Module 5: Data Visualisation

Case Study I

1. You are given a dataset, Hurricanes.csv, containing the number of hurricanes occurring in the United States along the coast of the Atlantic. Load the data from the dataset into your program and plot a Bar Graph of the data, taking the Year as the x-axis and the number of hurricanes occurring as the Y-axis.

2. The dataset given, records data of city temperatures over the years 2014 and 2015. Plot the histogram of the temperatures over this period for the cities of San Francisco and Moscow.

3. Plot a pie-chart of the number of models released by every manufacturer, recorded in the data provide. Also mention the name of the manufacture with the largest releases.

4. Create csv file from the data below and read in pandas data frame

Phase 1 -Reading Data
Phase 2 –Describe the data Describe the data on the unit price
Phase 3 –filter the data
Create new dataframe having columns 'name','net_price','date' and group all the records according to name
Phase 4 –Plotting graph

Plot the graph after calculating total sales by each customer. Customer name should be on x axis and total sales in y axis.

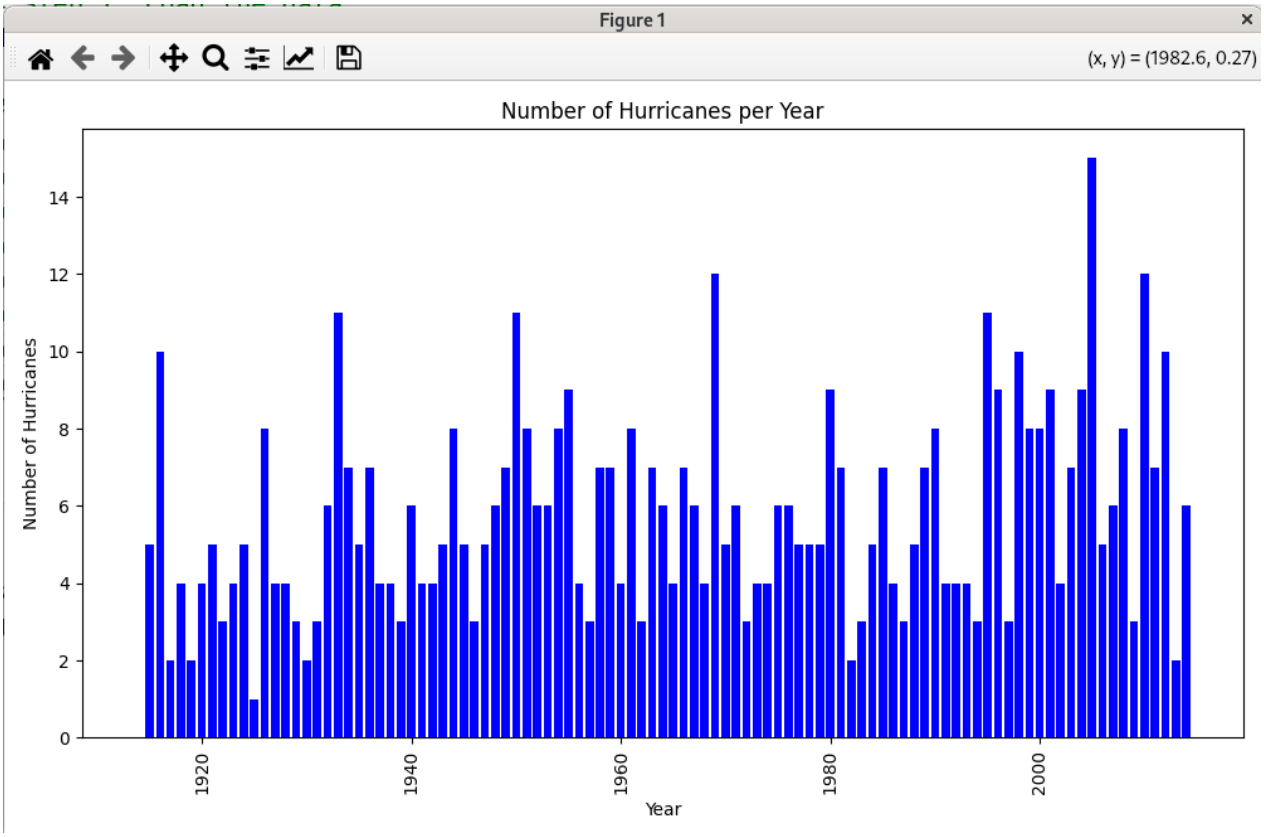
5. Let the x axis data points and y axis data points are
X = [1,2,3,4]
y = [20, 21, 20.5, 20.8]
5.1: Draw a Simple plot
5.2: Configure the line and markers in simple plot
5.3: configure the axes
5.4: Give title of Graph & labels of x axis and y axis
5.5: Give error bar if y_error = [0.12, 0.13, 0.2, 0.1]
5.6: define width, height as figsize=(4,5) DPI and adjust plot dpi=100
5.7: Give a font size of 14
5.8: Draw a scatter graph of any 50 random values of x and y axis
5.9: Create a dataframe from following data
'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
'last_name': ['Miller', 'Jacobson', 'Ali', 'Milner', 'Cooze'],
'female': [0, 1, 1, 0, 1],
'age': [42, 52, 36, 24, 73],
'preTestScore': [4, 24, 31, 2, 3],
'postTestScore': [25, 94, 57, 62, 70]

Draw a Scatterplot of preTestScore and postTestScore, with the size of each point determined by age

5.10: Draw a Scatterplot from the data in question 9 of preTestScore and postTestScore with the size = 300 and the color determined by sex

1

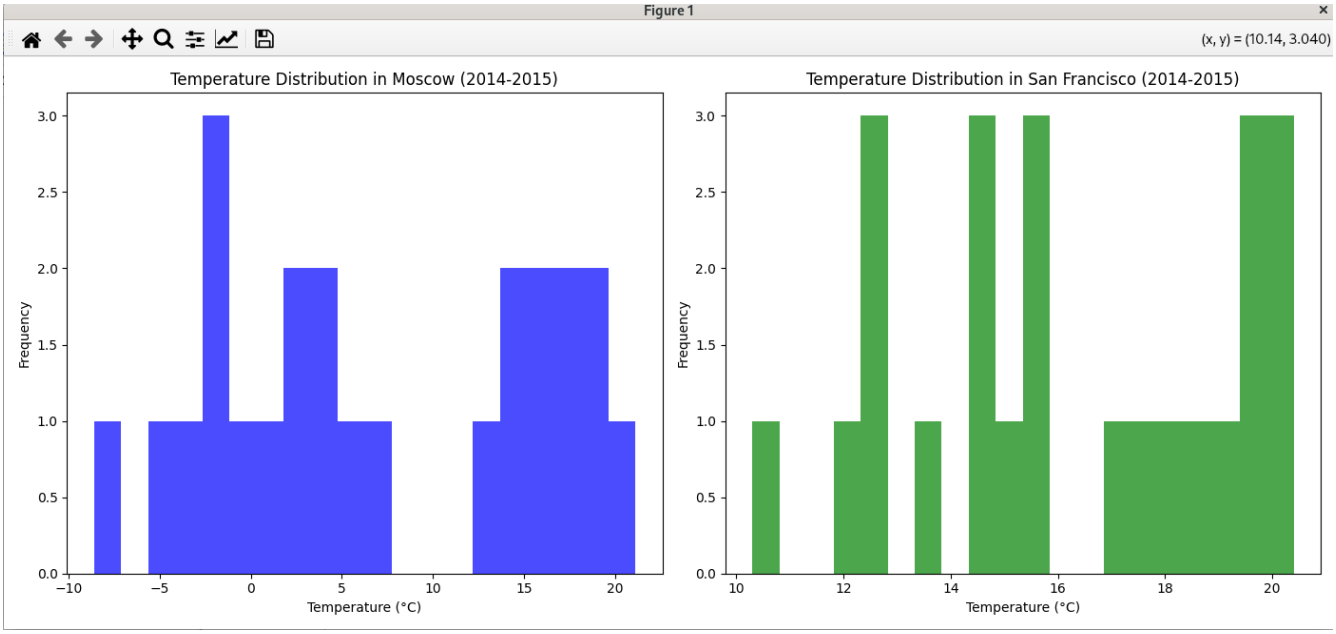
```
import pandas as pd
import matplotlib.pyplot as plt

# Step 1: Load the data
data = pd.read_csv('Hurricanes.csv')

# Step 2: Plot the data
plt.figure(figsize=(10, 6))  # Set the figure size for better readability
plt.bar(data['Year'], data['Hurricanes'], color='blue')  # Create a bar graph
plt.title('Number of Hurricanes per Year')  # Title of the graph
plt.xlabel('Year')  # Label for the x-axis
plt.ylabel('Number of Hurricanes')  # Label for the y-axis
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.tight_layout()  # Adjust layout to not cut off labels
plt.show()  # Display the plot
```



2

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

# Load the data
data = pd.read_csv('CityTemps.csv')

# Plot the histograms
plt.figure(figsize=(14, 6))

# Histogram for Moscow
plt.subplot(1, 2, 1)  # 1 row, 2 columns, 1st subplot
plt.hist(data['Moscow'], bins=20, color='blue', alpha=0.7)
plt.title('Temperature Distribution in Moscow (2014-2015)')
plt.xlabel('Temperature (°C)')
plt.ylabel('Frequency')

# Histogram for San Francisco
```

```
plt.subplot(1, 2, 2)  # 1 row, 2 columns, 2nd subplot
plt.hist(data['San Francisco'], bins=20, color='green', alpha=0.7)
plt.title('Temperature Distribution in San Francisco (2014-2015)')
plt.xlabel('Temperature (°C)')
plt.ylabel('Frequency')

plt.tight_layout()  # Adjust layout to not cut off labels
plt.show()  # Display the plots
```

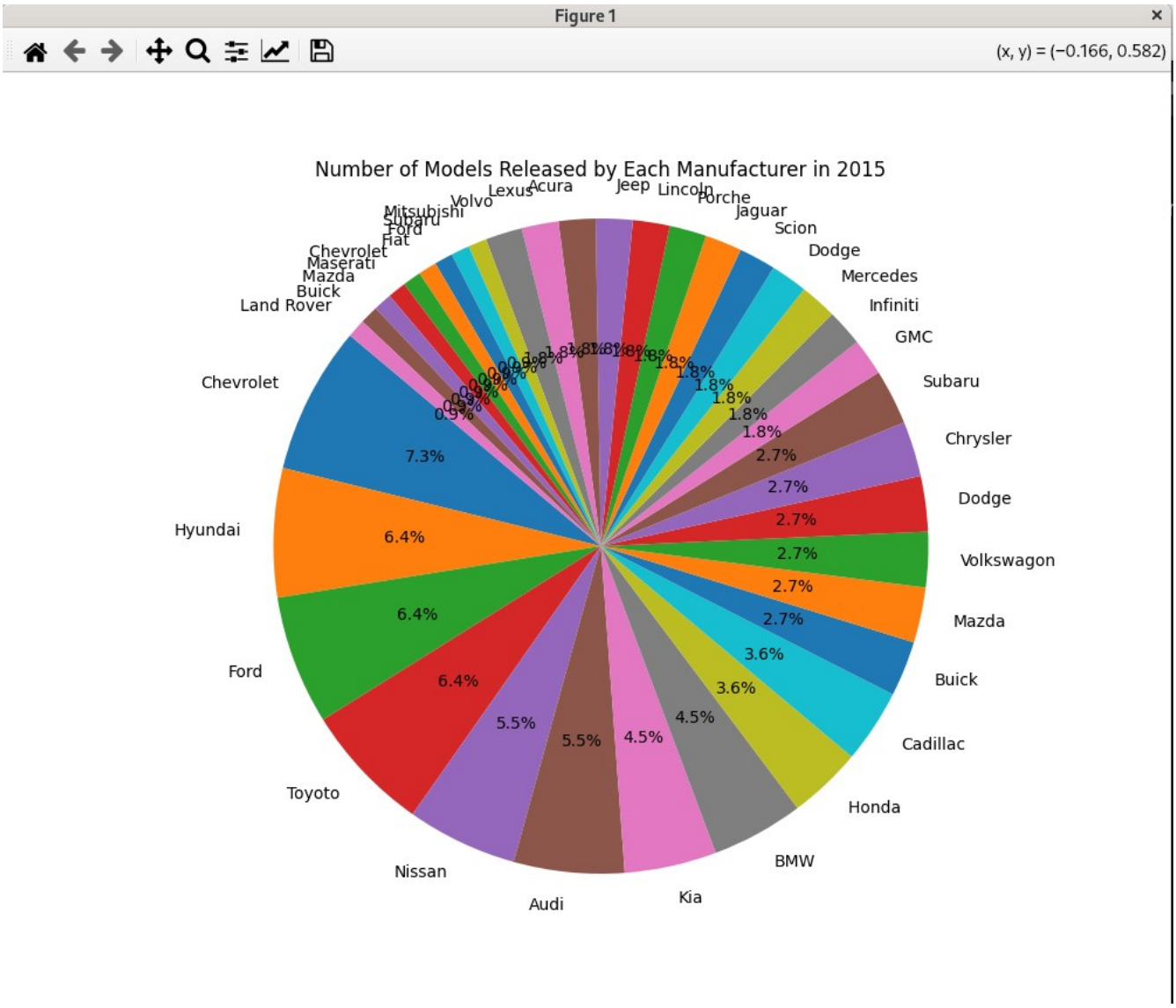


3

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

# Assuming the CSV data is loaded into a DataFrame
data = pd.read_csv('Cars2015.csv')

# Count the number of models for each manufacturer
model_counts = data['Make'].value_counts()

largest_releases_manufacturer = model_counts.idxmax()
print(f"The manufacturer with the largest releases is {largest_releases_manufacturer} with
{model_counts.max()} models.")

# Plot the pie chart
```

```
plt.figure(figsize=(10, 8))
plt.pie(model_counts, labels=model_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Number of Models Released by Each Manufacturer in 2015')
plt.axis('equal')  # Equal aspect ratio ensures the pie chart is circular
plt.show()
```

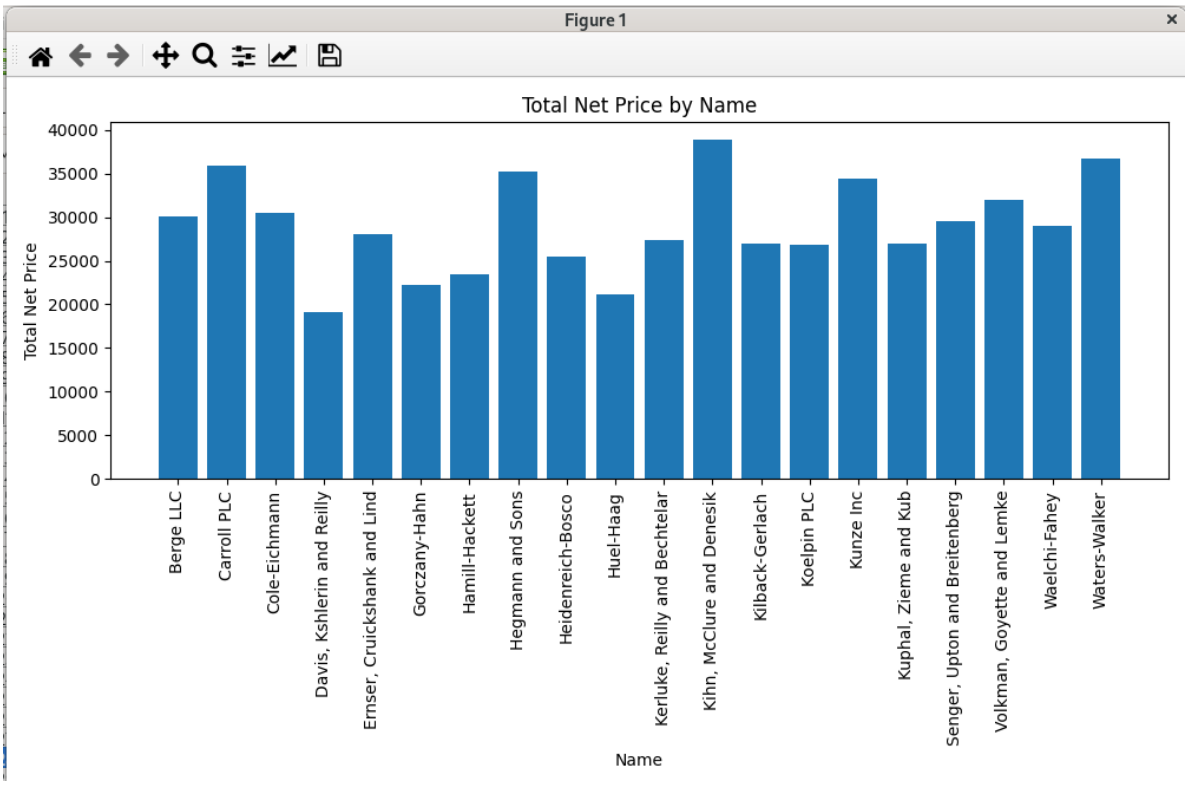
```
[john@squid use-cases_I-II]$
[john@squid use-cases_I-II]$ python3 case-study_I_3.py
The manufacturer with the largest releases is Chevrolet with 8 models.
```



4

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

# Assuming 'sample-salesv2.csv' is the file name
data = pd.read_csv('sample-salesv2.csv')

# Phase 2: Describe the data on the unit price
unit_price_description = data['unit price'].describe()
print(unit_price_description)

# Phase 3: Filter the data and create a new DataFrame
filtered_data = data[['name', 'net_price', 'date']]
grouped_data = filtered_data.groupby('name').sum().reset_index()
print(grouped_data)

# Phase 4: Plotting Graph
plt.figure(figsize=(10, 6))
plt.bar(grouped_data['name'], grouped_data['net_price'])
plt.xlabel('Name')
plt.ylabel('Total Net Price')
plt.title('Total Net Price by Name')
plt.xticks(rotation=90)

plt.tight_layout()  # Adjust layout to not cut off labels
plt.show()  # Display the plots
```

```
[john@squid use-cases_I-II]$ python3 case-study_I_4.py
count    1000.000000
mean       56.179630
std        25.331939
min        10.060000
25%        35.995000
50%        56.765000
75%        76.802500
max        99.970000
Name: unit price, dtype: float64
                            name  net_price                                                 date
0                     Berge LLC   30064.87  04-11-2013 09:4804-08-2014 23:0822-11-2013 17:...
1                   Carroll PLC   35934.31  27-09-2014 07:1324-05-2014 16:0312-01-2014 00:...
2                 Cole-Eichmann   30435.42  09-04-2014 16:1510-03-2014 06:2317-11-2013 23:...
3     Davis, Kshlerin and Reilly   19054.76  20-09-2014 09:4905-03-2014 09:2622-02-2014 13:...
4    Ernser, Cruickshank and Lind   28089.02  19-09-2014 13:2017-09-2014 19:1505-01-2014 10:...
5                 Gorczany-Hahn   22207.90  07-02-2014 03:3020-02-2014 07:4210-12-2013 20:...
6                Hamill-Hackett   23433.78  14-11-2013 22:1602-01-2014 12:4908-08-2014 13:...
7              Hegmann and Sons   35213.72  18-06-2014 19:2503-11-2013 18:3829-05-2014 05:...
8             Heidenreich-Bosco   25428.29  29-07-2014 02:1012-02-2014 07:2730-11-2013 23:...
9                   Huel-Haag   21087.88  23-09-2014 02:3615-03-2014 12:5005-06-2014 22:...
10  Kerluke, Reilly and Bechtelar   27389.43  01-03-2014 10:5104-12-2013 02:0720-05-2014 00:...
11     Kihn, McClure and Denesik   38935.29  11-01-2014 21:4813-11-2013 21:3821-03-2014 14:...
12               Kilback-Gerlach   26987.20  26-03-2014 20:5614-04-2014 16:5829-05-2014 22:...
13                  Koelpin PLC   26811.66  12-08-2014 08:0526-01-2014 01:5212-12-2013 02:...
14                   Kunze Inc   34406.54  19-02-2014 06:0320-02-2014 13:1821-04-2014 02:...
15          Kuphal, Zieme and Kub   27031.86  20-01-2014 20:3402-07-2014 08:3512-08-2014 22:...
16  Senger, Upton and Breitenberg   29577.46  10-02-2014 05:5528-04-2014 07:0109-07-2014 19:...
17    Volkman, Goyette and Lemke   32006.87  08-01-2014 02:4513-12-2013 03:1902-10-2014 04:...
18                 Waelchi-Fahey   28968.68  03-01-2014 08:1415-07-2014 21:0912-09-2014 21:...
19                 Waters-Walker   36778.96  17-11-2013 20:4115-07-2014 23:2104-09-2014 13:...
```



5

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

# Data points for X and y
X = [1, 2, 3, 4]
y = [20, 21, 20.5, 20.8]
y_error = [0.12, 0.13, 0.2, 0.1]

# 5.1: Draw a Simple plot
plt.figure(figsize=(4, 5), dpi=100)
plt.plot(X, y)
plt.show()

# 5.2: Configure the line and markers in simple plot
plt.figure(figsize=(4, 5), dpi=100)
plt.plot(X, y, linestyle='--', marker='o', color='b')
plt.show()

# 5.3: Configure the axes
plt.figure(figsize=(4, 5), dpi=100)
plt.plot(X, y, linestyle='--', marker='o', color='b')
plt.xlim(0, 5)
plt.ylim(19, 22)
plt.show()

# 5.4: Give title of Graph & labels of x axis and y axis
plt.figure(figsize=(4, 5), dpi=100)
plt.plot(X, y, linestyle='--', marker='o', color='b')
plt.xlim(0, 5)
plt.ylim(19, 22)
plt.title('Sample Plot', fontsize=14)
plt.xlabel('X Axis', fontsize=14)
plt.ylabel('Y Axis', fontsize=14)
plt.show()

# 5.5: Give error bar
plt.figure(figsize=(4, 5), dpi=100)
plt.errorbar(X, y, yerr=y_error, linestyle='--', marker='o', color='b')
plt.xlim(0, 5)
plt.ylim(19, 22)
plt.title('Sample Plot with Error Bars', fontsize=14)
plt.xlabel('X Axis', fontsize=14)
plt.ylabel('Y Axis', fontsize=14)
```

```
plt.show()

# 5.6: Define width, height as figsize=(4,5) DPI and adjust plot dpi=100
# (already applied in previous plots)

# 5.7: Give a font size of 14
# (already applied in previous plots)

# 5.8: Draw a scatter graph of any 50 random values of x and y axis
random_x = np.random.rand(50)
random_y = np.random.rand(50)
plt.figure(figsize=(4, 5), dpi=100)
plt.scatter(random_x, random_y)
plt.title('Random Scatter Plot', fontsize=14)
plt.xlabel('Random X', fontsize=14)
plt.ylabel('Random Y', fontsize=14)
plt.tight_layout()  # Adjust layout to not cut off labels
plt.show()

# 5.9: Create a dataframe and draw a scatterplot
data = {
    'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
    'last_name': ['Miller', 'Jacobson', 'Ali', 'Milner', 'Cooze'],
    'female': [0, 1, 1, 0, 1],
    'age': [42, 52, 36, 24, 73],
    'preTestScore': [4, 24, 31, 2, 3],
    'postTestScore': [25, 94, 57, 62, 70]
}
df = pd.DataFrame(data)

# Scatterplot of preTestScore and postTestScore, with the size of each point determined by age
plt.figure(figsize=(4, 5), dpi=100)
plt.scatter(df['preTestScore'], df['postTestScore'], s=df['age']*10, alpha=0.5)
plt.title('Pre Test Score vs Post Test Score', fontsize=14)
plt.xlabel('Pre Test Score', fontsize=14)
plt.ylabel('Post Test Score', fontsize=14)
plt.tight_layout()  # Adjust layout to not cut off labels
plt.show()

# 5.10: Scatterplot with size = 300 and the color determined by sex
plt.figure(figsize=(4, 5), dpi=100)
plt.scatter(df['preTestScore'], df['postTestScore'], s=300, c=df['female'], alpha=0.5, cmap='bwr')
plt.title('Pre Test Score vs Post Test Score by Gender', fontsize=14)
plt.xlabel('Pre Test Score', fontsize=14)
plt.ylabel('Post Test Score', fontsize=14)
plt.tight_layout()  # Adjust layout to not cut off labels
plt.show()
```

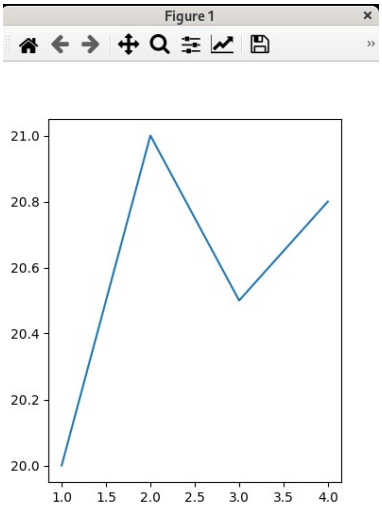


Figure 1
21.0
20.8
20.6
20.4
20.2
20.0
1.0 1.5 2.0 2.5 3.0 3.5 4.0

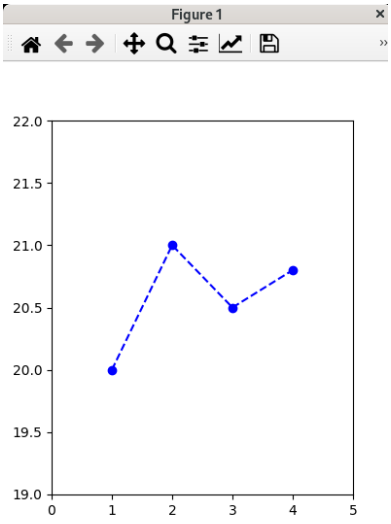


Figure 1
22.0
21.5
21.0
20.5
20.0
19.5
19.0
0 1 2 3 4 5

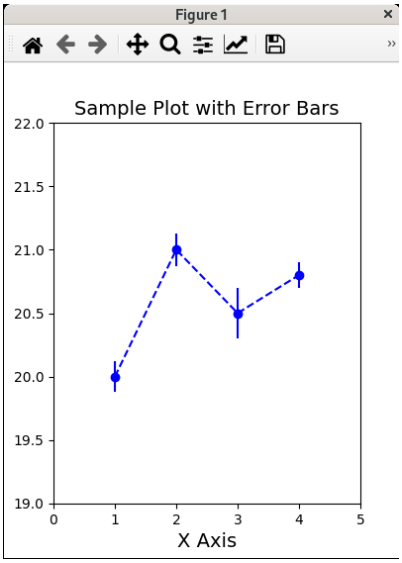


Figure 1
Sample Plot with Error Bars
22.0
21.5
21.0
20.5
20.0
19.5
19.0
0 1 2 3 4 5
X Axis

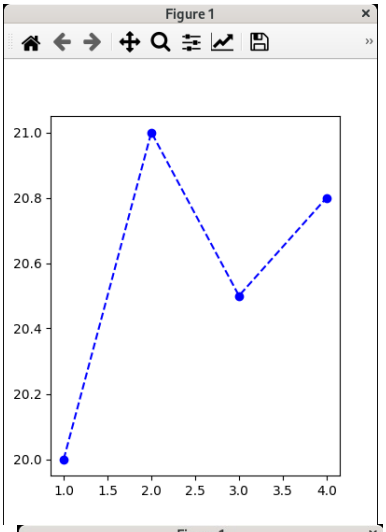


Figure 1
21.0
20.8
20.6
20.4
20.2
20.0
1.0 1.5 2.0 2.5 3.0 3.5 4.0

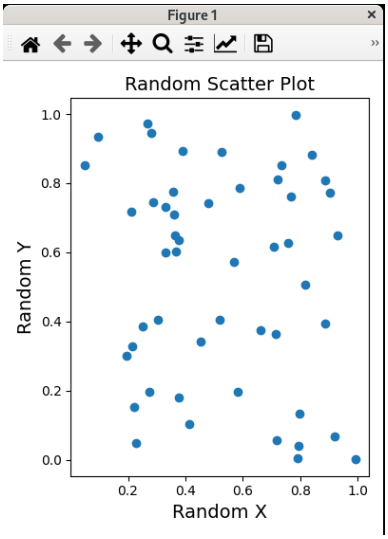


Figure 1
Random Scatter Plot
Random Y
1.0
0.8
0.6
0.4
0.2
0.0
0.2 0.4 0.6 0.8 1.0
Random X

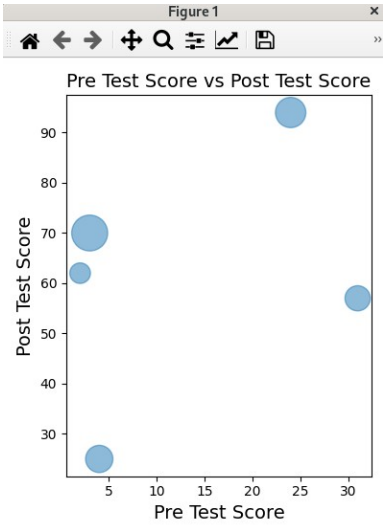


Figure 1
Pre Test Score vs Post Test Score
Post Test Score
90
80
70
60
50
40
30
5 10 15 20 25 30
Pre Test Score

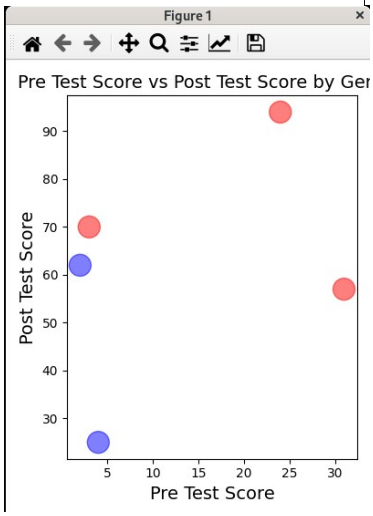


Figure 1
Pre Test Score vs Post Test Score by Ger
Post Test Score
90
80
70
60
50
40
30
5 10 15 20 25 30
Pre Test Score