

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

```
data = pd.read_csv('/content/Fitness_trackers.csv')
df = pd.DataFrame(data)
```

## ▼ Exploratory Data Analysis

It is the process of looking at a dataset to understand how it is structured, trying to understand the dataset using basic statistics and data visualizations that can give us insights into patterns in the data for further analysis

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 565 entries, 0 to 564
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Brand Name            565 non-null    object
 1   Device Type           565 non-null    object
 2   Model Name            565 non-null    object
 3   Color                 565 non-null    object
 4   Selling Price         565 non-null    object
 5   Original Price        565 non-null    object
 6   Display               565 non-null    object
 7   Rating (Out of 5)     514 non-null    float64
 8   Strap Material        565 non-null    object
 9   Average Battery Life (in days) 565 non-null    int64
10  Reviews               78 non-null     object
dtypes: float64(1), int64(1), object(9)
memory usage: 48.7+ KB
```

The dataset has mostly got non-numeric, categorical variables and a few numeric variables

```
unique_counts = df.nunique()
print(unique_counts)
```

```
Brand Name            20
Device Type           2
Model Name            384
Color                 146
Selling Price         244
Original Price        178
Display               7
Rating (Out of 5)     23
Strap Material        12
Average Battery Life (in days) 17
Reviews               46
dtype: int64
```

```
# Remove commas from 'Selling Price' and convert to float
df['Selling Price'] = df['Selling Price'].str.replace(',', '').astype(float)
```

```
# Count the occurrences of each color
display_counts = data['Display'].value_counts()
# Plotting a pie chart using matplotlib
plt.figure(figsize=(8,8))
plt.pie(display_counts, labels=display_counts.index, autopct='%1.1f%%', colors=sns.color_palette("pastel"))
plt.title('Distribution of Types of Displays')
plt.show()
```



AMOLED Display



 <Axes: xlabel='Strap Material'>

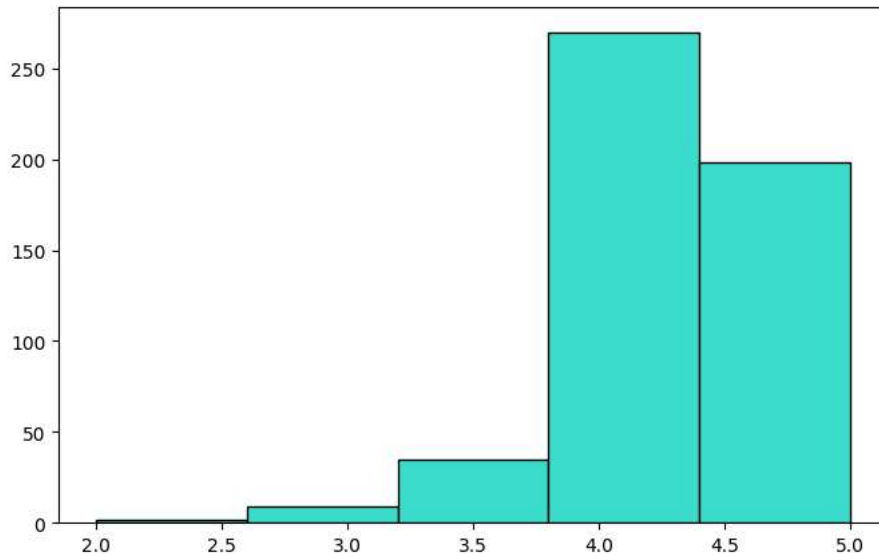


The bar chart gives insights into the distribution of different strap materials, helping to identify the most popular options

The bar chart gives insights into the distribution of different strap materials, helping to identify the most popular options

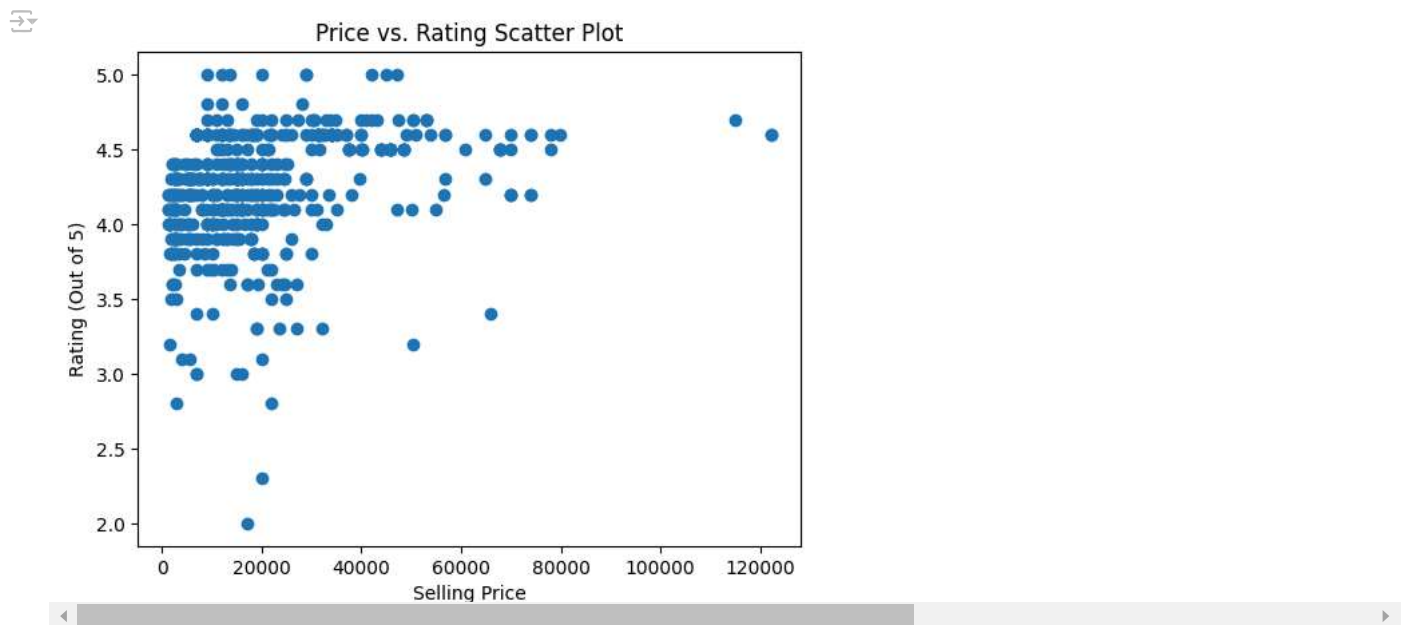
```
plt.figure(figsize=(8, 5))
plt.hist(df['Rating (Out of 5)'].dropna(), bins=5, color='turquoise', edgecolor='black')
```

```
(array([ 2.,  9., 35., 270., 198.]),
 array([2. , 2.6, 3.2, 3.8, 4.4, 5. ]),
 <BarContainer object of 5 artists>)
```



This histogram shows how ratings are spread across the dataset.

```
plt.scatter(df['Selling Price'].astype(float), df['Rating (Out of 5)'])
plt.xlabel('Selling Price')
plt.ylabel('Rating (Out of 5)')
plt.title('Price vs. Rating Scatter Plot')
plt.show()
```

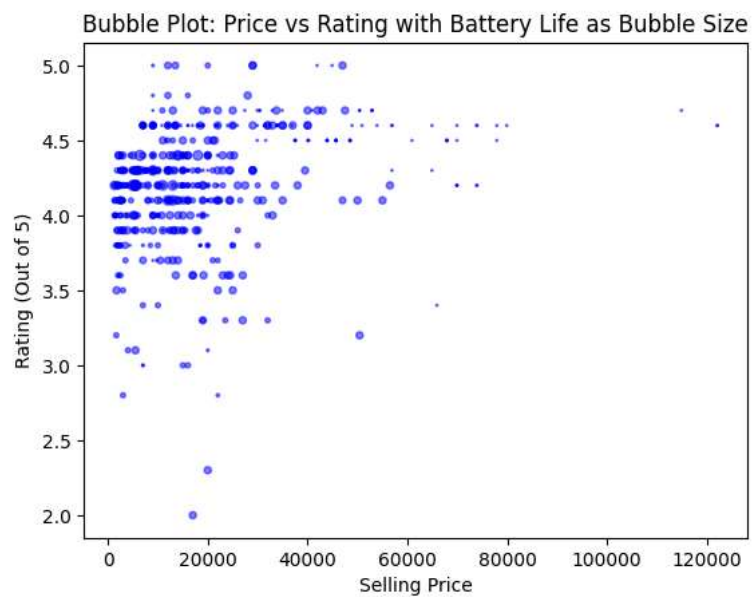


This visualization helps in identifying whether higher-priced products tend to have better or worse ratings. It might reveal trends, such as if there is a positive, negative, or no correlation between price and customer ratings

```
# Create the bubble plot
plt.scatter(df['Selling Price'], df['Rating (Out of 5)'],
            s=df['Average Battery Life (in days)'],
            alpha=0.5, color='blue')

# Adding labels and title
plt.xlabel('Selling Price')
plt.ylabel('Rating (Out of 5)')
plt.title('Bubble Plot: Price vs Rating with Battery Life as Bubble Size')
```

```
# Show plot  
plt.show()
```



This plot visualizes how the price of a product correlates with its rating while incorporating the additional factor of battery life. Larger bubbles indicate longer battery life, allowing you to identify products that are both well-rated and have longer battery life, and whether they are positioned in the higher or lower price range.