Time Series Analysis and Forecasting

Chapter 10: Comprehensive Review

Complete Analysis with Real Data

Outline

Course Overview: Methods Covered

Classical Methods

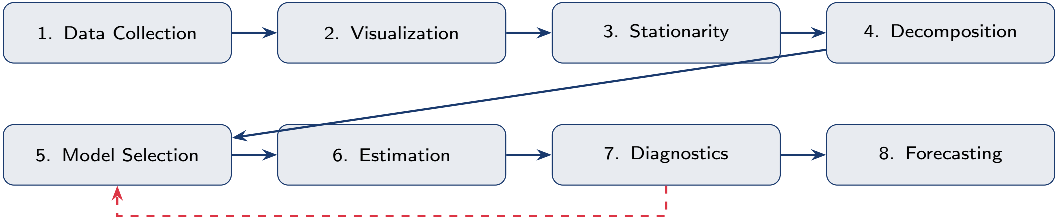
- Ch 1: Time Series Fundamentals
- Ch 2: ARMA Models
- Ch 3: ARIMA Models
- Ch 4: SARIMA Models
- Ch 5: GARCH Models

Advanced Methods

- Ch 6: VAR & Granger Causality
- Ch 7: Cointegration & VECM
- Ch 8: Modern Extensions
- Ch 9: Prophet & TBATS

Today: Apply ALL to Real Data!

The Complete Analysis Workflow



Key Principle

Model diagnostics may require returning to model selection (iterative process)

Real Datasets for This Chapter

Bitcoin

- Daily 2019-2024
- Volatility clustering
- ARIMA + GARCH

Sunspots

- Yearly 1900-2023
- 11-year cycle
- Fourier terms

- Monthly 2010-2023
- COVID-19 shock
- Prophet

Economic VAR

- Quarterly 2000-2023
- GDP, Inflation, etc.
- Multivariate VAR

Key Methodology: Train / Validation / Test Split

Training 70% Fit model parameters	**Validation 15%** Select hyperparams	**Test 15%** Final evaluation

Why This Matters:

- Training: Estimate model parameters
- Validation: Compare models, tune hyperparameters
- Test: Unbiased final performance (held out!)

Critical Rule

NEVER look at test data during model selection!

For time series: **NEVER** shuffle data — preserve temporal order.

Bitcoin: Data and Objective

Data: Bitcoin daily returns (2019–2024)

- Source: Yahoo Finance (BTC-USD)
- 1,826 observations
- Returns $= 100 \times \ln(P_t/P_{t-1})$

Objective: Forecast volatility (not returns!)

Why volatility?

- Returns are nearly unpredictable
- Volatility shows strong persistence
- Critical for risk management (VaR)

Data Split:

Set	Days	Period
Training	1,278	2019–2022
Validation	274	2022–2023
Test	274	2023–2024

Key Statistics

Mean return: 0.12%/day
Std dev: 3.8%/day
Kurtosis: 8.2 (fat tails!)

Bitcoin: Model Selection on Validation Set

Step 1: Fit GARCH variants on Training data
Step 2: Compare volatility forecasts on Validation set

Model	AIC	BIC	**Val MAE**
GARCH(1,1)	8542	8563	2.34
GARCH(2,1)	8540	8567	2.31
GJR-GARCH(1,1)	**8531**	**8558**	**2.18**
EGARCH(1,1)	8538	8565	2.25

$\Rightarrow$ Best: GJR-GARCH (captures asymmetry)

GJR-GARCH Model:

$$\sigma_t^2 = \omega + (\alpha + \gamma I_{t-1})\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

Where $I_{t-1} = 1$ if $\varepsilon_{t-1} < 0$

Leverage Effect

$\gamma > 0$: Negative returns increase volatility more than positive returns (fear vs greed)

Bitcoin: Final Test Set Evaluation

Step 3: Refit GJR-GARCH on Training+Validation, evaluate on Test

Test Set Results:

Metric	Value
Volatility MAE	2.21
Volatility RMSE	3.45

GJR-GARCH Parameters:

- $\alpha = 0.05$ (ARCH effect)
- $\gamma = 0.08$ (leverage effect)
- $\beta = 0.89$ (persistence)
- $\alpha + \gamma/2 + \beta = 0.98$

Interpretation:

- High persistence (≈ 0.98)
- Leverage effect significant ($\gamma > 0$)
- Volatility is predictable!

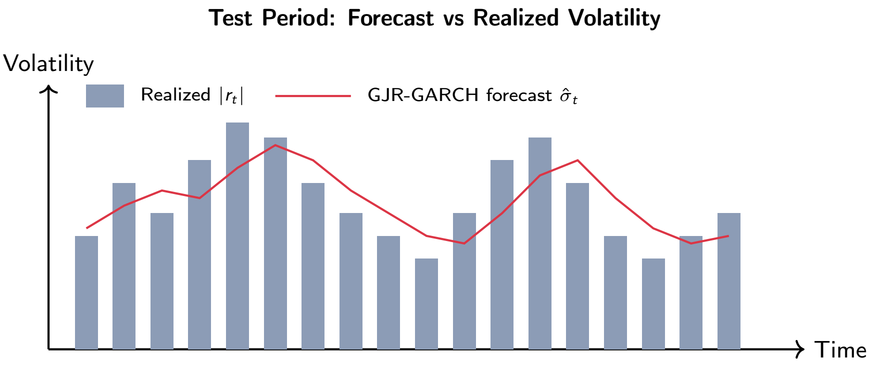
Practical Application

1-day ahead VaR (99%):

$$\text{VaR} = -2.33 \times \hat{\sigma}_{t+1}$$

If $\hat{\sigma} = 4\%$, then VaR $= -9.3\%$

Bitcoin: Volatility Forecast Visualization



- Model captures volatility clusters (high vol follows high vol)
- Forecast adapts to changing market conditions
- Test MAE $= 2.21$ (good for daily volatility forecasting)

Bitcoin: Summary

Methodology:

1. Split data: 70/15/15
2. Compare GARCH variants on validation
3. Select best model (GJR-GARCH)
4. Final evaluation on test set

Key Finding:

Returns are unpredictable, but volatility is forecastable!

Results:

Best Model	GJR-GARCH(1,1)
Validation MAE	2.18
Test MAE	2.21
Key insight	Leverage effect

Practical Use

- Value-at-Risk calculation
- Position sizing
- Options pricing

Sunspots: Data and Objective

Data: Yearly sunspot numbers (1900–2023)

- Source: statsmodels (Wolfer dataset)
- 124 yearly observations
- Famous 11-year Schwabe cycle

Objective: Forecast sunspot activity

Challenge:

- Standard SARIMA with $m = 11$ is complex
- Solution: Fourier terms as regressors

Data Split:

Set	Years	Period
Training	87	1900–1986
Validation	18	1987–2004
Test	19	2005–2023

Key Statistics

Mean: 76.4
Std dev: 56.8
Cycle period: ≈11 years

Sunspots: Fourier Terms for Long Seasonality

Idea: Capture 11-year cycle with sine/cosine regressors

$$y_t = \mu + \sum_{k=1}^{K}\left[a_k \sin\left(\frac{2\pi kt}{11}\right) + b_k \cos\left(\frac{2\pi kt}{11}\right)\right] + \text{ARMA errors}$$

How many harmonics (K)?

- K=1: Basic cycle shape
- K=2: Sharper peaks/troughs
- K=3,4: More detail
- Too many $\Rightarrow$ overfitting

Select K using validation set!

Why Fourier?

- Only 2K parameters (not 11)
- No seasonal differencing needed
- Works for any period length
- Combines with ARIMA naturally

Sunspots: Model Selection on Validation Set

Model: ARIMA(2,0,1) + K Fourier harmonics

K	AIC	BIC	**Val RMSE**	Val MAE
1	892	905	28.4	22.1
2	**878**	**896**	**24.2**	**18.9**
3	875	898	25.1	19.5
4	873	901	26.8	21.2

$\Rightarrow$ Best: K=2 Fourier harmonics

K=3,4 overfit: lower AIC but higher validation error!

Selected Model:
ARIMA(2,0,1) + 2 Fourier pairs

Parameters:

- $\phi_1 = 1.35$, $\phi_2 = -0.68$
- $\theta_1 = -0.42$
- 4 Fourier coefficients

Total: 8 parameters

Sunspots: Final Test Set Evaluation

Refit on Training+Validation, forecast Test period (2005–2023)

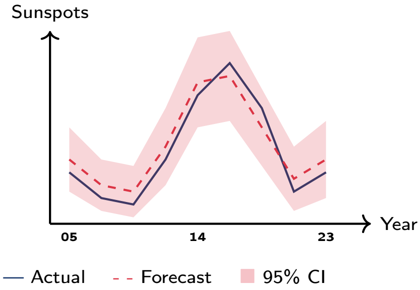
Test Set Results:

Metric	Value
Test RMSE	26.8
Test MAE	21.4
Test MAPE	42.3%

Interpretation:

- Model captures cycle timing
- Amplitude harder to predict
- Wide confidence intervals

Forecast vs Actual (2005–2023):



Sunspots: Summary

Methodology:

1. Split: 70/15/15 (years)
2. Use Fourier terms for 11-year cycle
3. Select K on validation (K=2 best)
4. Final evaluation on test

Key Finding:

Fourier terms efficiently capture long seasonal patterns without losing data to differencing.

Results:

Best Model	ARIMA(2,0,1)+Fourier(2)
Validation RMSE	24.2
Test RMSE	26.8
Key insight	K=2 optimal

Lesson

For long seasonal periods:

- Fourier terms > SARIMA
- Select K via cross-validation

Unemployment: Data and Challenge

Data: US Unemployment Rate (2010–2023)

- Source: FRED (UNRATE)
- 168 monthly observations
- Major structural break: COVID-19

The Challenge:

- April 2020: 3.5% → 14.7% (+10.3 pp)
- Largest single-month increase ever
- Traditional ARIMA struggles

Solution: Prophet with changepoint detection

Data Split:

Set	Months	Period
Training	118	2010–2019
Validation	25	2020–2021
Test	25	2022–2023

Note

Validation includes COVID shock — tests model's adaptability

Prophet: Changepoint Detection

Key Parameter: `changepoint_prior_scale`

What it controls:

- Flexibility of trend changes
- Low (0.01): Smooth, rigid trend
- High (0.5): Many changepoints allowed

For COVID data:

- Need flexibility for sudden jump
- But not too much (overfitting)
- Select via validation!

Prophet Model:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

Where:

- $g(t)$: Piecewise linear trend
- $s(t)$: Seasonality (Fourier)
- $h(t)$: Holiday effects
- Changepoints: Where $g(t)$ slope changes

Unemployment: Model Selection on Validation Set

Testing different `changepoint_prior_scale` values:

Scale	Changepoints	**Val RMSE**	Val MAE
0.01	2	2.85	2.31
0.05	4	1.92	1.54
0.10	**5**	**1.24**	**0.98**
0.30	8	1.31	1.05
0.50	12	1.48	1.19

$\Rightarrow$ Best: scale = 0.10

Detected changepoints:

- March 2020 (COVID start)
- April 2020 (peak)
- June 2020 (recovery start)

Why 0.10 wins:

- Flexible enough for COVID
- Not overfitting to noise
- Balances bias/variance

Key Insight

scale = 0.50 detected 12 changepoints — too many!
Model was chasing short-term noise.

Unemployment: Final Test Set Evaluation

Refit Prophet (scale=0.10) on Training+Validation, forecast Test (2022–2023)

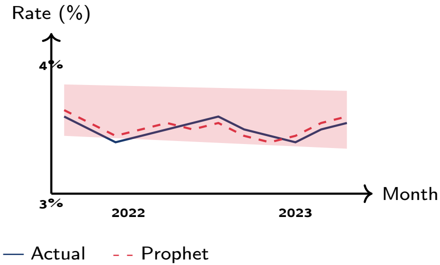
Test Set Results:

Metric	Value
Test RMSE	0.42
Test MAE	0.35
Test MAPE	9.8%

Comparison with ARIMA:

Model	Test RMSE
ARIMA(2,1,2)	0.89
Prophet	**0.42**

Forecast Visualization (2022–2023):



Unemployment: Summary

Methodology:

1. Split: 70/15/15 (months)
2. Tune `changepoint_prior_scale`
3. Select optimal scale on validation
4. Final evaluation on test

Key Finding:

Prophet handles structural breaks better than ARIMA through automatic changepoint detection.

Results:

Best Model	Prophet (scale=0.10)
Validation RMSE	1.24
Test RMSE	0.42
vs ARIMA	53% better

Lesson

For data with structural breaks:

- Prophet > traditional ARIMA
- Tune changepoint flexibility

VAR: Data and Objective

Data: US Economic Indicators (2001–2023)

- Source: FRED
- 92 quarterly observations
- 4 variables (GDP, Unemp, Inflation, Fed Rate)

Objective: Joint forecasting + relationship analysis

Why VAR?

- Variables influence each other
- Feedback loops (GDP $\leftrightarrow$ Unemployment)
- Policy transmission (Fed $\rightarrow$ Economy)

Data Split:

Set	Quarters	Period
Training	64	2001–2016
Validation	14	2017–2020
Test	14	2021–2023

Variables:

- GDP Growth (YoY %)
- Unemployment (%)
- Inflation (CPI YoY %)
- Fed Funds Rate (%)

VAR: Lag Order Selection on Validation

VAR(p) Model: $y_t = c + A_1 y_{t-1} + \cdots + A_p y_{t-p} + \varepsilon_t$

Selecting lag order p:

Lag	AIC	BIC	**Val RMSE**
1	12.4	13.1	1.85
2	**11.8**	**13.0**	**1.42**
3	11.6	13.4	1.51
4	11.5	13.8	1.68

$\Rightarrow$ Best: VAR(2)

Higher lags: lower AIC but higher validation error (overfitting)

VAR(2) Parameters:

- 4 variables $\times$ 2 lags = 8 coeff per equation
- + 4 intercepts
- Total: 36 parameters

BIC vs AIC

BIC selects p=2 (simpler)
AIC selects p=3 (complex)
Validation confirms BIC!

VAR: Granger Causality Results

Testing predictive relationships (on Training data):

Cause → Effect	F-stat	p-value	Sig.
GDP → Unemployment	4.82	0.012	**
Unemployment → GDP	1.45	0.243	
Inflation → Fed Rate	6.21	0.004	**
Fed Rate → Inflation	2.88	0.065	*
GDP → Inflation	3.12	0.052	*
Fed → Unemployment	2.15	0.127	

** $p<0.05$, * $p<0.10$

Interpretation:

- GDP leads unemployment (Okun's Law confirmed)
- Inflation leads Fed policy (Taylor Rule)
- Bidirectional: Fed ↔ Inflation

Caution

Granger causality = predictive, not true causality!

VAR: Final Test Set Evaluation

Refit VAR(2) on Training+Validation, forecast Test (2021–2023)

Test Set Results (by variable):

Variable	RMSE	MAE	vs AR(2)
GDP Growth	1.82	1.45	-18%
Unemployment	0.58	0.44	-25%
Inflation	1.24	0.98	-12%
Fed Rate	0.89	0.72	-31%
Average	**1.13**	**0.90**	**-22%**

VAR(2) beats univariate AR(2) by 22% on average!

Why VAR wins:

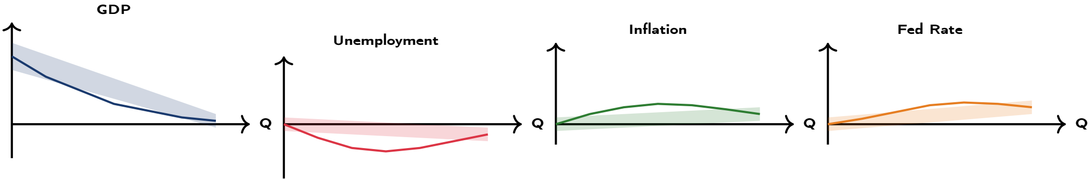
- Uses cross-variable information
- GDP helps predict unemployment
- Inflation helps predict Fed

Best Improvement

Fed Rate: -31% RMSE
Why? Inflation is a strong leading indicator for Fed policy.

VAR: Impulse Response Analysis

How does a 1% GDP shock affect other variables?



Interpretation (from IRF):

- GDP shock $\Rightarrow$ Unemployment falls for 4-6 quarters (Okun's Law)
- GDP shock $\Rightarrow$ Inflation rises, peaks at Q3-Q4 (demand-pull)
- GDP shock $\Rightarrow$ Fed Rate rises with lag (policy response)

VAR: Summary

Methodology:

1. Split: 70/15/15 (quarters)
2. Select lag order on validation (p=2)
3. Test Granger causality
4. Analyze IRF
5. Final forecast on test

Key Finding:
VAR captures economic interdependencies that univariate models miss.

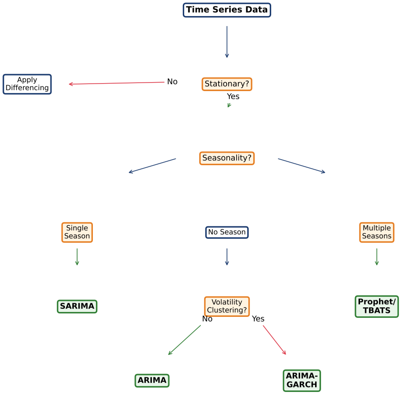
Results:

Best Model	VAR(2)
Validation RMSE	1.42
Test RMSE	1.13
vs Univariate	22% better

Applications

- Macroeconomic forecasting
- Policy impact analysis
- Portfolio risk (multi-asset)

Decision Framework



Model Selection Summary

Data Type	Characteristics	Recommended Model	Alternatives
Financial returns	No trend, volatility clustering	ARIMA-GARCH	EGARCH, GJR
Single seasonality	Trend + one seasonal period	SARIMA	ETS, Prophet
Long cycles	Sunspots, business cycles	AR + Fourier, TBATS	Spectral methods
Structural breaks	COVID, regime changes	Prophet	Intervention ARIMA
Multiple series	Interdependencies	VAR, VECM	Factor models

Forecast Evaluation Metrics

Point Forecast Metrics:

RMSE (Root Mean Square Error):

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

MAE (Mean Absolute Error):

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAPE (Mean Absolute % Error):

$$\frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

When to Use Each:

- **RMSE**: Penalizes large errors more
- **MAE**: Robust to outliers
- **MAPE**: Scale-independent

Cross-Validation

Always use time series CV:

- Rolling window
- Expanding window
- Never shuffle!

Course Summary: Complete Toolkit

Understanding the Data

- Visualization first!
- Test for stationarity (ADF, KPSS)
- Identify seasonality patterns
- Check for structural breaks

Classical Models

- ARIMA: Non-seasonal data
- SARIMA: Single seasonality
- GARCH: Volatility modeling

Modern Approaches

- Prophet: Interpretable, handles breaks
- TBATS: Multiple/long seasonalities
- VAR/VECM: Multiple time series

Best Practices

- Always check diagnostics
- Use cross-validation
- Compare multiple models
- Domain knowledge matters!

Final Recommendations

1. **Start Simple**: Begin with visualization and basic statistics
2. **Test Assumptions**: Stationarity, normality, independence
3. **Iterate**: Model $\rightarrow$ Diagnose $\rightarrow$ Improve
4. **Compare**: Never rely on a single model
5. **Validate**: Out-of-sample testing is essential
6. **Communicate**: Clear visualizations and interpretations

Remember

"All models are wrong, but some are useful." — George Box

The goal is not perfect prediction, but useful insights and reasonable forecasts.

Questions?

Questions?

Next Steps:

- Practice with the Jupyter notebook
- Apply these methods to your own data
- Compare different models on the same dataset

Course Materials: `github.com/danpele/Time-Series-Analysis`