```
from google.colab import files


uploaded = files.upload()
import pandas as pd
```

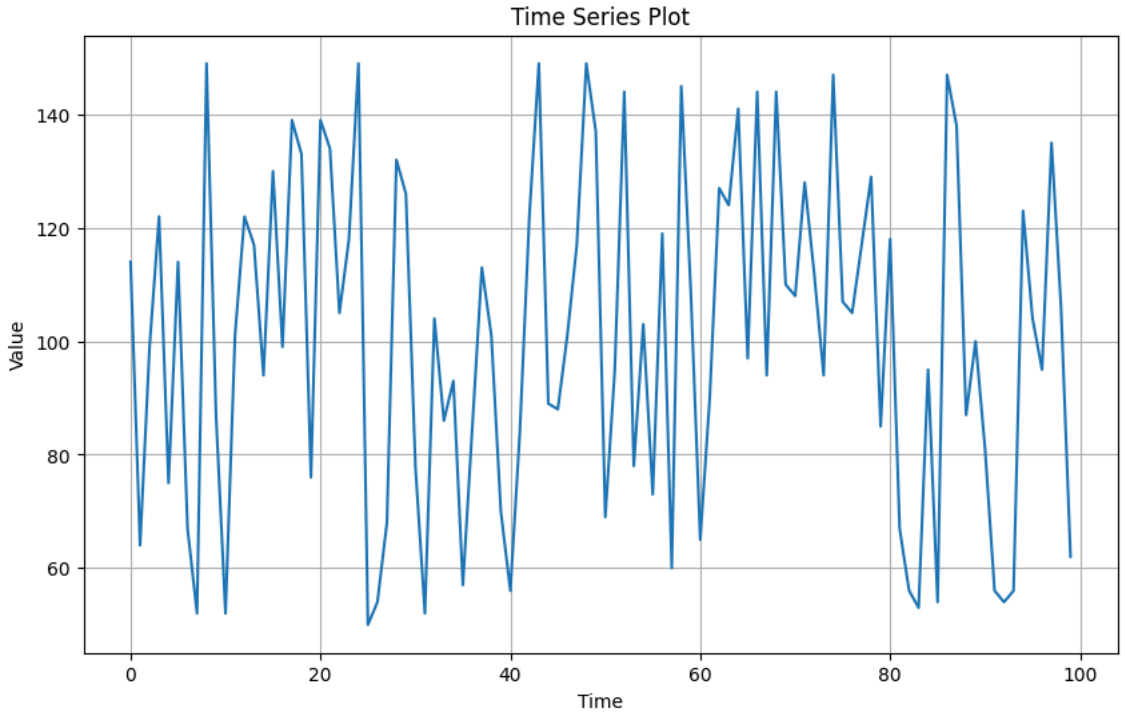
```
Choose Files time_series_data.csv
• time_series_data.csv(text/csv) - 1464 bytes, last modified: 3/4/2025 - 100% done
Saving time_series_data.csv to time_series_data (2).csv
```

```
import io

df = pd.read_csv(io.BytesIO(uploaded["time_series_data (2).csv"]))
print(df)
```

```
           Date  Value
0   2023-01-01    114
1   2023-01-02     64
2   2023-01-03     99
3   2023-01-04    122
4   2023-01-05     75
..         ...    ...
95  2023-04-06    104
96  2023-04-07     95
97  2023-04-08    135
98  2023-04-09    106
99  2023-04-10     62

[100 rows x 2 columns]
```

```
# prompt: Implement programs to check stationary of a time series data

import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import adfuller

# Assuming 'df' is your DataFrame and 'value_column' is the name of the column containing the time series data
# Replace 'value_column' with the actual column name in your DataFrame
value_column = 'Value'  # Changed from 'value' to 'Value' to match the actual column name

# Perform Augmented Dickey-Fuller test
result = adfuller(df[value_column])

# Print the test results
print('ADF Statistic: %f' % result[0])
print('p-value: %f' % result[1])
print('Critical Values:')
for key, value in result[4].items():
    print('\t%s: %.3f' % (key, value))

# Plot the time series data
plt.figure(figsize=(10, 6))
plt.plot(df.index, df[value_column])
plt.xlabel("Time")
plt.ylabel(value_column)
plt.title("Time Series Plot")
plt.grid(True)
plt.show()

# Interpret the results
if result[1] <= 0.05:
    print("\nThe time series is stationary.")
else:
    print("\nThe time series is non-stationary.")
```

```
ADF Statistic: -8.747765
p-value: 0.000000
Critical Values:
        1%: -3.498
        5%: -2.891
        10%: -2.583
```




```
The time series is stationary.
```