In [5]:

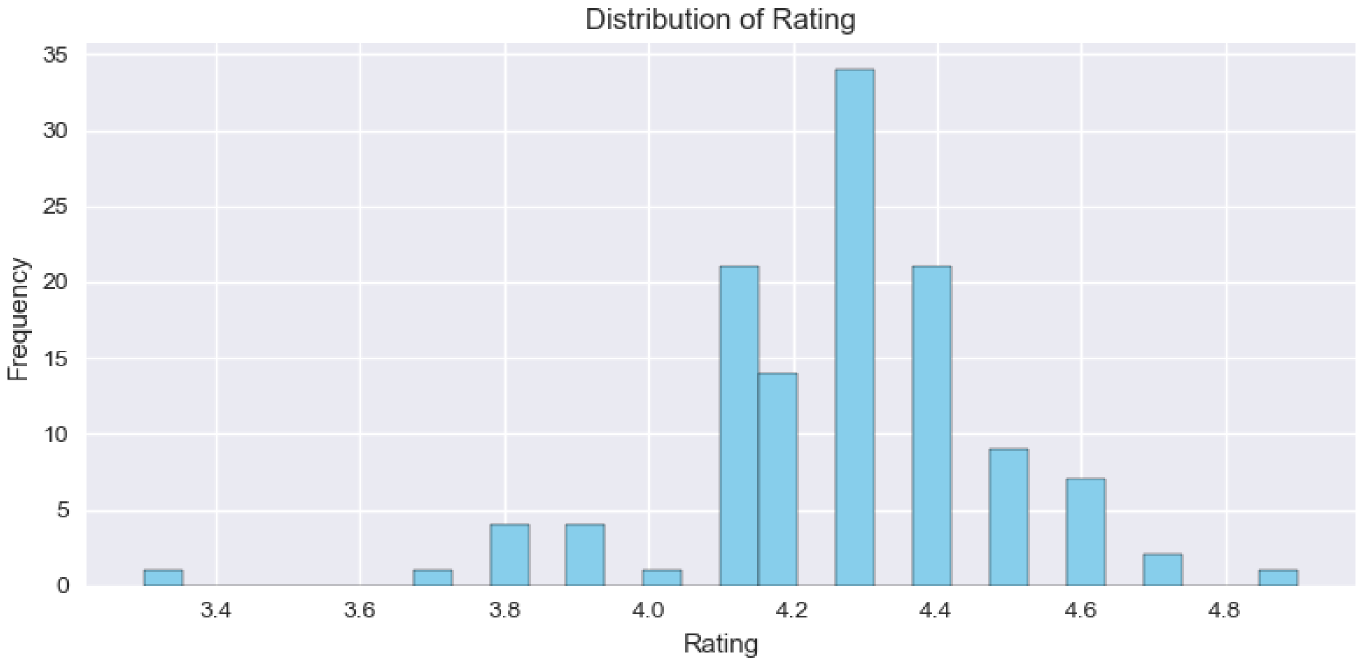
```
# ============================
# TASK 3: Data Visualization for Large Dataset
# ============================

# 1. Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import os

plt.style.use('seaborn-v0_8')
pd.set_option('display.max_columns', None)

# 2. Load Dataset
# Make sure your CSV is in the same folder as this notebook or provide full path
csv_files = [f for f in os.listdir() if f.endswith('.csv')]
if not csv_files:
    raise FileNotFoundError("No CSV file found. Please add your dataset.")
df = pd.read_csv(csv_files[0])
print("Dataset Loaded:", csv_files[0])
print("Dataset Shape:", df.shape)

# 3. Detect Date Column
date_cols = [col for col in df.columns if 'date' in col.lower()]
if date_cols:
    df[date_cols[0]] = pd.to_datetime(df[date_cols[0]], errors='coerce')
    date_col = date_cols[0]
else:
    date_col = None

# 4. Downsample function for large datasets
def downsample(df, step=10):
    return df.iloc[::step] if len(df) > 1000 else df

# 5. Closing Price Trend
if date_col and 'Close' in df.columns:
    df_sample = downsample(df)
    plt.figure(figsize=(12,5))
    plt.plot(df_sample[date_col], df_sample['Close'], color='blue')
    plt.title("Closing Price Trend")
    plt.xlabel("Date")
    plt.ylabel("Close Price")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

# 6. OHLC Overview
if date_col and all(col in df.columns for col in ['Open', 'High', 'Low', 'Close']):
    df_sample = downsample(df)
    plt.figure(figsize=(12,5))
    plt.plot(df_sample[date_col], df_sample['High'], label='High', color='green')
    plt.plot(df_sample[date_col], df_sample['Low'], label='Low', color='red')
    plt.fill_between(df_sample[date_col], df_sample['Open'], df_sample['Close'], color='
    plt.title("Open-High-Low-Close Overview")
    plt.xlabel("Date")
    plt.ylabel("Price")
    plt.legend()
    plt.xticks(rotation=45)
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

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

# 7. Volume Chart
if date_col and 'Volume' in df.columns:
    df_sample = downsample(df)
    plt.figure(figsize=(12,4))
    plt.bar(df_sample[date_col], df_sample['Volume'], color='orange')
    plt.title("Trading Volume Over Time")
    plt.xlabel("Date")
    plt.ylabel("Volume")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

# 8. Histograms of Numeric Columns (Top 6)
num_cols = df.select_dtypes(include=['float64', 'int64']).columns
plot_cols = num_cols[:6]
for col in plot_cols:
    plt.figure(figsize=(8,4))
    plt.hist(df[col], bins=30, color='skyblue', edgecolor='black')
    plt.title(f"Distribution of {col}")
    plt.xlabel(col)
    plt.ylabel("Frequency")
    plt.tight_layout()
    plt.show()

# 9. Correlation Heatmap using matplotlib
if len(num_cols) > 1:
    plt.figure(figsize=(8,5))
    corr = df[plot_cols].corr()
    plt.imshow(corr, cmap='coolwarm', interpolation='none')
    plt.colorbar()
    plt.xticks(range(len(plot_cols)), plot_cols, rotation=45)
    plt.yticks(range(len(plot_cols)), plot_cols)
    for i in range(len(plot_cols)):
        for j in range(len(plot_cols)):
            plt.text(j, i, f"{corr.iloc[i, j]:.2f}", ha='center', va='center', color='bl
    plt.title("Correlation Heatmap")
    plt.tight_layout()
    plt.show()

# 10. Completion Message
print("Data Visualization Completed Successfully.")
```

```
Dataset Loaded: flipkart_laptop_data.csv
Dataset Shape: (120, 3)
```

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF



```
Data Visualization Completed Successfully.
```

In [ ]:

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF