In [1]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\hardik gohil\anaconda3\lib\site-pack
ages (1.25.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import numpy as np
```

In [3]:
```
def read_xy_from_file(filename):
    yi, xi = [], [] # Create two empty lists, to be filled in later in the function

    with open("linear-data-set-for-regression.csv", "r") as file:
        a_line = file.readline() # read the first line, and ignore it. It is the the h
        a_line = file.readline() # read the second line and then enter a 'while' loop

        while a_line:           # So long as a line has been successfully read ...
            yt, xt = a_line.strip().split(",")  # strip the line of leading and traili
            yi.append(float(yt))                # convert the string to a float and ad
            xi.append(float(xt))                # convert the string to a float and ad
            a_line = file.readline()            # read the next line, and re-enter the

    # When control comes here, it means that 'a_line' is empty, there was nothing more
    return yi, xi
```

In [4]:
```
yi, xi = read_xy_from_file("linear-data-set-for-regression.csv")
y = np.array(yi)
x = np.array(xi)
print(f"Successfully read {len(y)} records from the file")
```

```
Successfully read 99 records from the file
```

In [5]:
```
pip install matplotlib
```

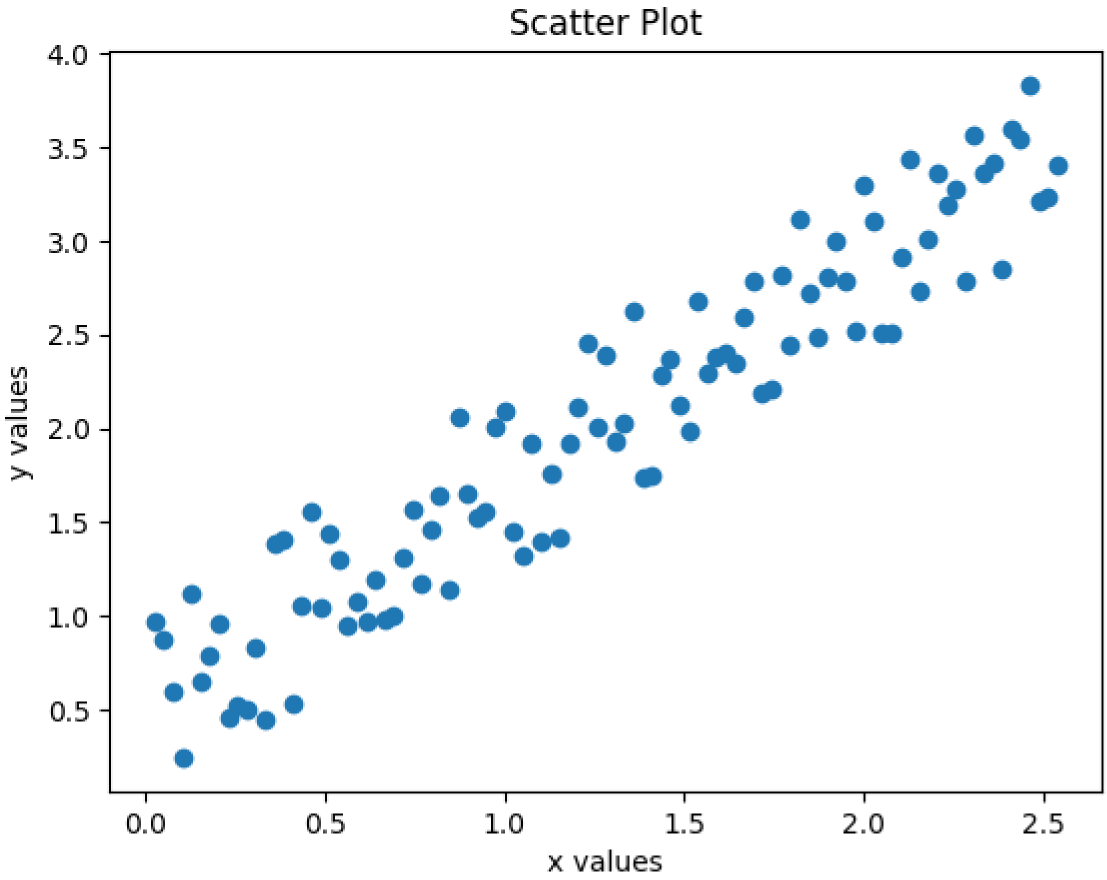
```
Requirement already satisfied: matplotlib in c:\users\hardik gohil\anaconda3\lib\site
-packages (3.7.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hardik gohil\anaconda3\l
ib\site-packages (from matplotlib) (4.41.0)
Requirement already satisfied: numpy>=1.20 in c:\users\hardik gohil\anaconda3\lib\sit
e-packages (from matplotlib) (1.25.0)
Requirement already satisfied: cycler>=0.10 in c:\users\hardik gohil\anaconda3\lib\si
te-packages (from matplotlib) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hardik gohil\anaconda3\li
b\site-packages (from matplotlib) (1.1.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\hardik gohil\anacond
a3\lib\site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: packaging>=20.0 in c:\users\hardik gohil\anaconda3\lib
\site-packages (from matplotlib) (23.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hardik gohil\anaconda3\lib\s
ite-packages (from matplotlib) (9.4.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hardik gohil\anaconda
3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hardik gohil\anaconda3\l
ib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: six>=1.5 in c:\users\hardik gohil\anaconda3\lib\site-p
ackages (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [6]:

```
import matplotlib.pyplot as plt
```

In [7]:

```
# Let's try a simple scatter plot

plt.scatter(x=x, y=y) # Plots a scatter plot

plt.title("Scatter Plot")
plt.ylabel("y values")
plt.xlabel("x values")
plt.show()
```



In [8]:

```
xbar = np.mean(x)
ybar = np.mean(y)
xybar = np.mean(x*y)
xsqbar = np.mean(x*x)
```

In [9]:

```
a = (xybar - xbar*ybar)/(xsqbar - xbar*xbar)
b = (ybar*xsqbar - xbar*xybar)/(xsqbar - xbar*xbar)
```

In [10]:

```
print(a,b)
```

```
1.1719511804459168 0.5156466449319416
```

In [11]:

```
ycap = a*x + b
```

In [18]:

```
e = ycap - y
```

In [19]:

```
from scipy.stats import normaltest
```

```
# Perform the normality test
statistic, p_value = normaltest(e)

print("Statistic:", statistic)
print("P-value:", p_value)

# Interpret the result
alpha = 0.05  # Significance level
if p_value < alpha:
    print("The prediction errors do not follow a normal distribution (reject H0)")
else:
    print("The prediction errors may follow a normal distribution (fail to reject H0)'
```

```
Statistic: 17.521894547562955
P-value: 0.00015673606627631648
The prediction errors do not follow a normal distribution (reject H0)
```

In [20]:

```
SST = np.sum((y-ybar)**2)
SSR = np.sum((ycap-ybar)**2)
SSE = np.sum(e*e)
Rsq = SSR/SST
print(SST,SSR,SSE,Rsq)
```

```
81.10703596594 73.00789919833157 8.099136767608572 0.9001426119059563
```

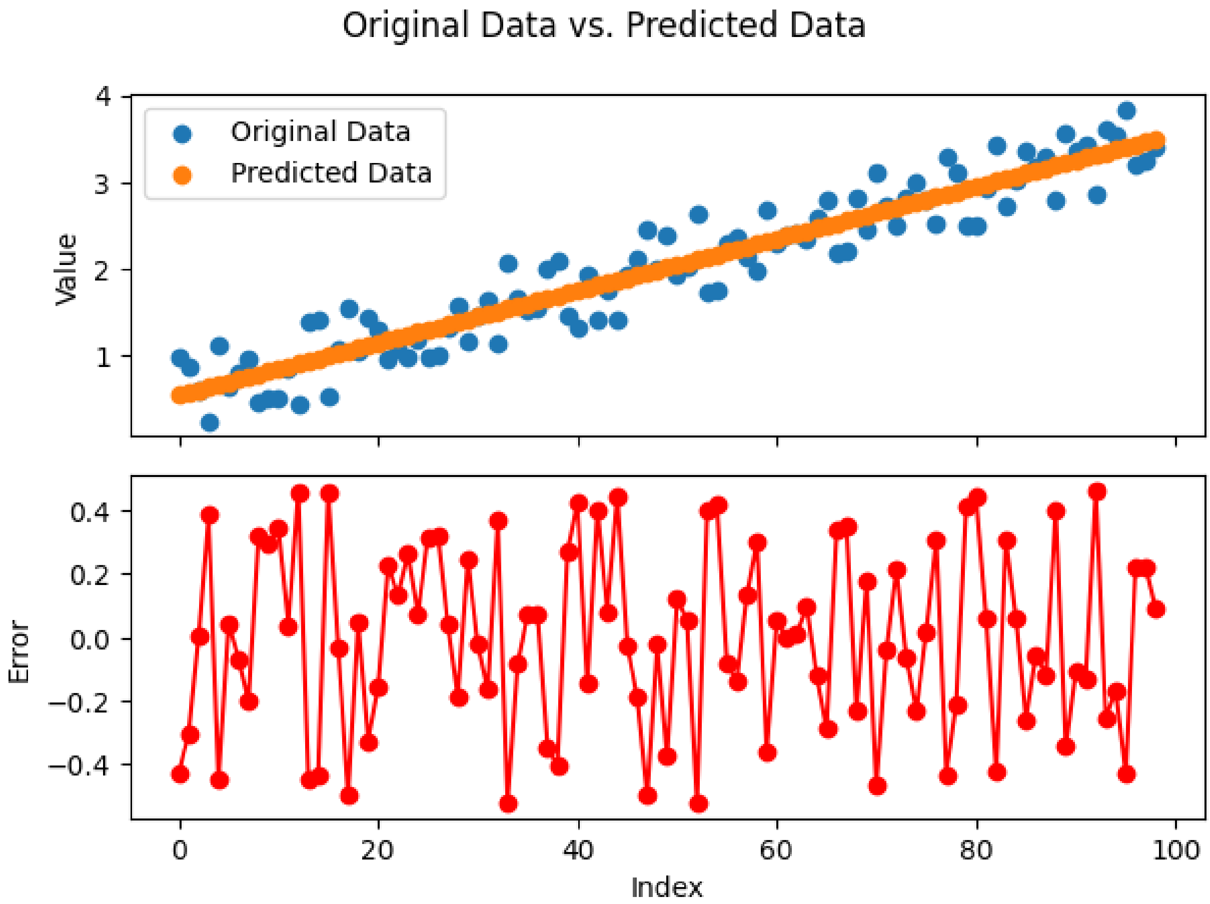
In [14]:

```
print(SST,SSR + SSE)
```

```
81.10703596594 81.10703596594014
```

In [21]:

```
# Create a figure with two subplots
fig, (ax1, ax2) = plt.subplots(2, 1, sharex=True)

# Subplot 1: Scatter plot of original data and predicted points
ax1.scatter(range(len(y)), y, label='Original Data')
ax1.scatter(range(len(ycap)), ycap, label='Predicted Data')
ax1.set_ylabel('Value')
ax1.legend()

# Subplot 2: Error plot
ax2.plot(range(len(e)), e, marker='o', linestyle='-', color='r')
ax2.set_xlabel('Index')
ax2.set_ylabel('Error')

# Set title for the whole figure
fig.suptitle('Original Data vs. Predicted Data')

# Adjust layout
plt.tight_layout()

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



In [25]:

```
report = f"""Metrics Report
-------------------
xbar: {xbar}
ybar: {ybar}
xybar: {xybar}
xsqbar: {xsqbar}
a: {a}
b: {b}
p value: {p_value}
SST: {SST}
SSR: {SSR}
SSE: {SSE}
Rsq: {Rsq}
"""
with open("metrics.txt","w") as file:
    file.write(report)
```

Metrics report

image.png

In [1]:

```
Metrics Report
-------------------
xbar: 1.282051282050505
ybar: 2.0181481583232324
xybar: 3.216622204305499
xsqbar: 2.1805829498060683
a: 1.1719511804459168
```

```
b: 0.5156466449319416
p value: 0.00015673606627631648
SST: 81.10703596594
SSR: 73.00789919833157
SSE: 8.099136767608572
Rsq: 0.9001426119059563
```

```
  Cell In[1], line 1
    Metrics Report
            ^
SyntaxError: invalid syntax
```

In [ ]: