```
((venv_des) ) (base) ➜  DES_A3_W3 git:(main) ✗ /Users/kofibadu/Desktop/DES/DES_A3_W3/venv_des/bin/python /Users/kofibadu/Desktop/DES/DES_A3_W3/test_spx_analysis.py
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2666 entries, 0 to 2665
Data columns (total 6 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Date     2666 non-null   datetime64[ns]
 1   SPX      2666 non-null   float64
 2   GLD      2666 non-null   float64
 3   USO      2666 non-null   float64
 4   SLV      2666 non-null   float64
 5   EUR/USD  2666 non-null   float64
dtypes: datetime64[ns](1), float64(5)
memory usage: 125.1 KB
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2666 entries, 2015-01-02 to 2025-08-14
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   SPX      2666 non-null   float64
 1   GLD      2666 non-null   float64
 2   USO      2666 non-null   float64
 3   SLV      2666 non-null   float64
 4   EUR/USD  2666 non-null   float64
dtypes: float64(5)
memory usage: 125.0 KB
test_correlation_matrix_properties (__main__.TestDataAnalysis.test_correlation_matrix_properties)
Test properties of correlation matrix ... ok
test_descriptive_statistics (__main__.TestDataAnalysis.test_descriptive_statistics)
Test that descriptive statistics are reasonable ... ok
test_correlation_calculation (__main__.TestDataCleaning.test_correlation_calculation)
Test that correlation matrix can be calculated ... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    5 non-null      datetime64[ns]
 1   SPX     5 non-null      float64
 2   GLD     5 non-null      float64
 3   SLV     5 non-null      float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 292.0 bytes
ok
test_data_continuity (__main__.TestDataCleaning.test_data_continuity)
Test for reasonable data continuity (no extreme jumps) ... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    5 non-null      datetime64[ns]
```

```
test_dataframe_not_empty (__main__.TestDataCleaning.test_dataframe_not_empty)
Test that the loaded DataFrame is not empty ... ok
test_date_range_validity (__main__.TestDataCleaning.test_date_range_validity)
Test that dates fall within expected range ... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    5 non-null      datetime64[ns]
 1   SPX     5 non-null      float64
 2   GLD     5 non-null      float64
 3   SLV     5 non-null      float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 292.0 bytes
ok
test_duplicate_dates_handling (__main__.TestDataCleaning.test_duplicate_dates_handling)
Test handling of duplicate dates ... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    6 non-null      datetime64[ns]
 1   SPX     6 non-null      float64
 2   GLD     6 non-null      float64
 3   SLV     6 non-null      float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 324.0 bytes
ok
test_empty_dataframe_handling (__main__.TestDataCleaning.test_empty_dataframe_handling)
Test handling of empty DataFrame ... ok
test_missing_data_handling (__main__.TestDataCleaning.test_missing_data_handling)
Test behavior with missing data ... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    5 non-null      datetime64[ns]
 1   SPX     4 non-null      float64
 2   GLD     5 non-null      float64
 3   SLV     5 non-null      float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 292.0 bytes
ok
test_no_all_null_columns (__main__.TestDataCleaning.test_no_all_null_columns)
Test that no columns are entirely null ... ok
test_numeric_columns_have_valid_ranges (__main__.TestDataCleaning.test_numeric_columns_have_valid_ranges)
Test that numeric columns have reasonable value ranges ... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    5 non-null      datetime64[ns]
 1   SPX     5 non-null      float64
 2   GLD     5 non-null      float64
```

```
---  ------  --------------  -----
 0   Date    5 non-null      datetime64[ns]
 1   SPX     5 non-null      float64
 2   GLD     5 non-null      float64
 3   SLV     5 non-null      float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 292.0 bytes
ok
test_process_date_fn_datetime_conversion (__main__.TestDataCleaning.test_process_date_fn_datetime_conversion)
Test that Date column is properly converted to datetime and set as index ... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    5 non-null      datetime64[ns]
 1   SPX     5 non-null      float64
 2   GLD     5 non-null      float64
 3   SLV     5 non-null      float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 292.0 bytes
ok
test_process_date_fn_handles_invalid_dates (__main__.TestDataCleaning.test_process_date_fn_handles_invalid_dates)
Test handling of invalid date formats ... /Users/kofibadu/Desktop/DES/DES_A3_W3/spx_analysis.py:12: UserWarning: Could not infer format,
so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent and as-expected, please specify
a format.
  df['Date'] = pd.to_datetime(df['Date'])
ok
test_process_date_fn_preserves_data (__main__.TestDataCleaning.test_process_date_fn_preserves_data)
Test that processing preserves original data values ... <class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Date    5 non-null      datetime64[ns]
 1   SPX     5 non-null      float64
 2   GLD     5 non-null      float64
 3   SLV     5 non-null      float64
dtypes: datetime64[ns](1), float64(3)
memory usage: 292.0 bytes
ok
test_read_fn (__main__.TestDataCleaning.test_read_fn) ... ok
test_read_fn_columns_exist (__main__.TestDataCleaning.test_read_fn_columns_exist)
Test that required columns exist in the loaded data ... ok
test_read_fn_data_types (__main__.TestDataCleaning.test_read_fn_data_types)
Test that data types are as expected after reading ... ok
test_read_fn_file_not_found (__main__.TestDataCleaning.test_read_fn_file_not_found)
Test handling of non-existent file ... ok

----------------------------------------------------------------------
Ran 19 tests in 0.033s

OK
```