Generate print hello world using rot13 Close

Generate is available for a limited time for unsubscribed users. Upgrade to Colab Pro

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

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

```
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

```
column_data_types = df.dtypes
column_data_types
```

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

```
# To find the number of rows and columns
df.shape
```

```
(180, 9)
```

```
df.head(20)
```

	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
5	KP281	20	Female	14	Partnered	3	3	32973	66
6	KP281	21	Female	14	Partnered	3	3	35247	75
7	KP281	21	Male	13	Single	3	3	32973	85
8	KP281	21	Male	15	Single	5	4	35247	141
9	KP281	21	Female	15	Partnered	2	3	37521	85
10	KP281	22	Male	14	Single	3	3	36384	85
11	KP281	22	Female	14	Partnered	3	2	35247	66
12	KP281	22	Female	16	Single	4	3	36384	75
13	KP281	22	Female	14	Single	3	3	35247	75
14	KP281	23	Male	16	Partnered	3	1	38658	47
15	KP281	23	Male	16	Partnered	3	3	40932	75
16	KP281	23	Female	14	Single	2	3	34110	103
17	KP281	23	Male	16	Partnered	4	3	39795	94
18	KP281	23	Female	16	Single	4	3	38658	113
19	KP281	23	Female	15	Partnered	2	2	34110	38

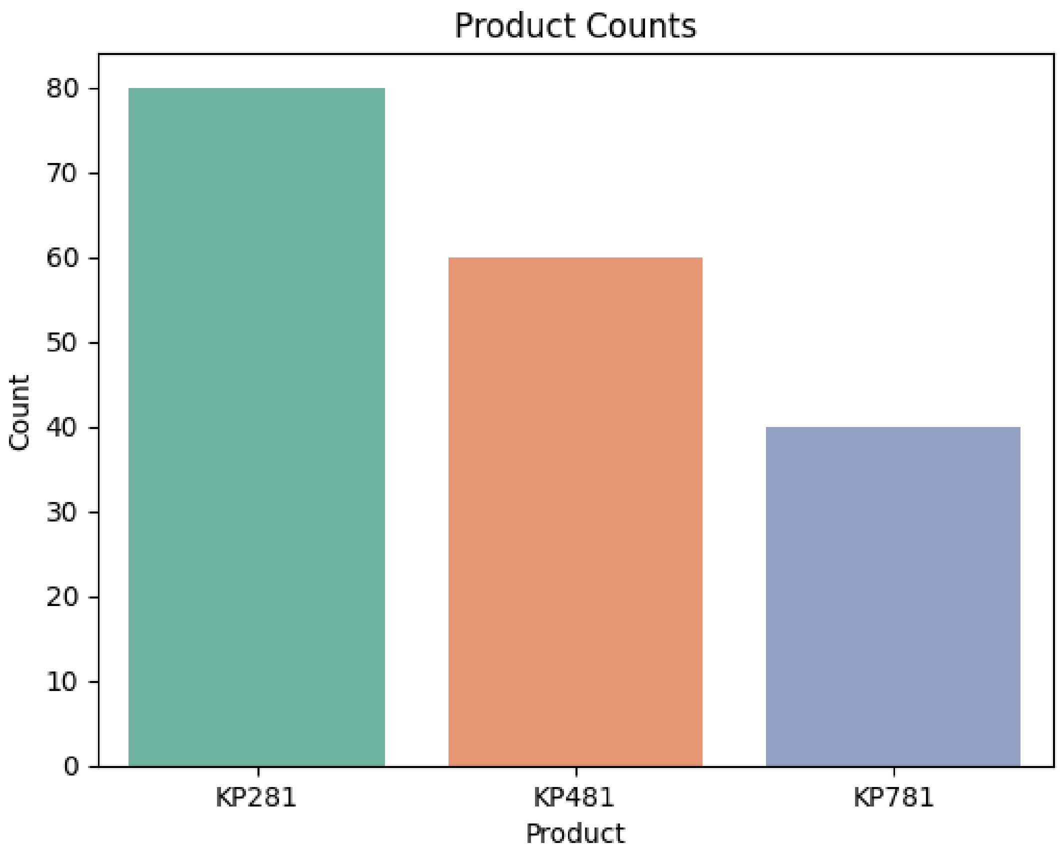
Next steps: Generate code with df View recommended plots

```
# To Check any null values in the columns of the data.
df.isnull().sum()
```

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

```
#To find the volume of Products
sns.countplot(data=df, x='Product', palette='Set2')
plt.xlabel("Product")
plt.ylabel("Count")
plt.title("Product Counts")
```

```
<ipython-input-32-ad625b7bfd7c>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

  sns.countplot(data=df, x='Product', palette='Set2')
Text(0.5, 1.0, 'Product Counts')
```

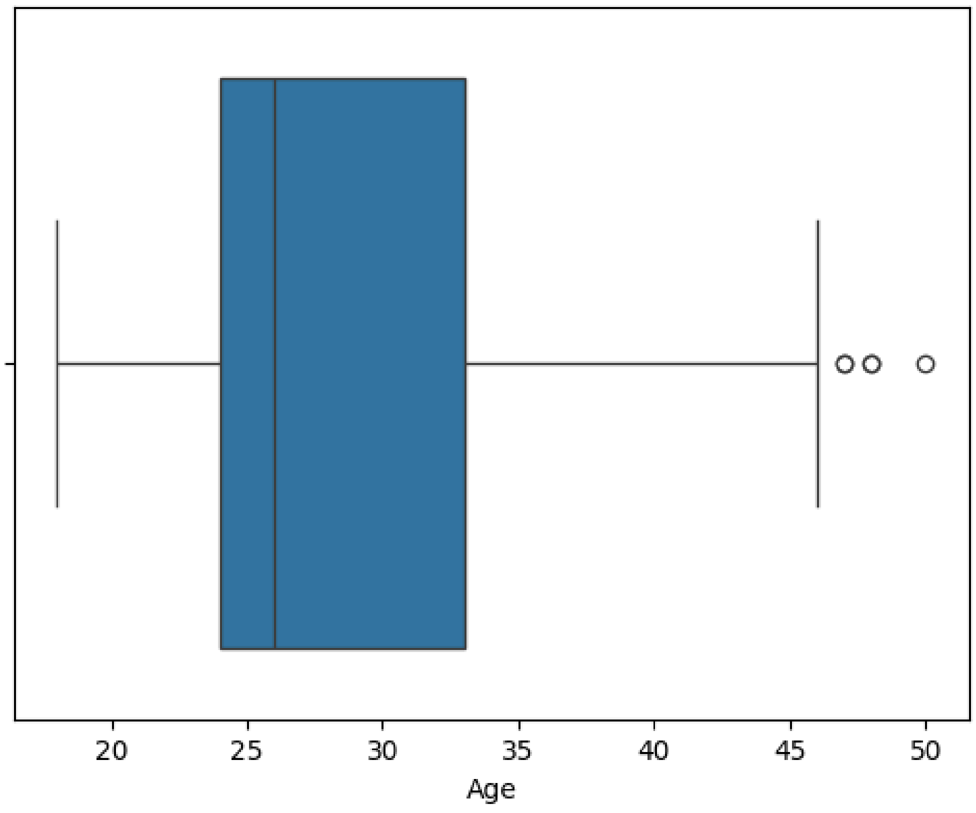


```
# Calculating IQR for Age
Age_p25 = np.percentile(df["Age"], 25)
Age_p75 = np.percentile(df["Age"], 75)
Age_iqr = (Age_p75 - Age_p25)
print(Age_iqr)
```

```
9.0
```

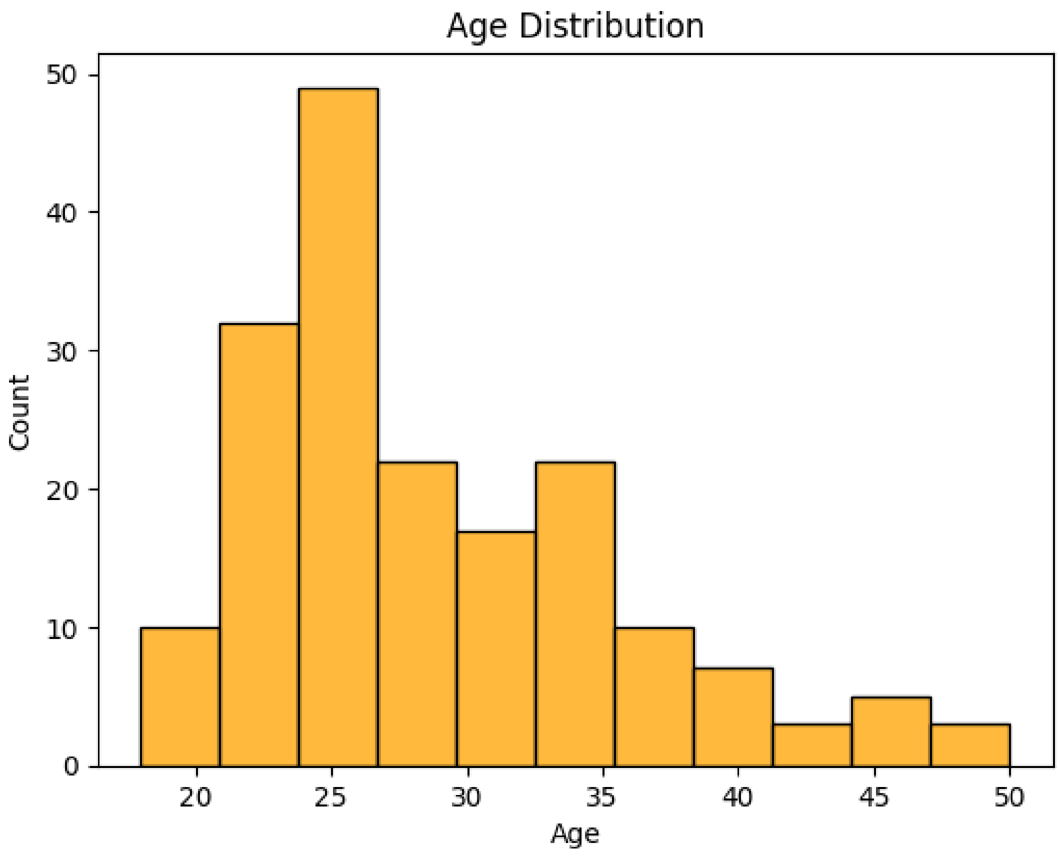
```
sns.boxplot(data = df["Age"], orient = "h")
```

```
<Axes: xlabel='Age'>
```



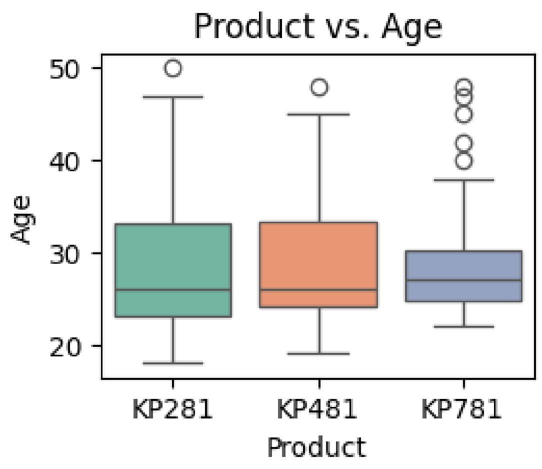
```
#Histplot for Age count
df_age = df["Age"]
sns.histplot(data=df_age, color='orange')
plt.title("Age Distribution")
```

```
Text(0.5, 1.0, 'Age Distribution')
```



```
#Boxplot for Product vs Age Distribution
plt.subplot(2, 2, 1)
sns.boxplot(data=df, x="Product", y="Age", palette="Set2")
plt.title("Product vs. Age")
```

```
<ipython-input-36-5ef890462d23>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

  sns.boxplot(data=df, x="Product", y="Age", palette="Set2")
Text(0.5, 1.0, 'Product vs. Age')
```

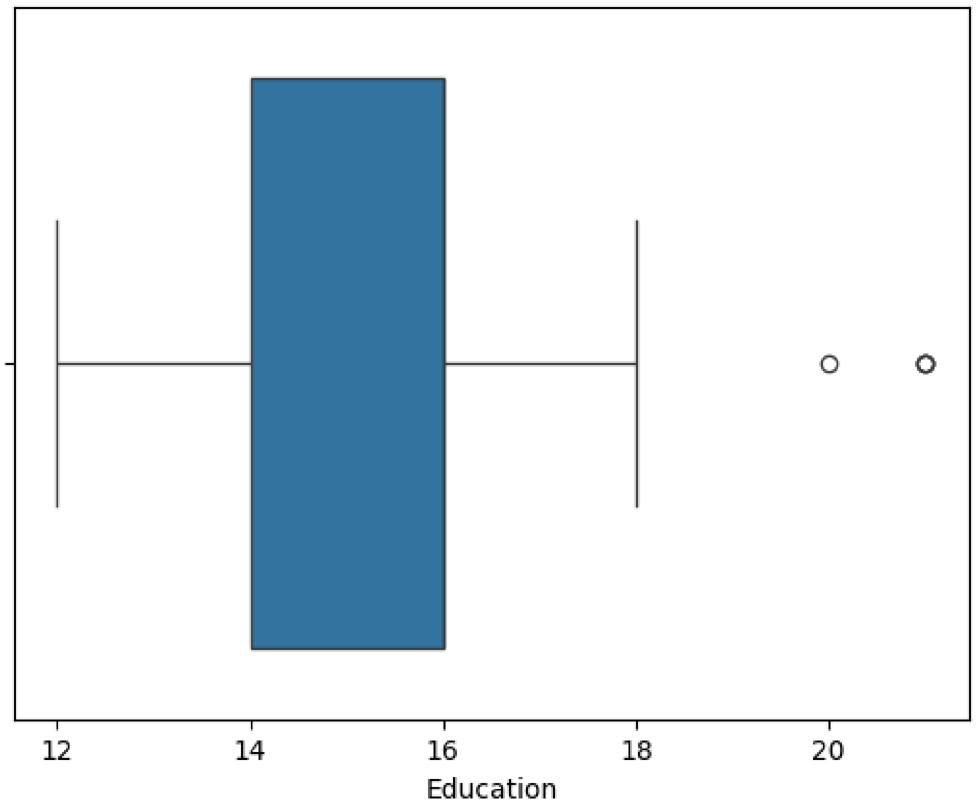


```
# Calculating IQR for Education
Education_p25 = np.percentile(df["Education"], 25)
Education_p75 = np.percentile(df["Education"], 75)
Education_iqr = (Education_p75 - Education_p25)
print(Education_iqr)
```

```
2.0
```

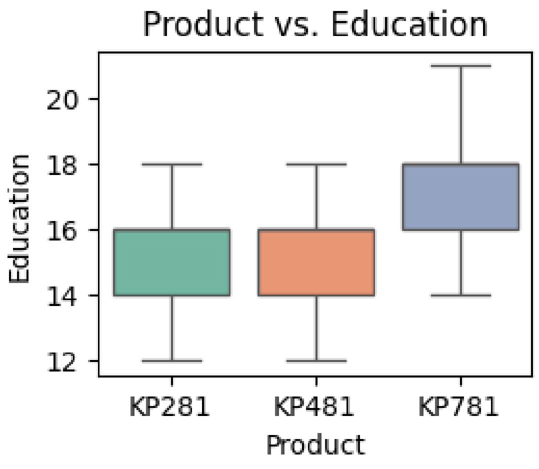
```
sns.boxplot(data = df["Education"], orient = "h")
```

```
<Axes: xlabel='Education'>
```



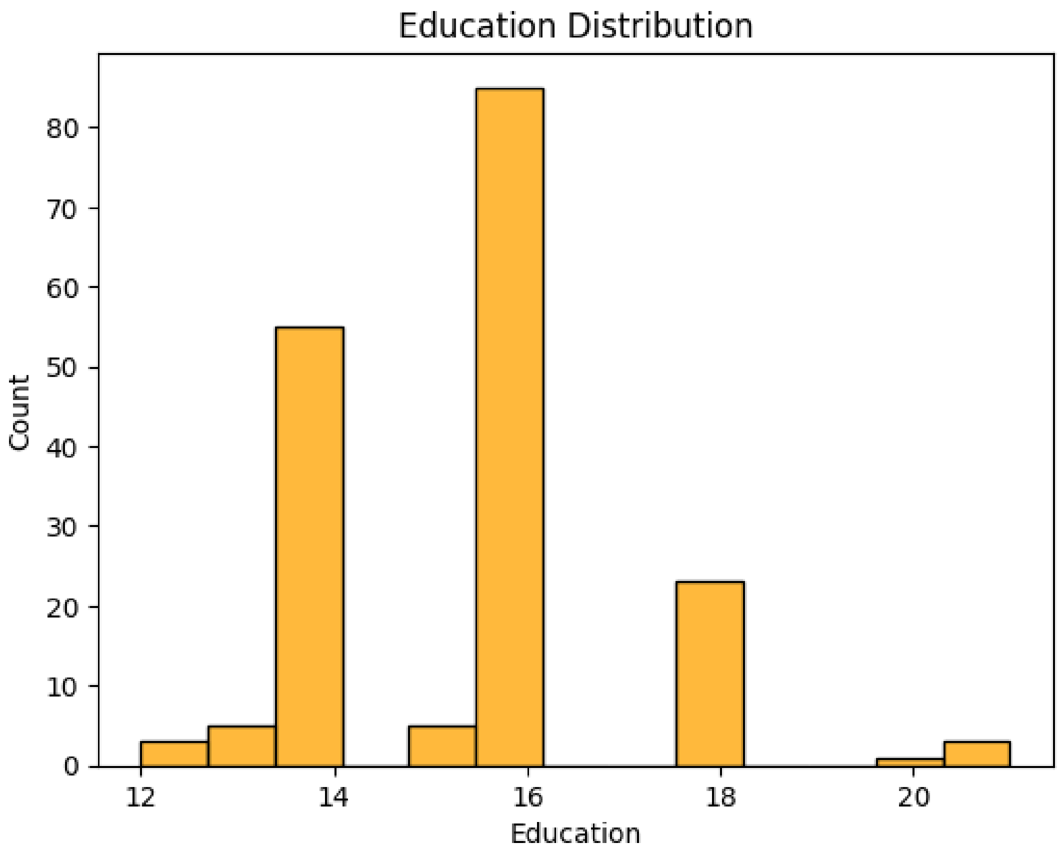
```
#Boxplot for Product vs Education Distribution
plt.subplot(2, 2, 1)
sns.boxplot(data=df, x="Product", y="Education", palette="Set2")
plt.title("Product vs. Education")
```

```
<ipython-input-37-e3e65c1e7d23>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

  sns.boxplot(data=df, x="Product", y="Education", palette="Set2")
Text(0.5, 1.0, 'Product vs. Education')
```



```
#Histplot for Education count
df_Education = df["Education"]
sns.histplot(data=df_Education, color='orange')
plt.title("Education Distribution")
```

```
Text(0.5, 1.0, 'Education Distribution')
```

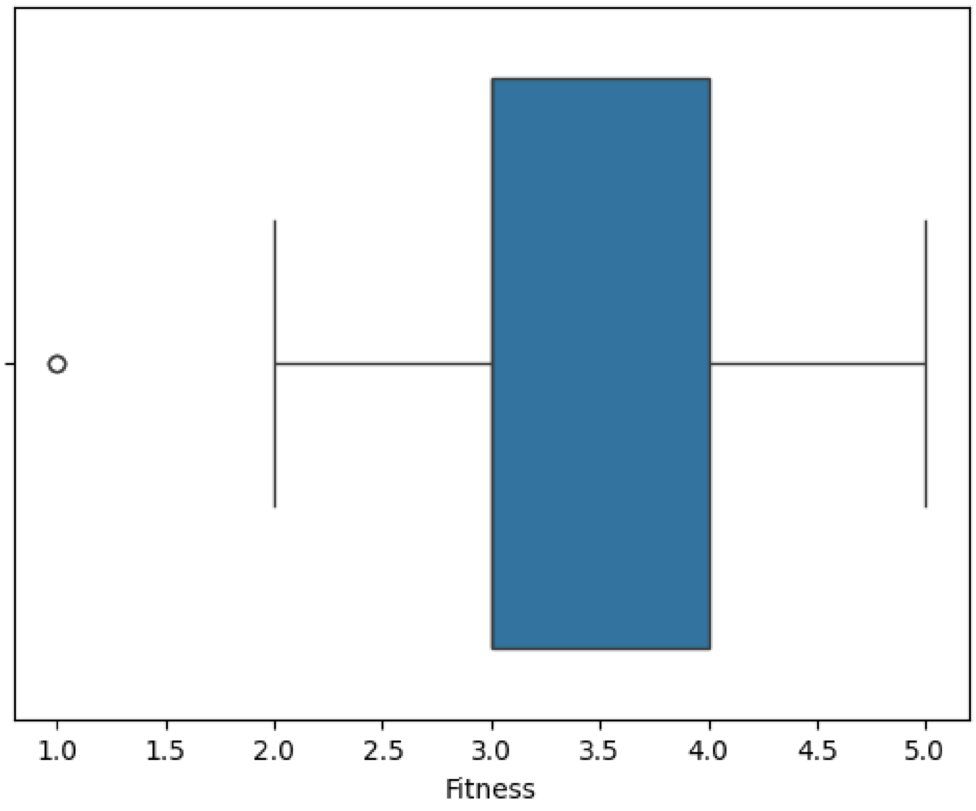


```
# Calculating IQR for Fitness
Fitness_p25 = np.percentile(df["Fitness"], 25)
Fitness_p75 = np.percentile(df["Fitness"], 75)
Fitness_iqr = (Fitness_p75 - Fitness_p25)
print(Fitness_iqr)
```

```
1.0
```

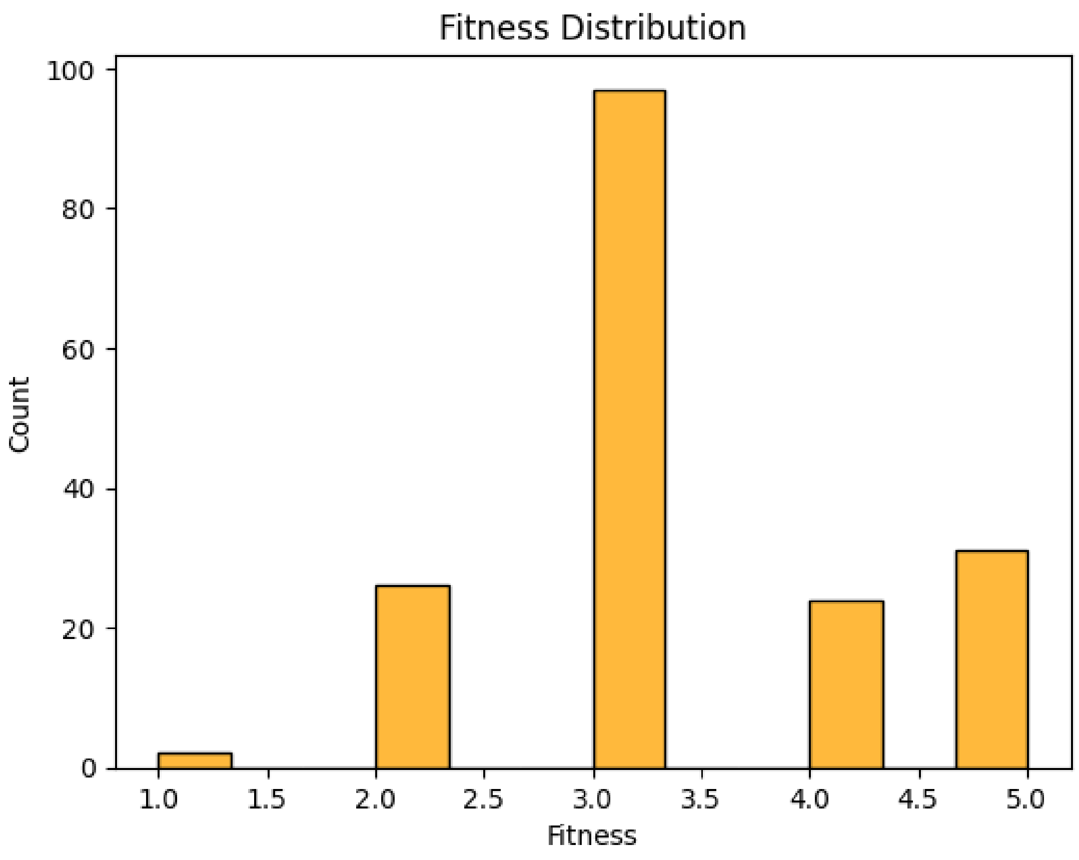
```
sns.boxplot(data = df["Fitness"], orient = "h")
```

```
<Axes: xlabel='Fitness'>
```



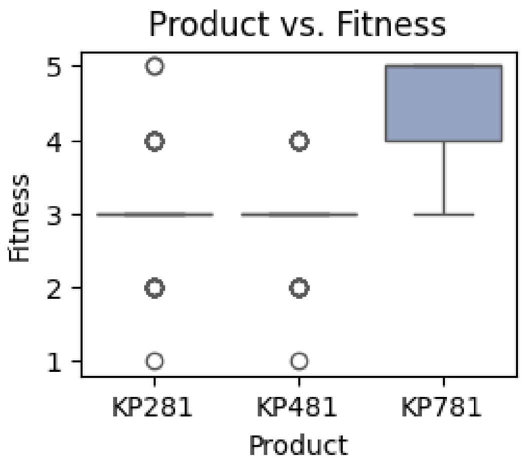
```
#Histplot for Fitness count
df_Fitness = df["Fitness"]
sns.histplot(data=df_Fitness, color='orange')
plt.title("Fitness Distribution")
```

```
Text(0.5, 1.0, 'Fitness Distribution')
```



```
#Boxplot for Product vs Fitness Distribution
plt.subplot(2, 2, 1)
sns.boxplot(data=df, x="Product", y="Fitness", palette="Set2")
plt.title("Product vs. Fitness")
```

```
<ipython-input-38-914a7a08d5e3>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

  sns.boxplot(data=df, x="Product", y="Fitness", palette="Set2")
Text(0.5, 1.0, 'Product vs. Fitness')
```

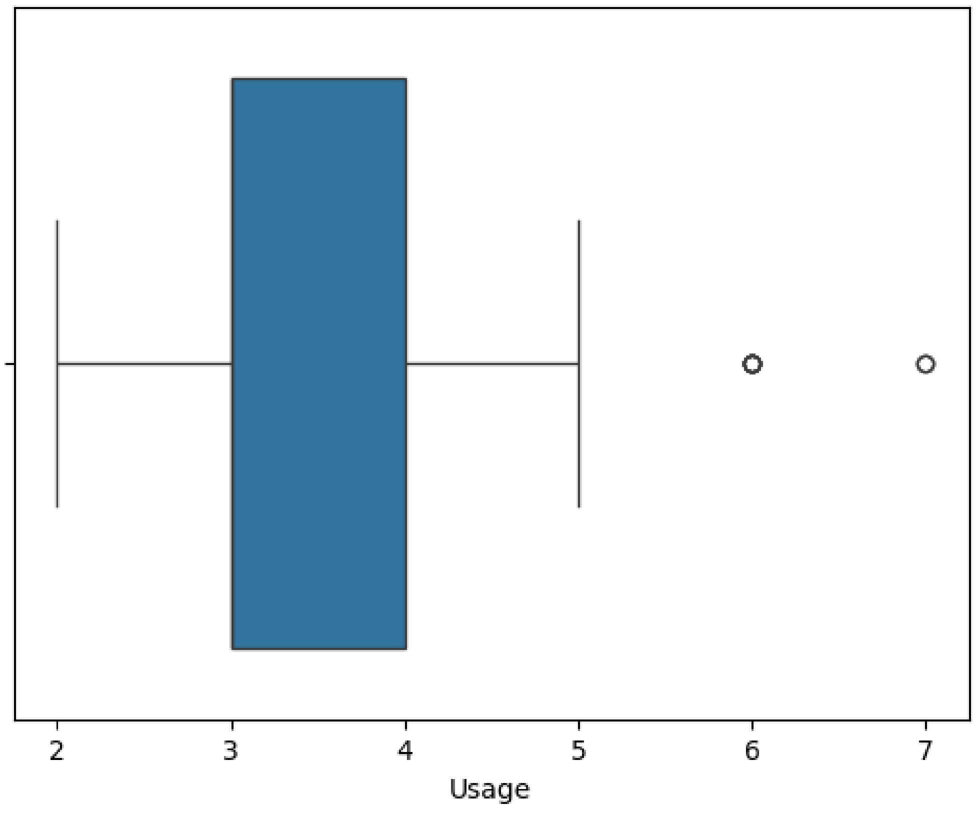


```
Usage_p25 = np.percentile(df["Usage"], 25)
Usage_p75 = np.percentile(df["Usage"], 75)
Usage_iqr = (Usage_p75 - Usage_p25)
print(Usage_iqr)
```

```
1.0
```

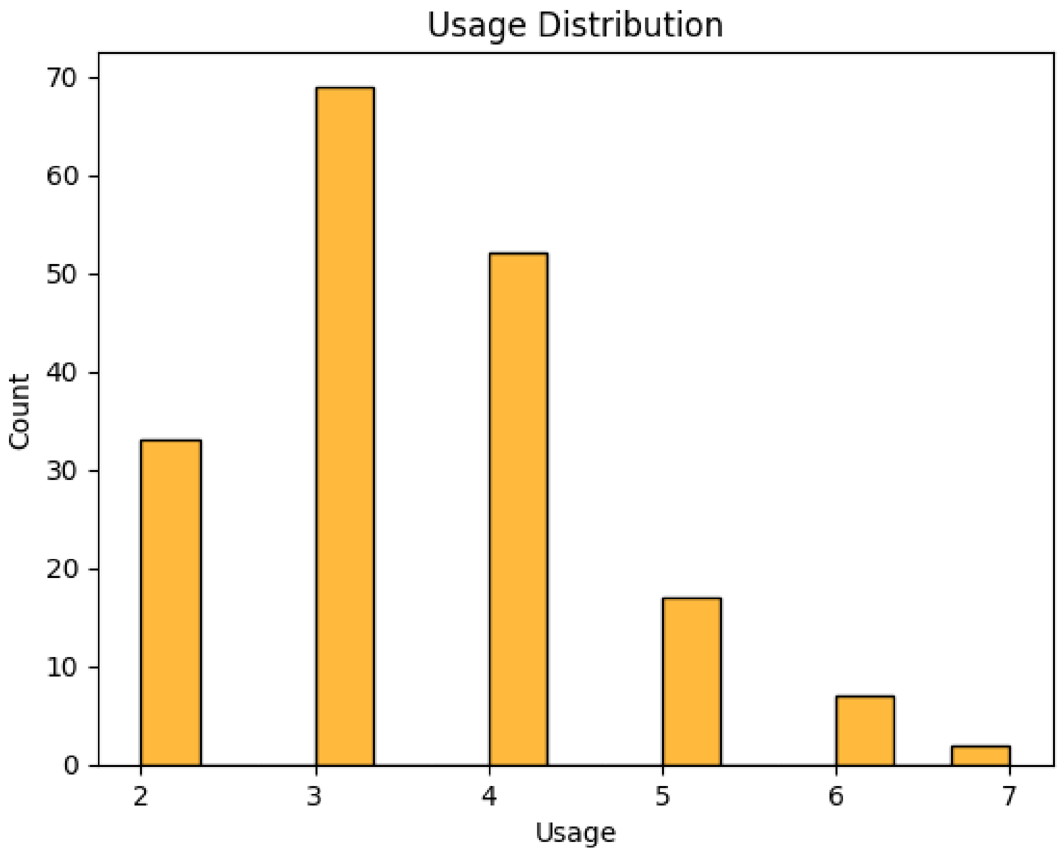
```
sns.boxplot(data = df["Usage"], orient = "h")
```

```
<Axes: xlabel='Usage'>
```



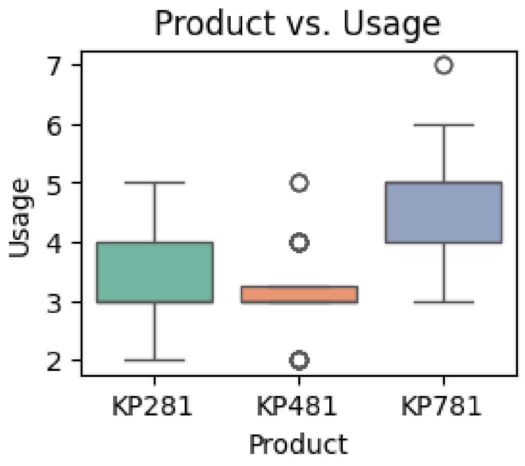
```
#Histplot for Usage count
df_Usage = df["Usage"]
sns.histplot(data=df_Usage, color='orange')
plt.title("Usage Distribution")
```

```
Text(0.5, 1.0, 'Usage Distribution')
```



```
#Boxplot for Product vs Usage Distribution
plt.subplot(2, 2, 1)
sns.boxplot(data=df, x="Product", y="Usage", palette="Set2")
plt.title("Product vs. Usage")
```

```
<ipython-input-39-85b0b889c180>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

  sns.boxplot(data=df, x="Product", y="Usage", palette="Set2")
Text(0.5, 1.0, 'Product vs. Usage')
```

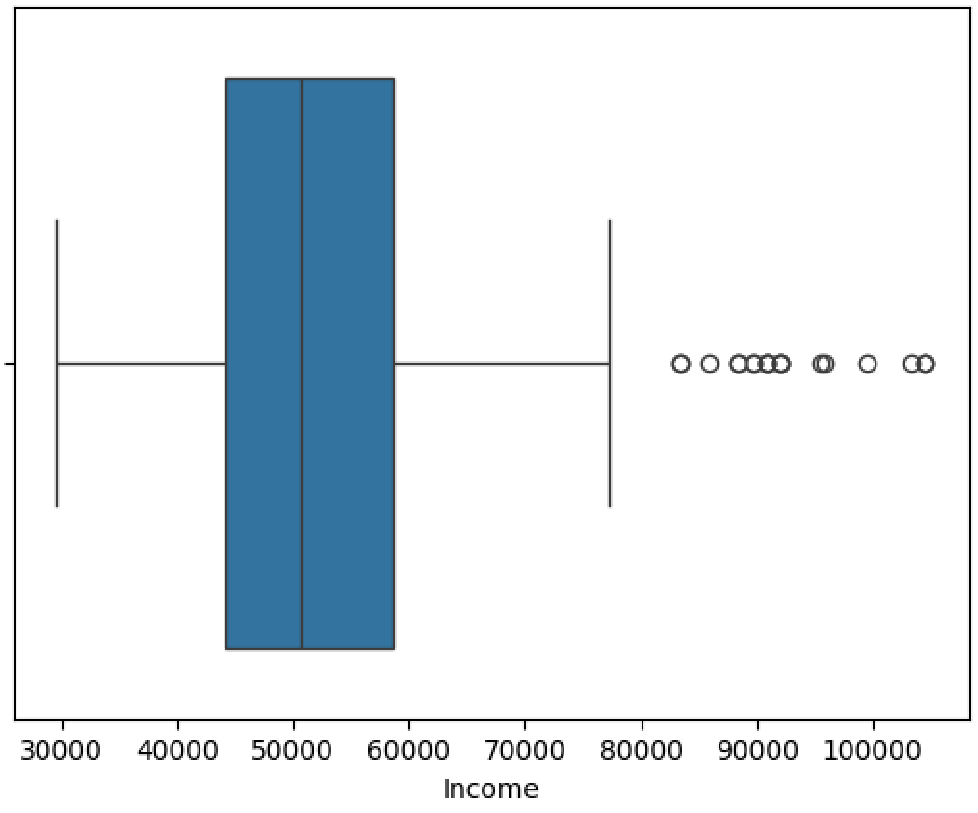


```
# Calculating IQR for Income
Income_p25 = np.percentile(df["Income"], 25)
Income_p75 = np.percentile(df["Income"], 75)
Income_iqr = (Income_p75 - Income_p25)
print(Income_iqr)
```

```
14609.25
```

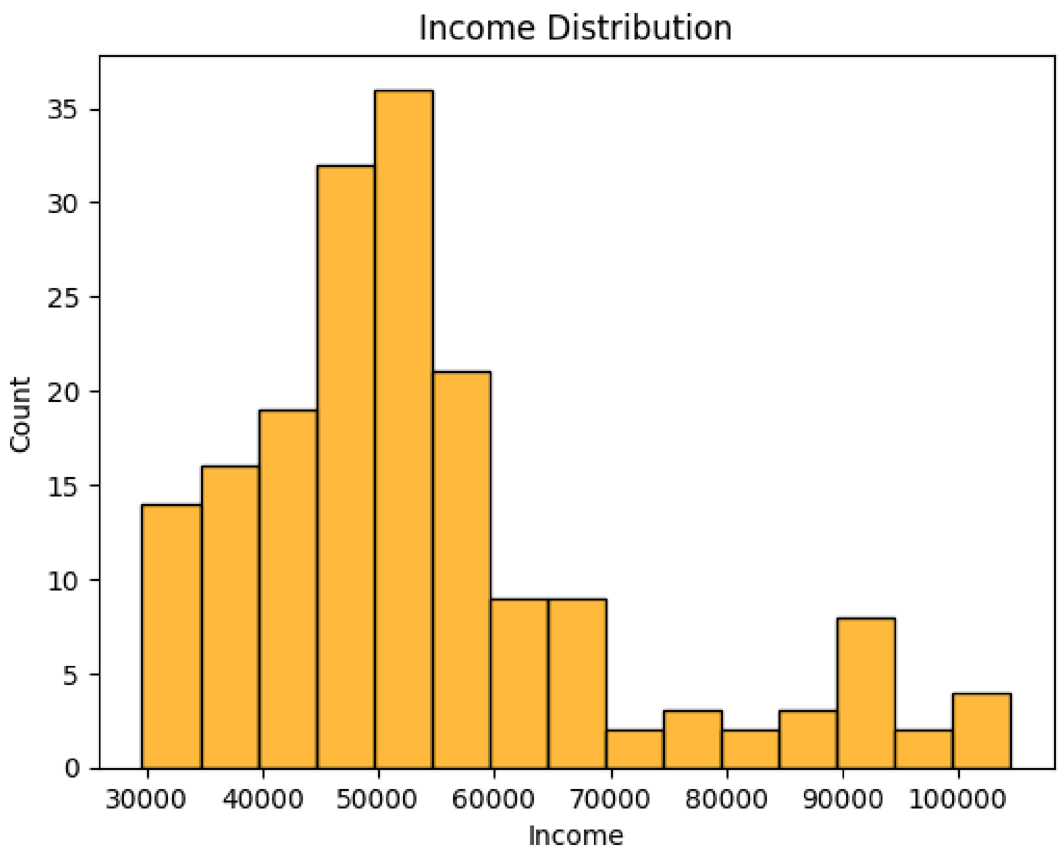
```
sns.boxplot(data = df["Income"], orient = "h")
```

```
<Axes: xlabel='Income'>
```



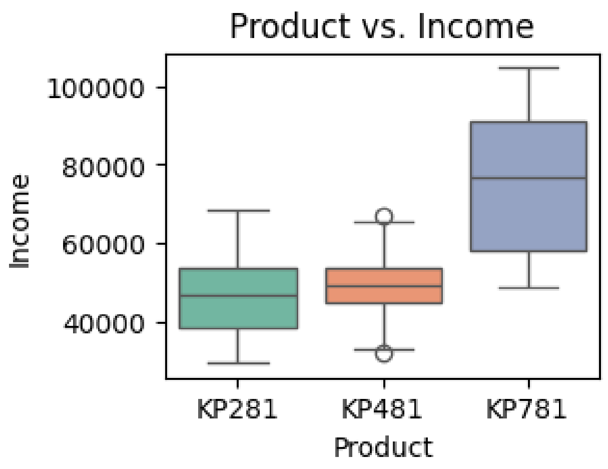
```
#Histplot for Income count
df_Income = df["Income"]
sns.histplot(data=df_Income, color='orange')
plt.title("Income Distribution")
```

```
Text(0.5, 1.0, 'Income Distribution')
```



```
#Boxplot for Product vs Income Distribution
plt.subplot(2, 2, 1)
sns.boxplot(data=df, x="Product", y="Income", palette="Set2")
plt.title("Product vs. Income")
```

```
<ipython-input-40-643e1d7038e4>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

  sns.boxplot(data=df, x="Product", y="Income", palette="Set2")
Text(0.5, 1.0, 'Product vs. Income')
```

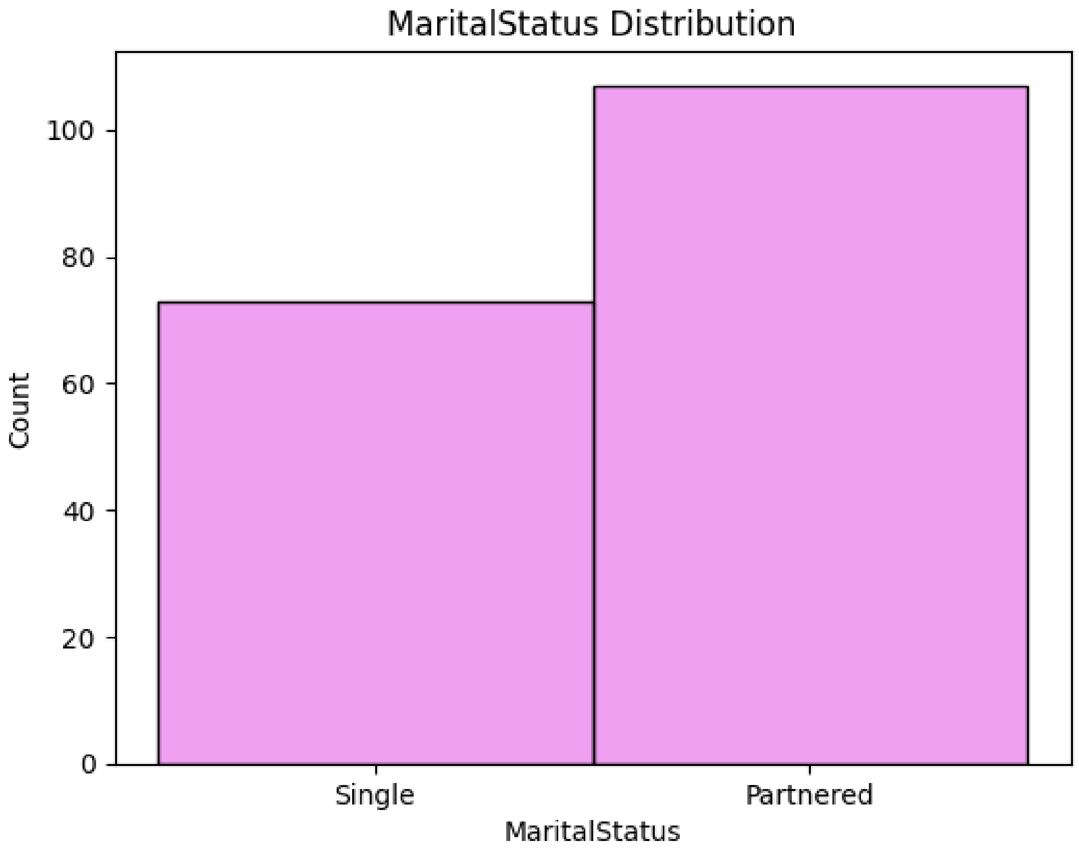


```
# Calculate the difference between mean and median for each numerical column
difference_mean_median = df.select_dtypes(include=['int64', 'float64']).apply(lambda x: x.me

# Print the difference
print(difference_mean_median)
```

```
Age             2.788889
Education      -0.427778
Usage           0.455556
Fitness         0.311111
Income       3123.077778
Miles           9.194444
dtype: float64
```

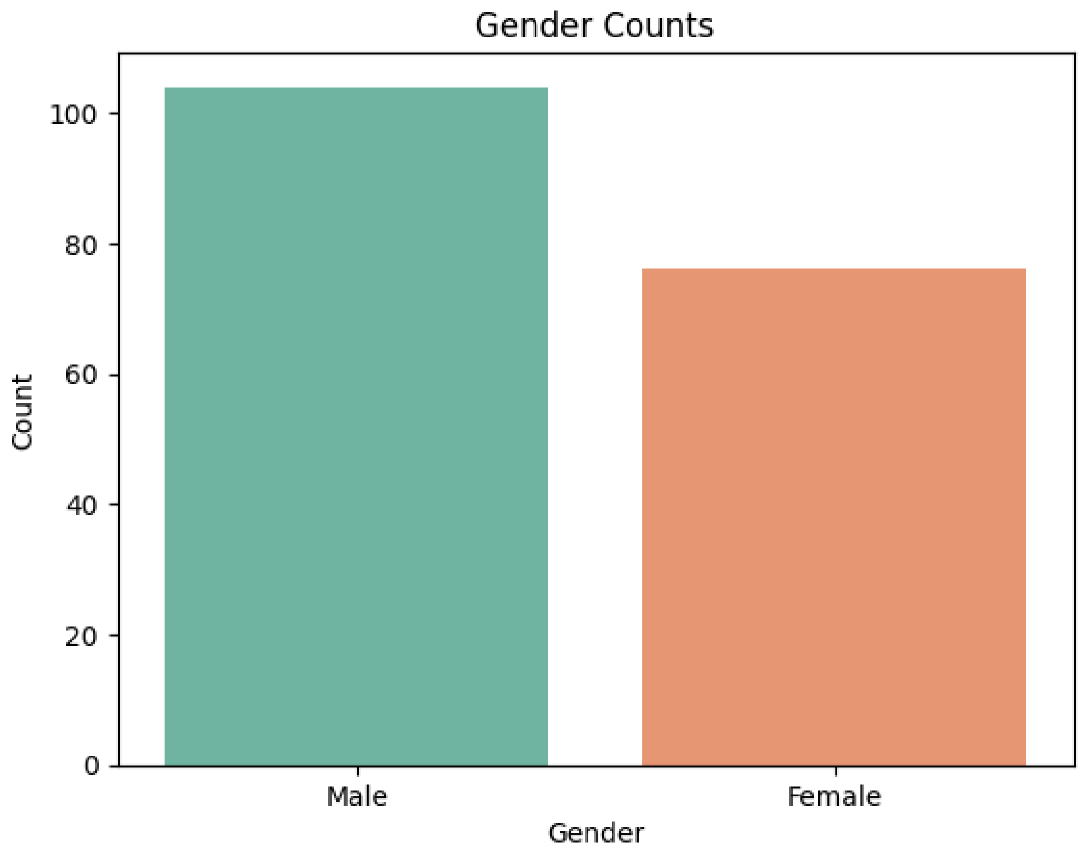
```
# MaritalStatus Distribution
df_MaritalStatus = df["MaritalStatus"]
sns.histplot(data=df_MaritalStatus, color='violet')
plt.title("MaritalStatus Distribution")
```

```
Text(0.5, 1.0, 'MaritalStatus Distribution')
```

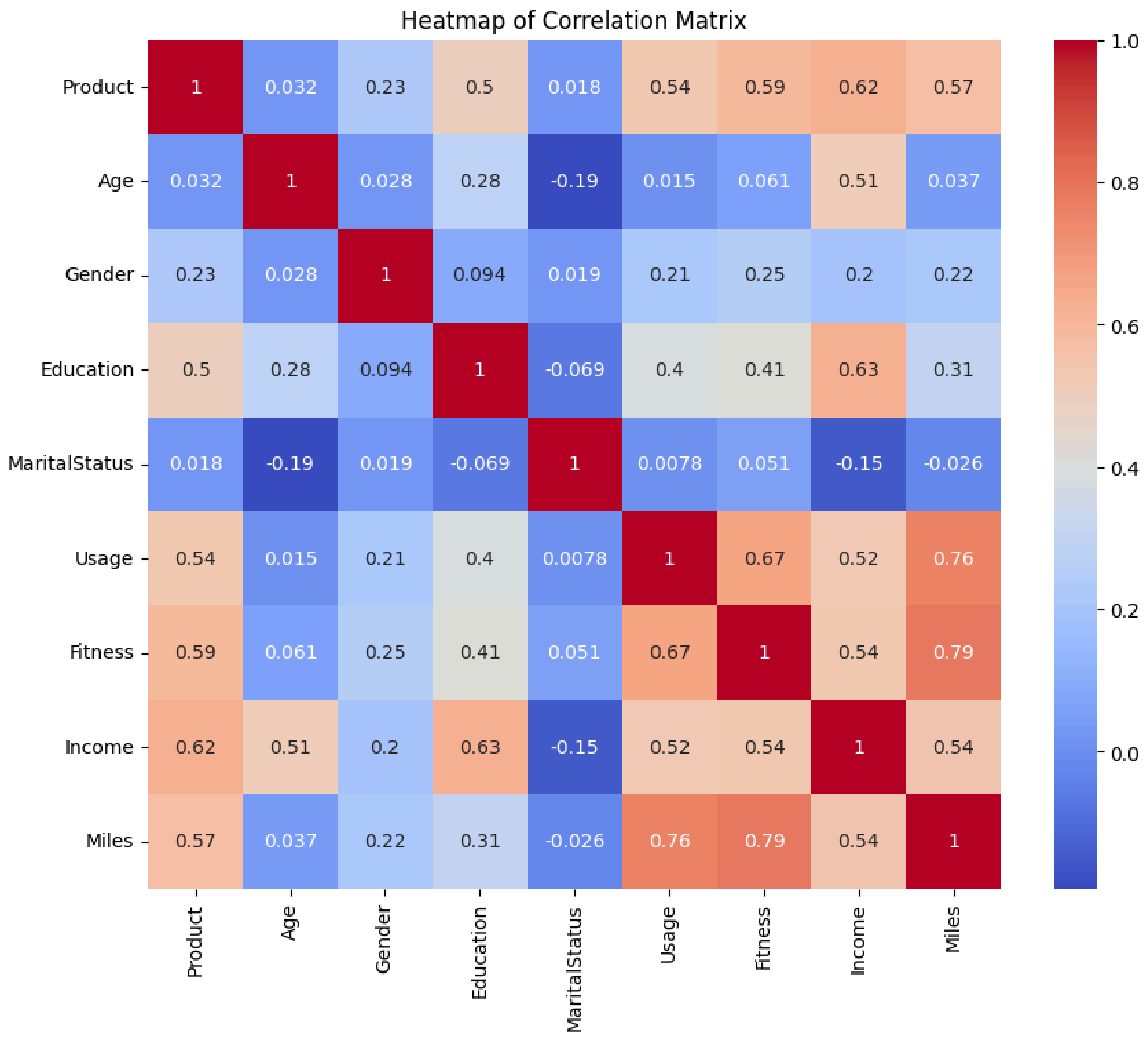


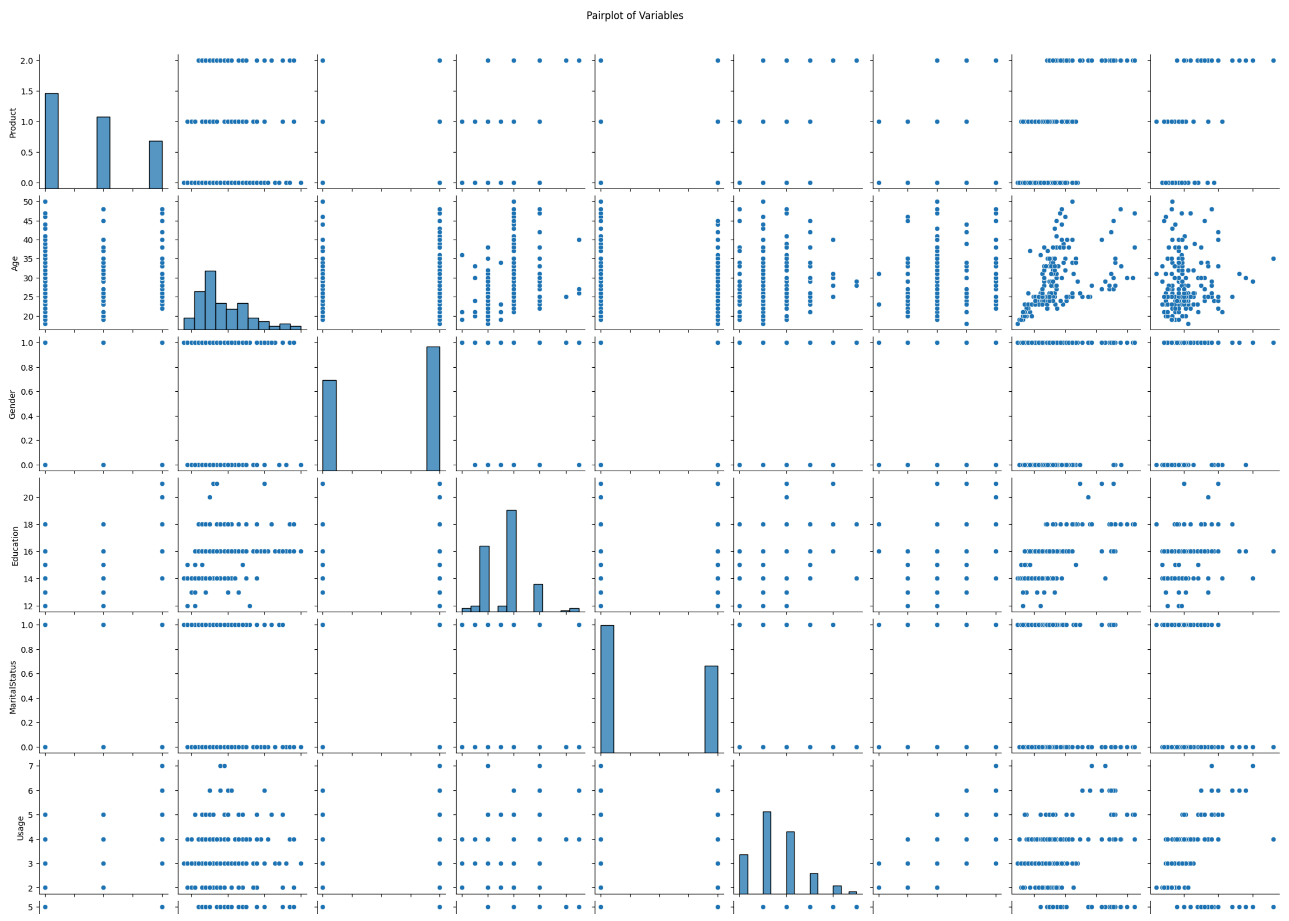
```
# Gender Count
sns.countplot(data=df, x='Gender', palette='Set2')
plt.xlabel("Gender")
plt.ylabel("Count")
plt.title("Gender Counts")
```

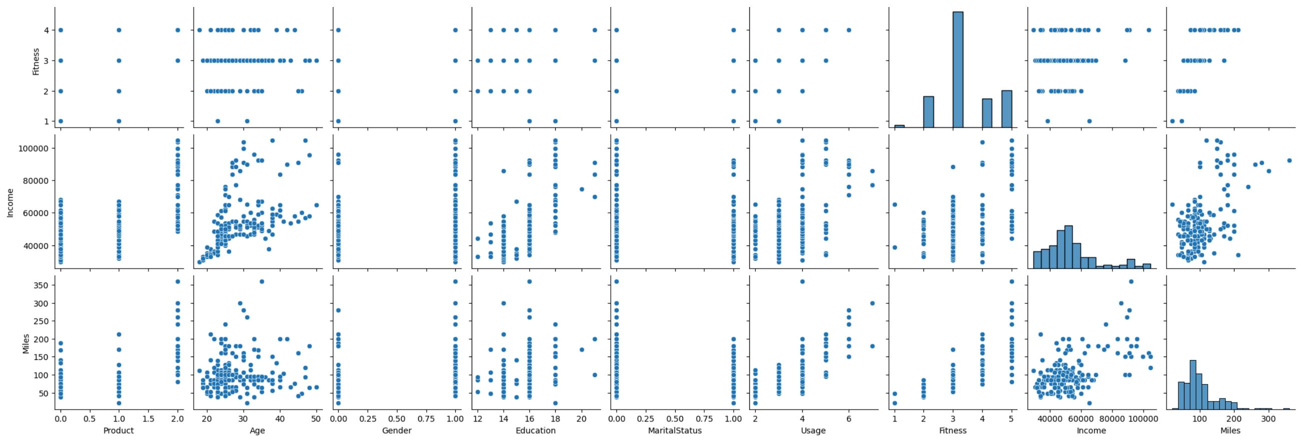
```
<ipython-input-43-0f5118677da0>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

  sns.countplot(data=df, x='Gender', palette='Set2')
Text(0.5, 1.0, 'Gender Counts')
```



```
# Calculate the correlation matrix
correlation_matrix = df.corr()

# Plot the correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Heatmap of Correlation Matrix')
plt.show()

# Generate pairplots
sns.pairplot(df)
plt.suptitle('Pairplot of Variables', y=1.02)
plt.show()
```


Heatmap of Correlation Matrix
Product 1 0.032 0.23 0.5 0.018 0.54 0.59 0.62 0.57
Age 0.032 1 0.028 0.28 -0.19 0.015 0.061 0.51 0.037
Gender 0.23 0.028 1 0.094 0.019 0.21 0.25 0.2 0.22
Education 0.5 0.28 0.094 1 -0.069 0.4 0.41 0.63 0.31
MaritalStatus 0.018 -0.19 0.019 -0.069 1 0.0078 0.051 -0.15 -0.026
Usage 0.54 0.015 0.21 0.4 0.0078 1 0.67 0.52 0.76
Fitness 0.59 0.061 0.25 0.41 0.051 0.67 1 0.54 0.79
Income 0.62 0.51 0.2 0.63 -0.15 0.52 0.54 1 0.54
Miles 0.57 0.037 0.22 0.31 -0.026 0.76 0.79 0.54 1
Product Age Gender Education MaritalStatus Usage Fitness Income Miles
1.0 0.8 0.6 0.4 0.2 0.0


Pairplot of Variables
Product
Age
Gender
Education
MaritalStatus
Usage



```
# Count the occurrences of each product
product_count = df['Product'].value_counts()

# Calculate the total number of customers
customers_count = df.shape[0]


#Calcualte the percentage of each product
product_percentages = (product_count / customers_count) * 100
```