

# Aerofit - Descriptive Statistics & Probability

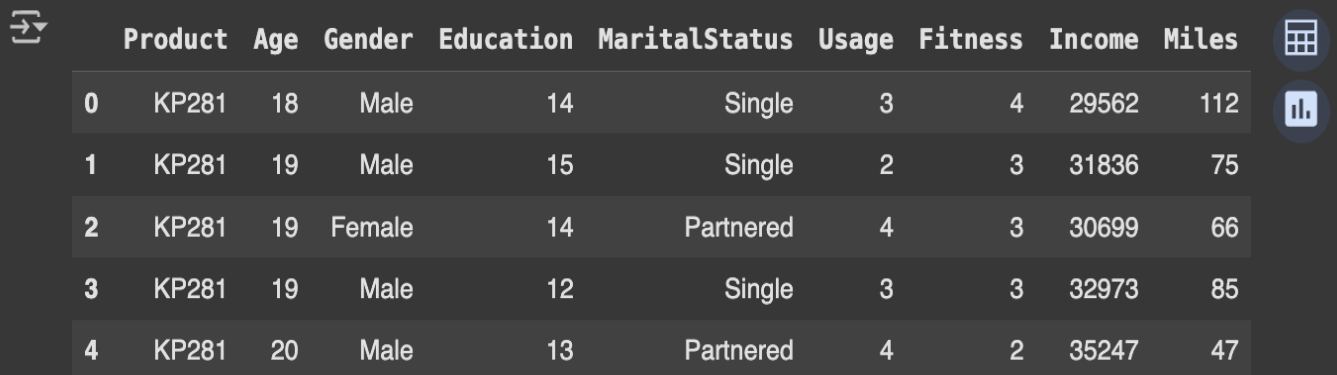
## Defining Problem Statement and Analysing basic metrics:

**Problem Statement:** Analyze the data and generate insights to create a customer profile for each Aerofit Treadmill product. By this, 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.

Importing the data and checking how the original data looks like:

```
[1] import pandas as pd
```

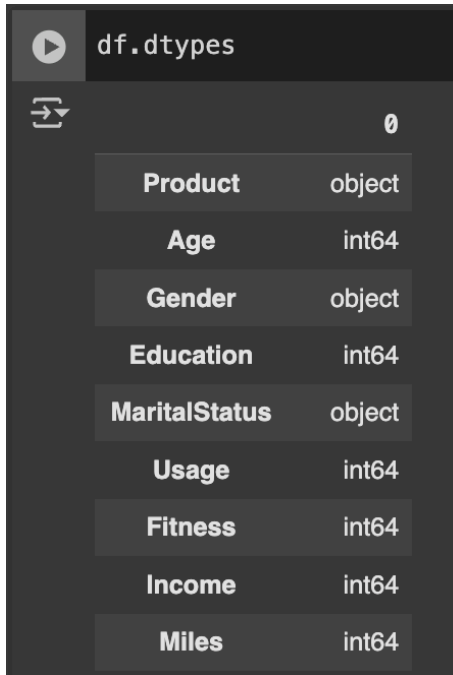
```
file_id = '1DE73ZwMT3WYc1NDeECFE8o8pkszuEn-V'  
url = f'https://drive.google.com/uc?id={file_id}'  
  
#https://drive.google.com/file/d/1DE73ZwMT3WYc1NDeECFE8o8pkszuEn-V/view?usp=sharing  
df = pd.read_csv(url)  
df.head() # Show first few rows
```



|   | 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    |

There are a total 9 features for the data set.

Checking the data types of all the attributes:



A screenshot of a Jupyter Notebook cell. The top bar shows a play button icon and the code `df.dtypes`. Below the code bar, there is a table icon and the number `0`. The main content of the cell is a table with two columns: attribute names and their corresponding data types.

|                      |        |
|----------------------|--------|
| <b>Product</b>       | object |
| <b>Age</b>           | int64  |
| <b>Gender</b>        | object |
| <b>Education</b>     | int64  |
| <b>MaritalStatus</b> | object |
| <b>Usage</b>         | int64  |
| <b>Fitness</b>       | int64  |
| <b>Income</b>        | int64  |
| <b>Miles</b>         | int64  |

Except Product, Gender and MaritalStatus, all the attributes are integer values.

Shape of the dataset:



A screenshot of a Jupyter Notebook cell. The top bar shows a play button icon and the code `[75] df.shape`. Below the code bar, there is a table icon and the output `(180, 9)`.

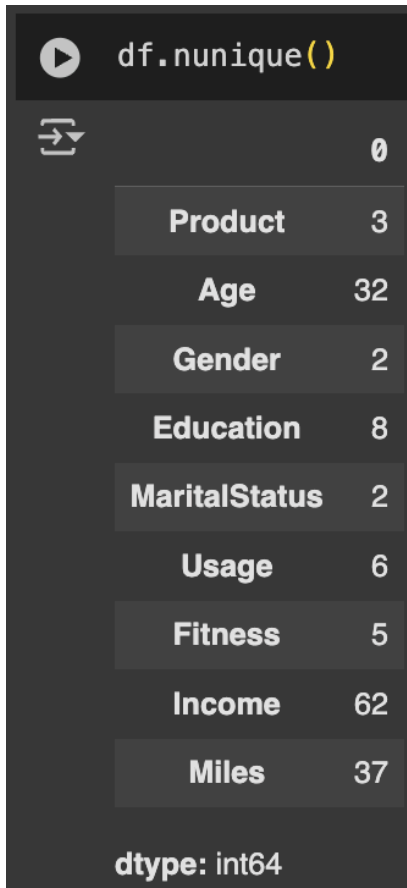
```
[75] df.shape
```

`(180, 9)`

So we have data belonging to 180 purchases.

## Non-Graphical Analysis:

Lets check how my unique values are there for each attribute:

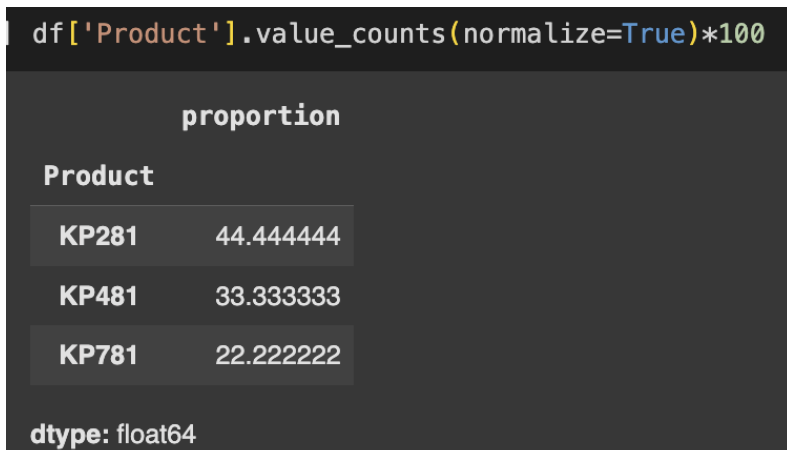


A Jupyter Notebook cell showing the execution of `df.nunique()`. The output is a Series with 11 attributes and their unique counts. The dtype is `int64`.

| Attribute     | Count |
|---------------|-------|
| Product       | 3     |
| Age           | 32    |
| Gender        | 2     |
| Education     | 8     |
| MaritalStatus | 2     |
| Usage         | 6     |
| Fitness       | 5     |
| Income        | 62    |
| Miles         | 37    |

dtype: int64

Total [no.of](#) products are 3, Gender are 2 and MaritalStatus are 2.



A Jupyter Notebook cell showing the execution of `df['Product'].value_counts(normalize=True)*100`. The output is a Series showing the proportion of each product. The dtype is `float64`.

| Product | proportion |
|---------|------------|
| KP281   | 44.444444  |
| KP481   | 33.333333  |
| KP781   | 22.222222  |

dtype: float64

Above output shows that out of the three treadmills, KP281 was the most bought one and KP781 is the least bought one.

44.44% of the customers have purchased KP2821, 33.33% of the customers have purchased KP481 and 22.22% of the customers have purchased KP781.

```
df.isna().sum()
```

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

dtype: int64

There are no missing values in the original data.

## Univariate Analysis:

Checking the impact of each attribute in purchasing the treadmill:

Product vs Income:

```
[80] df.groupby('Product')['Income'].describe()
```

|         | count | mean      | std          | min     | 25%      | 50%     | 75%     | max      |
|---------|-------|-----------|--------------|---------|----------|---------|---------|----------|
| Product |       |           |              |         |          |         |         |          |
| KP281   | 80.0  | 46418.025 | 9075.783190  | 29562.0 | 38658.00 | 46617.0 | 53439.0 | 68220.0  |
| KP481   | 60.0  | 48973.650 | 8653.989388  | 31836.0 | 44911.50 | 49459.5 | 53439.0 | 67083.0  |
| KP781   | 40.0  | 75441.575 | 18505.836720 | 48556.0 | 58204.75 | 76568.5 | 90886.0 | 104581.0 |

As the data shows, the high income groups are more likely to buy the most expensive treadmill and lower income groups.

Product vs Age:

```
[11] df.groupby('Product')['Age'].describe()
```

|         | count | mean  | std      | min  | 25%   | 50%  | 75%   | max  |
|---------|-------|-------|----------|------|-------|------|-------|------|
| Product |       |       |          |      |       |      |       |      |
| KP281   | 80.0  | 28.55 | 7.221452 | 18.0 | 23.00 | 26.0 | 33.00 | 50.0 |
| KP481   | 60.0  | 28.90 | 6.645248 | 19.0 | 24.00 | 26.0 | 33.25 | 48.0 |
| KP781   | 40.0  | 29.10 | 6.971738 | 22.0 | 24.75 | 27.0 | 30.25 | 48.0 |

It seems that age has no effect on purchasing patterns.

Product vs Gender:

```
[12] df.groupby('Product')['Gender'].describe()
```

|         | count | unique | top  | freq |
|---------|-------|--------|------|------|
| Product |       |        |      |      |
| KP281   | 80    | 2      | Male | 40   |
| KP481   | 60    | 2      | Male | 31   |
| KP781   | 40    | 2      | Male | 33   |

For every product Males are the most frequent customers.

Product vs Education:

```
[61] df.groupby('Product')['Education'].describe()
```

|         | count | mean      | std      | min  | 25%  | 50%  | 75%  | max  |
|---------|-------|-----------|----------|------|------|------|------|------|
| Product |       |           |          |      |      |      |      |      |
| KP281   | 80.0  | 15.037500 | 1.216383 | 12.0 | 14.0 | 16.0 | 16.0 | 18.0 |
| KP481   | 60.0  | 15.116667 | 1.222552 | 12.0 | 14.0 | 16.0 | 16.0 | 18.0 |
| KP781   | 40.0  | 17.325000 | 1.639066 | 14.0 | 16.0 | 18.0 | 18.0 | 21.0 |

Customers who are more educated are more likely to buy the expensive product KP781.

Product vs MaritalStatus:

Let's see the purchasing behaviour of customers based on their marital status.

```

desc = df.groupby('Product')['MaritalStatus'].describe()
desc['Top_Percent'] = (desc['freq'] / desc['count']) * 100

print(desc[['count', 'top', 'freq', 'Top_Percent']].round(2))

```

|         | count | top       | freq | Top_Percent |
|---------|-------|-----------|------|-------------|
| Product |       |           |      |             |
| KP281   | 80    | Partnered | 48   | 60.0        |
| KP481   | 60    | Partnered | 36   | 60.0        |
| KP781   | 40    | Partnered | 23   | 57.5        |

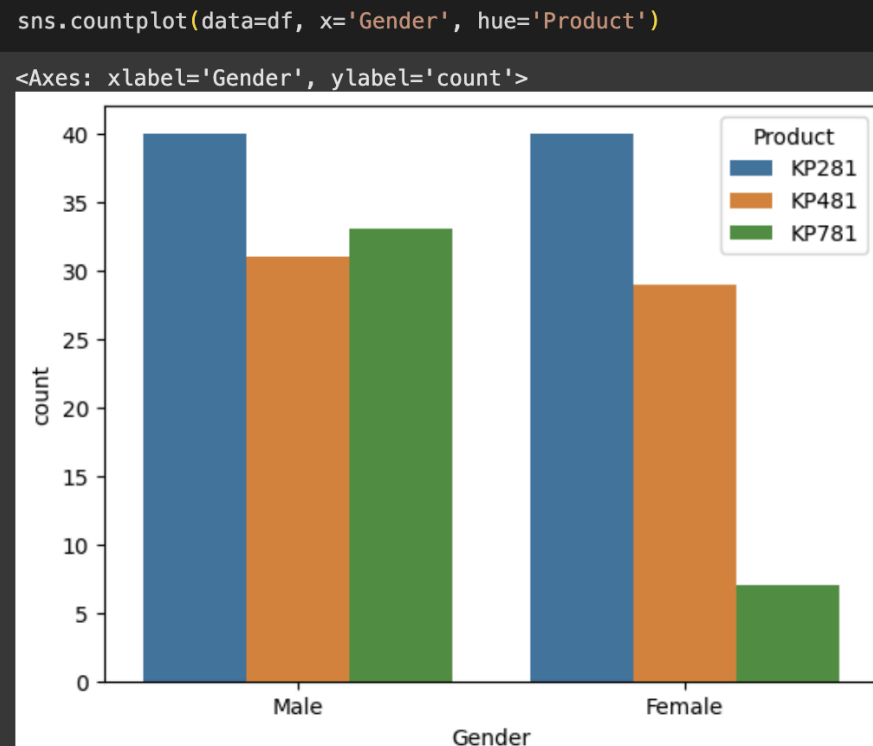
For every product, partnered people are the frequent customers.

Product vs Miles:

```
df.groupby('Product')['Miles'].describe()
```

|         | count | mean       | std       | min  | 25%   | 50%   | 75%   | max   |
|---------|-------|------------|-----------|------|-------|-------|-------|-------|
| Product |       |            |           |      |       |       |       |       |
| KP281   | 80.0  | 82.787500  | 28.874102 | 38.0 | 66.0  | 85.0  | 94.0  | 188.0 |
| KP481   | 60.0  | 87.933333  | 33.263135 | 21.0 | 64.0  | 85.0  | 106.0 | 212.0 |
| KP781   | 40.0  | 166.900000 | 60.066544 | 80.0 | 120.0 | 160.0 | 200.0 | 360.0 |

People who use the treadmill the most are more likely to buy KP781 product.



[No.](#) of Males and Females who bought KP281 are the same. i.e., 40 each.  
Males dominated in buying KP481 with a little margin.  
Most of the customers who bought KP781 are males.

Identifying Outliers using Boxplots:

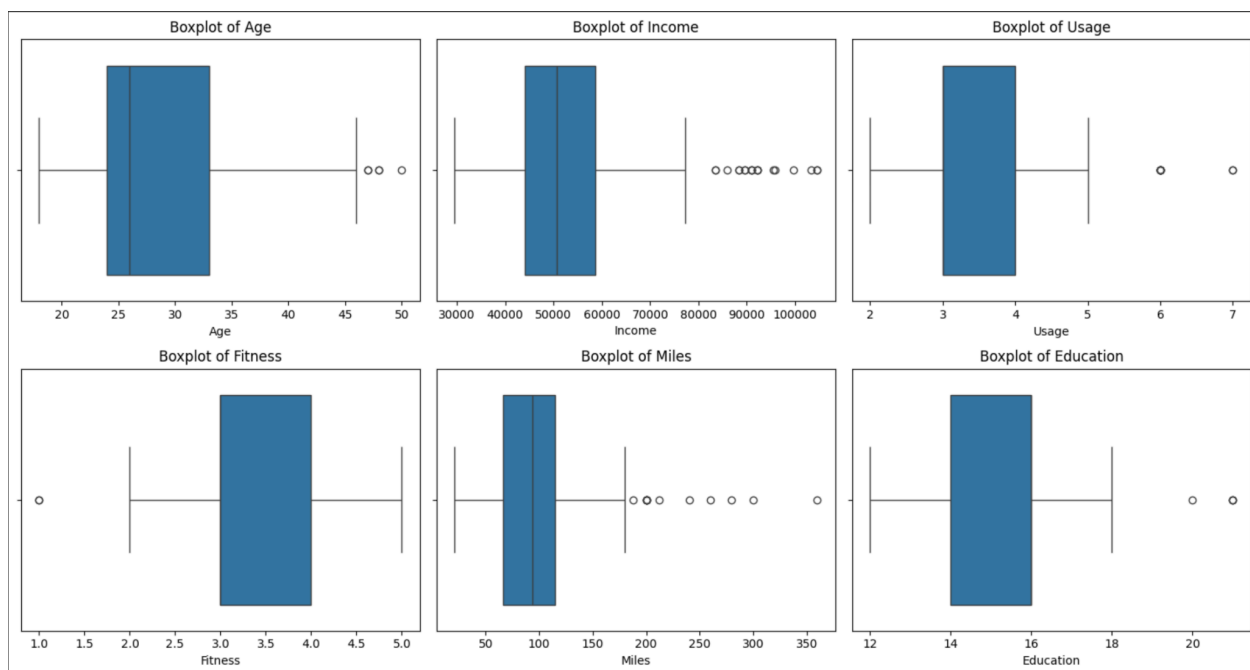
```
cols = ['Age', 'Income', 'Usage', 'Fitness', 'Miles', 'Education']

rows = 2
cols_per_row = 3

fig, axes = plt.subplots(rows, cols_per_row, figsize=(15, 8))
axes = axes.flatten()

for i, col in enumerate(['Age', 'Income', 'Usage', 'Fitness', 'Miles', 'Education']):
    sns.boxplot(x=df[col], ax=axes[i])
    axes[i].set_title(f'Boxplot of {col}')

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



In the columns Income and Miles we can find outliers.

Let's remove them and plot again.  
For that I'm clipping the data between 0.05 to 0.95.

```

import numpy as np

df_clipped = df.copy()
for col in cols:
    lower = df[col].quantile(0.05)
    upper = df[col].quantile(0.95)
    df_clipped[col] = np.clip(df[col], lower, upper)

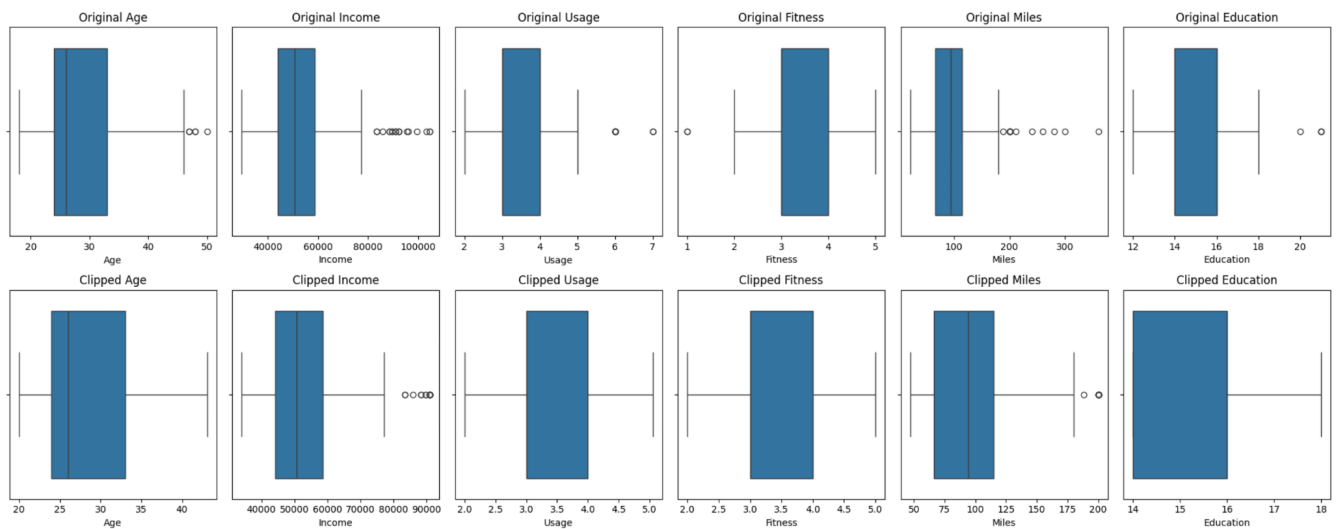
fig, axes = plt.subplots(2, len(cols), figsize=(20, 8))
axes = axes.flatten()

for i, col in enumerate(cols):
    sns.boxplot(x=df[col], ax=axes[i])
    axes[i].set_title(f'Original {col}')

# Plot clipped data
for i, col in enumerate(cols):
    sns.boxplot(x=df_clipped[col], ax=axes[i + len(cols)])
    axes[i + len(cols)].set_title(f'Clipped {col}')

plt.tight_layout()
plt.show()

```



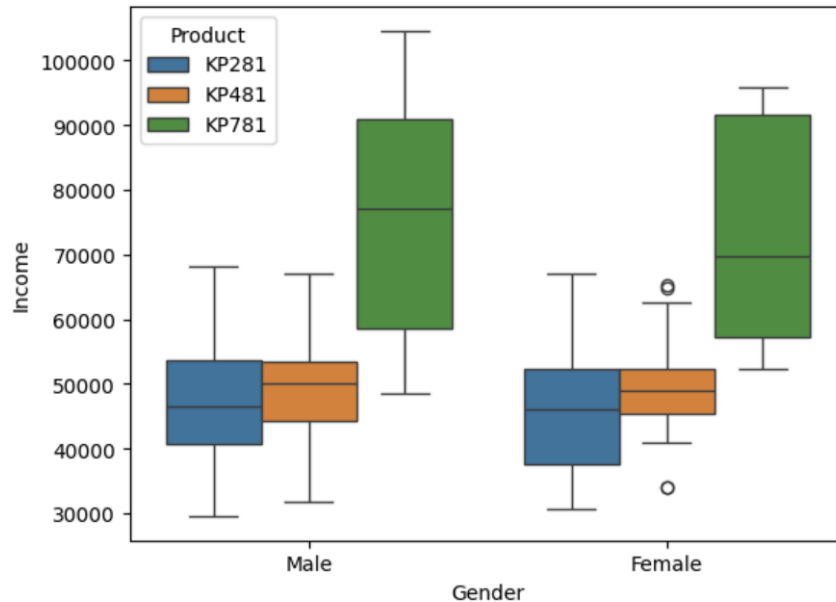
Outliers are reduced from all the columns. But in the Income and Miles columns it seems that the removal of remaining outliers may result in losing the data. So I think the dataset is good now.



## Bivariate and Multivariate Analysis:

```
import seaborn as sns
sns.boxplot(x='Gender', y='Income', hue='Product', data=df)
```

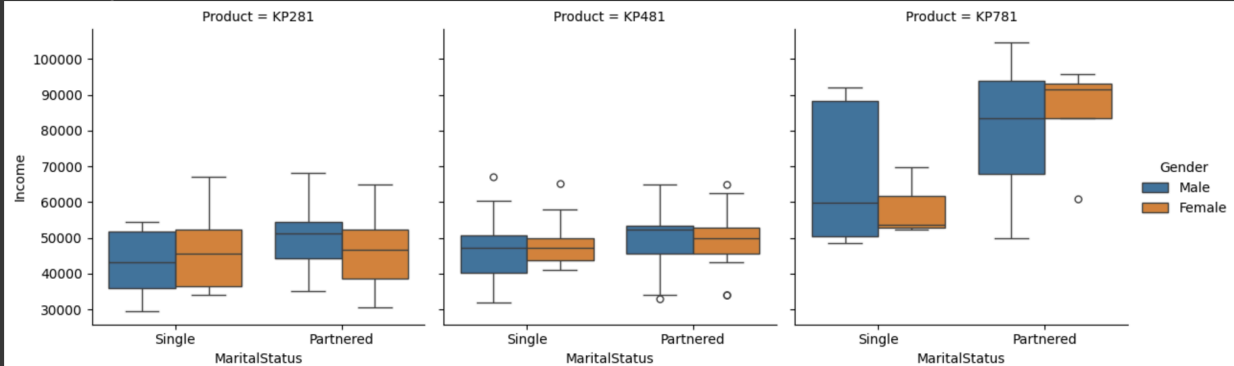
<Axes: xlabel='Gender', ylabel='Income'>



The difference in the income between male and female does have an effect in buying KP281 and KP481, but no effect on KP781.

```
sns.catplot(x='MaritalStatus', y='Income', hue='Gender', col='Product',
            kind='box', data=df, height=4, aspect=1)
```

<seaborn.axisgrid.FacetGrid at 0x7cd2ce99e750>



```
df.groupby('Product')[['Income', 'MaritalStatus', 'Gender']].agg(lambda x: x.value_counts().to_dict())
```

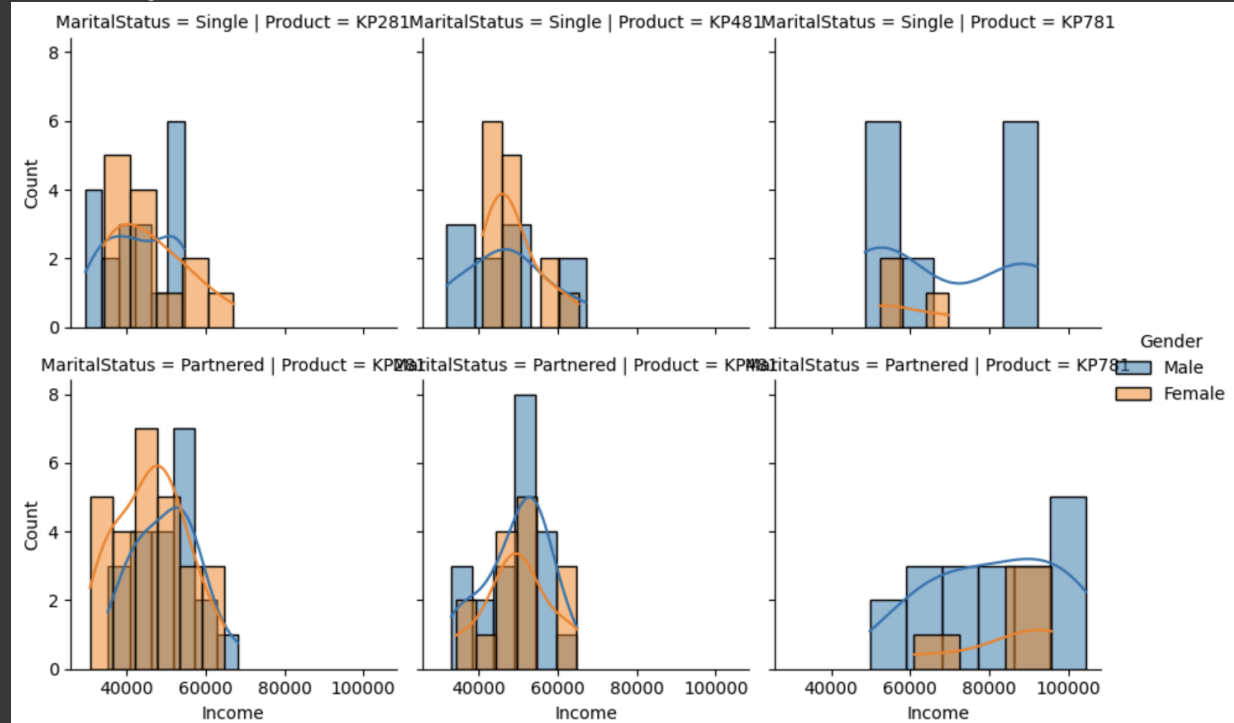
|                | Income  | MaritalStatus                   | Gender                     |
|----------------|---|---------------------------------|----------------------------|
| <b>Product</b> |   |                                 |                            |
| KP281          | {46617: 7, 54576: 7, 52302: 6, 45480: 5, 35247... | {'Partnered': 48, 'Single': 32} | {'Male': 40, 'Female': 40} |
| KP481          | {45480: 9, 50028: 5, 53439: 5, 43206: 4, 51165... | {'Partnered': 36, 'Single': 24} | {'Male': 31, 'Female': 29} |
| KP781          | {90886: 3, 92131: 3, 48556: 2, 64741: 2, 88396... | {'Partnered': 23, 'Single': 17} | {'Male': 33, 'Female': 7}  |

High income males bought kp781 more than high income females.

Overall, partnered customers are more compared to single customers.

```
g = sns.FacetGrid(df, col='Product', row='MaritalStatus', hue='Gender')
g.map(sns.histplot, 'Income', kde=True)
g.add_legend()
```

<seaborn.axisgrid.FacetGrid at 0x7cd2c221bf50>

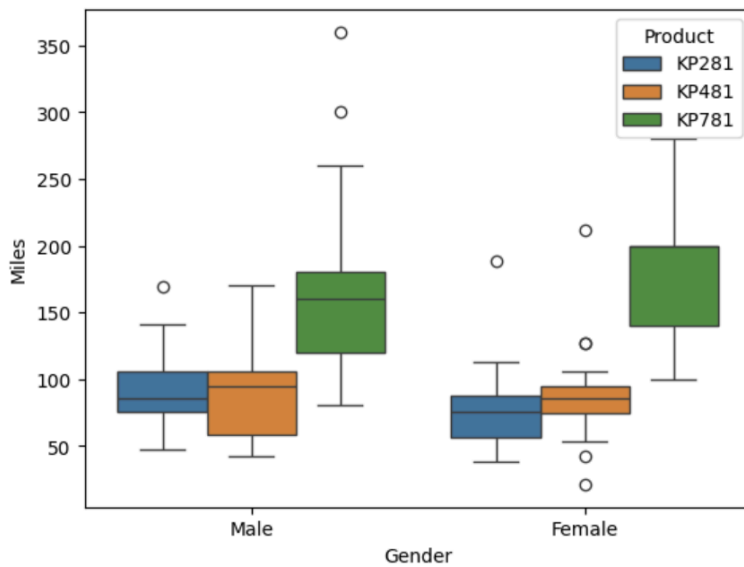


Histogram shows partnered female customers are more likely to buy KP281.

For KP482 partnered males are best customers and for KP781 Male customers bought far more products than females in both single and partnered categories.

```
sns.boxplot(x='Gender', y='Miles', hue='Product', data=df)
```

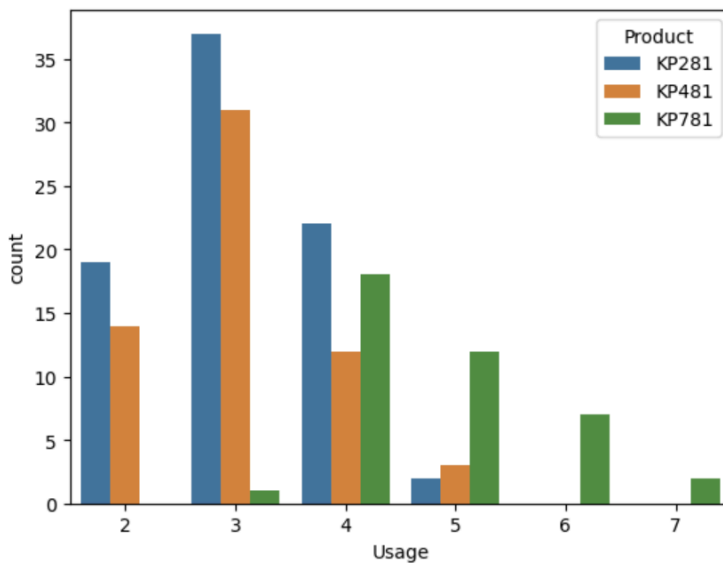
```
<Axes: xlabel='Gender', ylabel='Miles'>
```



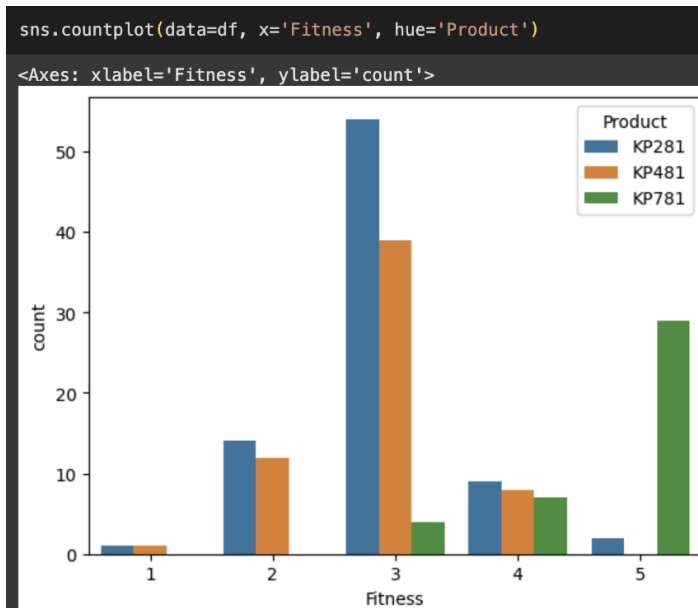
Irrespective of gender, people who expect to walk/run more in a week are buying KP781 more.

```
sns.countplot(data=df, x='Usage', hue='Product')
```

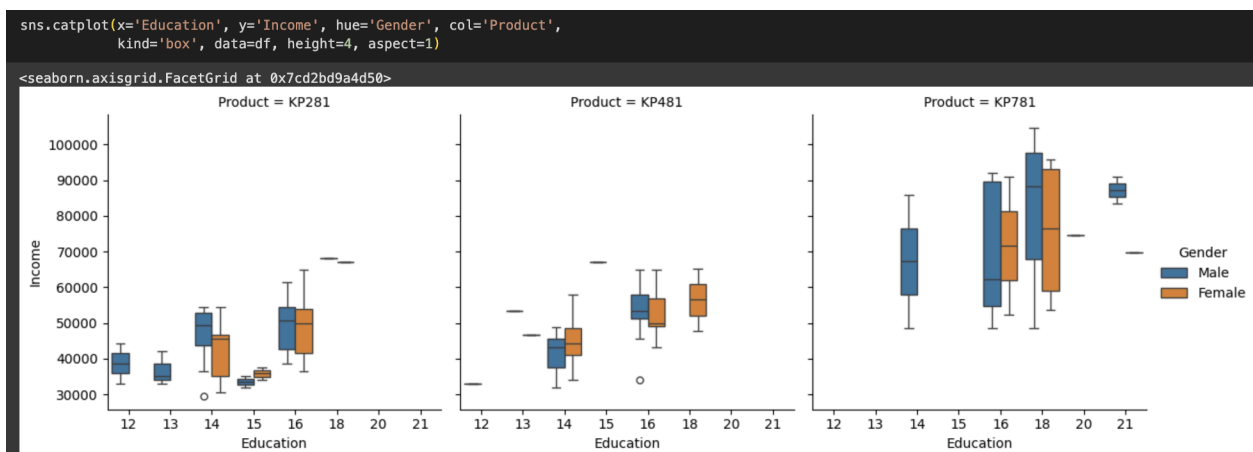
```
<Axes: xlabel='Usage', ylabel='count'>
```



Customers who expect to use it frequently i.e., at least 5 days a week are more likely to buy KP781.



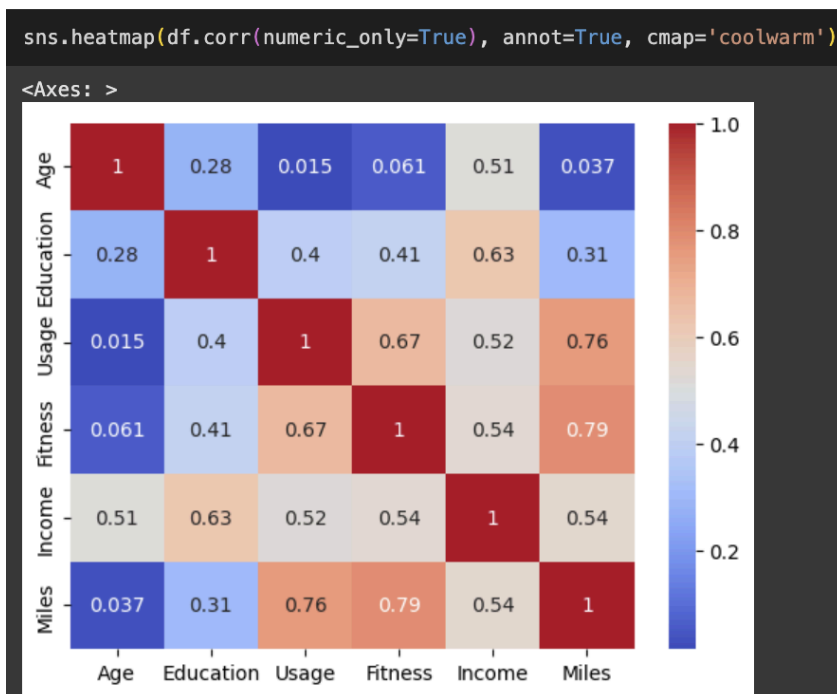
KP781 is bought by those who think they are fit.



Customers having more income and being highly educated are more likely to buy KP781.

Low income groups and comparatively less educated people are more likely to buy KP281.

## Correlation Heatmap:



High/strongest correlation metrics:

Fitness vs Miles

Usage vs Miles

Fitness vs Usage

Education vs Income

Weakest correlation is between:

Age vs Education

Age vs Usage

Age vs Miles

Education vs Miles

Education vs Fitness

## Marginal Probability:

```
marginal_probability = product_counts/product_counts.sum() *100  
marginal_probability.columns = ['Marginal Prob(%)']  
marginal_probability.round(2)
```

| Marginal Prob(%) |       |
|------------------|-------|
| Product          |       |
| KP281            | 44.44 |
| KP481            | 33.33 |
| KP781            | 22.22 |

## Conditional Probability:

```
gender_product = pd.crosstab(df['Gender'], df['Product'], normalize='index') * 100
print(gender_product.round(2))
```

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

Most bought product being female is KP281

Least bought product being female is KP781

Most bought product being male is KP281

Least bought product being male is KP481

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

| Product | KP281     | KP481     | KP781     |
|---------|-----------|-----------|-----------|
| Gender  |           |           |           |
| Female  | 22.222222 | 16.111111 | 3.888889  |
| Male    | 22.222222 | 17.222222 | 18.333333 |

Male are more likely to buy more products overall compared to females.

```
MStatus_product = pd.crosstab(index=df['MaritalStatus'], columns=[df['Product'], df['Gender']], normalize=True)*100
print(MStatus_product.round(2))
```

| Product       | KP281  |       | KP481  |       | KP781  |       |
|---------------|--------|-------|--------|-------|--------|-------|
| Gender        | Female | Male  | Female | Male  | Female | Male  |
| MaritalStatus |        |       |        |       |        |       |
| Partnered     | 15.00  | 11.67 | 8.33   | 11.67 | 2.22   | 10.56 |
| Single        | 7.22   | 10.56 | 7.78   | 5.56  | 1.67   | 7.78  |

KP281: Most bought - Partnered Female

KP281: Least bought - Single Female

KP481: Most bought - Partnered male

KP481: Least bought - Single male

KP781: Most bought - Partnered male

KP781: Least bought - Single Female

To analyse through Income, [no.of](#) rows for income are 62.

```
income_product = pd.crosstab(df['Income'], df['Product'], normalize='index') * 100
print(income_product.round(2))
```

| Product | KP281 | KP481 | KP781 |
|---------|-------|-------|-------|
| Income  |       |       |       |
| 29562   | 100.0 | 0.0   | 0.0   |
| 30699   | 100.0 | 0.0   | 0.0   |
| 31836   | 50.0  | 50.0  | 0.0   |
| 32973   | 60.0  | 40.0  | 0.0   |
| 34110   | 40.0  | 60.0  | 0.0   |
| ...     | ...   | ...   | ...   |
| 95508   | 0.0   | 0.0   | 100.0 |
| 95866   | 0.0   | 0.0   | 100.0 |
| 99601   | 0.0   | 0.0   | 100.0 |
| 103336  | 0.0   | 0.0   | 100.0 |
| 104581  | 0.0   | 0.0   | 100.0 |

[62 rows x 3 columns]

So bins are created to categorize the Income column and analysed along with Gender.

```
income_bins = [29000, 44000, 50500, 58600, 105000]
income_labels = ['Low', 'Lower-Mid', 'Upper-Mid', 'High']
df['IncomeGroup'] = pd.cut(df['Income'], bins=income_bins, labels=income_labels)
```

```
pd.crosstab(index=df['IncomeGroup'], columns=[df['Product'], df['Gender']], normalize=True)*100
```

| Product     | KP281    |          | KP481    |          | KP781    |           |
|-------------|----------|----------|----------|----------|----------|-----------|
| Gender      | Female   | Male     | Female   | Male     | Female   | Male      |
| IncomeGroup |          |          |          |          |          |           |
| Low         | 8.333333 | 8.333333 | 3.888889 | 4.444444 | 0.000000 | 0.000000  |
| Lower-Mid   | 6.666667 | 4.444444 | 6.666667 | 4.444444 | 0.000000 | 2.777778  |
| Upper-Mid   | 5.000000 | 7.777778 | 3.333333 | 5.555556 | 1.111111 | 2.222222  |
| High        | 2.222222 | 1.666667 | 2.222222 | 2.777778 | 2.777778 | 13.333333 |

KP281: Most bought - Low income Male and Female

KP281: Least bought - High income Male

KP481: Most bought - Low-Mid income female

KP481: Least bought - High income female

KP781: Most bought - High income Male

KP781: Least bought - Low income Male, female and Low-Mid income female

Similarly bins are created for Miles:

```
miles_bins = [20, 65, 93, 114, 361]
miles_labels = ['Low', 'Low-Mid', 'Mid-High', 'High']
df['MilesGroups'] = pd.cut(df['Miles'], bins = miles_bins, labels = miles_labels)
```

```
miles_product = pd.crosstab(index=df['MilesGroups'], columns=[df["Product"],df['Gender']], normalize=True)*100
print(miles_product.round(2))
```

| Product     | KP281  |      | KP481  |      | KP781  |       |
|-------------|--------|------|--------|------|--------|-------|
| Gender      | Female | Male | Female | Male | Female | Male  |
| MilesGroups |        |      |        |      |        |       |
| Low         | 6.11   | 3.89 | 3.89   | 6.11 | 0.00   | 0.00  |
| Low-Mid     | 10.56  | 9.44 | 6.11   | 1.67 | 0.00   | 0.56  |
| Mid-High    | 5.00   | 6.11 | 4.44   | 6.67 | 1.11   | 3.33  |
| High        | 0.56   | 2.78 | 1.67   | 2.78 | 2.78   | 14.44 |

KP281: Most bought - Low-mid usage, female

KP281: Least bought - high usage, female

KP481: Most bought - mid-high usage, male

KP481: least bought - Low-mid usage, male and high usage, female

KP781: Most bought - High usage male

KP781: least bought - Low usage female & male and low-mid usage female.

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

| Product | KP281  |       | KP481  |       | KP781  |       |
|---------|--------|-------|--------|-------|--------|-------|
| Gender  | Female | Male  | Female | Male  | Female | Male  |
| Fitness |        |       |        |       |        |       |
| 1       | 0.00   | 0.56  | 0.56   | 0.00  | 0.00   | 0.00  |
| 2       | 5.56   | 2.22  | 3.33   | 3.33  | 0.00   | 0.00  |
| 3       | 14.44  | 15.56 | 10.00  | 11.67 | 0.56   | 1.67  |
| 4       | 1.67   | 3.33  | 2.22   | 2.22  | 0.56   | 3.33  |
| 5       | 0.56   | 0.56  | 0.00   | 0.00  | 2.78   | 13.33 |

Customers who rate themselves as fit are more likely to buy kp781 and most of them are males.



## **Business Insights:**

**Product Popularity:** Among the three treadmill models — KP281, KP481, and KP781 — KP281 emerged as the most purchased product, indicating strong customer preference for entry-level or mid-range options.

**Customer Segmentation:** Our analysis revealed that partnered individuals and males were more likely to purchase treadmills, suggesting that household or shared fitness goals may influence buying decisions.

**Income Influence:** Customers with mid to upper-mid income levels showed higher purchase rates, especially for KP481 and KP781, highlighting the role of affordability and perceived value in product selection.

**Usage Patterns:** Higher usage and fitness scores were positively correlated with purchases of advanced models like KP781, indicating that more active or health-conscious users tend to invest in premium equipment.

**Outlier Management:** Clipping continuous variables between the 5th and 95th percentiles helped refine our analysis by removing extreme values, ensuring more accurate insights into customer behavior.

**Education Impact:** Years of education showed a moderate influence on product choice, with more educated customers leaning toward higher-end models, possibly due to greater health awareness or disposable income.

**Seasonal Trends:** While not explicitly time-based, the data hints at consistent purchasing behavior across demographics, suggesting that AeroFit's marketing strategy can focus on lifestyle rather than seasonal spikes.

**Product Differentiation:** Each treadmill model attracted distinct customer profiles, reinforcing the importance of targeted marketing and product positioning based on demographic and behavioral data.

**Data-Driven Strategy:** This case study highlights the power of data analytics in understanding customer preferences, optimizing product offerings, and guiding strategic decisions in the fitness equipment industry.

**Fit for the Future:** As health and wellness continue to gain importance, leveraging these insights will help AeroFit stay ahead of trends, meet evolving customer needs, and inspire more people to embrace active living.

## **Recommendations:**

**Promote KP281 More Aggressively** Since KP281 is the most popular model, focus marketing efforts on it. Offer bundle deals, highlight customer reviews, and position it as the best value for money.

**Target Partnered Customers** Most buyers are in relationships. Create ads that show couples working out together or offer “family fitness” packages to appeal to this segment.

**Focus on Male Customers, But Don’t Ignore Females** Males are buying more treadmills, so tailor messaging to their fitness goals. At the same time, create campaigns that encourage female buyers — like wellness challenges or women-focused fitness plans.

**Offer Financing for High-End Models** Customers with higher income tend to buy premium models. Make it easier for mid-income customers to upgrade by offering EMI options or limited-time discounts on KP781.

**Highlight Fitness Benefits in Ads** People who are already fit or use treadmills often are buying more. Use testimonials and success stories to show how AeroFit helps users stay active and healthy.

**Create Education-Based Campaigns** More educated customers are leaning toward better models. Use informative content — like blog posts, videos, or webinars — to explain the benefits of regular treadmill use.

**Launch Seasonal Promotions** Even though purchases are steady, launching offers around New Year, summer, and festival seasons can boost sales. Think “New Year, New You” or “Summer Fit Sale.”

**Differentiate Products Clearly** Each model attracts different types of buyers. Make sure your website and ads clearly explain who each treadmill is best for — beginners, regular users, or fitness enthusiasts.

**Use Data to Guide Inventory and Marketing** Keep track of which models sell best in which regions or among which groups. Use this info to stock wisely and run targeted ads.

**Encourage Word-of-Mouth and Referrals** Happy customers are your best promoters. Offer referral bonuses or loyalty discounts to encourage them to spread the word.