In [5]:
```python
import pandas as pd
import numpy as np
data = {
    "M1": [45,75,np.NaN,3],
    "M2": [36,48,75,1000],
    "M3": [75,np.NaN,25,11],
    "M4": [33,np.NaN,72,31]
}
df = pd.DataFrame(data)
print("Original DataFrame:")
print(df)
```

```
Original DataFrame:
      M1    M2    M3    M4
0   45.0    36  75.0  33.0
1   75.0    48   NaN   NaN
2    NaN    75  25.0  72.0
3    3.0  1000  11.0  31.0
```

In [6]:
```python
df_dropped = df.dropna()
print("\nDataFrame after dropping missing values:")
print(df_dropped)
```

```
DataFrame after dropping missing values:
      M1    M2    M3    M4
0   45.0    36  75.0  33.0
3    3.0  1000  11.0  31.0
```

In [7]:
```python
df_fillzero=df.fillna(0)
print("\nDataFrame after Filling Zeros for the missing values:")
print(df_fillzero)
```

```
DataFrame after Filling Zeros for the missing values:
      M1    M2    M3    M4
0   45.0    36  75.0  33.0
1   75.0    48   0.0   0.0
2    0.0    75  25.0  72.0
3    3.0  1000  11.0  31.0
```

In [8]:
```python
df_replaced = df.replace(np.NaN,32)
print("\nDataFrame after replacing specific values:")
print(df_replaced)
```

```
DataFrame after replacing specific values:
     M1    M2    M3    M4
0  45.0    36  75.0  33.0
1  75.0    48  32.0  32.0
2  32.0    75  25.0  72.0
3   3.0  1000  11.0  31.0
```

In [9]:
```python
df_dropped_column = df_replaced.drop(columns=['M4'])
print("\nDataFrame after dropping column 'M4':")
print(df_dropped_column)
```

```
DataFrame after dropping column 'M4':
     M1    M2    M3
0  45.0    36  75.0
1  75.0    48  32.0
2  32.0    75  25.0
3   3.0  1000  11.0
```

In [ ]:

```
In [1]: import pandas as pd
        import numpy as np
        data = {
            'A': [1, 2, np.nan, 4, 5],
            'B': [np.nan, 2, 3, np.nan, 5],
            'C': [1, np.nan, np.nan, 4, 5],
            'D': [1, 2, 3, 4, np.nan]
        }
        df = pd.DataFrame(data)

        print("Original DataFrame:")
        print(df)
```

```
Original DataFrame:
     A    B    C    D
0  1.0  NaN  1.0  1.0
1  2.0  2.0  NaN  2.0
2  NaN  3.0  NaN  3.0
3  4.0  NaN  4.0  4.0
4  5.0  5.0  5.0  NaN
```

```
In [3]: df_forward_fill = df.ffill()
        print("\nDataFrame after forward fill:")
        print(df_forward_fill)
```

```
DataFrame after forward fill:
     A    B    C    D
0  1.0  NaN  1.0  1.0
1  2.0  2.0  1.0  2.0
2  2.0  3.0  1.0  3.0
3  4.0  3.0  4.0  4.0
4  5.0  5.0  5.0  4.0
```

```
In [4]: df_backward_fill = df.bfill()
        print("\nDataFrame after backward fill:")
        print(df_backward_fill)
```

```
DataFrame after backward fill:
     A    B    C    D
0  1.0  2.0  1.0  1.0
1  2.0  2.0  4.0  2.0
2  4.0  3.0  4.0  3.0
3  4.0  5.0  4.0  4.0
4  5.0  5.0  5.0  NaN
```

In [5]:
```python
nan_bool = df.isnull()
print("\nBoolean DataFrame where NaN values are True:")
print(nan_bool)
```

```
Boolean DataFrame where NaN values are True:
       A      B      C      D
0  False   True  False  False
1  False  False   True  False
2   True  False   True  False
3  False   True  False  False
4  False  False  False   True
```

In [6]:
```python
df_drop_na = df.dropna()
print("\nDataFrame after dropping rows with missing values:")
print(df_drop_na)
```

```
DataFrame after dropping rows with missing values:
Empty DataFrame
Columns: [A, B, C, D]
Index: []
```

In [ ]:

```
In [2]: import pandas as pd
        df1 = pd.DataFrame({'A': ['A1', 'A2', 'A3'], 'B': ['B1', 'B2', 'B3']})
        df2 = pd.DataFrame({'C': ['C1', 'C2', 'C3'], 'D': ['D1', 'D2', 'D3']}, index=[0,
        joined_df = df1.join(df2)
        print(joined_df)
```

```
    A   B   C   D
0  A1  B1  C1  D1
1  A2  B2  C2  D2
2  A3  B3  C3  D3
```

```
In [3]: melted_df = pd.melt(joined_df, id_vars=['A'])
        print(melted_df)
```

```
    A variable value
0  A1        B    B1
1  A2        B    B2
2  A3        B    B3
3  A1        C    C1
4  A2        C    C2
5  A3        C    C3
6  A1        D    D1
7  A2        D    D2
8  A3        D    D3
```

```
In [ ]:
```

```python
In [1]: import pandas as pd
        df3 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Jana', 'Smith', 'Ashwin']})
        df4 = pd.DataFrame({'ID': [2, 3, 4], 'Score': [85, 92, 78]})

        right_merge = pd.merge(df3, df4, on='ID', how='right')
        left_merge = pd.merge(df3, df4, on='ID', how='left')
        inner_merge = pd.merge(df3, df4, on='ID', how='inner')
        outer_merge = pd.merge(df3, df4, on='ID', how='outer')
```

```python
In [2]: concat_df = pd.concat([df3, df4], axis=0)
        print(right_merge)
        print(left_merge)
        print(inner_merge)
        print(outer_merge)
        print(concat_df)
```

```
   ID    Name  Score
0   2   Smith     85
1   3  Ashwin     92
2   4     NaN     78
   ID    Name  Score
0   1    Jana    NaN
1   2   Smith   85.0
2   3  Ashwin   92.0
   ID    Name  Score
0   2   Smith     85
1   3  Ashwin     92
   ID    Name  Score
0   1    Jana    NaN
1   2   Smith   85.0
2   3  Ashwin   92.0
3   4     NaN   78.0
   ID    Name  Score
0   1    Jana    NaN
1   2   Smith    NaN
2   3  Ashwin    NaN
0   2     NaN   85.0
1   3     NaN   92.0
2   4     NaN   78.0
```

```
In [ ]:
```

```
In [2]: import pandas as pd
        data = {'Category': ['A', 'B', 'A', 'B', 'C', 'A'],
                'Value': [10, 20, 15, 25, 30, 10]}

        df = pd.DataFrame(data)
        grouped_df = df.groupby('Category').sum()

        df1 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Jana', 'Ash', 'Smith']})
        df2 = pd.DataFrame({'ID': [2, 3, 4], 'Score': [85, 92, 78]})
```
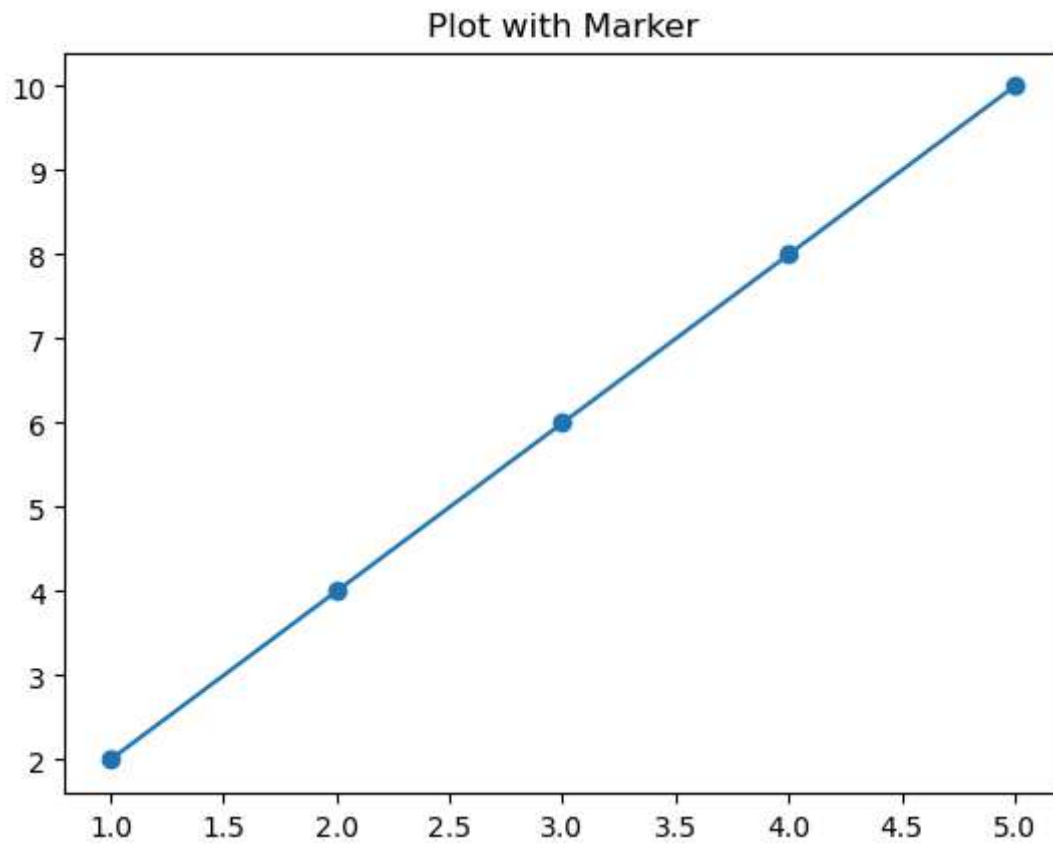
```
In [4]: merged_df = pd.merge(df1, df2, on='ID', how='inner')

        print(grouped_df)
        print(merged_df)
```
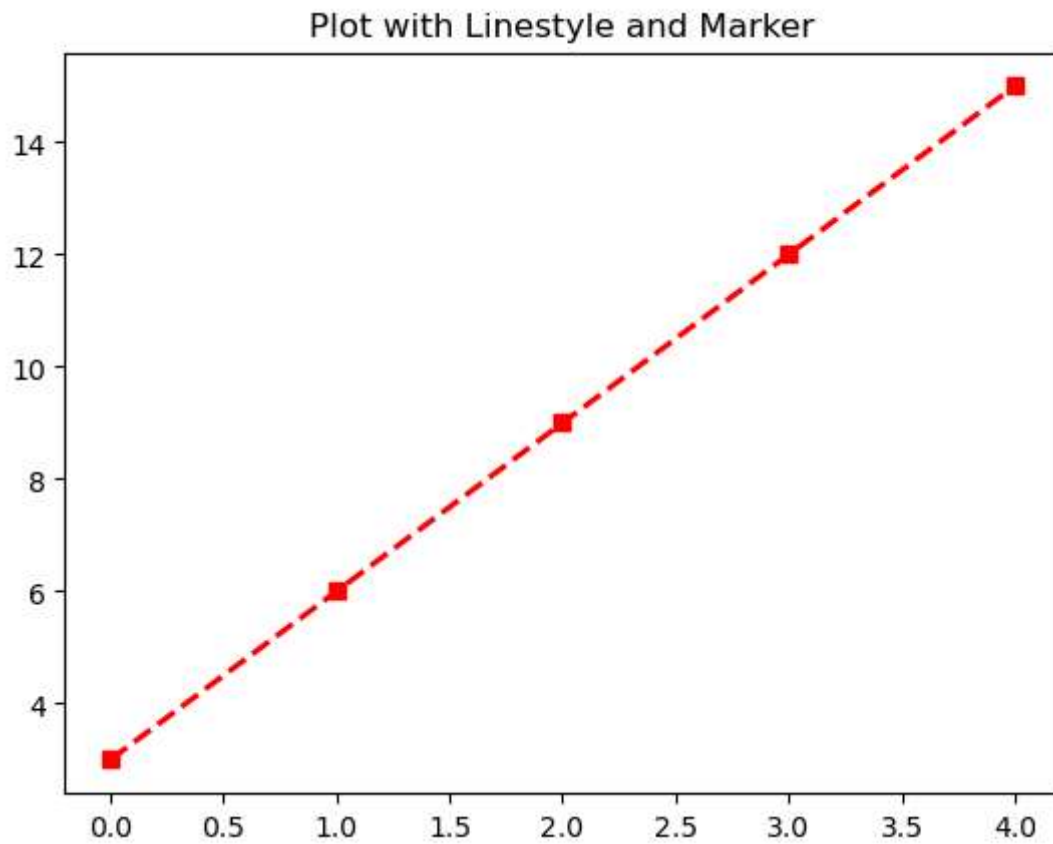
```
          Value
Category
A            35
B            45
C            30
   ID   Name  Score
0   2    Ash     85
1   3  Smith     92
```

```
In [ ]:
```

In [9]:
```python
import numpy as np
import matplotlib.pyplot as plt
x = np.array([1, 2, 3, 4, 5])
y = np.array([2, 4, 6, 8, 10])
plt.plot(x, y, marker='o')
plt.title('Plot with Marker')
plt.show()
```
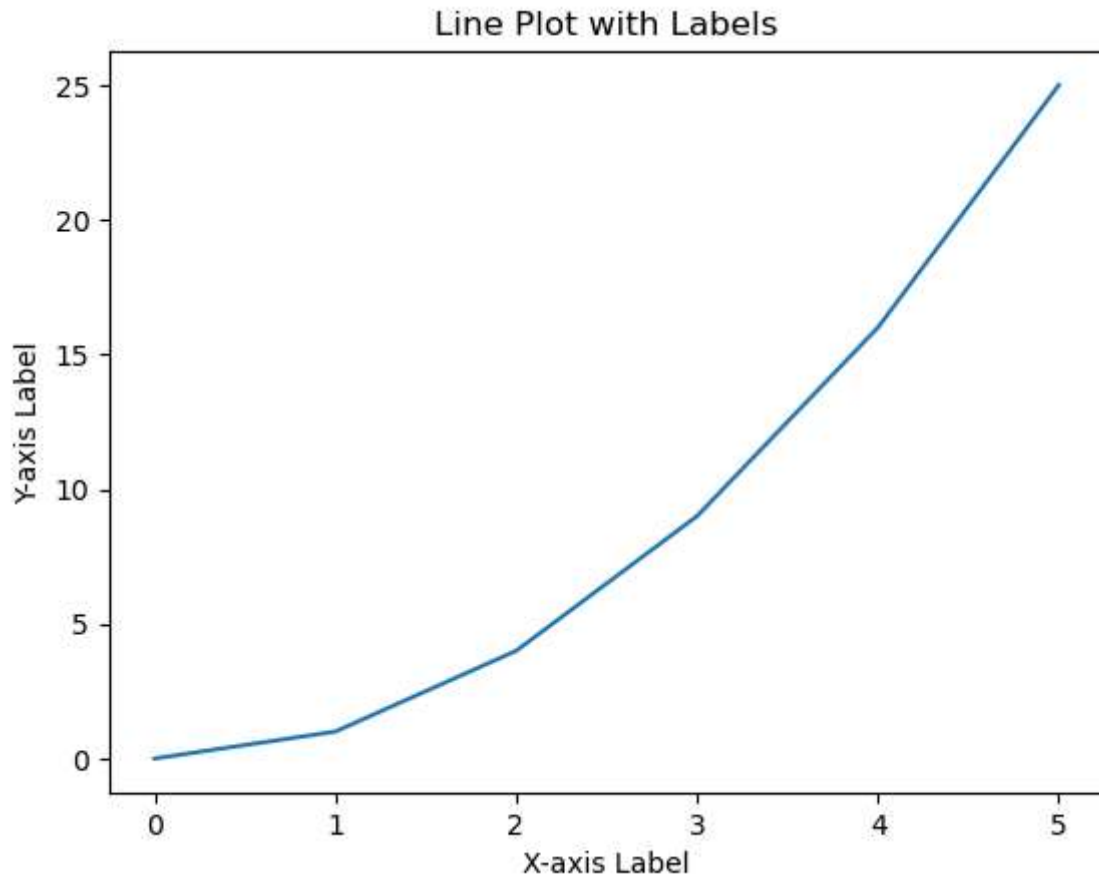

Plot with Marker

In [10]:
```python
y2 = np.array([3, 6, 9, 12, 15])
plt.plot(y2, linestyle='--', marker='s', linewidth=2, color='r')
plt.title('Plot with Linestyle and Marker')
plt.show()
```


Plot with Linestyle and Marker

In [11]:
```python
x2 = np.array([1, 2, 3, 4, 5])
y3 = np.array([5, 10, 15, 20, 25])
y4 = np.array([2, 4, 6, 8, 10])
plt.plot(x2, y3, label='Line 1', color='g')
plt.plot(x2, y4, label='Line 2', color='b')
plt.grid(True)
plt.legend()
plt.title('Multiple Lines with Grid')
plt.show()
```
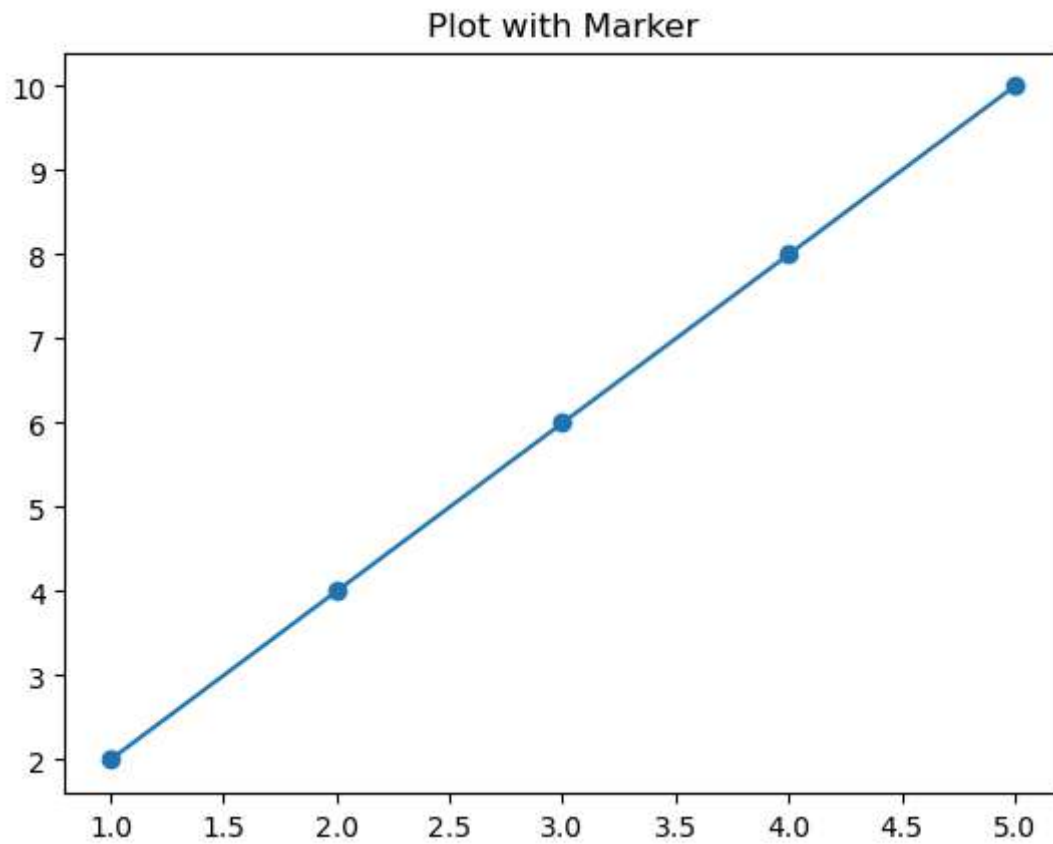

Multiple Lines with Grid

In [12]:
```python
x3 = np.array([0, 1, 2, 3, 4, 5])
y5 = np.array([0, 1, 4, 9, 16, 25])
plt.plot(x3, y5)
plt.title('Line Plot with Labels')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.show()
```
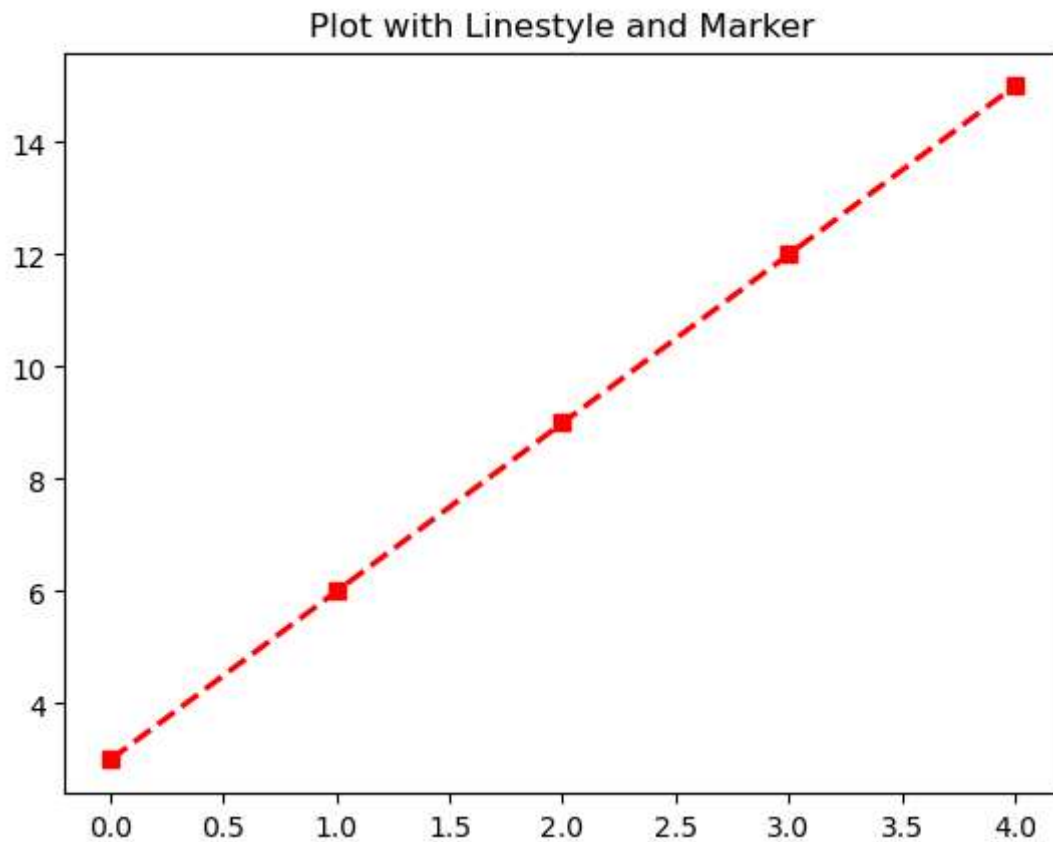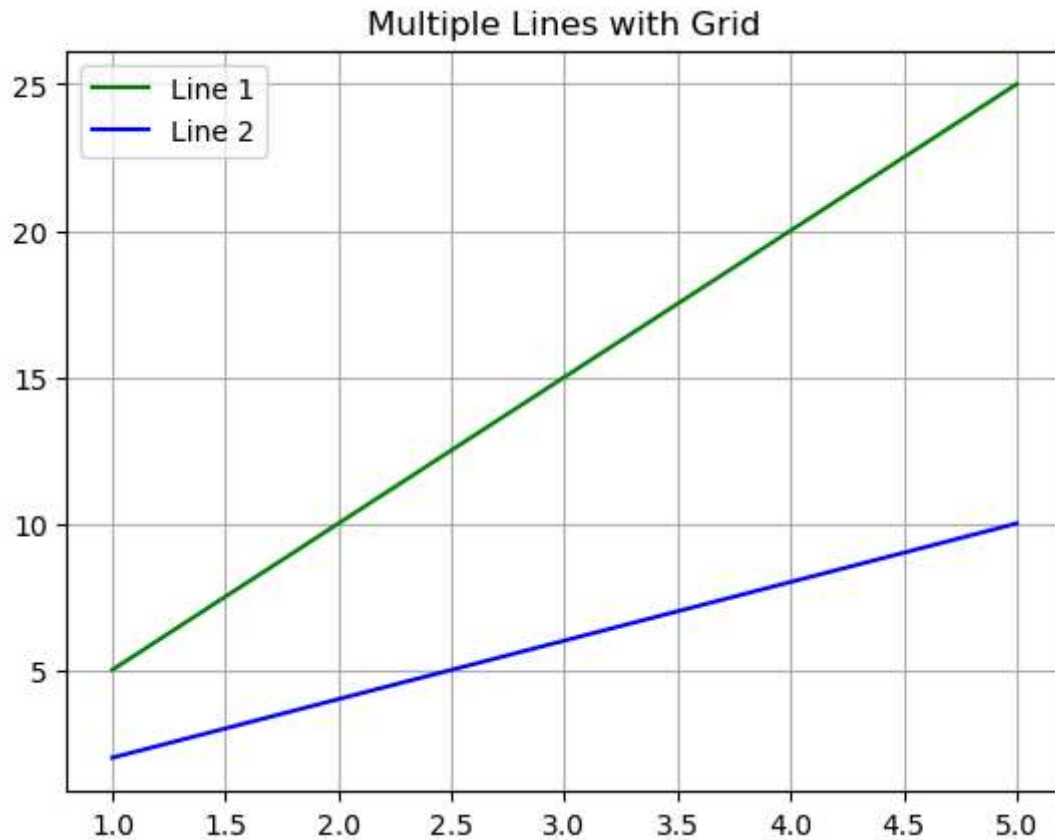

In [ ]:

In [9]:
```python
import numpy as np
import matplotlib.pyplot as plt
x = np.array([1, 2, 3, 4, 5])
y = np.array([2, 4, 6, 8, 10])
plt.plot(x, y, marker='o')
plt.title('Plot with Marker')
plt.show()
```
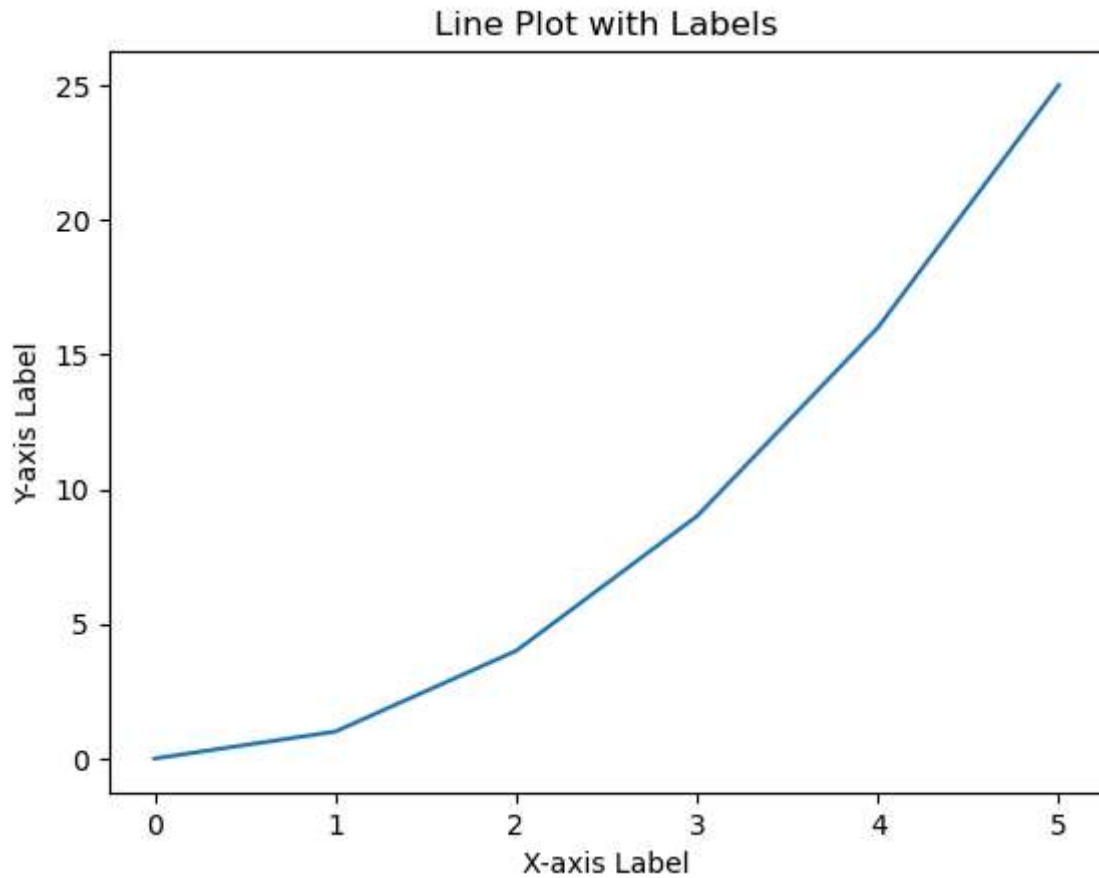

Plot with Marker

```
In [10]: y2 = np.array([3, 6, 9, 12, 15])
         plt.plot(y2, linestyle='--', marker='s', linewidth=2, color='r')
         plt.title('Plot with Linestyle and Marker')
         plt.show()
```


Plot with Linestyle and Marker

In [11]:
```python
x2 = np.array([1, 2, 3, 4, 5])
y3 = np.array([5, 10, 15, 20, 25])
y4 = np.array([2, 4, 6, 8, 10])
plt.plot(x2, y3, label='Line 1', color='g')
plt.plot(x2, y4, label='Line 2', color='b')
plt.grid(True)
plt.legend()
plt.title('Multiple Lines with Grid')
plt.show()
```


Multiple Lines with Grid

In [12]:
```python
x3 = np.array([0, 1, 2, 3, 4, 5])
y5 = np.array([0, 1, 4, 9, 16, 25])
plt.plot(x3, y5)
plt.title('Line Plot with Labels')
plt.xlabel('X-axis Label')
plt.ylabel('Y-axis Label')
plt.show()
```


In [ ]: