Assignment

a) Read test and label

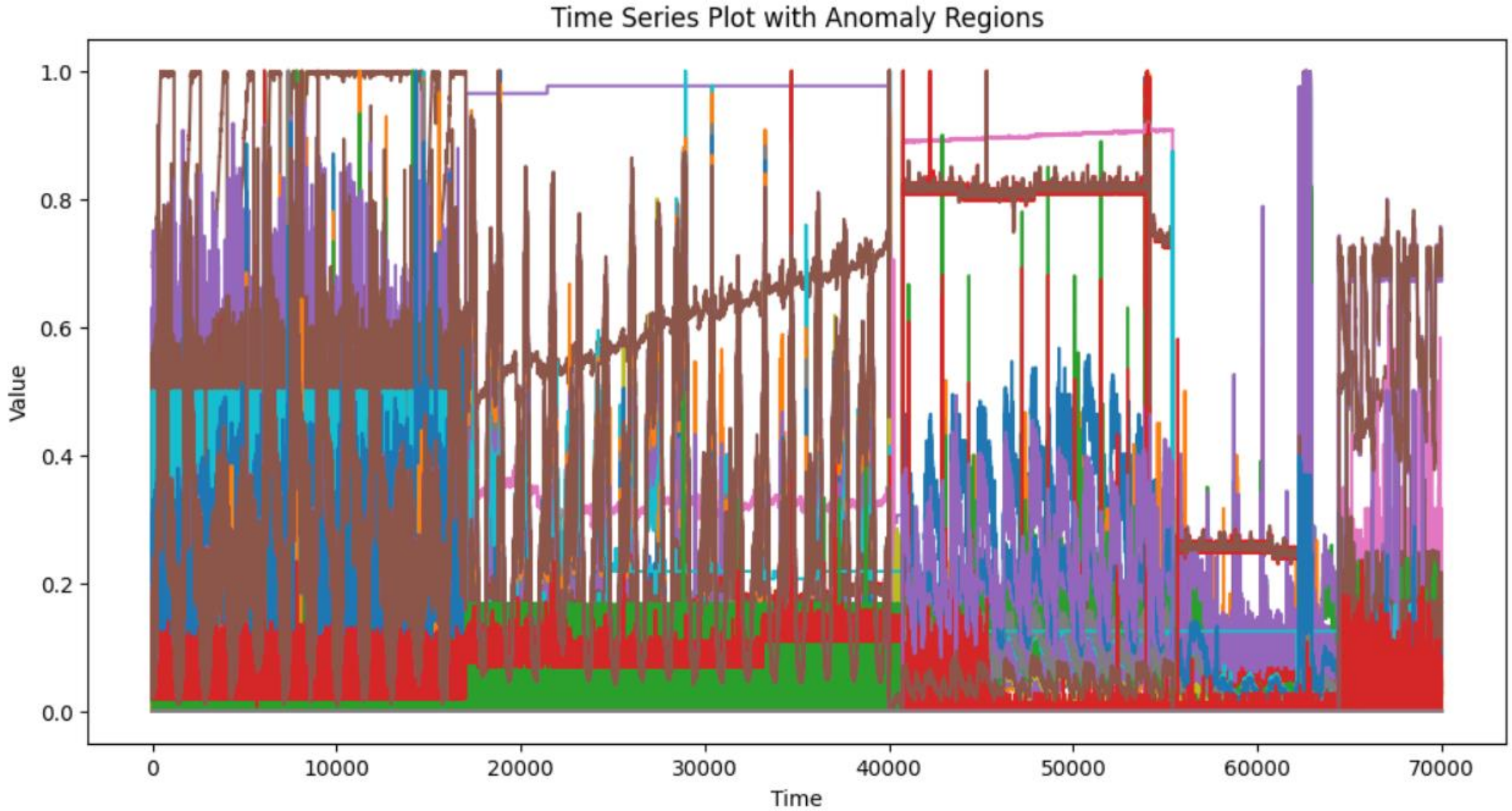
[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

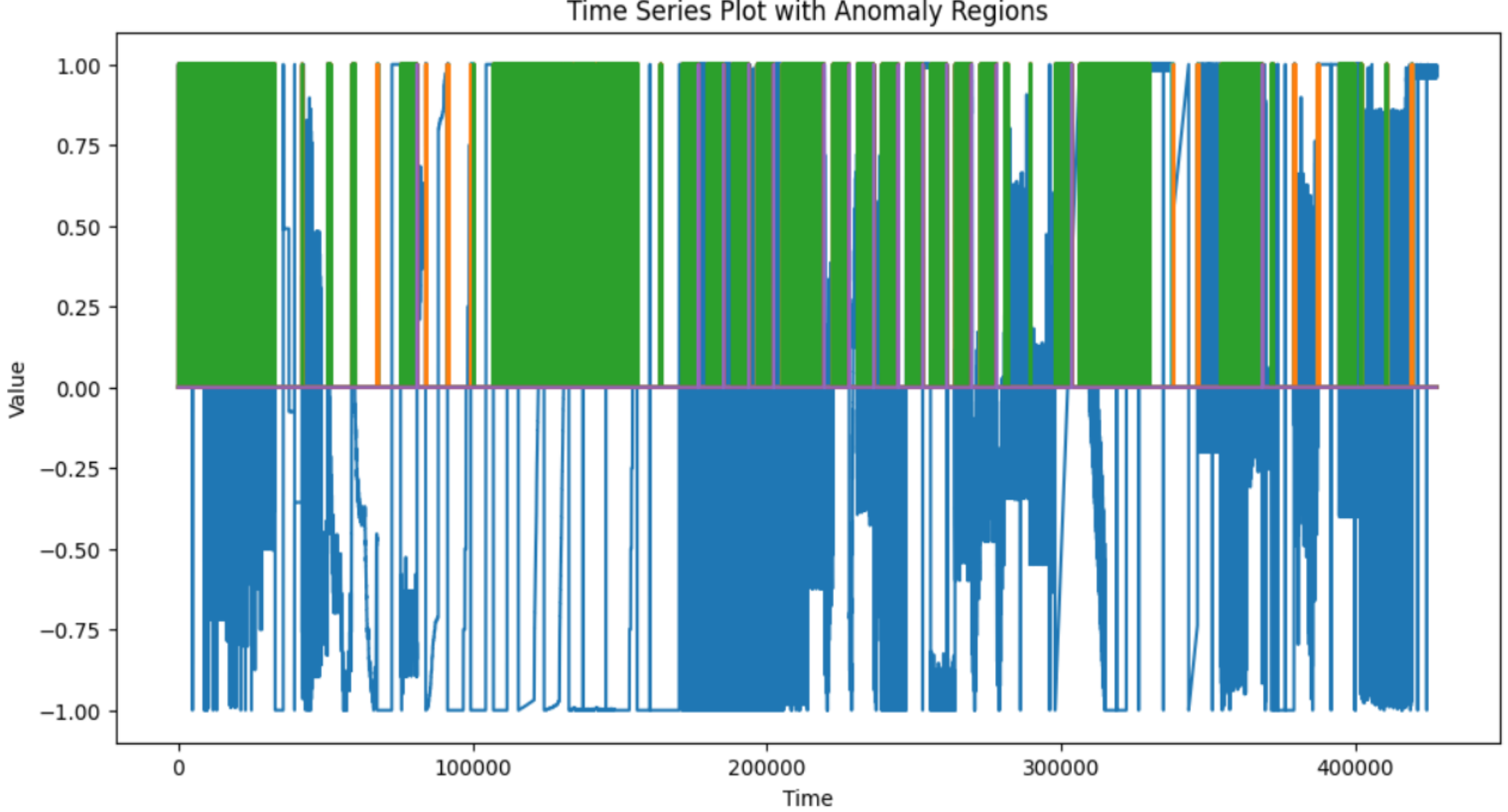
[3]:
```
# Defining file paths
file_paths = {
    "set1": ("Downloads/files/files/test.csv", "Downloads/files/files/test_label.csv"),
    "set2": ("Downloads/files/files/smap_test.csv", "Downloads/files/files/smap_test_label.csv"),
    "set3": ("Downloads/files/files/msl_test.csv", "Downloads/files/files/msl_test_label.csv"),
    "set4": ("Downloads/files/files/psm_test.csv", "Downloads/files/files/psm_test_label.csv")
}

# Read test and label files
def read_files(file_paths):
    data = {}
    for set_name, (test_file, label_file) in file_paths.items():
        test_data = pd.read_csv(test_file)
        label_data = pd.read_csv(label_file)
        data[set_name] = {"test_data": test_data, "label_data": label_data}
    return data

data = read_files(file_paths)
```

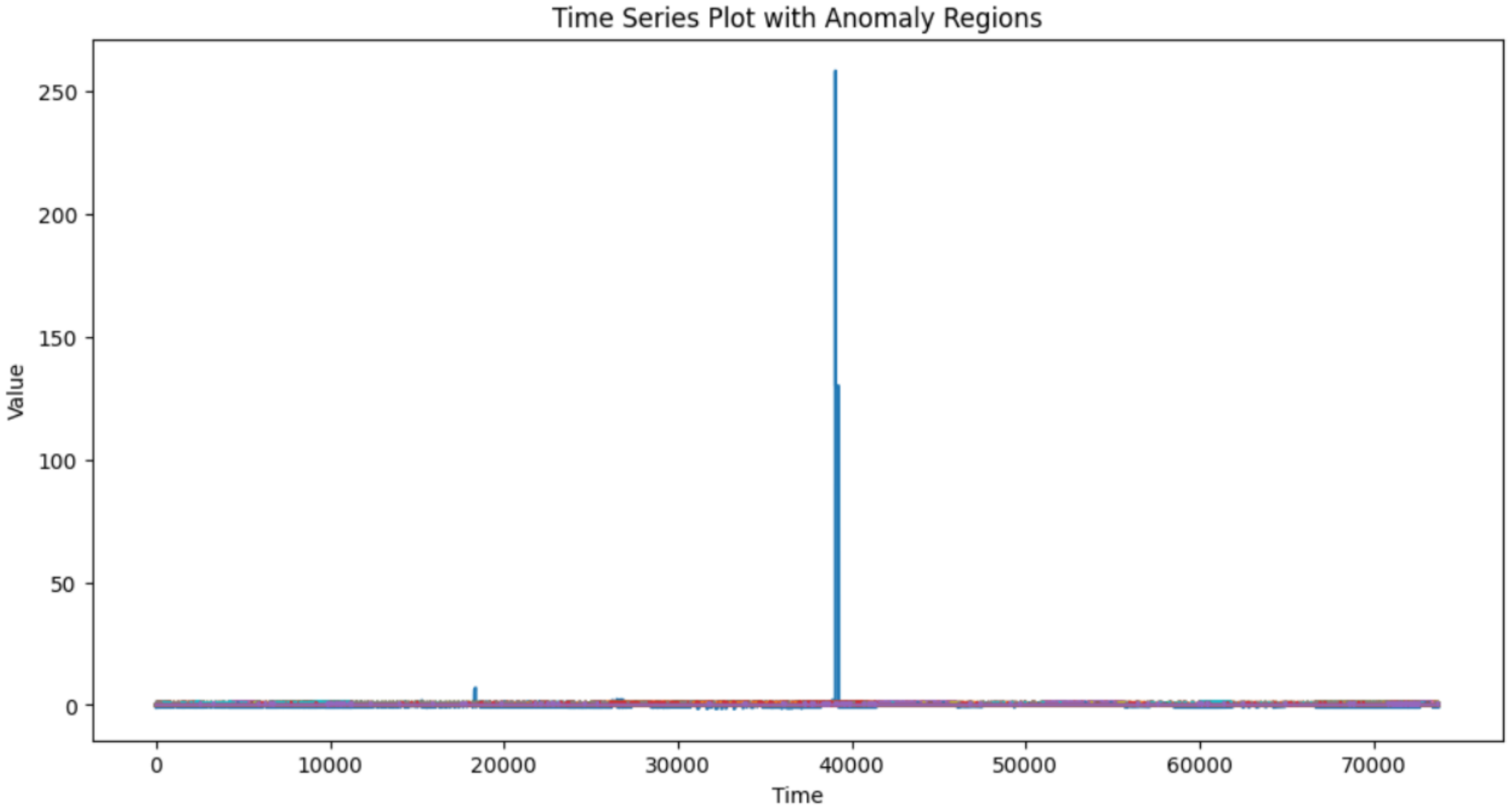
b) Draw time series plots with anomaly regions

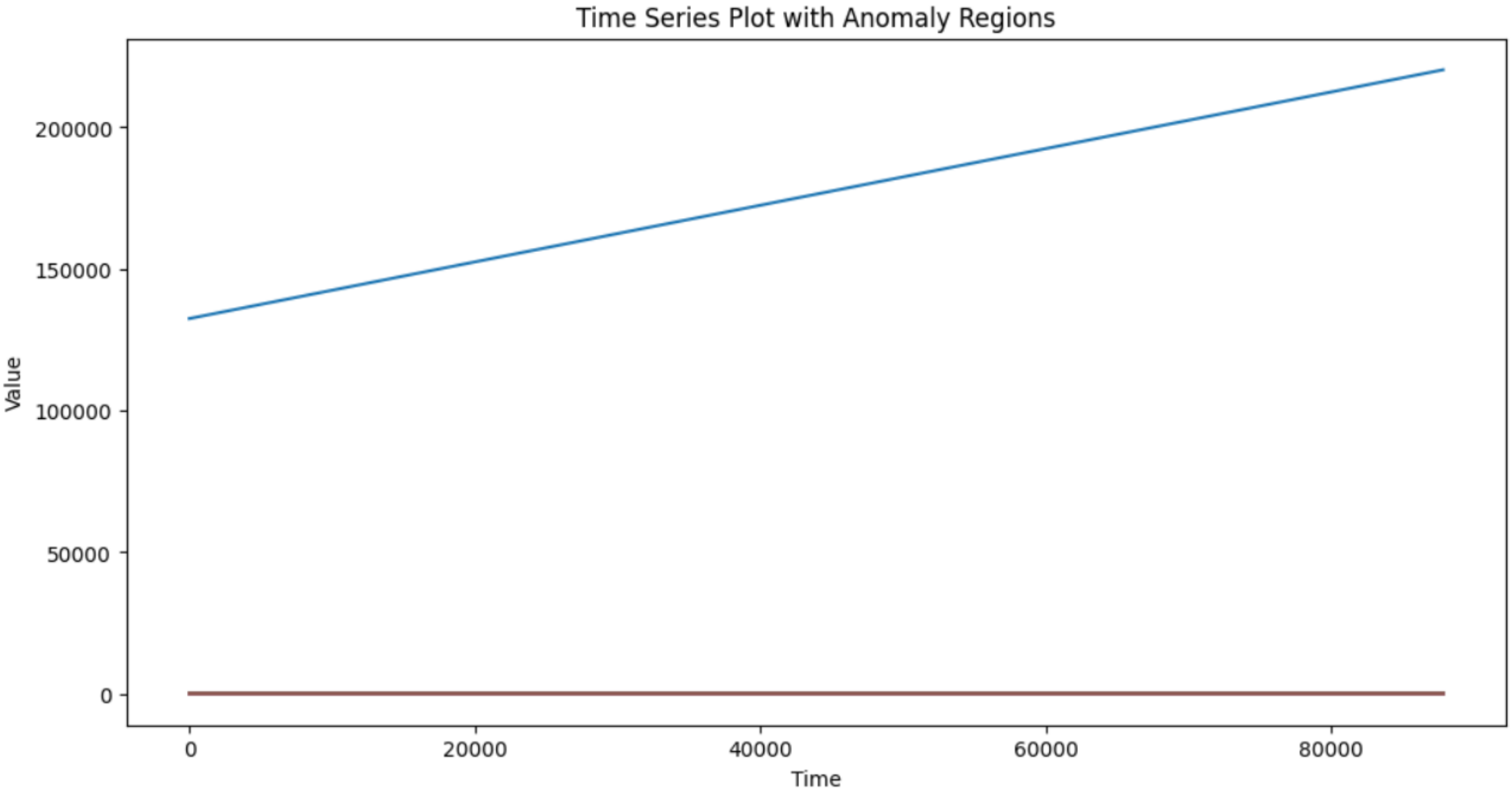
[4]:
```python
def draw_time_series_with_anomalies(test_data, label_data):
    try:
        plt.figure(figsize=(12, 6))
        plt.plot(test_data.index, test_data.values, label='Time Series Data')
        plt.xlabel('Time')
        plt.ylabel('Value')
        plt.title('Time Series Plot with Anomaly Regions')
        for index, row in label_data.iterrows():
            plt.axvspan(row['start_time'], row['end_time'], color='red', alpha=0.3, label='Anomaly Region')
        plt.legend()
        plt.show()
    except KeyError as e:
        print(f"Error: {e}. Make sure the label_data DataFrame contains columns 'start_time' and 'end_time'.")

# Example usage
for set_name, data_dict in data.items():
    test_data = data_dict["test_data"]
    label_data = data_dict["label_data"]
    draw_time_series_with_anomalies(test_data, label_data)
```

```
Error: 'start_time'. Make sure the label_data DataFrame contains columns 'start_time' and 'end_time'.
Error: 'start_time'. Make sure the label_data DataFrame contains columns 'start_time' and 'end_time'.
Error: 'start_time'. Make sure the label_data DataFrame contains columns 'start_time' and 'end_time'.
Error: 'start_time'. Make sure the label_data DataFrame contains columns 'start_time' and 'end_time'.
```




Time Series Plot with Anomaly Regions
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
Value
0
100000
200000
300000
400000
Time


Time Series Plot with Anomaly Regions
Value
Time
250
200
150
100
50
0
0
10000
20000
30000
40000
50000
60000
70000


Time Series Plot with Anomaly Regions
200000
150000
100000
50000
0
Value
0
20000
40000
60000
80000
Time

c) Perform EDA and find out root cause

[5]:
```
def perform_eda(test_data):
    print(test_data.describe())
    print(test_data.info())
    test_data.plot()
    plt.show()

if __name__ == "__main__":
    perform_eda(test_data)
```

d) Find out the variables which are the root cause for the anomaly

[6]:
```
def find_root_cause(test_data, label_data):
    try:

        correlation_matrix = test_data.corr()
        root_cause_variables = correlation_matrix['value'][abs(correlation_matrix['value']) > 0.5].index.tolist()
        print("Variables which are root cause for anomaly:", root_cause_variables)
    except KeyError as e:
        print("Error:", e)

if __name__ == "__main__":

    find_root_cause(test_data, label_data)
```

```
Error: 'value'
```

[ ]: