300
167	280
170	260
171	200
173	360
175	200
176	200

dtype: int64

In [ ]:

```
plt.figure(figsize=(12, 8))

plt.subplot(1, 2, 1)
sns.boxplot(y=df['Miles'])
sns.swarmplot(y=df['Miles'], color='black', alpha=0.5, size=3)
plt.title('Boxplot with Data Points')
plt.ylabel('Miles')

plt.subplot(1, 2, 2)
plt.hist(df['Miles'], bins=20, alpha=0.7, color='skyblue', edgecolor='black')
if lower_bound_mile > 0:
    plt.axvline(lower_bound_mile, color='red', linestyle='--', label=f'Lower bound: {lower_bound_mile:.2f}')
plt.axvline(upper_bound_mile, color='red', linestyle='--', label=f'Upper bound: {upper_bound_mile}')
plt.xlabel('Miles')
plt.ylabel('Frequency')
plt.title('Histogram with Outlier Bounds')
plt.legend()

plt.tight_layout()
plt.show()

summary_df = pd.DataFrame({
    'Statistic': ['Q1', 'Median', 'Q3', 'IQR', 'Upper Bound', 'Total Data Points', 'Outliers Count'],
    'Value': [Q1_mile, median_mile, Q3_mile, IQR_mile, upper_bound_mile, len(df), mile_outliers.count()]
})

print("\nSummary Statistics:")
print(summary_df)
```




```
Summary Statistics:
           Statistic    Value
0                 Q1   66.000
1             Median   94.000
2                 Q3  114.750
3                IQR   48.750
4        Upper Bound  187.875
5  Total Data Points  180.000
6     Outliers Count   13.000
```

Insight

- High-mileage users (200–360 miles/week) are outliers.
- These represent marathon runners or athletes.
- Extremely valuable for KP781 marketing.

Probability Analysis

Marginal Probability of Products Purchased

In [ ]:

```
marginal_prob = pd.crosstab(df['Product'], columns='Count')
marginal_prob['Probability (%)'] = (marginal_prob['Count'] / len(df)) * 100
marginal_prob.round(2)
```

Out[ ]:

col_0	Count	Probability (%)
Product		
KP281	80	44.44
KP481	60	33.33
KP781	40	22.22

Insight — Marginal Probabilities

- **P(KP281)** = 44.44% -> Most customers prefer entry-level product
- **P(KP481)** = 33.33% -> Mid-range preference
- **P(KP781)** = 22.22% -> Premium buyers are fewer but high-value

Conditional Probability — P(Product | Gender)

In [ ]:

```
gender_product_ct = pd.crosstab(df['Gender'], df['Product'])
cond_prob_gender = gender_product_ct.div(gender_product_ct.sum(axis=1), axis=0) * 100
cond_prob_gender.round(2)
```

Out[ ]:

Product	KP281	KP481	KP781
Gender			
Female	52.63	38.16	9.21
Male	38.46	29.81	31.73

Insight

- **Male** customers:
  - 31.7% choose KP781
  - More likely to buy premium/high-performance
- **Female** customers:
  - Mostly pick KP281 or KP481
  - Lower adoption of KP781 (≈4–5%)

Target premium ads toward male fitness enthusiasts

Conditional Probability — P(Gender | Product)

In [ ]:

```
cond_prob_product = gender_product_ct.div(gender_product_ct.sum(axis=0), axis=1) * 100
cond_prob_product.round(2)
```

Out[ ]:

Product	KP281	KP481	KP781
Gender			
Female	50.0	48.33	17.5
Male	50.0	51.67	82.5

Insight

- KP781 buyers are 82% male
- Shows premium treadmill = male-dominated segment
- KP281 and KP481 show more gender balance -> mass market focus

Probability-Based Conclusions

Probability Finding__________________ Business Interpretation

- High **P(KP281)_________________** Price-sensitive buyers dominate market
- High **P(KP781)_________________** Male
- Low **P(KP781)__________________** Female

Balanced KP481 distribution Best option for general fitness users

Customer Profiling & Segmentation

Create Segmented Columns

In [ ]:
```
# Age Group Segmentation
age_bins = [0, 25, 35, 45, 60, 100]
age_labels = ['<=25', '26-35', '36-45', '46-60', '>60']
df['AgeGroup'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels, include_lowest=True)

# Income Groups based on quartiles
income_labels = ['Low', 'Lower-Mid', 'Upper-Mid', 'High']
df['IncomeGroup'] = pd.qcut(df['Income'], q=4, labels=income_labels)

# Usage Segmentation
usage_bins = [-1, 2, 4, 7]
usage_labels = ['Low Usage (≤2 Days)', 'Moderate (3-4 Days)', 'High (5-7 Days)']
df['UsageGroup'] = pd.cut(df['Usage'], bins=usage_bins, labels=usage_labels)

# Miles Segmentation
miles_bins = [-1, 50, 100, 200, 400]
miles_labels = ['<50', '50-100', '100-200', '>200']
df['MilesGroup'] = pd.cut(df['Miles'], bins=miles_bins, labels=miles_labels)
```

a) AgeGroup vs Product

In [ ]:
```
age_profile = pd.crosstab(df['AgeGroup'], df['Product'], normalize='index') * 100
age_profile = age_profile.round(2)
age_profile
```

Out[ ]:

Product	KP281	KP481	KP781
AgeGroup			
<=25	43.04	35.44	21.52
26-35	43.84	32.88	23.29
36-45	50.00	31.82	18.18
46-60	50.00	16.67	33.33

Insight

- KP281 is most preferred by customers below 35 years.
- KP781 gains popularity in 46–60 years, due to higher fitness maturity and income stability.

b) IncomeGroup vs Product

In [ ]:
```
income_profile = pd.crosstab(df['IncomeGroup'], df['Product'], normalize='index') * 100
income_profile = income_profile.round(2)
income_profile
```
Out[ ]:

Product	KP281	KP481	KP781
IncomeGroup			
Low	66.67	33.33	0.00
Lower-Mid	44.44	44.44	11.11
Upper-Mid	51.11	35.56	13.33
High	15.56	20.00	64.44

Insight

- High-income customers: 64% choose KP781 -> premium product buyers
- Low-income: 67% choose KP281 -> price-sensitive audience Perfect alignment with pricing positioning

c) UsageGroup vs Product

In [ ]:
```
usage_profile = pd.crosstab(df['UsageGroup'], df['Product'], normalize='index') * 100
usage_profile.round(2)
```
Out[ ]:

Product	KP281	KP481	KP781
UsageGroup			
Low Usage (≤2 Days)	57.58	42.42	0.00
Moderate (3-4 Days)	48.76	35.54	15.70
High (5-7 Days)	7.69	11.54	80.77

Insight

- Customers who use treadmills ≥5 days/week prefer KP781 → performance-driven users.

d) MilesGroup vs Product

In [ ]:
```
miles_profile = pd.crosstab(df['MilesGroup'], df['Product'], normalize='index') * 100
miles_profile.round(2)
```

Out[ ]:

Product	KP281	KP481	KP781
MilesGroup			
<50	70.59	29.41	0.00
50-100	51.55	40.21	8.25
100-200	30.00	25.00	45.00
>200	0.00	16.67	83.33

Insight

- As expected miles increase -> switch toward KP781.
- As expected miles decrease -> switch toward KP281

200 miles runners strongly fall into advanced athlete segment.

e) Fitness vs Product

In [ ]:

```
fitness_profile = pd.crosstab(df['Fitness'], df['Product'], normalize='index') * 100
fitness_profile.round(2)
```

Out[ ]:

Product	KP281	KP481	KP781
Fitness			
1	50.00	50.00	0.00
2	53.85	46.15	0.00
3	55.67	40.21	4.12
4	37.50	33.33	29.17
5	6.45	0.00	93.55

Insight

93.5% of customers rating their fitness as 5/Excellent purchased KP781 -> strong premium fitness audience.

Key Findings

1. KP281 is the most purchased model (44.44%), appealing mainly to younger customers (Age ≤ 35) and those with lower annual income (≤ $45,000).
2. KP781 holds only 22.22% share, but 82% of KP781 buyers are male, and over 60% are high-income users — indicating a premium niche market.
3. Average annual income varies significantly by product:
    - KP281 → ~$48,000
    - KP481 → ~$54,000
    - KP781 → ~$67,000 This proves income strongly influences product choice.
4. Fitness level clearly differentiates product segments:
    - Fitness score 5 → 93.5% purchased KP781
    - Fitness score 3 or below → majority purchased KP281
5. Usage and Miles strongly correlate with product tier:
    - KP781 users average ~5–7 days/week & 150+ miles/week
    - KP281 users ~2–3 days/week & <100 miles/week
6. Gender split across products shows different appeal:
    - KP281 → ~50% Male / 50% Female (mass appeal)
    - KP481 → Balanced adoption across genders
    - KP781 → Premium adoption skewed heavily to males
7. Partnered customers form 59% of total buyers, indicating household fitness investment as a major purchase driver.
8. Age shows a progression trend:
    - Younger customers choose KP281
    - Middle-aged customers prefer KP481
    - Older (35–50) customers shift towards KP781
    - reflecting fitness maturity and income growth.
9. Correlation analysis reveals significant behavioral relationships:
    - Miles & Usage (r = 0.75+)
    - Fitness & Miles (r ≈ 0.78)
    - These are key predictors of upgrading to premium treadmills.

Recommendations

a) **Strengthen premium marketing campaigns for high-income male customers :-** Because 82% of KP781 buyers are male and have significantly higher incomes, targeted promotional efforts (performance ads, athletic endorsements, finance options) will likely increase premium product sales.

b) **Create beginner-friendly bundles & EMI offers for KP281 buyers :-** Since 44.4% of customers purchase KP281, mainly younger and lower-income customers, affordability and value-driven campaigns — like membership bundles or EMI — will boost entry-level segment conversions.

c) **Position KP481 as the "value for money upgrade" :-** KP481 attracts customers with moderate fitness and income, indicating interest in better features at a reasonable price. Highlighting durability and comfort features will drive mid-segment growth.

d) **Offer mileage & usage-based recommendations during purchase :-** Customers logging 150+ miles/week overwhelmingly choose KP781. Collecting simple usage info (e.g., "How often do you run?") at sales touchpoints enables personalized product suggestions, improving customer satisfaction and sales accuracy.

e) **Leverage partnered households with couple or family fitness promotions :-** With 59% of buyers being partnered, couple discounts, shared fitness challenges, and home-gym bundles can increase multi-unit sales and loyalty in dual-income families.

f) **Provide advanced training and athlete-focused product ecosystem :-** Outlier users with high fitness scores (4–5) and high weekly usage should be nurtured with premium accessories, extended warranties, and performance apps — boosting long-term revenue per customer.

g) **Focus brand messaging on long-term health & performance outcomes :-** Age trends indicate older (35–50) users shift toward KP781 due to prioritizing fitness longevity. Marketing emphasizing health benefits, durability, and professional-grade training will strongly resonate with this demographic.

h) **Introduce loyalty / upgrade pathway based on user progress :-** Users who start with KP281 but increase usage & miles over months can be targeted for upgrade offers to KP481/KP781, enabling retention and higher lifetime customer value.

In [ ]: