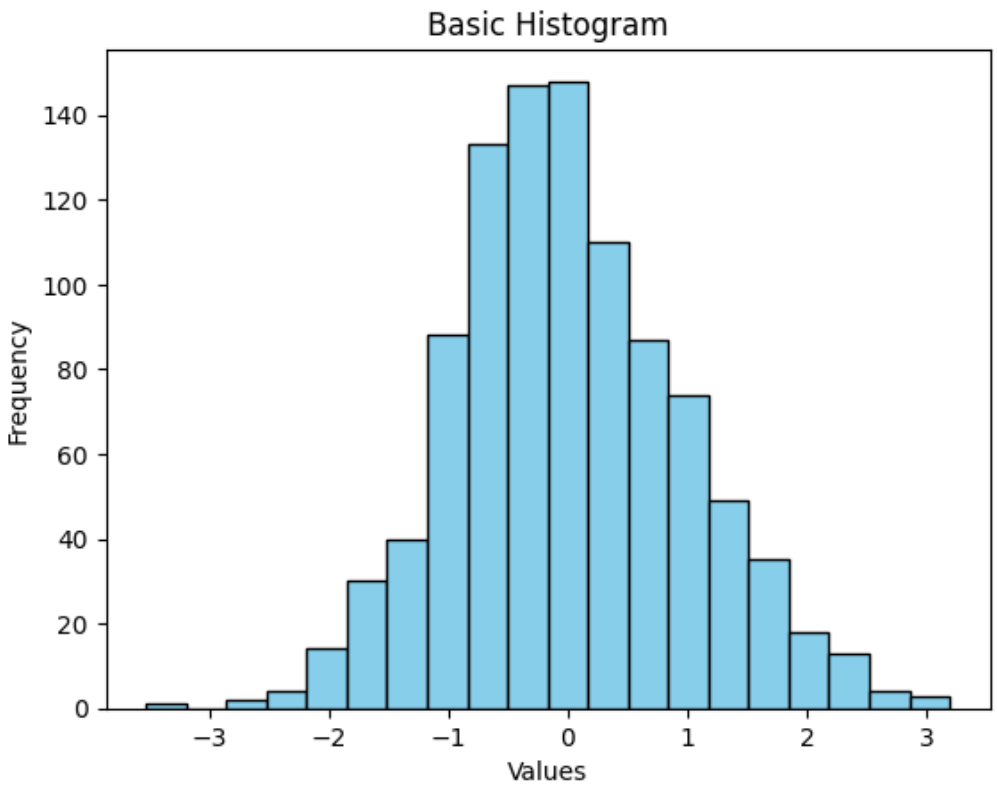
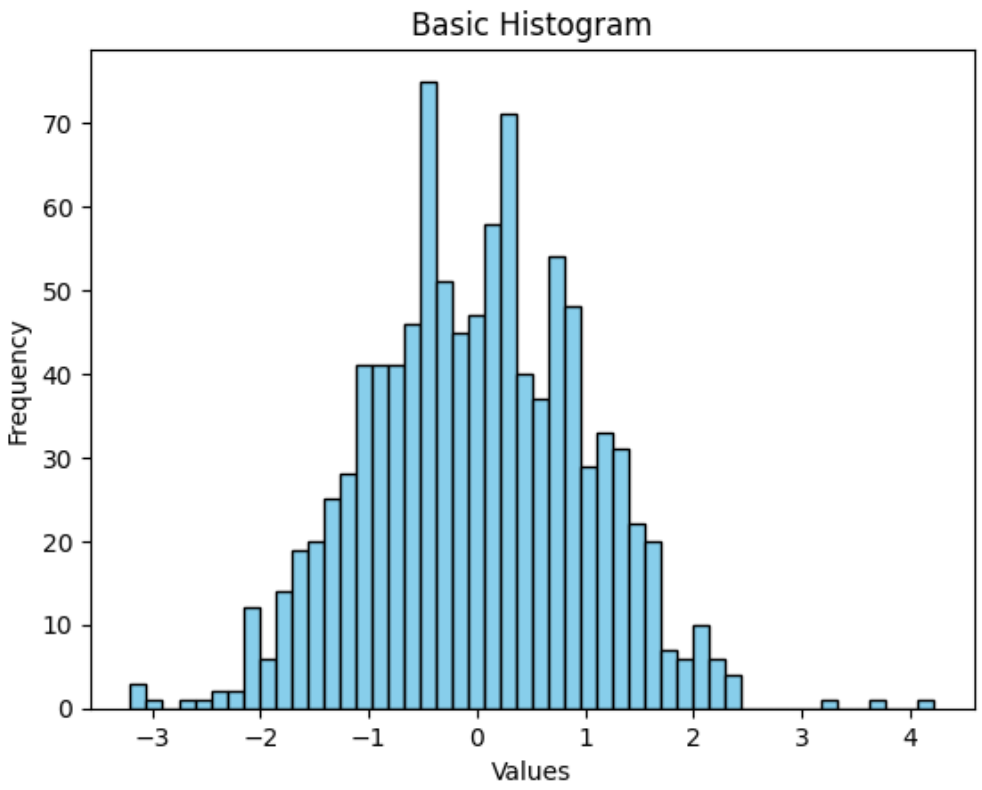
In [23]:
```python
import matplotlib.pyplot as plt
import numpy as np

# Generate random data for the histogram
data = np.random.randn(1000)

# Plotting a basic histogram
plt.hist(data, bins=20, color='skyblue', edgecolor='black')

# Adding labels and title
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Basic Histogram')

# Display the plot
plt.show()
```

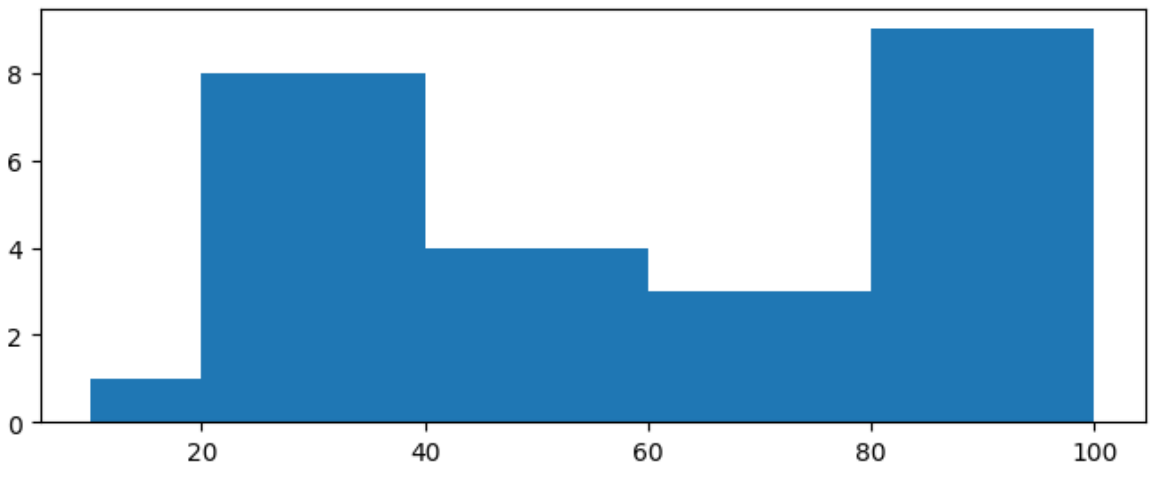


In [24]:
```python
import matplotlib.pyplot as plt
import numpy as np

# Generate random data for the histogram
data = np.random.randn(1000)

# Plotting a basic histogram
plt.hist(data, bins=50, color='skyblue', edgecolor='black')

# Adding labels and title
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Basic Histogram')

# Display the plot
plt.show()
```

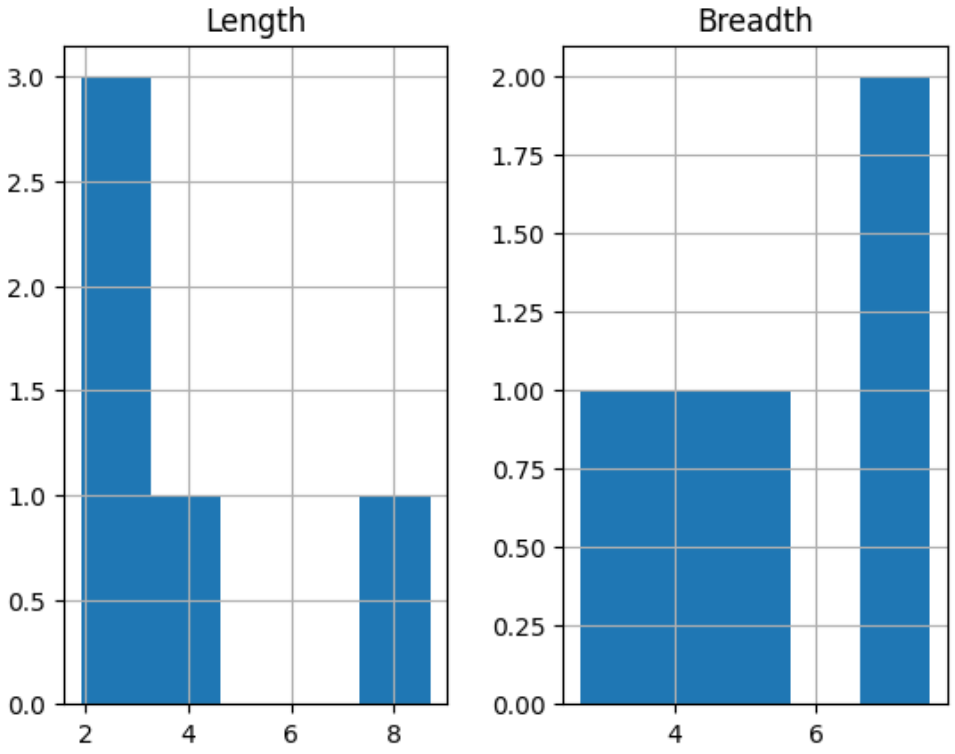


In [25]:
```python
import numpy as np
from matplotlib import pyplot as plt

# Creating a sample dataset
array = np.array([23, 56, 87, 87, 98,
             12, 76, 98, 34, 87,
             67, 23, 87, 56, 34,
             26, 85, 47, 35, 86,
             76, 45, 86, 34, 37])

# Creating a histogram using .hist() function
figure, axis = plt.subplots(figsize = (8, 3))
#axis.hist(array)
axis.hist(array, bins = [10,20, 40, 60, 80, 100])
```

```
# Showing the plot
plt.show()
```

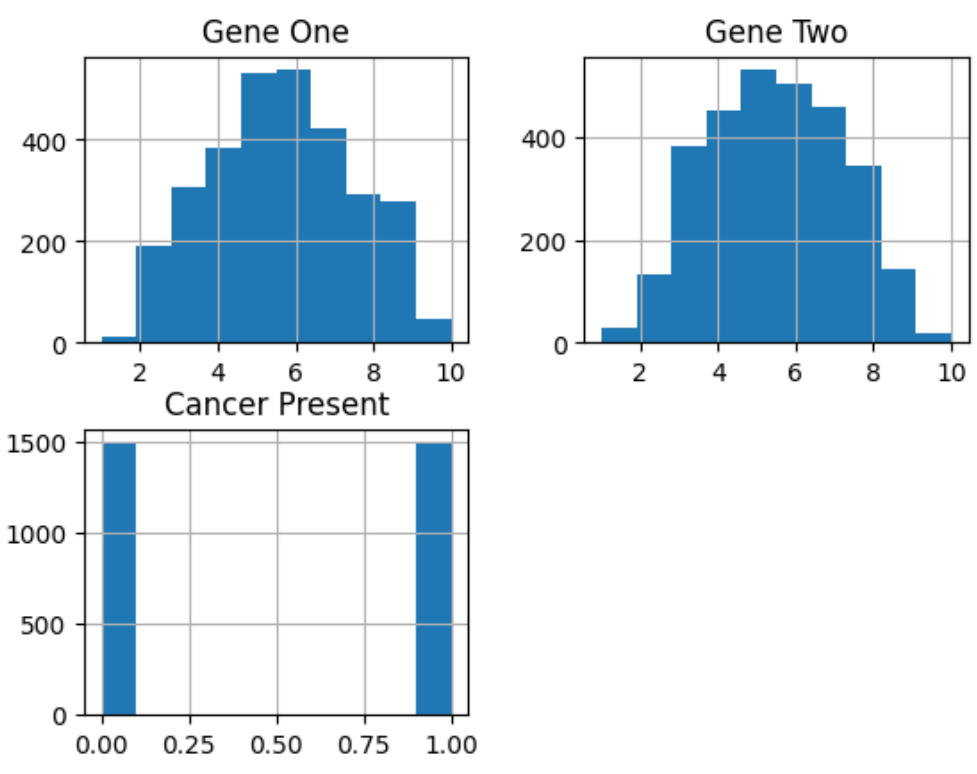


In [26]:
```
# Importing pandas library
import pandas as pd

# Creating a Data frame
values = pd.DataFrame({
    'Length': [2.7, 8.7, 3.4, 2.4, 1.9],
    'Breadth': [4.24, 2.67, 7.6, 7.1, 4.9]
})

# Creating Histograms of columns 'Length'
# and 'Breadth' using Dataframe.hist()
# function
hist = values.hist(bins=5)
```

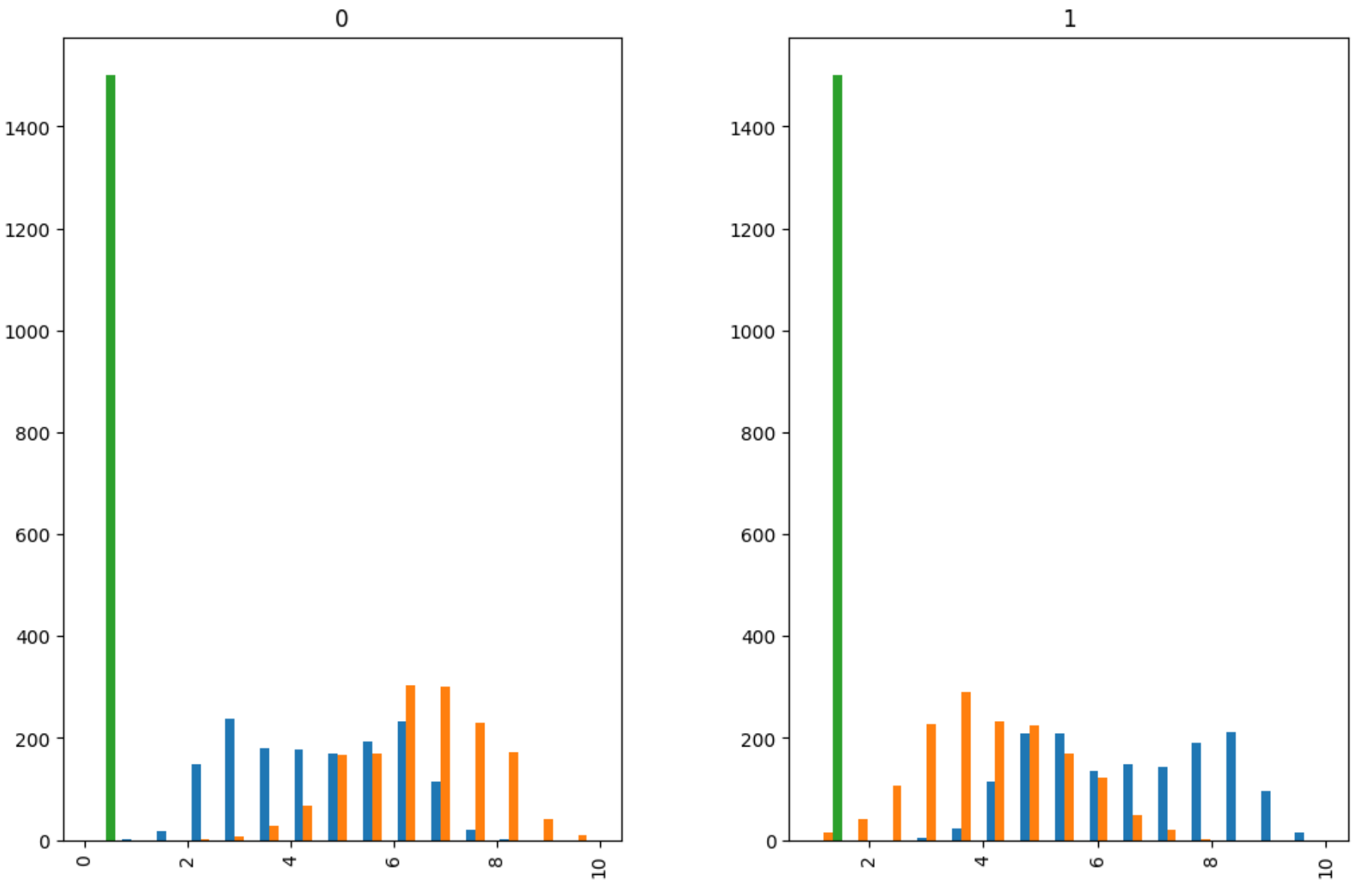


In [27]:
```
# import libraries and packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# reading the CSV file
df = pd.read_csv('gene_expression.csv')

# displaying the DataFrame
print(df)

# creating a basic histogram
df.hist()
plt.show()
```

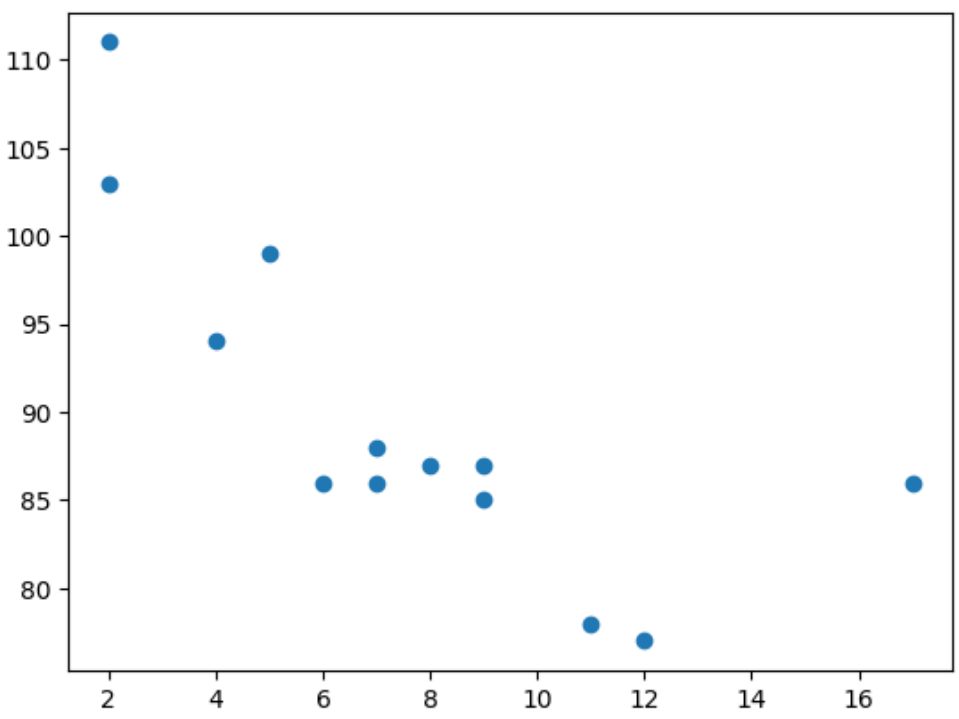
```
      Gene One  Gene Two  Cancer Present
0          4.3       3.9               1
1          2.5       6.3               0
2          5.7       3.9               1
3          6.1       6.2               0
4          7.4       3.4               1
...        ...       ...             ...
2995       5.0       6.5               1
2996       3.4       6.6               0
2997       2.7       6.5               0
2998       3.3       5.6               0
2999       4.6       8.2               0

[3000 rows x 3 columns]
```



In [16]:
```
# creating a basic histogram
df.hist(by='Cancer Present', figsize=[12, 8], bins=15)
plt.show()
```

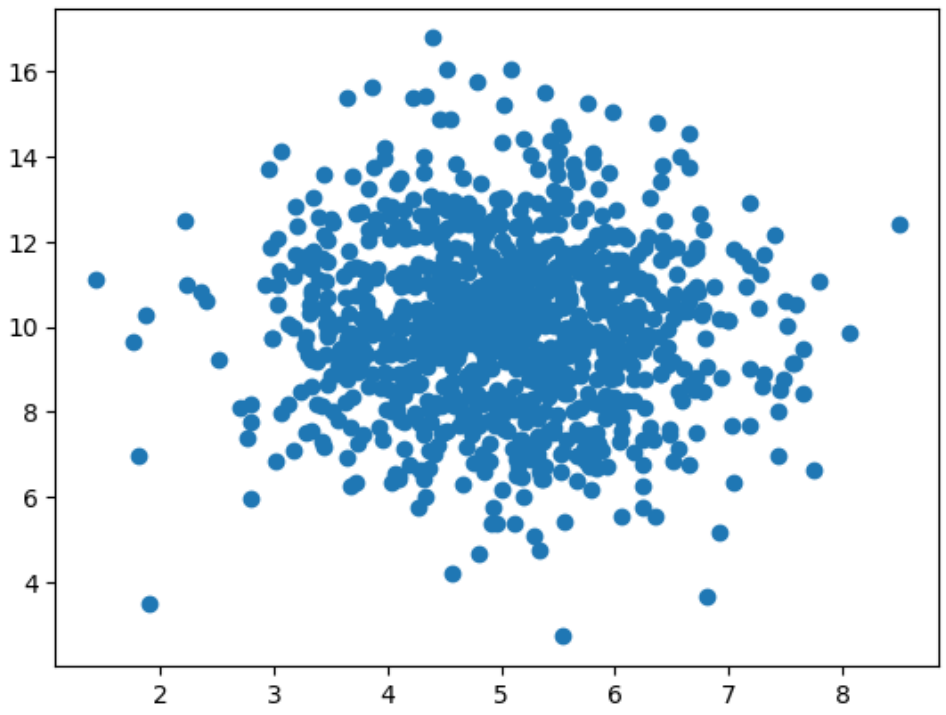


In [5]:
```
x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

plt.scatter(x, y)
plt.show()
```

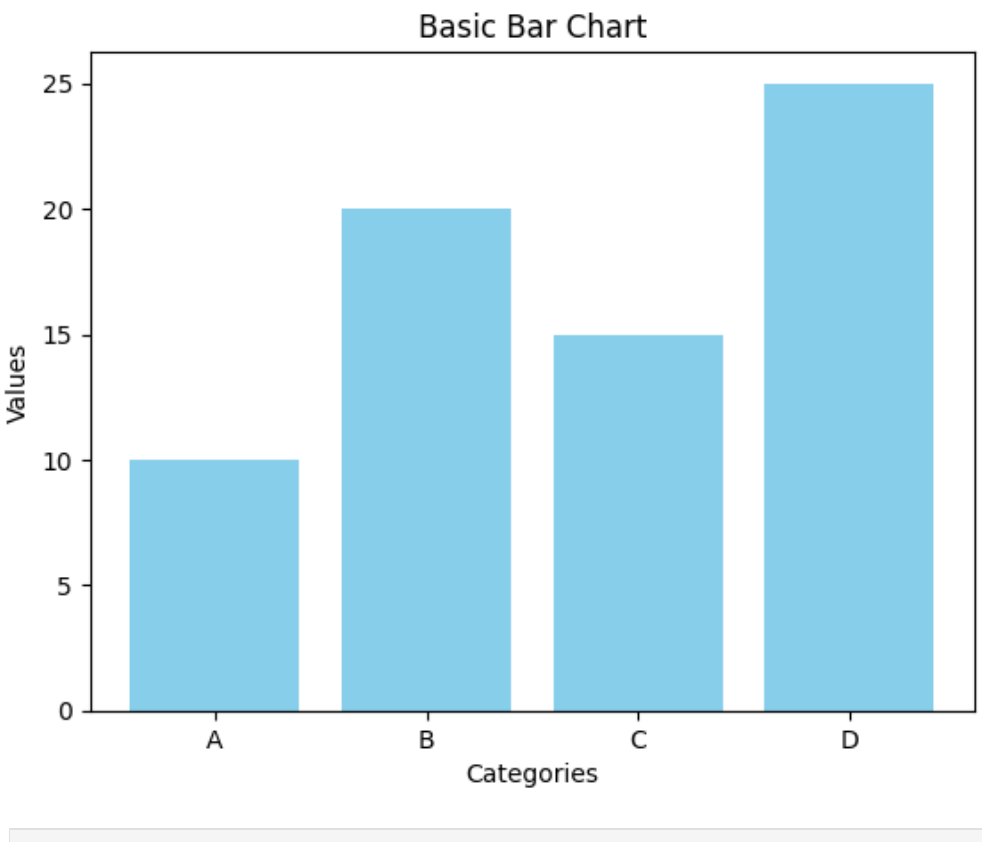


In [6]:
```
import numpy
import matplotlib.pyplot as plt
```

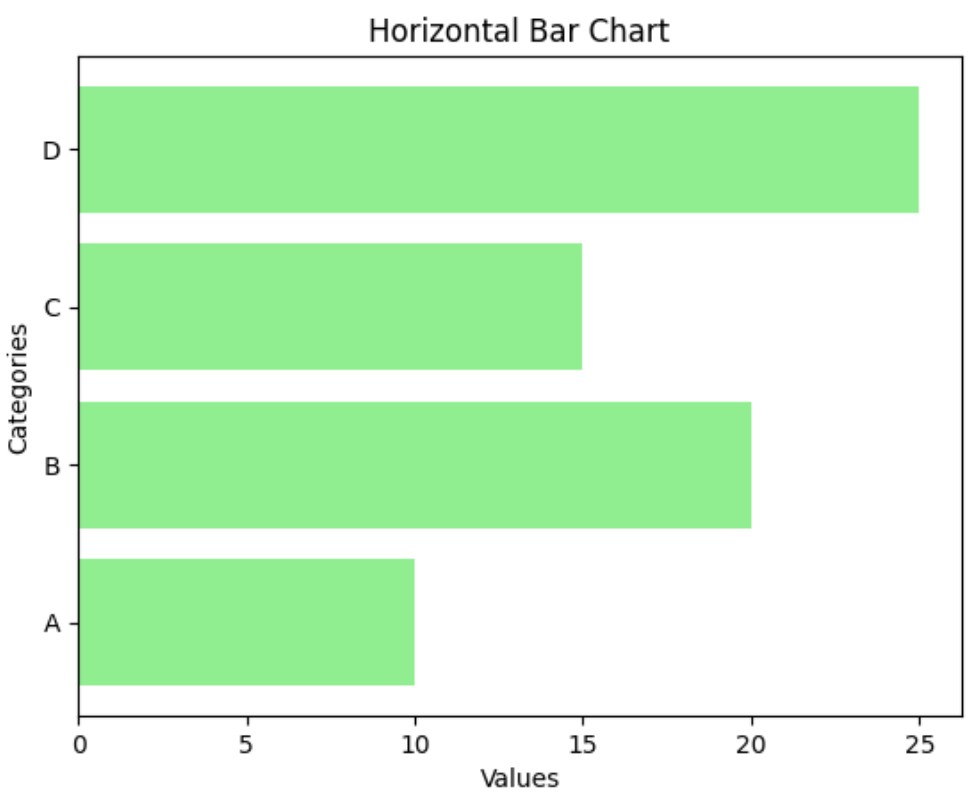
```
x = numpy.random.normal(5.0, 1.0, 1000)
y = numpy.random.normal(10.0, 2.0, 1000)


plt.scatter(x, y)
plt.show()
```



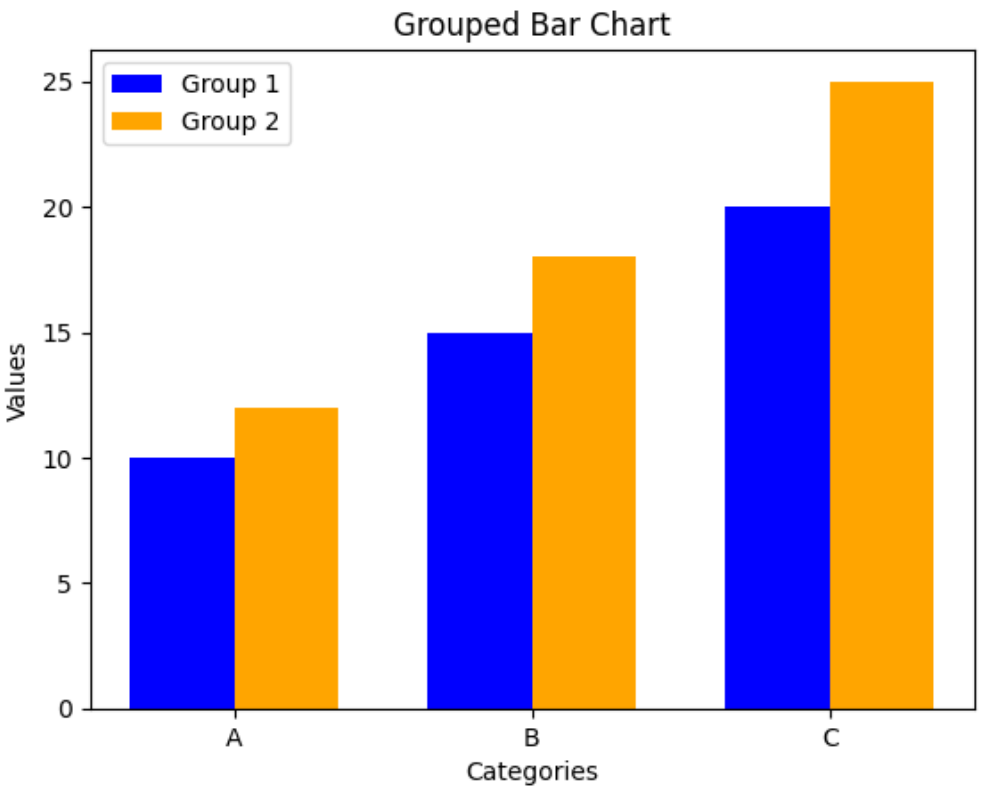
In [28]:
```
import matplotlib.pyplot as plt

# Data
categories = ['A', 'B', 'C', 'D']
values = [10, 20, 15, 25]

# Create Bar Chart
plt.bar(categories, values, color='skyblue')

# Add Labels and Title
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Basic Bar Chart')

# Show the Plot
plt.show()
```



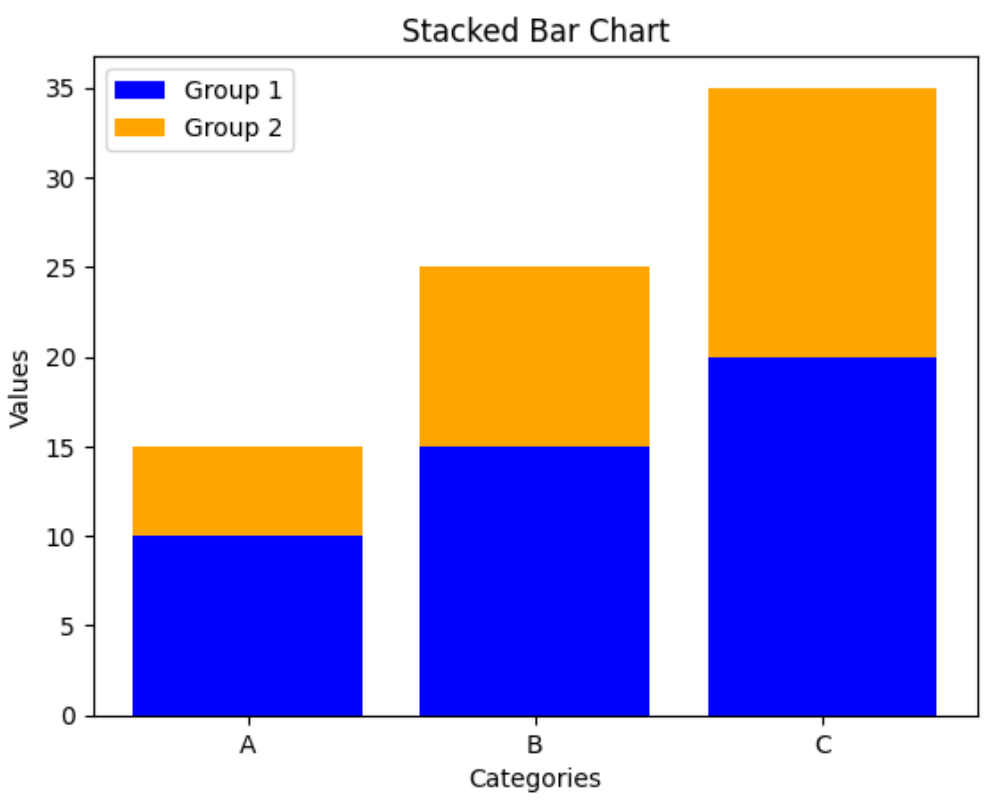
In [8]:
```
import matplotlib.pyplot as plt

# Data
categories = ['A', 'B', 'C', 'D']
values = [10, 20, 15, 25]

# Horizontal Bar Chart
plt.barh(categories, values, color='lightgreen')

# Add Labels and Title
plt.xlabel('Values')
plt.ylabel('Categories')
plt.title('Horizontal Bar Chart')

# Show the Plot
plt.show()
```



In [22]:
```python
import numpy as np
import matplotlib.pyplot as plt

# Data
categories = ['A', 'B', 'C']
values1 = [10, 15, 20]
values2 = [12, 18, 25]

# Position of bars
x = np.arange(len(categories))
width = 0.35

# Plot Bars
plt.bar(x - width/2, values1, width, label='Group 1', color='blue')
plt.bar(x + width/2, values2, width, label='Group 2', color='orange')

# Add Labels and Title
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Grouped Bar Chart')
plt.xticks(x, categories)
plt.legend()

# Show the Plot
plt.show()
```



In [30]:
```python
import numpy as np
import matplotlib.pyplot as plt

# Data
categories = ['A', 'B', 'C']
values1 = [10, 15, 20]
values2 = [5, 10, 15]

# Create Stacked Bars
plt.bar(categories, values1, label='Group 1', color='blue')
plt.bar(categories, values2, label='Group 2', color='orange', bottom=values1)

# Add Labels and Title
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Stacked Bar Chart')
plt.legend()

# Show the Plot
plt.show()
```



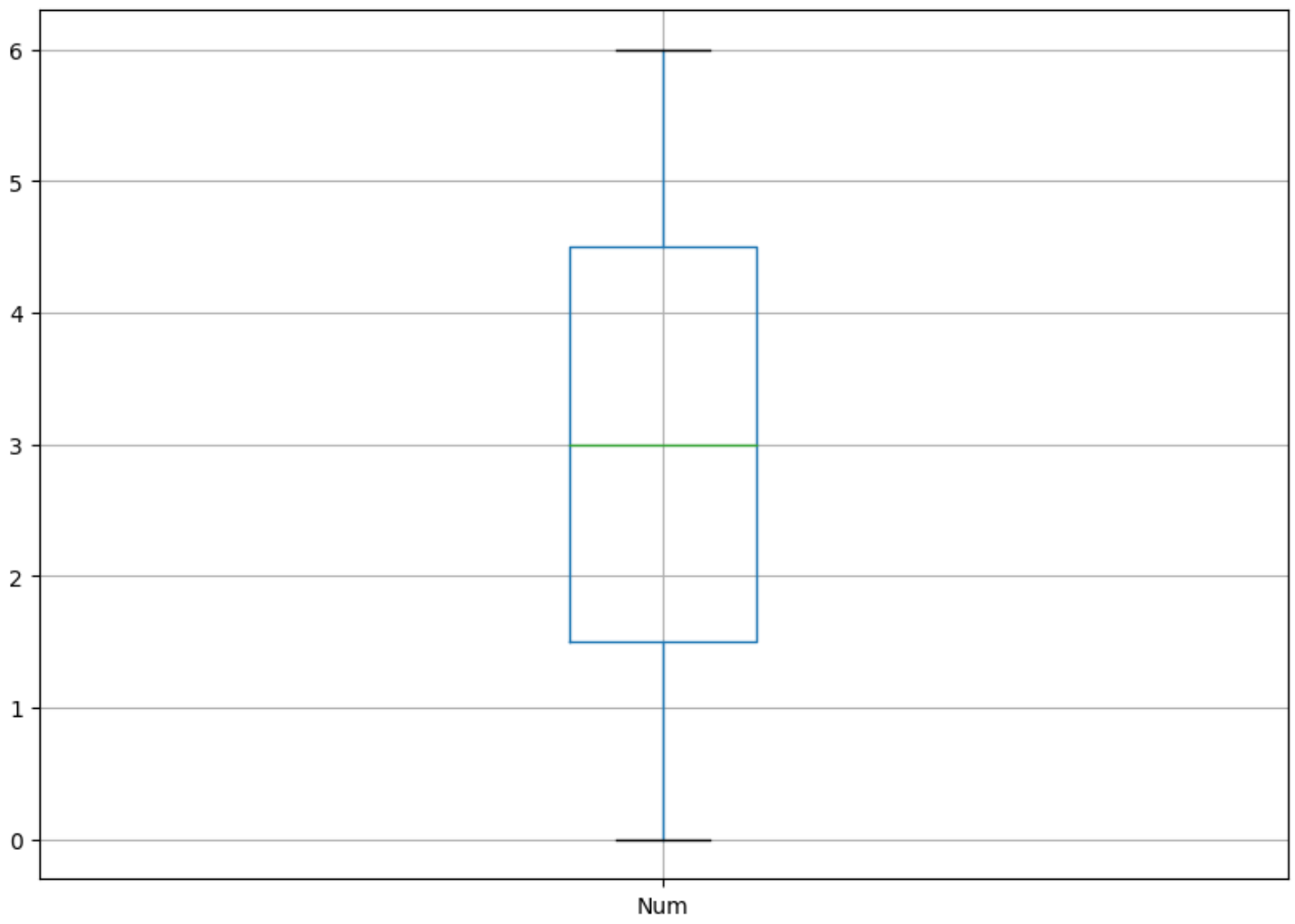
In [1]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
data = [0, 1, 2, 3, 4, 5, 6]
df = pd.DataFrame(data, columns = ['Num'])
df
```

Out[1]:

	Num
0	0
1	1
2	2
3	3
4	4
5	5
6	6

In [2]:
```
plt.figure(figsize = (10, 7))

df.boxplot()
```

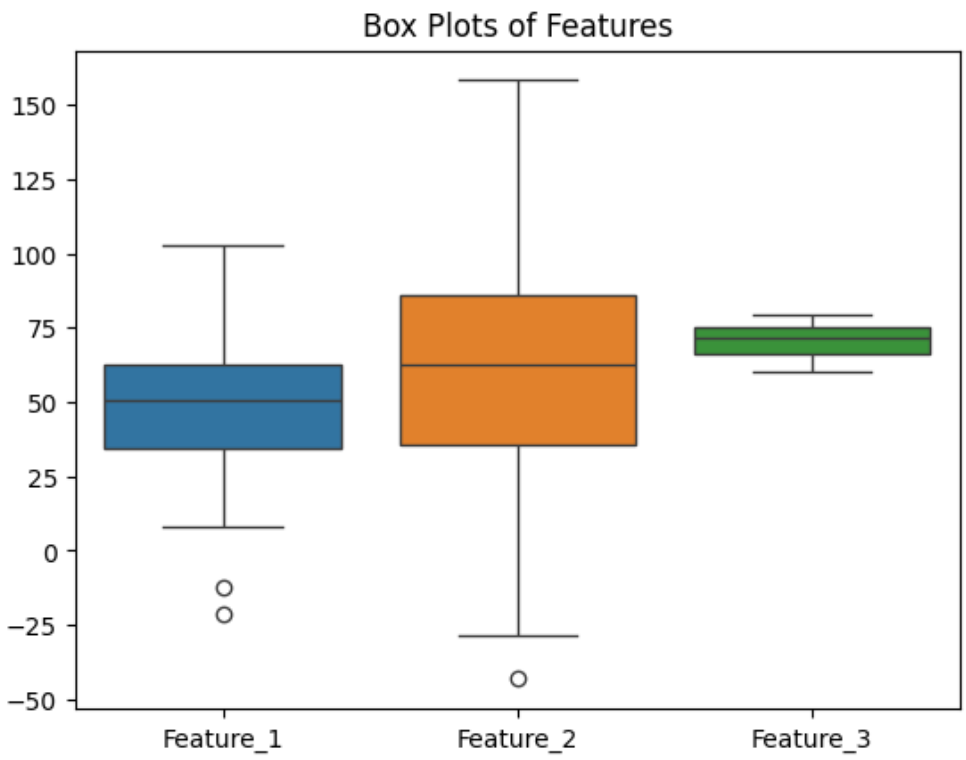
Out[2]: <Axes: >



In [4]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd

# Generate simple data
data = {
    'Feature_1': np.random.normal(50, 20, 100),
    'Feature_2': np.random.normal(60, 40, 100),
    'Feature_3': np.random.uniform(80, 60, 100)
}
df = pd.DataFrame(data)
print(df)
```

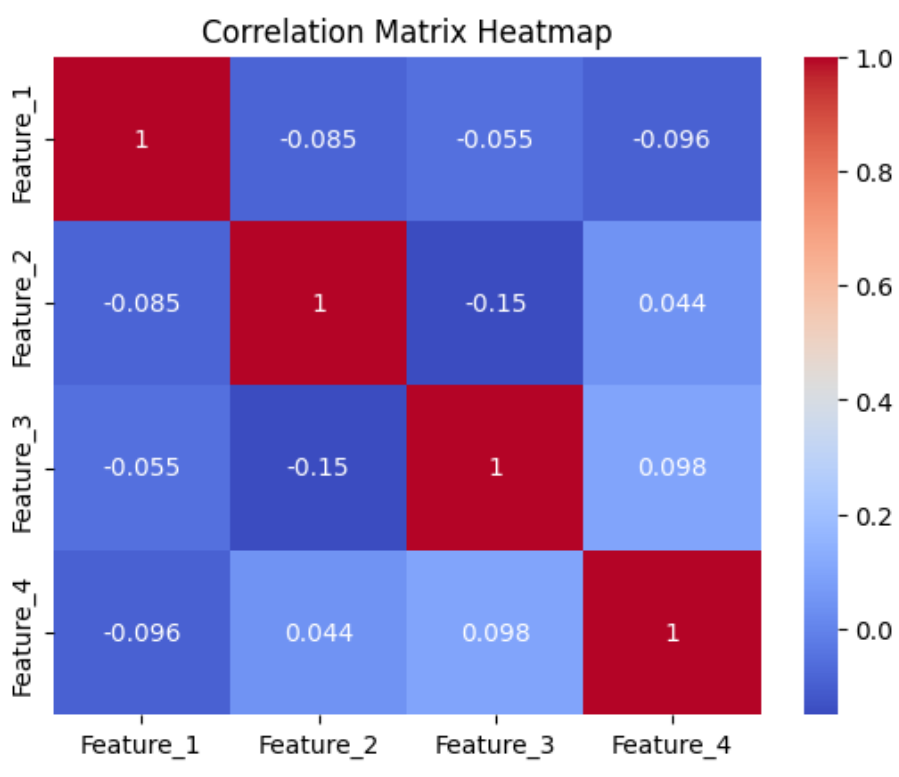
```
# Plot simple box plot
sns.boxplot(data=df)
plt.title("Box Plots of Features")
plt.show()
# Detect outliers using IQR method
for column in df.columns:
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[column] < lower_bound) | (df[column] > upper_bound)]
    print(f"Outliers in {column}:\n{outliers}\n")
```

```
    Feature_1   Feature_2  Feature_3
0   55.198202   71.568923  62.896730
1   32.653712  120.414576  75.324035
2   48.028596   70.253470  72.408271
3   50.636442   18.986758  71.558774
4   28.267491   -0.847092  78.593385
..        ...         ...        ...
95  55.012117   81.769109  72.251833
96  73.834227   50.781617  69.081590
97  72.983071   51.014158  71.621766
98 -21.337866   59.627151  75.281177
99  53.380478   34.775142  70.923037

[100 rows x 3 columns]
```




```
Outliers in Feature_1:
    Feature_1   Feature_2  Feature_3
10 -12.617317  108.712867  69.234657
98 -21.337866   59.627151  75.281177

Outliers in Feature_2:
    Feature_1  Feature_2  Feature_3
14  22.233103 -43.221552  75.470077

Outliers in Feature_3:
Empty DataFrame
Columns: [Feature_1, Feature_2, Feature_3]
Index: []
```

In [5]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Generate simple data

data = {
    'Feature_1': np.random.normal(50, 10, 100),
    'Feature_2': np.random.normal(30, 5, 100),
    'Feature_3': np.random.normal(10, 2, 100),
    'Feature_4': np.random.uniform(20, 40, 100)
}

df = pd.DataFrame(data)

# Compute correlation matrix
correlation_matrix = df.corr()

# Plot heatmap
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title("Correlation Matrix Heatmap")
plt.show()
```


Correlation Matrix Heatmap
Feature_1
Feature_2
Feature_3
Feature_4
1
-0.085
-0.055
-0.096
-0.085
1
-0.15
0.044
-0.055
-0.15
1
0.098
-0.096
0.044
0.098
1
1.0
0.8
0.6
0.4
0.2
0.0

In [ ]: