Replicating Excel Charts in Python

Alexander Joel Molinar Sr

October 18, 2024

1 Replicating Excel Charts in Python

Excel offers a wide range of chart types for data visualization. In this notebook, we'll recreate various Excel charts using Python libraries like `matplotlib`, `seaborn`, and `plotly`. We'll cover:

1. Column Charts
2. Bar Charts
3. Line Charts
4. Pie Charts
5. Scatter Plots
6. Area Charts
7. Bubble Charts
8. Histograms
9. Box and Whisker Plots
10. Waterfall Charts

Let's get started!

1.1 Importing Necessary Libraries

We'll use several libraries for data manipulation and visualization.

[1]:
```
# Install libraries if not already installed
# !pip install pandas numpy matplotlib seaborn plotly

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

%matplotlib inline

# Set a style for the plots
sns.set_style('whitegrid')
```

1.2 Generating Sample Data

We'll create some sample datasets to use in our charts.

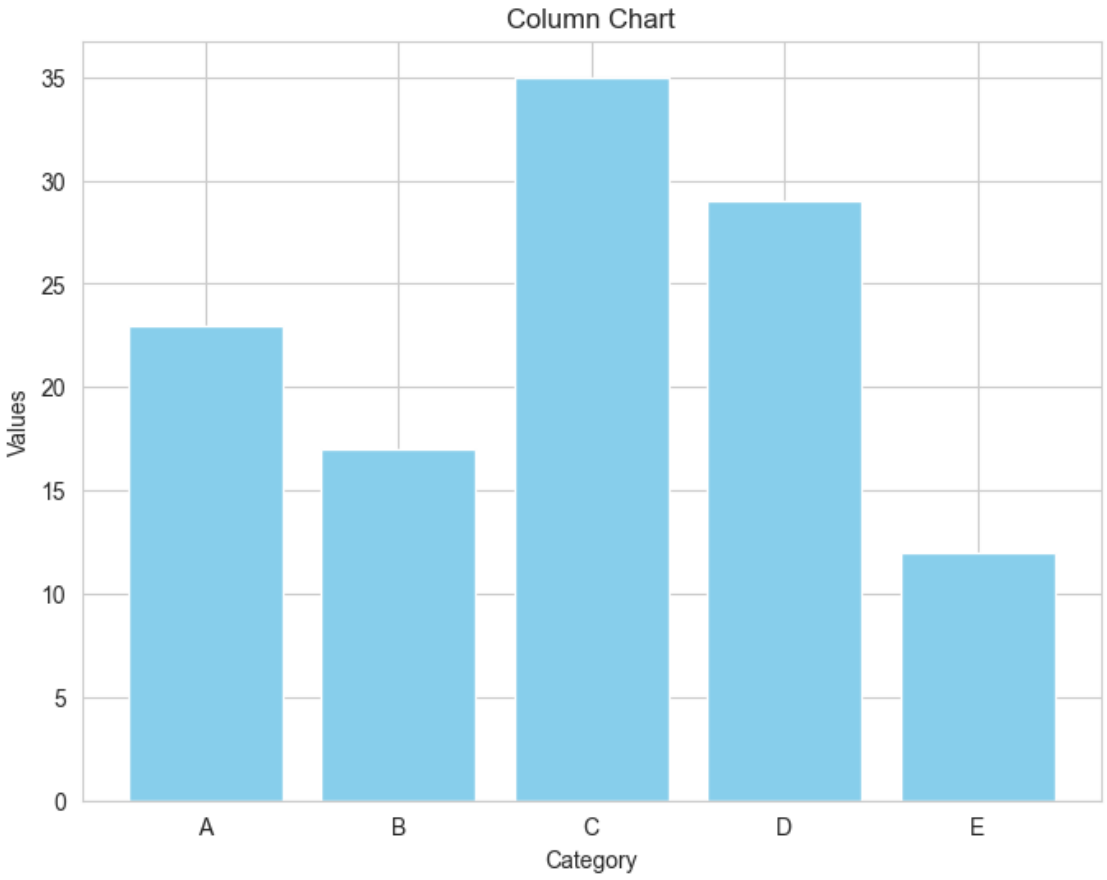
[2]:
```python
# Sample data for charts

# Data for Column and Bar Charts
categories = ['A', 'B', 'C', 'D', 'E']
values = [23, 17, 35, 29, 12]

# Data for Line Chart
dates = pd.date_range(start='2021-01-01', periods=12, freq='ME')
line_values = np.random.randint(100, 200, size=12)

# Data for Pie Chart
pie_labels = ['Category A', 'Category B', 'Category C', 'Category D']
pie_sizes = [15, 30, 45, 10]

# Data for Scatter Plot
np.random.seed(0)
x = np.random.randn(100)
y = x * 2 + np.random.randn(100)

# Data for Area Chart
area_values = np.cumsum(np.random.randn(100) + 10)

# Data for Bubble Chart
bubble_x = np.random.rand(50) * 100
bubble_y = np.random.rand(50) * 100
bubble_size = np.random.rand(50) * 1000

# Data for Histogram
hist_data = np.random.randn(1000)

# Data for Box Plot
box_data = [np.random.normal(0, std, 100) for std in range(1, 4)]

# Data for Waterfall Chart
waterfall_data = {
    'Metric': ['Sales', 'Returns', 'Marketing', 'Operating Expenses', 'Profit'],
    'Amount': [100000, -5000, -15000, -30000, 50000]
}
waterfall_df = pd.DataFrame(waterfall_data)
```

```
C:\Users\AlexanderMolinar\AppData\Local\Temp\ipykernel_47840\1862676425.py:8:
FutureWarning: 'M' is deprecated and will be removed in a future version, please
use 'ME' instead.
  dates = pd.date_range(start='2021-01-01', periods=12, freq='M')
```

1.3 1. Column Chart

Column charts are useful for comparing data across categories.

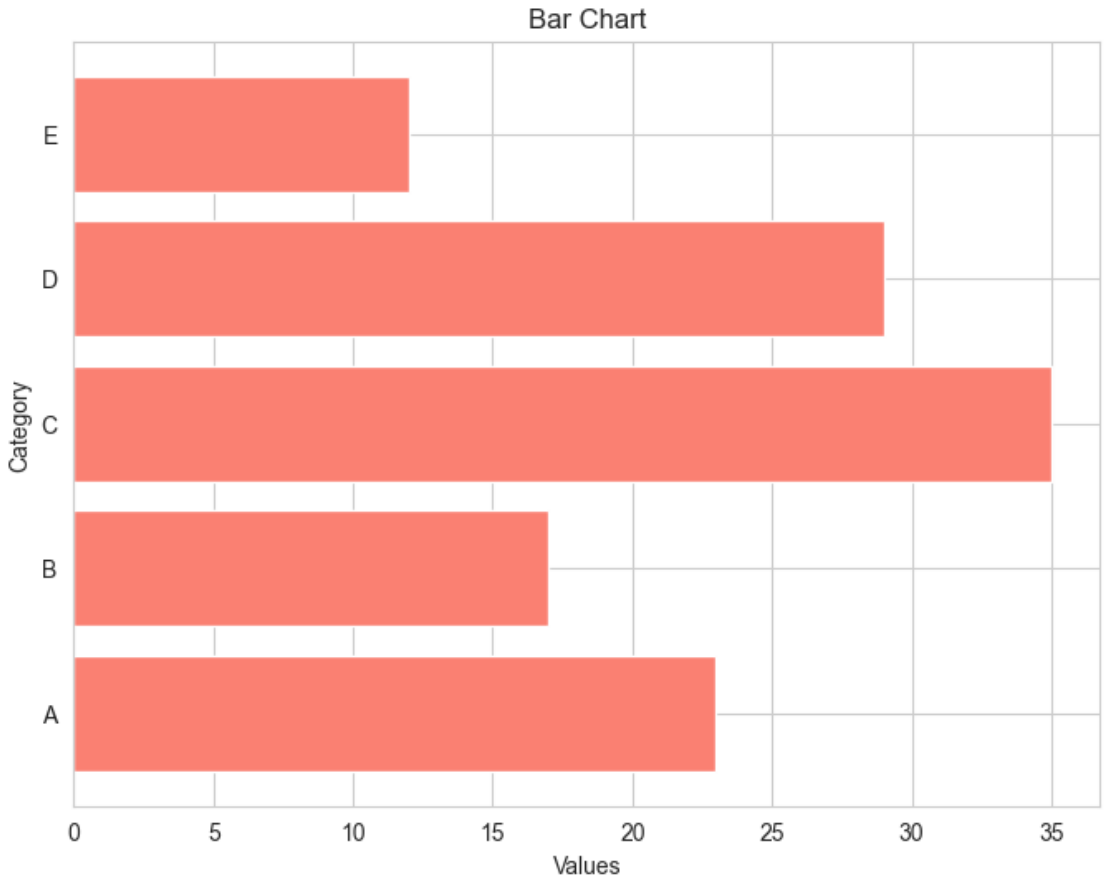
[3]:
```
# Column Chart using Matplotlib
plt.figure(figsize=(8,6))
plt.bar(categories, values, color='skyblue')
plt.title('Column Chart')
plt.xlabel('Category')
plt.ylabel('Values')
plt.show()
```



1.4 2. Bar Chart

Bar charts are similar to column charts but display data horizontally.

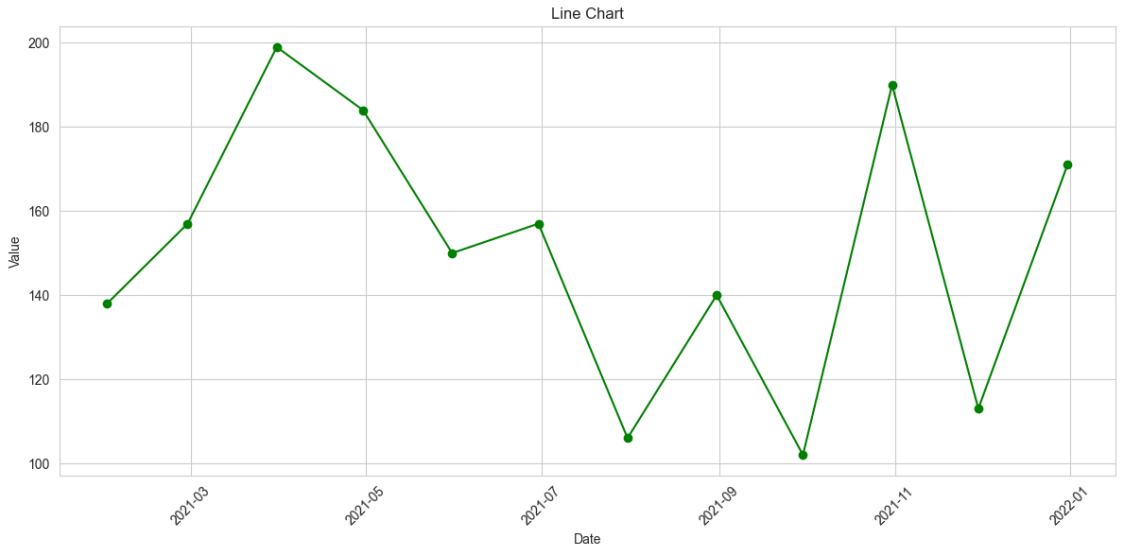
[4]:
```
# Bar Chart using Matplotlib
plt.figure(figsize=(8,6))
plt.barh(categories, values, color='salmon')
plt.title('Bar Chart')
plt.xlabel('Values')
plt.ylabel('Category')
plt.show()
```



1.5 3. Line Chart

Line charts are great for showing trends over time.

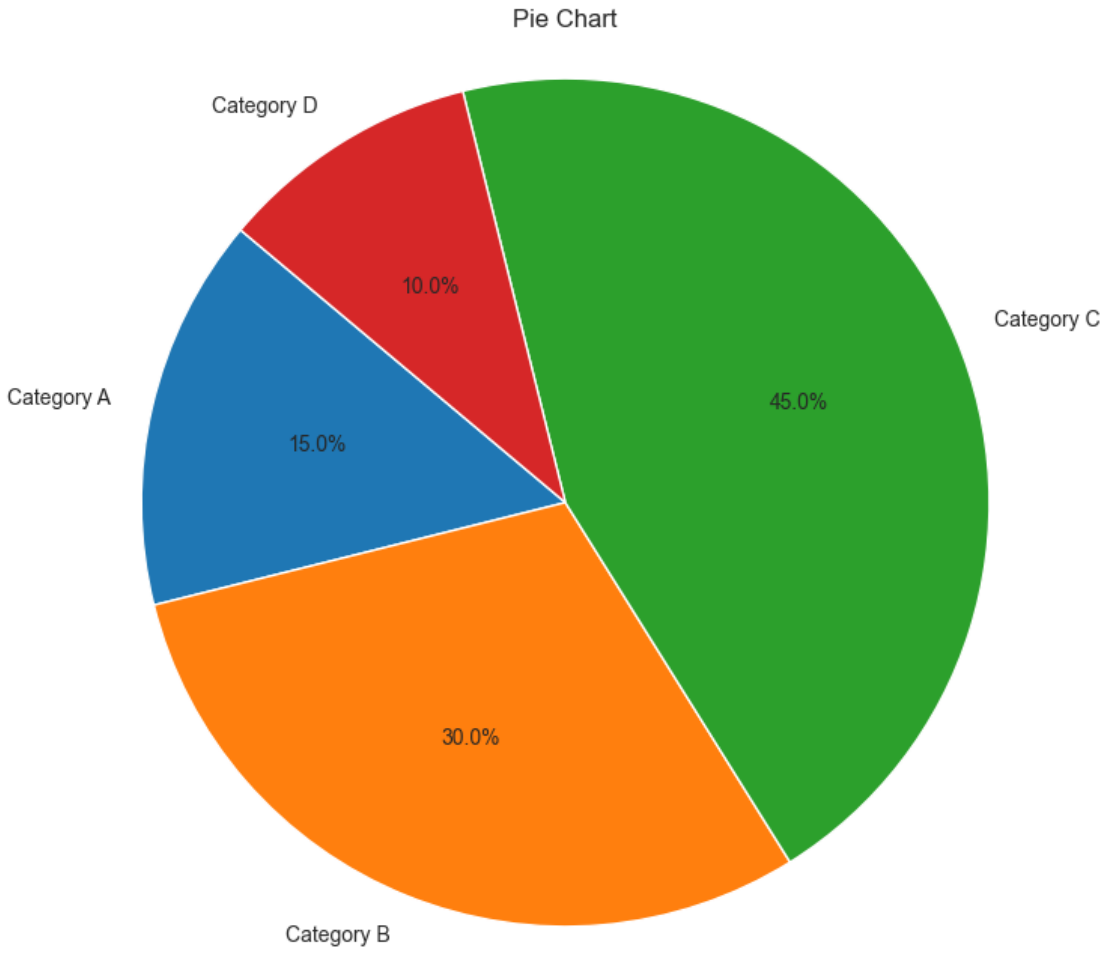
[5]:
```
# Line Chart using Matplotlib
plt.figure(figsize=(12,6))
plt.plot(dates, line_values, marker='o', linestyle='-', color='green')
plt.title('Line Chart')
plt.xlabel('Date')
plt.ylabel('Value')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



1.6 4. Pie Chart

Pie charts show proportions of a whole.

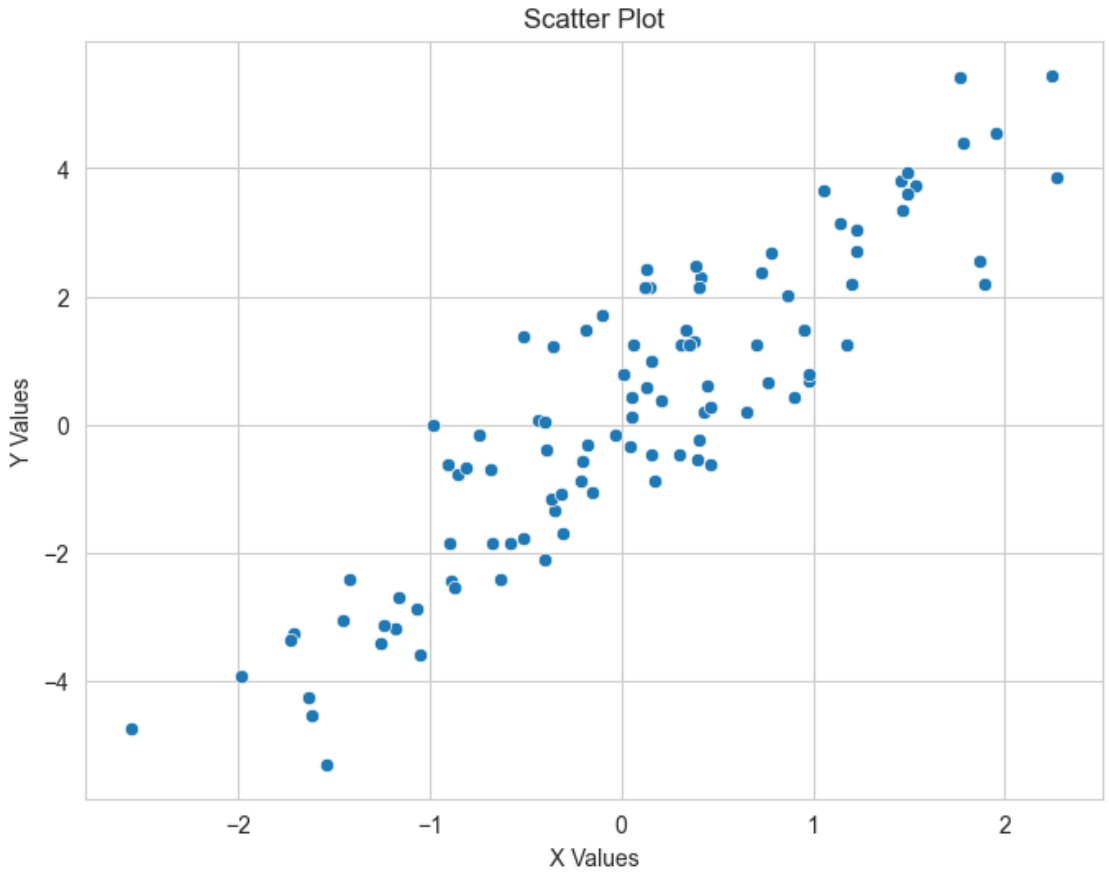
[6]:
```
# Pie Chart using Matplotlib
plt.figure(figsize=(8,8))
plt.pie(pie_sizes, labels=pie_labels, autopct='%1.1f%%', startangle=140)
plt.title('Pie Chart')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



1.7 5. Scatter Plot

Scatter plots display values for typically two variables for a set of data.

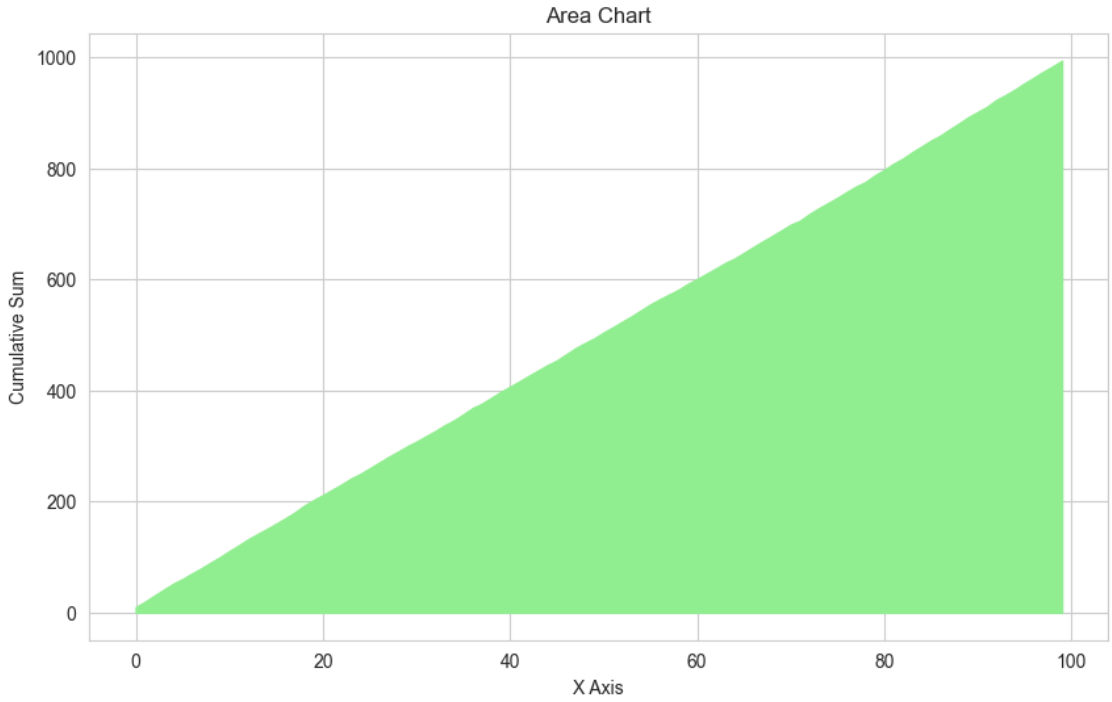
[7]:
```
# Scatter Plot using Seaborn
plt.figure(figsize=(8,6))
sns.scatterplot(x=x, y=y)
plt.title('Scatter Plot')
plt.xlabel('X Values')
plt.ylabel('Y Values')
plt.show()
```



1.8 6. Area Chart

Area charts are used to represent cumulated totals over time.

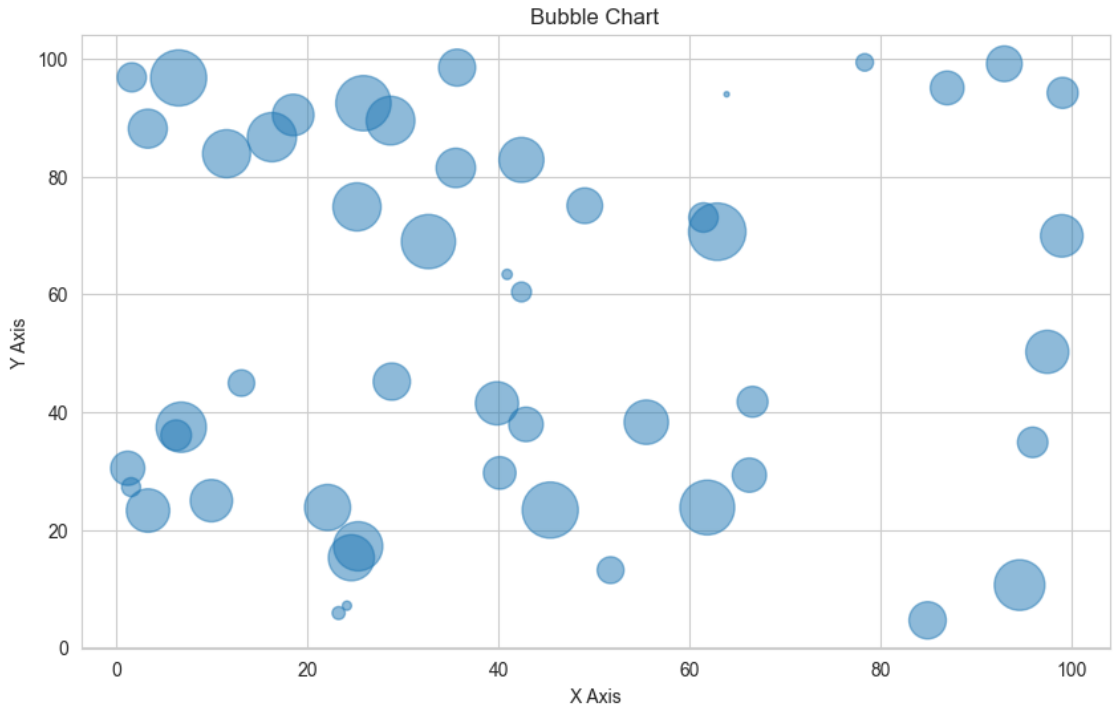
[8]:
```
# Area Chart using Matplotlib
plt.figure(figsize=(10,6))
plt.fill_between(range(len(area_values)), area_values, color='lightgreen')
plt.title('Area Chart')
plt.xlabel('X Axis')
plt.ylabel('Cumulative Sum')
plt.show()
```



1.9 7. Bubble Chart

Bubble charts are a variation of scatter plots with an additional dimension represented by the size of the bubbles.

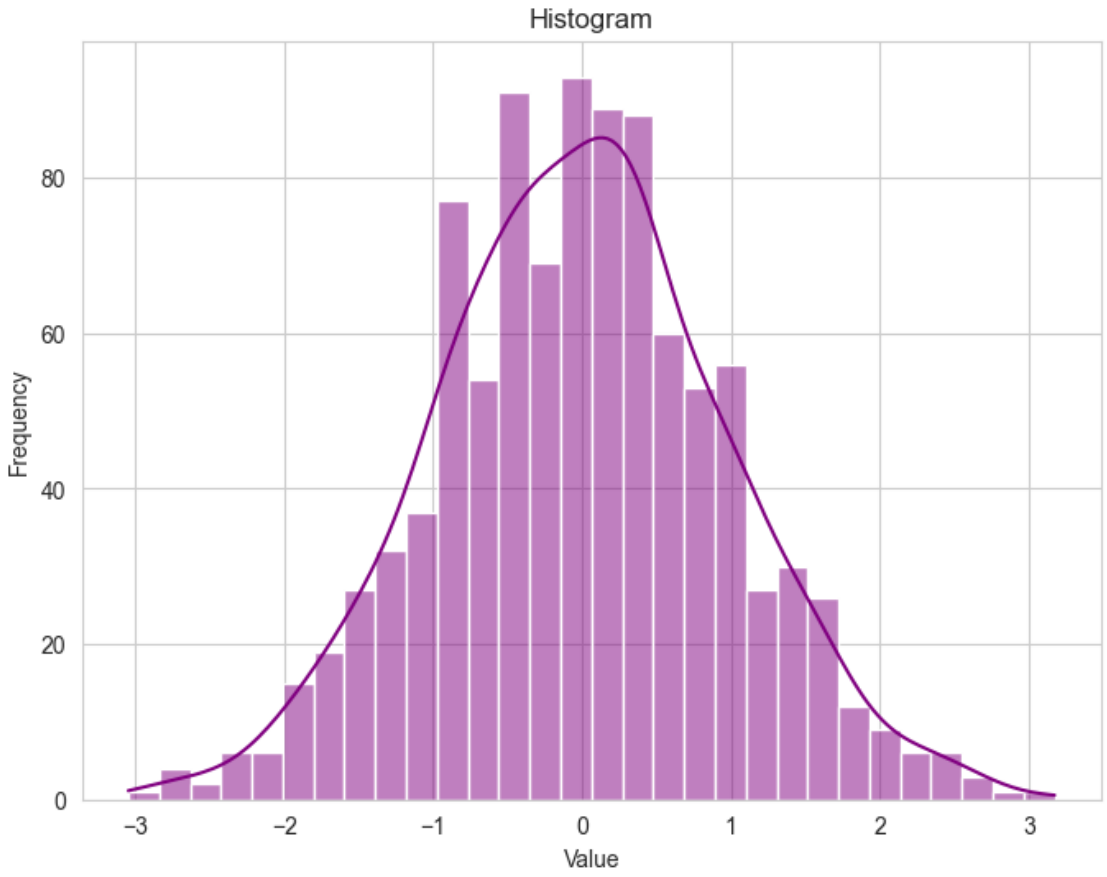
[9]:
```
# Bubble Chart using Matplotlib
plt.figure(figsize=(10,6))
plt.scatter(bubble_x, bubble_y, s=bubble_size, alpha=0.5)
plt.title('Bubble Chart')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.show()
```



1.10 8. Histogram

Histograms are used to represent the distribution of numerical data.

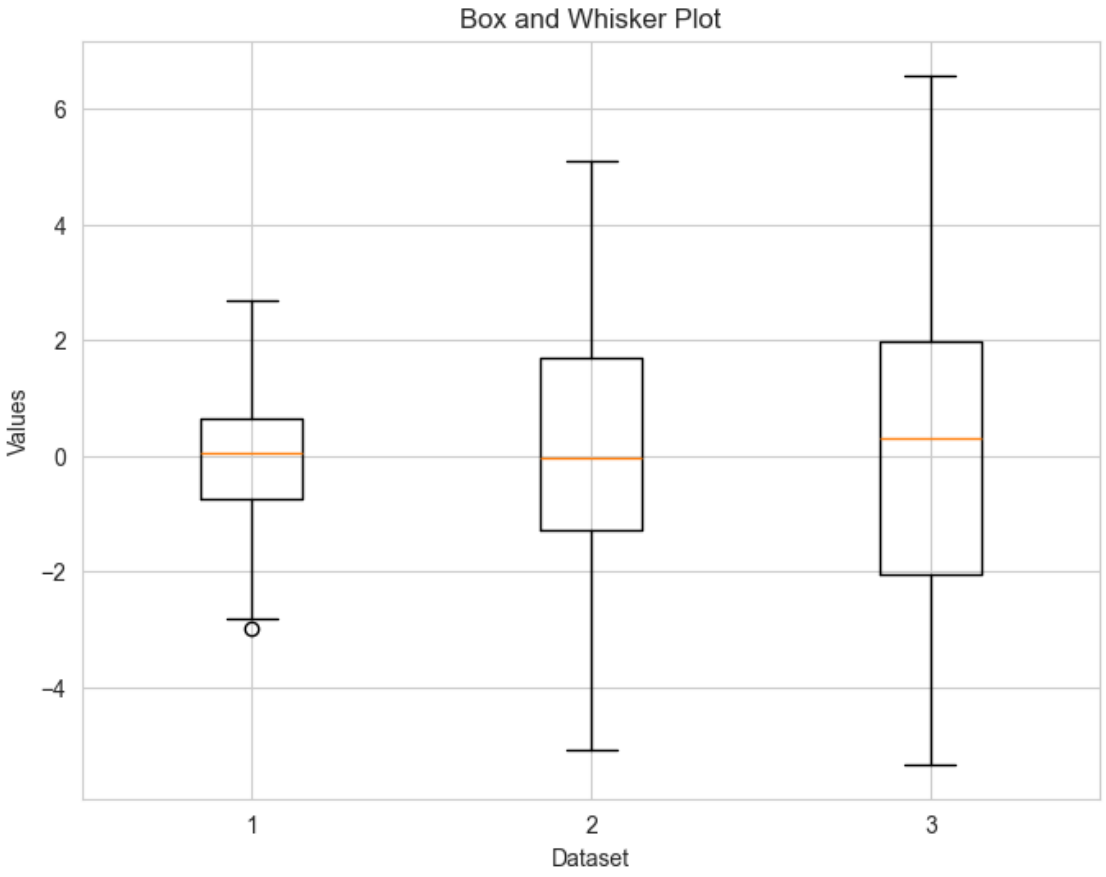
[10]:
```
# Histogram using Seaborn
plt.figure(figsize=(8,6))
sns.histplot(hist_data, bins=30, kde=True, color='purple')
plt.title('Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



1.11 9. Box and Whisker Plot

Box plots are used to show the distribution of quantitative data.

[11]:
```
# Box Plot using Matplotlib
plt.figure(figsize=(8,6))
plt.boxplot(box_data)
plt.title('Box and Whisker Plot')
plt.xlabel('Dataset')
plt.ylabel('Values')
plt.show()
```



1.12 10. Waterfall Chart

Waterfall charts are used to show how an initial value is affected by intermediate positive or negative values.

[12]:
```
import plotly.graph_objects as go

fig = go.Figure()

fig.add_trace(go.Waterfall(
    x = [["2016", "2017", "2017", "2017", "2017", "2018", "2018", "2018",␣
 ↪"2018"],
         ["initial", "q1", "q2", "q3", "total", "q1", "q2", "q3", "total"]],
    measure = ["absolute", "relative", "relative", "relative", "total",␣
 ↪"relative", "relative", "relative", "total"],
    y = [1, 2, 3, -1, None, 1, 2, -4, None],
    base = 1000
))
```

```
fig.update_layout(
    waterfallgroupgap = 0.5,
)

fig.show()
```

1.13 Conclusion

We've demonstrated how to replicate various Excel charts using Python libraries. Python offers powerful tools for data visualization, and with libraries like `matplotlib`, `seaborn`, and `plotly`, you can create interactive and highly customizable charts.