Time Series Relationship Analysis Framework

Overview

This document provides a generalized framework for analyzing relationships between two time series, derived from our SPY vs RETAILIRSA analysis. The framework can be applied to any pair of economic/financial series.

Framework Summary

```
┌─────────────────────────────────────────────────────────────────┐
│                     RELATIONSHIP ANALYSIS                       │
├─────────────────────────────────────────────────────────────────┤
│  0. QUALITATIVE ANALYSIS (NEW - DO FIRST)                       │
│     ├── Literature review (academic, professional, public)      │
│     ├── Market interpretation and usage                         │
│     └── Historical context and recent examples                  │
├─────────────────────────────────────────────────────────────────┤
│  1. DATA PREPARATION                                            │
│     ├── Load raw series                                         │
│     ├── Create derivatives (MoM, QoQ, YoY, Direction)           │
│     └── Align to common frequency and date range                │
├─────────────────────────────────────────────────────────────────┤
│  2. CORRELATION ANALYSIS                                        │
│     ├── Level correlation (often spurious)                      │
│     ├── Change correlation (more meaningful)                    │
│     └── Direction correlation                                   │
├─────────────────────────────────────────────────────────────────┤
│  3. LEAD-LAG ANALYSIS                                           │
│     ├── Cross-correlation at various lags                       │
│     ├── Identify if relationship is leading, lagging, or        │
│     │   contemporaneous                                         │
│     └── Statistical significance testing                        │
├─────────────────────────────────────────────────────────────────┤
│  4. CAUSALITY TESTING                                           │
│     ├── Granger causality (does X help predict Y?)              │
│     └── Bidirectional testing (X→Y and Y→X)                     │
├─────────────────────────────────────────────────────────────────┤
│  5. PREDICTIVE MODELING                                         │
│     ├── ML models with lagged features                          │
│     ├── Time series cross-validation                            │
│     └── Feature importance analysis                             │
├─────────────────────────────────────────────────────────────────┤
│  6. REGIME ANALYSIS                                             │
│     ├── Define regimes based on Series X                        │
│     ├── Measure Series Y performance in each regime             │
│     └── Statistical comparison of regime performance            │
├─────────────────────────────────────────────────────────────────┤
│  7. VISUALIZATION                                               │
│     ├── Time series overlay                                     │
```

```
│       ├── Scatter plots with regression                                │
│       ├── Regime background coloring                                   │
│       └── Annotated examples                                           │
├────────────────────────────────────────────────────────────────────────┤
│  8. DOCUMENTATION                                                      │
│       ├── Store in docs/analysis_reports/                              │
│       ├── Use consistent naming convention                             │
│       └── Embed visualizations with relative paths                     │
└────────────────────────────────────────────────────────────────────────┘
```

Step 0: Qualitative Analysis (REQUIRED - Do First)

Before any quantitative analysis, conduct a comprehensive qualitative review. This provides essential context for interpreting results and ensures your analysis aligns with market understanding.

0.1 What the Indicator Means

Document the fundamental definition and purpose:

Aspect	Description
Definition	What does the indicator measure?
Source	Who publishes it? (e.g., Federal Reserve, Census Bureau)
Frequency	Monthly, weekly, daily?
Release Timing	How long after the reference period?
Historical Range	Record high, record low, typical values
Revisions	Is data revised? How significantly?

0.2 Market Interpretation and Usage

Research how market participants interpret and use the indicator:

Key Questions to Answer:

1. Is it a leading, coincident, or lagging indicator?
2. What economic conditions does it signal?
3. How do institutional investors use it?
4. Is it part of any composite indices (e.g., LEI, CEI)?
5. What threshold levels are considered significant?

Example Template:

```
#### How Investors Use [Indicator Name]

1. **Primary Signal**: What does rising/falling indicate?
2. **Secondary Uses**: Risk assessment, regime identification, etc.
```

```
3. **Combined With**: What other indicators complement this one?
4. **Caution**: Known limitations or misinterpretations
```

0.3 Literature Review

Conduct a comprehensive review from multiple source types:

Academic Research

Search for peer-reviewed papers examining the indicator's relationship with asset returns:

- Use Google Scholar, SSRN, JSTOR
- Look for seminal papers (e.g., Fama for economic indicators)
- Note methodology and findings
- Cite with proper hyperlinks

Example Citations:

```
- **Fama (1981)** in "Stock Returns, Real Activity, Inflation, and Money"
  established the relationship between stock returns and industrial
production
  ([American Economic Review](https://www.aeaweb.org/))

- **Hong et al.** found industry portfolios can lead the market by up to
  two months ([NYU Stern Research](https://pages.stern.nyu.edu/))
```

Professional/Institutional Research

Include analysis from respected financial institutions:

- Federal Reserve research notes
- IMF working papers
- Major bank research (Goldman, JPM, etc.)
- Financial media analysis (CNBC, Bloomberg)

Public Domain Analysis

Capture recent market commentary and blog analysis:

- Financial blogs (Advisor Perspectives, Zero Hedge)
- Social media insights (FinTwit, LinkedIn)
- Industry publications

0.4 Recent Examples

Provide 2-3 recent real-world examples of the indicator in action:

Example Format:

```
**2020-2021 [Event Name]**: Description of what happened and how the
indicator behaved. [Source link](https://example.com) documented the
impact on markets.
```

0.5 Key Insights Summary Table

Consolidate findings in a reference table:

```
| Finding | Source | Implication for Analysis |
|---------|--------|-------------------------|
| Indicator is coincident | NBER | Cannot predict returns directly |
| Spikes 6 months before recession | Historical data | Useful for regime
detection |
| Relationship unstable over time | Academic paper | Be cautious with
backtests |
```

0.6 Limitations to Note

Document known limitations upfront:

1. **Data Issues**: Publication lag, revision frequency
2. **Coverage**: What the indicator doesn't capture
3. **Structural Changes**: How the indicator's meaning may have changed
4. **Recent Anomalies**: COVID, financial crises, etc.

Step 1: Data Preparation

1.1 Load and Align Series

```python
import pandas as pd
import numpy as np

def prepare_series(series_x, series_y, freq='ME'):
    """
    Align two series to common frequency and date range.

    Args:
        series_x: First time series (e.g., indicator)
        series_y: Second time series (e.g., asset price)
        freq: Target frequency ('ME' for month-end, 'W' for weekly, etc.)

    Returns:
        DataFrame with aligned series
    """
    # Resample to target frequency
    x = series_x.resample(freq).last()
```

```
    y = series_y.resample(freq).last()

    # Combine and align
    df = pd.DataFrame({'X': x, 'Y': y})
    df = df.dropna()

    return df
```

1.2 Create Derivative Series

For any series, create these standard derivatives:

Derivative	Formula	Purpose
MoM	`pct_change(1)`	Short-term momentum
QoQ	`pct_change(3)`	Medium-term trend
YoY	`pct_change(12)`	Long-term trend, seasonality-adjusted
Direction	`sign(value)` or `sign(change)`	Binary regime indicator
Z-Score	`(x - rolling_mean) / rolling_std`	Normalized level

```python
def create_derivatives(df, col, prefix):
    """Create standard derivative series."""
    series = df[col]

    # Percentage changes
    df[f'{prefix}_MoM'] = series.pct_change(1) * 100
    df[f'{prefix}_QoQ'] = series.pct_change(3) * 100
    df[f'{prefix}_YoY'] = series.pct_change(12) * 100

    # Direction indicators
    df[f'{prefix}_MoM_Dir'] = np.sign(df[f'{prefix}_MoM'])
    df[f'{prefix}_YoY_Dir'] = np.sign(df[f'{prefix}_YoY'])

    # Z-score (60-month rolling)
    rolling_mean = series.rolling(60).mean()
    rolling_std = series.rolling(60).std()
    df[f'{prefix}_ZScore'] = (series - rolling_mean) / rolling_std

    return df
```

Step 2: Correlation Analysis

2.1 Correlation Matrix

```python
from scipy import stats

def correlation_analysis(df, x_cols, y_cols):
    """
    Compute correlation matrix with significance testing.
    """
    results = []

    for x_col in x_cols:
        for y_col in y_cols:
            valid = df[[x_col, y_col]].dropna()
            if len(valid) < 30:
                continue

            corr, pval = stats.pearsonr(valid[x_col], valid[y_col])

            results.append({
                'X': x_col,
                'Y': y_col,
                'correlation': corr,
                'p_value': pval,
                'significant': pval < 0.05,
                'n_obs': len(valid)
            })

    return pd.DataFrame(results)
```

2.2 Interpretation Guidelines

Correlation Type	Typical Finding	Interpretation
Level vs Level	Often high	Usually **spurious** (common trends)
Change vs Change	Lower but meaningful	**Contemporaneous** relationship
Direction vs Direction	Moderate	**Regime** relationship

Key Insight from SPY/RETAILIRSA: The -0.77 level correlation was spurious (opposite trends). The meaningful relationship was the -0.13 to -0.34 change correlation.

Step 3: Lead-Lag Analysis

3.1 Cross-Correlation Function

```python
def lead_lag_analysis(df, x_col, y_col, max_lag=12):
    """
    Test correlations at various leads and lags.

    Negative lag = X leads Y (X at t predicts Y at t+|lag|)
    Positive lag = Y leads X
```

```
    Zero lag = Contemporaneous
    """
    results = []

    for lag in range(-max_lag, max_lag + 1):
        if lag < 0:
            x = df[x_col].shift(-lag)  # X leads
            y = df[y_col]
        else:
            x = df[x_col]
            y = df[y_col].shift(-lag)  # Y leads

        valid = pd.DataFrame({'x': x, 'y': y}).dropna()
        if len(valid) < 30:
            continue

        corr, pval = stats.pearsonr(valid['x'], valid['y'])

        results.append({
            'lag': lag,
            'correlation': corr,
            'p_value': pval,
            'abs_corr': abs(corr),
            'interpretation': 'X leads Y' if lag < 0 else 'Y leads X' if
lag > 0 else 'Contemporaneous'
        })

    return pd.DataFrame(results)
```

3.2 Interpretation

Finding	Meaning	Implication
Peak at lag=0	Contemporaneous	No predictive value, move together
Peak at lag<0	X leads Y	X may predict Y (test further)
Peak at lag>0	Y leads X	Y may predict X
Flat across lags	No relationship	Series are independent

Key Insight from SPY/RETAILIRSA: Peak at lag=0 confirmed the relationship is contemporaneous, not predictive.

Step 4: Causality Testing

4.1 Granger Causality

```
from statsmodels.tsa.stattools import grangercausalitytests

def granger_test(df, x_col, y_col, max_lag=6):
```

```
    """
    Test if X Granger-causes Y (and vice versa).

    Granger causality: X helps predict Y beyond Y's own history.
    """
    data = df[[y_col, x_col]].dropna()

    results = []

    # Test X -> Y
    try:
        gc = grangercausalitytests(data[[y_col, x_col]], maxlag=max_lag,
verbose=False)
        for lag in range(1, max_lag + 1):
            f_stat = gc[lag][0]['ssr_ftest'][0]
            p_val = gc[lag][0]['ssr_ftest'][1]
            results.append({
                'direction': f'{x_col} -> {y_col}',
                'lag': lag,
                'f_statistic': f_stat,
                'p_value': p_val,
                'significant': p_val < 0.05
            })
    except:
        pass

    # Test Y -> X
    try:
        gc = grangercausalitytests(data[[x_col, y_col]], maxlag=max_lag,
verbose=False)
        for lag in range(1, max_lag + 1):
            f_stat = gc[lag][0]['ssr_ftest'][0]
            p_val = gc[lag][0]['ssr_ftest'][1]
            results.append({
                'direction': f'{y_col} -> {x_col}',
                'lag': lag,
                'f_statistic': f_stat,
                'p_value': p_val,
                'significant': p_val < 0.05
            })
    except:
        pass

    return pd.DataFrame(results)
```

4.2 Interpretation

Result	Meaning
X → Y significant	X helps predict Y (potential leading indicator)
Y → X significant	Y helps predict X (reverse causality)

Result	Meaning
Both significant	Bidirectional feedback loop
Neither significant	No predictive relationship

Key Insight from SPY/RETAILIRSA: No Granger causality in either direction (all p > 0.05).

Step 5: Predictive Modeling

5.1 ML Prediction Framework

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import TimeSeriesSplit
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score

def predictive_model_test(df, feature_cols, target_col, lags=[1, 3, 6,
12]):
    """
    Test if lagged features can predict target using ML.
    """
    # Create lagged features
    X_cols = []
    for lag in lags:
        for col in feature_cols:
            lag_col = f'{col}_lag{lag}'
            df[lag_col] = df[col].shift(lag)
            X_cols.append(lag_col)

    # Prepare data
    df_model = df[X_cols + [target_col]].dropna()
    X = df_model[X_cols]
    y = df_model[target_col]

    # Time series cross-validation
    tscv = TimeSeriesSplit(n_splits=5)
    scores = []

    for train_idx, test_idx in tscv.split(X):
        X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
        y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]

        scaler = StandardScaler()
        X_train_s = scaler.fit_transform(X_train)
        X_test_s = scaler.transform(X_test)

        model = RandomForestRegressor(n_estimators=100, max_depth=5,
random_state=42)
        model.fit(X_train_s, y_train)
        y_pred = model.predict(X_test_s)
```

```
        scores.append(r2_score(y_test, y_pred))

    return {
        'mean_r2': np.mean(scores),
        'std_r2': np.std(scores),
        'predictive': np.mean(scores) > 0
    }
```

5.2 Interpretation

R^2 Value	Interpretation
$R^2 > 0.1$	Weak but potentially useful predictive power
$R^2 \approx 0$	No predictive value
$R^2 < 0$	**Worse than guessing the mean** (no relationship)

Key Insight from SPY/RETAILIRSA: All models had negative R^2, confirming no predictive power.

Step 6: Regime Analysis

6.1 Define Regimes

```
def define_regimes(df, x_col, method='median'):
    """
    Create regime indicators based on Series X.

    Methods:
    - 'median': Above/below median
    - 'direction': Positive/negative change
    - 'zscore': High/normal/low based on z-score
    """
    if method == 'median':
        median = df[x_col].median()
        df['Regime'] = np.where(df[x_col] > median, 'High', 'Low')

    elif method == 'direction':
        df['Regime'] = np.where(df[x_col] > 0, 'Rising', 'Falling')

    elif method == 'zscore':
        zscore = (df[x_col] - df[x_col].rolling(60).mean()) /
df[x_col].rolling(60).std()
        df['Regime'] = pd.cut(zscore, bins=[-np.inf, -1, 1, np.inf],
                              labels=['Low', 'Normal', 'High'])

    return df
```

6.2 Compare Performance by Regime

```
def regime_performance(df, regime_col, target_col):
    """
    Calculate target series statistics by regime.
    """
    results = []

    for regime in df[regime_col].dropna().unique():
        subset = df[df[regime_col] == regime][target_col].dropna()

        if len(subset) < 10:
            continue

        results.append({
            'regime': regime,
            'n_periods': len(subset),
            'mean': subset.mean(),
            'std': subset.std(),
            'sharpe': subset.mean() / subset.std() * np.sqrt(12) if
subset.std() > 0 else 0,
            'positive_pct': (subset > 0).mean() * 100,
            'median': subset.median()
        })

    return pd.DataFrame(results)
```

6.3 Interpretation

Even without predictive power, regime analysis is valuable:

Regime Finding	Actionable Insight
Regime A has higher Sharpe	**Filter**: Favor exposure during Regime A
Regimes have similar returns	No value as filter
One regime has negative returns	**Avoid** during that regime

Key Insight from SPY/RETAILIRSA: Falling inventories regime had Sharpe 0.98 vs Rising 0.52 - useful as a **filter** even though not predictive.

Step 7: Visualization

7.1 Essential Plots

1. **Time Series Overlay**: Both series on same chart (dual y-axis if needed)
2. **Scatter Plot**: X vs Y with regression line and correlation
3. **Lead-Lag Plot**: Correlation vs lag number
4. **Regime Background**: Target series with regime colors as background

5. **Box Plot by Regime**: Distribution of target in each regime

7.1.1 Embedding Images in Documentation

Always embed generated visualizations in analysis documents using markdown syntax:

```
![Alt text description](../data/filename.png)

*Caption explaining what the visualization shows*
```

Best Practices for Image Embedding:

Practice	Description
Use relative paths	`../data/image.png` from docs folder
Add alt text	Describes the image for accessibility
Include caption	Explains colors, axes, key takeaways
Save to data folder	Keep visualizations with data artifacts
Use descriptive filenames	`spy_ip_regime_background.png` not `fig1.png`

Standard Image Naming Convention:

```
{target}_{indicator}_{plot_type}.png

Examples:
- spy_ip_regime_background.png      # Full timeline with regime colors
- spy_ip_regime_examples.png        # Validated example subplots
- spy_retailirsa_correlation.png    # Scatter plot with correlation
- spy_retailirsa_leadlag.png        # Lead-lag analysis plot
```

Example Document Structure with Embedded Images:

```
## Visualizations

### Full Timeline with Regime Background

![SPY Price with IP Regime Background]
(../data/spy_ip_regime_background.png)

*Green = IP Rising (YoY > 0), Pink = IP Falling (YoY < 0), Gray =
Recession*

### Validated Examples

![SPY vs IP Regime Examples](../data/spy_ip_regime_examples.png)
```

```
The four panels show:
1. Bull market with IP Rising regime
2. Recession with negative returns
3. Recovery period with IP Rising
4. Counter-example: positive returns despite falling IP
```

Benefits of Embedded Images:

1. **Self-contained documentation** - Readers see visuals alongside analysis
2. **Version control** - Images tracked with code in git
3. **Reproducibility** - Clear link between analysis and output
4. **PDF export** - Most markdown renderers include images in exports

7.2 Annotated Examples - CRITICAL PRACTICE

IMPORTANT: Example Selection and Validation

When presenting visual examples, you MUST follow this validation process:

Step 1: State Your Proposition

Clearly define what you're trying to demonstrate:

- "Regime A is associated with positive returns"
- "Rising X leads to falling Y"
- "Indicator spikes precede market crashes"

Step 2: Find Supporting Examples

Search for periods where the proposition holds true:

```python
def find_supporting_examples(df, regime_col, target_col, proposition):
    """
    Find time periods that support the stated proposition.

    Example: Find periods where 'Falling Inv' regime had strong positive returns.
    """
    results = []

    # Group by year or custom periods
    for year in df.index.year.unique():
        subset = df[df.index.year == year]

        # Count regime months
        regime_counts = subset[regime_col].value_counts()

        # Calculate target performance
        target_perf = subset[target_col].sum()  # or mean, or total return
```

```
        results.append({
            'period': year,
            'dominant_regime': regime_counts.idxmax(),
            'regime_pct': regime_counts.max() / len(subset) * 100,
            'target_performance': target_perf
        })

    # Filter for examples that SUPPORT the proposition
    # e.g., periods where dominant regime matches expected outcome
    return pd.DataFrame(results)
```

Step 3: Validate Visual-Description Alignment

Before finalizing any visualization, verify:

Check	Question	Action if Fails
Visual Match	Does the plot show what the title claims?	Change title OR choose different period
Data Match	Do the numbers support the description?	Verify with actual data counts
Proposition Support	Does this example demonstrate the claim?	Find a better example

```
def validate_example(df_subset, regime_col, expected_regime, target_col,
expected_direction):
    """
    Validate that an example supports the proposition.

    Returns True if:
    1. The expected regime dominates the period
    2. The target moved in the expected direction
    """
    # Check regime dominance
    regime_counts = df_subset[regime_col].value_counts()
    dominant = regime_counts.idxmax()
    dominant_pct = regime_counts.max() / len(df_subset) * 100

    # Check target direction
    target_change = df_subset[target_col].iloc[-1] -
df_subset[target_col].iloc[0]
    actual_direction = 'positive' if target_change > 0 else 'negative'

    # Validation
    regime_valid = (dominant == expected_regime) and (dominant_pct > 50)
    direction_valid = (actual_direction == expected_direction)

    print(f"Regime: {dominant} ({dominant_pct:.0f}%) - Expected:
```

```
{expected_regime} -> {'✓' if regime_valid else '✗'}")
    print(f"Target: {actual_direction} - Expected: {expected_direction} ->
{'✓' if direction_valid else '✗'}")

    return regime_valid and direction_valid
```

Step 4: Write Accurate Descriptions

DO: Match description to what the visual actually shows

```
Title: "2009-2010: Falling Inventories = Bullish"
Visual: Green background (falling inv) dominates, SPY rises
Data: 17 falling months, 1 rising month, SPY +63.8%
✓ Description matches visual matches data
```

DON'T: Claim something the visual contradicts

```
Title: "2014-2016: Green dominates = Strong SPY"
Visual: Pink background (rising inv) dominates
Data: 31 rising months, 5 falling months
✗ Description contradicts visual - WRONG
```

Example Validation Checklist

Before including any example in a report:

- [ ] Counted actual regime months in the period
- [ ] Verified dominant regime matches what visual shows
- [ ] Confirmed target behavior matches proposition
- [ ] Title accurately describes what viewer will see
- [ ] Example genuinely supports (not contradicts) the finding

Anti-Pattern Warning

Never assume a period supports your thesis without checking the data.

Common mistakes:

1. Assuming bull markets always have "bullish" regime → may have opposite
2. Picking arbitrary periods without verifying regime composition
3. Writing descriptions based on expectations rather than actual data
4. Not validating that visual colors match the verbal description

---

Decision Framework

After completing the analysis, use this decision tree:

```
        ┌─────────────────────────┐
        │  Correlation Analysis   │
        └───────────┬─────────────┘
                    │
        ┌───────────▼─────────────┐
        │  Is correlation > 0.3?  │
        └───────────┬─────────────┘
               No   │   Yes
        ┌───────────┴───────────┐
        ▼                       ▼
Weak/No relationship    Check if spurious
        │               (level vs change)
        │                       │
        │               ┌───────▼───────┐
        │               │  Lead-Lag     │
        │               │  Analysis     │
        │               └───────┬───────┘
        │                       │
        │               ┌───────▼───────┐
        │               │  Peak at      │
        │               │  lag ≠ 0?     │
        │               └───────┬───────┘
        │                  No   │   Yes
        │               ┌───────┴───────┐
        │               ▼               ▼
        │       Contemporaneous    Leading indicator
        │       (no prediction)    (test causality)
        │               │               │
        │               │       ┌───────▼───────┐
        │               │       │ Granger test  │
        │               │       │ significant?  │
        │               │       └───────┬───────┘
        │               │          No   │   Yes
        │               │       ┌───────┴───────┐
        │               │       ▼               ▼
        │               │   No causality   Predictive!
        │               │       │          Build model
        └───────────────┼───────┘
                        │
                ┌───────▼───────┐
                │ Regime        │
                │ Analysis      │
                └───────┬───────┘
                        │
                ┌───────▼───────┐
                │ Regime        │
                │ differences?  │
                └───────┬───────┘
                   No   │   Yes
                ┌───────┴───────┐
                ▼               ▼
```

```
                  No value        Use as FILTER
                                  (not predictor)
```

Summary: What We Learned from SPY/RETAILIRSA

Analysis Step	Finding	Generalization
Level Correlation	-0.77 (spurious)	Always check if due to common trends
Change Correlation	-0.13 to -0.34	Change correlations more meaningful
Lead-Lag	Peak at lag=0	Contemporaneous = no prediction
Granger Causality	Not significant	Confirms no predictive power
ML Prediction	Negative R^2	Even ML can't find signal
Regime Analysis	**Sharpe 0.98 vs 0.52**	**Useful as filter, not predictor**

Key Takeaway

A series can be valuable without being predictive.

RETAILIRSA cannot predict SPY returns, but it provides useful regime context:

- **Falling inventories** = favorable environment for stocks
- **Rising inventories** = cautious environment
- **Recession** = avoid

Use it as a **filter** (adjust exposure based on regime) rather than a **signal** (time entries/exits).

Step 8: Documentation and Organization

8.1 Folder Structure

All analysis reports MUST be stored in a dedicated folder:

```
docs/
├── analysis_reports/              # All indicator analysis documents
│   ├── spy_retailirsa_analysis.md
│   ├── spy_industrial_production_analysis.md
│   └── spy_[indicator]_analysis.md
├── 11_time_series_relationship_framework.md  # This framework
└── [other docs...]
```

8.2 Naming Convention

Use consistent naming for analysis reports and associated files:

Report Files:

```
docs/analysis_reports/{target}_{indicator}_analysis.md

Examples:
- spy_retailirsa_analysis.md
- spy_industrial_production_analysis.md
- spy_cpi_analysis.md
```

Visualization Files:

```
data/{target}_{indicator}_{plot_type}.png

Examples:
- spy_ip_regime_background.png      # Full timeline with regime colors
- spy_ip_regime_examples.png        # Validated example subplots
- spy_retailirsa_correlation.png    # Scatter plot with correlation
- spy_retailirsa_leadlag.png        # Lead-lag analysis plot
```

8.3 Document Structure

Every analysis report MUST follow this structure:

```
# {Target} vs {Indicator} Analysis

## Overview
- Brief description
- Data period

---

## Qualitative Analysis: Understanding {Indicator}
### What is {Indicator}?
### Market Interpretation and Usage
### Key Insights from Literature
### Limitations as a Stock Market Indicator

---

## Key Findings Summary
### 1. Level Relationship
### 2. Change Relationship
### 3. Predictive Power
### 4. Regime-Based Insights

## Detailed Analysis
### Correlation Matrix
### Lead-Lag Analysis
```

```
### Granger Causality Tests
### ML Predictive Model Results
### Regime Analysis

## Visualizations
![Embedded images with relative paths](../../data/image.png)

## Validated Visual Examples

## Economic Interpretation

## Practical Applications

## Files Created

## Conclusion
```

8.4 Image Embedding

Embed all visualizations in the analysis document:

```
## Visualizations

### Full Timeline with Regime Background

![SPY Price with {Indicator} Regime Background]
(../../data/{target}_{indicator}_regime_background.png)

*Color legend: Green = [regime A], Pink = [regime B], Gray = Recession*

### Validated Examples

![{Target} vs {Indicator} Regime Examples]
(../../data/{target}_{indicator}_regime_examples.png)
```

Note: Use `../../data/` path from `docs/analysis_reports/` subfolder.

8.5 Citation Requirements

All qualitative analysis sections MUST include:

1. **Hyperlinked citations** to source material
2. **Mix of source types**: Academic, professional, public
3. **Recent examples** with dates and context
4. **Summary table** of key findings

Example Citation Format:

```
```

```
According to the [Federal Reserve's analysis]
(https://www.federalreserve.gov/...),
this indicator is considered a coincident measure of economic activity.

Research by [Hong et al.](https://pages.stern.nyu.edu/~rengle/...) found
that
industry portfolios can lead the market by up to two months.
```

Reusable Code Location

All analysis code is available in:

- `src/ml/retail_spy_analysis/relationship_analysis.py` - Core analysis functions
- `src/ml/retail_spy_analysis/visualizations.py` - Plotting functions

These can be adapted for any pair of time series.

Complete Analysis Reports

Completed analyses following this framework:

- SPY vs RETAILIRSA Analysis
- SPY vs Industrial Production Analysis