Social Upheaval Composite Index

A quantitative framework for measuring societal upheaval in the United States across decades, designed to objectively assess periods of political instability, institutional crisis, and social fragmentation.

Project Overview

This project develops a multi-dimensional composite index that measures social upheaval across five key components:

- **Political Violence & Instability** (25%)
- **Institutional Trust Erosion** (20%)
- **Economic Stress & Inequality** (15%)
- **External Threats & Conflicts** (20%)
- **Social Fragmentation** (20%)

The index provides a standardized 0-100 scale for comparing upheaval levels across different decades in American history.

Mathematical Framework

Composite Index Formula

$$CUI_d = w_pv \times PV_d + w_it \times IT_d + w_es \times ES_d + w_et \times ET_d + w_sf \times SF_d$$

Where:

- CUI_d = Composite Upheaval Index for decade d
- Component weights: Political Violence (25%), Institutional Trust (20%), Economic Stress (15%), External Threats (20%), Social Fragmentation (20%)

Component Calculations

Each component uses weighted event counts with caps to prevent outlier dominance:

Political Violence:

Institutional Trust:

$$PV_d = min(20A_d + 15T_d + 10R_d + 5P_d + 8G_d, 100)$$

$$IT_d = min(25S_d + 10C_d + 15I_d + 12E_d, 100)$$

Economic Stress:

$$ES_d = min(min(M_d \times U_d, 40) + max(\Delta G_d \times 30, 0) + 15F_d, 100)$$

See `/docs/mathematical_framework.md` *for complete formulation.*

Project Structure

```
social_upheaval_index/
├── data/
│   ├── raw/                # Original data sources
│   ├── processed/          # Cleaned, standardized data
│   └── validation/         # Historical benchmark data
├── src/
│   ├── collectors/         # Data collection scripts
│   ├── calculators/        # Index computation modules
│   ├── validators/         # Validation and testing
│   └── utils/              # Helper functions
├── notebooks/
│   ├── data_collection.ipynb   # Interactive data gathering
│   ├── validation.ipynb        # Index validation analysis
│   └── optimization.ipynb      # Weight optimization
├── tests/
├── docs/
├── requirements.txt
└── README.md
```

Installation

1. **Clone the repository**

```
git clone <repository-url>
cd social_upheaval_index
```

1. **Create virtual environment**

```
python -m venv upheaval_env
source upheaval_env/bin/activate  # On Windows: upheaval_env\Scripts\activ
ate
```

1. **Install dependencies**

```
pip install -r requirements.txt
```

Dependencies

Core Analysis

- `pandas>=1.5.0` - Data manipulation and analysis
- `numpy>=1.21.0` - Numerical computing
- `scipy>=1.9.0` - Statistical functions and optimization

Data Collection

- `requests>=2.28.0` - HTTP requests for web data
- `beautifulsoup4>=4.11.0` - Web scraping
- `lxml>=4.9.0` - XML/HTML parser

Visualization & Analysis

- `matplotlib>=3.6.0` - Plotting and visualization
- `seaborn>=0.11.0` - Statistical visualization

- `plotly>=5.10.0` - Interactive plots

Development & Testing

- `jupyter>=1.0.0` - Interactive notebooks
- `pytest>=7.0.0` - Testing framework
- `black>=22.0.0` - Code formatting

Quick Start

1. Calculate Upheaval Index for a Decade

```
from src.calculators.composite_index import CompositeUpheavalIndex

# Initialize calculator
calculator = CompositeUpheavalIndex()

# Example: 1960s data
decade_1960s = {
    'political_violence': {
        'assassinations_major': 4,  # JFK, MLK, RFK, Malcolm X
        'terrorist_attacks_domestic': 2,
        'riots_major': 4,  # Watts, Newark, Detroit, post-MLK
        'protests_large': 5,
        'government_crises': 1
    },
    'institutional_trust': {
        'major_scandals': 1,
        'supreme_court_controversial': 2,
        'intelligence_scandals': 1,
        'electoral_controversies': 0
    }
    # ... other components
}
```

```
# Calculate composite score
result = calculator.calculate(decade_1960s)
print(f"1960s Upheaval Index: {result['composite_score']:.2f}")
print(f"Component breakdown: {result['components']}")
```

2. Validate Against Historical Consensus

```
from src.validators.historical_validator import HistoricalValidator

validator = HistoricalValidator()
correlation = validator.test_historical_ranking()
print(f"Historical validation correlation: {correlation:.3f}")
```

3. Optimize Component Weights

```
from src.validators.weight_optimizer import WeightOptimizer

optimizer = WeightOptimizer()
optimal_weights = optimizer.find_optimal_weights()
print(f"Optimized weights: {optimal_weights}")
```

Data Collection

Current Test Decades

The project focuses on 5 decades for initial validation:

- **1950s** (Expected LOW) - Post-war stability
- **1960s** (Expected HIGH) - Assassinations, civil rights, Vietnam
- **1970s** (Expected HIGH) - Watergate, economic crisis
- **1990s** (Expected MEDIUM) - Economic boom but some scandals
- **2010s** (Expected MEDIUM-HIGH) - Financial crisis aftermath, polarization

Data Sources

- **Primary**: Government databases (FBI, BLS, NBER, Census)
- **Secondary**: Academic datasets, historical archives
- **Validation**: Wikipedia, news archives, scholarly sources

See `/docs/data_sources.md` *for complete source documentation.*

Validation Framework

1. Historical Consensus Test

Compares calculated rankings against expert historical consensus on most turbulent decades.

2. Component Sensitivity Analysis

Tests impact of removing individual components on overall rankings.

3. Weight Optimization

Uses correlation maximization to find optimal component weightings.

4. Robustness Testing

Validates stability across different aggregation methods and data sources.

Usage Examples

Basic Analysis

```
# Load processed data for multiple decades
from src.utils.data_loader import load_decade_data

decades_data = load_decade_data(['1960s', '1970s', '1950s'])
calculator = CompositeUpheavalIndex()

results = {}
for decade, data in decades_data.items():
    results[decade] = calculator.calculate(data)
```

```
# Rank decades by upheaval level
ranked = sorted(results.items(), key=lambda x: x[1]['composite_score'], revers
e=True)
for decade, score in ranked:
    print(f"{decade}: {score['composite_score']:.2f}")
```

Comparative Analysis

```
import matplotlib.pyplot as plt
from src.visualizers.upheaval_plots import create_comparison_chart

# Compare component contributions across decades
create_comparison_chart(results)
plt.show()
```
