Time Series Analysis and Forecasting

Chapter 4: SARIMA Models

Daniel Traian PELE

Bucharest University of Economic Studies
IDA Institute Digital Assets
Blockchain Research Center
AI4EFin Artificial Intelligence for Energy Finance
Romanian Academy, Institute for Economic Forecasting
MSCA Digital Finance

Learning Objectives

By the end of this chapter, you will be able to:

1. **Identify** seasonal patterns in time series data
2. **Apply** seasonal differencing to remove seasonal unit roots
3. **Build** and estimate SARIMA models with seasonal components
4. **Interpret** seasonal ACF/PACF patterns for model identification
5. **Evaluate** forecasts using rolling window methods for seasonal data
6. **Apply** the complete methodology on real data (airline passengers)

Data Sources and Software Tools

Data Sources

- **FRED** (Federal Reserve)
  - Retail sales, industrial production
- **Yahoo Finance**
  - Stock prices, exchange rates
- **Eurostat / INS**
  - Seasonal macroeconomic data
- **Statsmodels datasets**
  - Airline passengers (Box-Jenkins)

Python

- `statsmodels` — SARIMA models
- `pmdarima` — automatic selection
- `pandas-datareader` — FRED download
- `matplotlib` — visualization
- `scipy` — statistical tests

Resources

- `github.com/QuantLet/TSA/TSA_ch4`
- `quantlet.com`

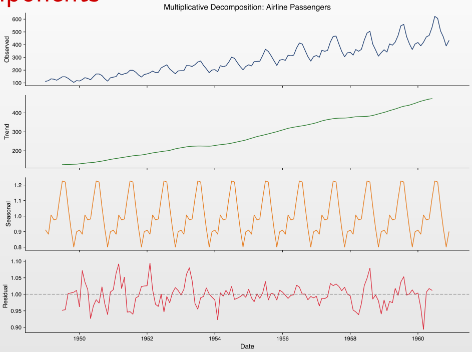
Chapter Structure

Why SARIMA? Seasonality Is Everywhere

- Retail sales exhibit clear **annual patterns**: December peaks, January troughs
- Standard ARIMA models cannot capture these **repeating seasonal cycles**
- Ignoring seasonality leads to systematic forecast errors

TSA_ch4_motivation_seasonal

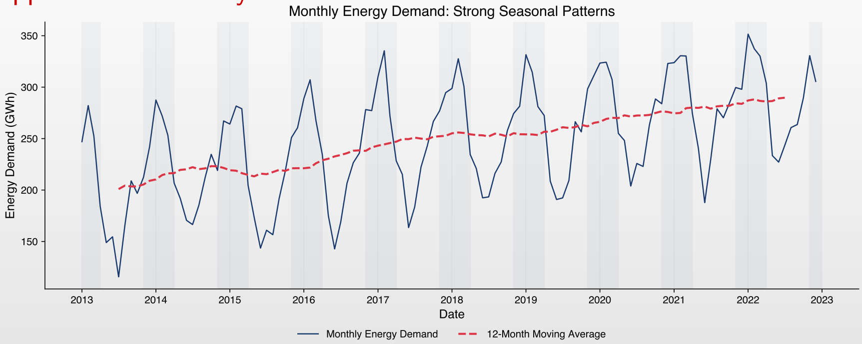
Understanding Seasonal Components



- Seasonal time series = **Trend** + **Seasonal Pattern** + **Residuals**
- Decomposition helps visualize each component separately
- SARIMA models capture both trend dynamics and seasonal behavior

TSA_ch4_motivation_decomposition

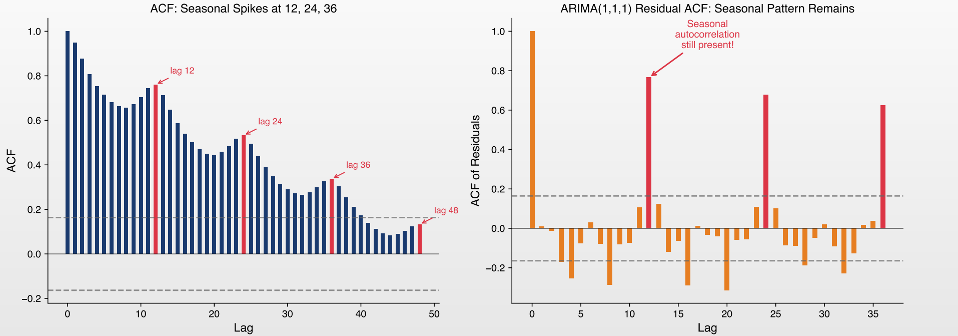
Real-World Application: Monthly Patterns



- Energy demand shows strong **monthly seasonality** (heating/cooling cycles)
- Pattern repeats predictably each year with slight variations
- Utility companies use SARIMA forecasts for capacity planning

TSA_ch4_motivation_monthly

Why Do We Need SARIMA?



- **Left**: Seasonal ACF shows spikes at lags 12, 24, 36... (annual pattern)
- **Right**: ARIMA residuals still show seasonal autocorrelation $\Rightarrow$ model is incomplete
- SARIMA adds **seasonal AR and MA terms** to capture these patterns

TSA_ch4_motivation_why_sarima

What We'll Learn Today

Concepts

- Identifying seasonal patterns
- Seasonal differencing operator
- SARIMA$(p, d, q)(P, D, Q)_s$ notation
- The famous "Airline Model"
- Model selection for seasonal data

Skills

- Diagnose seasonality from ACF/PACF
- Determine seasonal period s
- Choose (P, D, Q) seasonal orders
- Implement SARIMA in Python/R
- Forecast seasonal time series

Key Insight

- SARIMA = ARIMA applied at **two frequencies**: non-seasonal (short-term) and seasonal (long-term)

What is Seasonality?

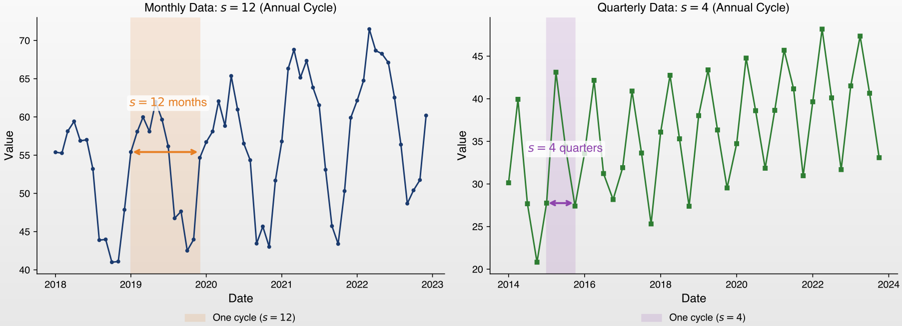
Definition 1 (Seasonality)

A time series exhibits **seasonality** when it shows regular, periodic fluctuations that repeat over a fixed period s (the seasonal period).

Common Seasonal Periods

- Monthly data: $s = 12$ (annual cycle)
- Quarterly data: $s = 4$ (annual cycle)
- Weekly data: $s = 52$ (annual) or $s = 7$ (weekly pattern)
- Daily data: $s = 7$ (weekly pattern)

Seasonality: Visual Illustration



- **Left**: Monthly data with $s = 12$ (annual cycle); **Right**: Quarterly data with $s = 4$
- The pattern repeats every s periods $\Rightarrow$ this regularity is exploited by SARIMA

TSA_ch4_seasonality

Examples of Seasonal Data

Economic Series

- Retail sales (holiday peaks)
- Tourism (summer/winter)
- Agricultural production
- Energy consumption
- Employment (hiring cycles)

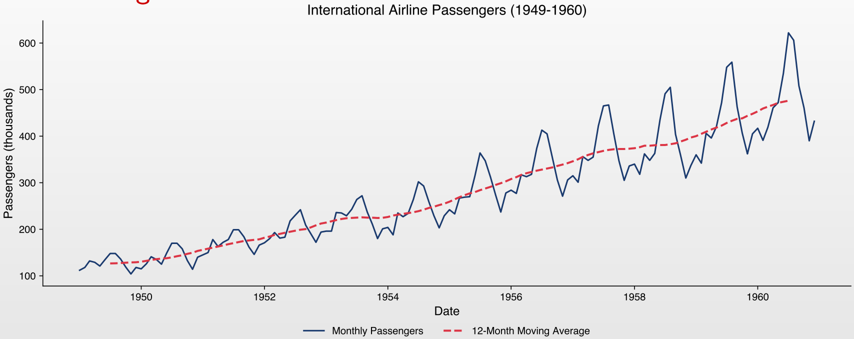
Other Domains

- Weather/temperature
- Website traffic
- Hospital admissions
- Transportation usage
- Electricity demand

Why It Matters

Ignoring seasonality leads to biased forecasts and invalid inference!

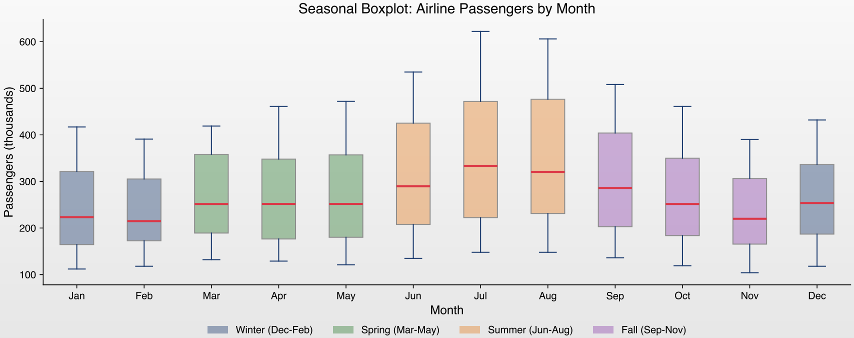
Example: Airline Passengers Data



- Monthly international airline passengers (1949–1960)
- Clear **upward trend** and **growing seasonal amplitude**
- Summer peaks reflect vacation travel patterns

TSA_ch4_airline_data

Visualizing Seasonal Patterns



- Box plot reveals consistent seasonal pattern across years
- July–August: highest passenger counts (summer travel)
- November–February: lowest counts (winter months)

TSA_ch4_seasonal_boxplot

Deterministic vs Stochastic Seasonality

Deterministic Seasonality

- **Fixed pattern**: $Y_t = \sum_{j=1}^{s} \gamma_j D_{jt} + \varepsilon_t$
  - D_{jt} are seasonal dummies
- Pattern constant over time
- Same amplitude every year
- Removed by regression on dummies
- ACF: sharp cutoff at seasonal lags
- **Example:** University enrollment peaks every September by the same amount

Stochastic Seasonality

- **Evolving pattern**: $\Delta_s Y_t = Y_t - Y_{t-s}$
  - Exhibits dependence structure
- Pattern evolves over time
- Amplitude may grow or shrink
- Requires seasonal differencing
- ACF: slow decay at seasonal lags
- **Example:** Retail sales peaks grow larger each December

How to decide?

- Slow ACF decay at lags $s, 2s, 3s, \ldots \Rightarrow$ stochastic (use Δ_s)
- Sharp cutoff $\Rightarrow$ deterministic (use dummies)
- Use HEGY or Canova-Hansen tests to confirm

Additive vs Multiplicative Seasonality

Additive: $Y_t = T_t + S_t + \varepsilon_t$

- Seasonal amplitude **constant**
- No transformation needed
- Ex: temperatures, university enrollment

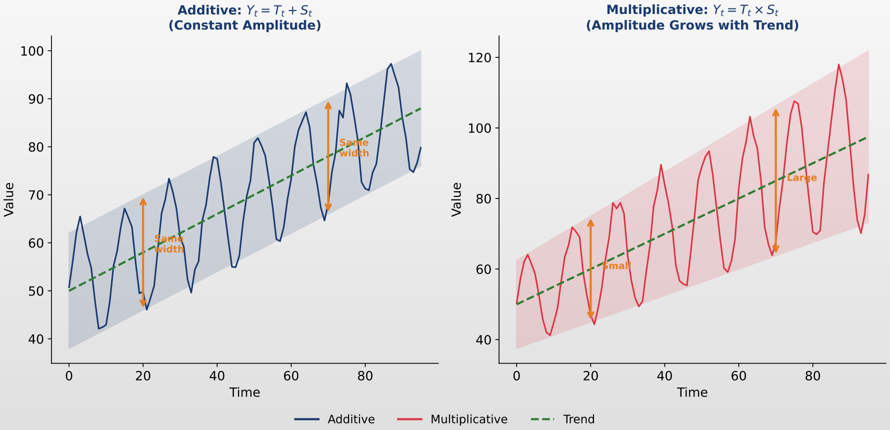
Multiplicative: $Y_t = T_t \cdot S_t \cdot \varepsilon_t$

- Amplitude **grows with level**
- Requires log transform (Box-Cox)
- Ex: Airline, retail sales, GDP

First practical decision

- Amplitude grows with the trend? $\Rightarrow$ multiplicative $\Rightarrow$ apply log/Box-Cox *before* differencing

Additive vs Multiplicative Seasonality



Detecting Seasonality

Visual Methods

- Time series plot – look for repeating patterns
- Seasonal subseries plot – compare same seasons across years
- ACF plot – spikes at seasonal lags $(s, 2s, 3s, \ldots)$

Statistical Tests

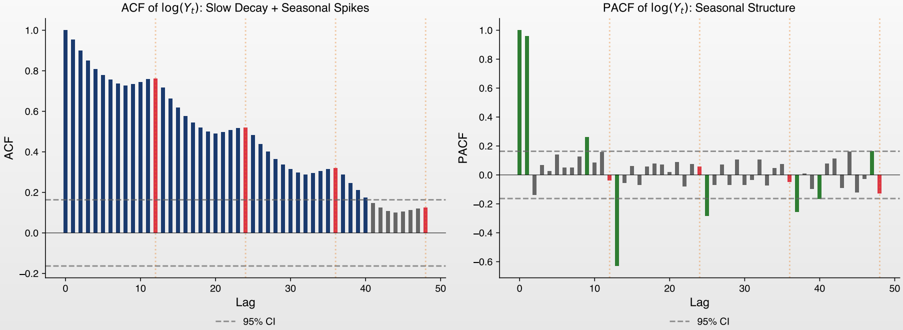
- Seasonal unit root tests (HEGY, Canova-Hansen, OCSB[a])
- F-test for seasonal dummy variables
- Kruskal-Wallis test (non-parametric)

ACF Signature

- Strong seasonality: ACF shows significant spikes at lags $s, 2s, 3s, \ldots$

[a] Osborn-Chui-Smith-Birchenhall — the default test in `auto_arima`

ACF Reveals Seasonal Structure



- **Slow decay** at all lags indicates non-stationarity (trend)
- **Spikes at lags 12, 24, 36** confirm seasonal pattern ($s = 12$)
- ACF at seasonal lags: slow decay $\Rightarrow$ needs seasonal differencing

TSA_ch4_acf_seasonality

F-Test for Seasonal Dummy Variables: Intuition

What does this test do?

- **Goal**: test whether mean values differ significantly across seasons
- **Logic**: if the mean in January $\neq$ February $\neq \ldots \neq$ December $\Rightarrow$ seasonality
- **Method**: compare a model WITH seasonal dummy variables vs. a model WITHOUT

Models compared

- **Restricted**: $Y_t = \alpha + \varepsilon_t$ **Unrestricted**: $Y_t = \alpha + \sum_{j=1}^{s-1} \gamma_j D_{jt} + \varepsilon_t$
- where $D_{jt} = 1$ if observation t is in season j, 0 otherwise

Key idea

- If adding seasonal dummy variables **significantly reduces** prediction errors, then seasonality is present

F-Test for Seasonal Dummy Variables: Formula and Example

F-statistic formula

- **Formula**: $F = \frac{(SSR_R - SSR_U)/(s-1)}{SSR_U/(n-s)} \sim F_{s-1,n-s}$
  - SSR_R: sum of squared residuals from the restricted model (no dummies)
  - SSR_U: sum of squared residuals from the unrestricted model (with dummies)
  - $s-1$: number of restrictions (monthly: 11, quarterly: 3)

Numerical example (Monthly data, n=120)

- $SSR_R = 15000$, $SSR_U = 8500$, $s = 12$
- $F = \frac{(15000-8500)/11}{8500/108} = \frac{590.9}{78.7} = 7.51$
- Critical value $F_{0.05,11,108} \approx 1.87$. Since $7.51 > 1.87$: **Reject** $H_0 \Rightarrow$ Seasonality present!

Kruskal-Wallis Test: Intuition

What does this test do?

- **Non-parametric test**: checks whether observations from different seasons come from the same distribution
- **Mechanism**: ranks all observations from smallest to largest
- **Check**: whether ranks are uniformly distributed across seasons
- **Conclusion**: if one season consistently has higher/lower ranks $\Rightarrow$ seasonality

Why use it instead of the F-test?

- **No normality assumption** – works with any distribution
- **Robust to outliers** – extreme values do not distort results

Limitation

- Less powerful than the F-test when data ARE normally distributed

Kruskal-Wallis Test: Formula and Example

Test statistic

- $H = \frac{12}{N(N+1)} \sum_{j=1}^{s} \frac{R_j^2}{n_j} - 3(N+1)$ where N = total obs., n_j = obs. in season j, R_j = rank sum

Example: Quarterly sales (n=20, s=4)

- Data ranked 1–20. Rank sums: Q1: $R_1 = 15$, Q2: $R_2 = 35$, Q3: $R_3 = 70$, Q4: $R_4 = 90$
- $H = \frac{12}{20\times21}\left(\frac{15^2}{5} + \frac{35^2}{5} + \frac{70^2}{5} + \frac{90^2}{5}\right) - 3(21) = 19.6$
- Critical value $\chi^2_{0.05,3} = 7.81$. Since $19.6 > 7.81$: **Reject** $H_0 \Rightarrow$ Seasonality!

In Python

- **Implementation**: `scipy.stats.kruskal(q1, q2, q3, q4)`

HEGY Test: What Problem Does It Solve?

Key question

- **Problem**: given a seasonal series, we need to determine the type of differencing
- **Regular differencing** $(1 - L)$? $\Rightarrow$ set $d = 1$; **Seasonal differencing** $(1 - L^s)$? $\Rightarrow$ set $D = 1$
- **HEGY**: tests for both types of unit roots simultaneously!

Why not just use ADF?

- **ADF**: tests only for a regular unit root at frequency zero
- **Limitation**: seasonal data may have unit roots at seasonal frequencies that ADF misses!

HEGY tests multiple frequencies

- **Quarterly**: tests at 0, π, $\pm\pi/2$
- **Monthly**: tests at 0, π, $\pm\pi/6$, $\pm\pi/3$, $\pm\pi/2$, $\pm2\pi/3$, $\pm5\pi/6$

HEGY Test: Auxiliary Regression (Quarterly)

HEGY auxiliary regression

- **Quarterly data** $(s = 4)$: $\Delta_4 y_t = \pi_1 z_{1,t-1} + \pi_2 z_{2,t-1} + \pi_3 z_{3,t-2} + \pi_4 z_{4,t-2} + \sum_{j=1}^{k} \phi_j \Delta_4 y_{t-j} + \varepsilon_t$

Transformed variables

- z_{1t}: $(1 + L + L^2 + L^3)y_t = y_t + y_{t-1} + y_{t-2} + y_{t-3}$
- z_{2t}: $-(1 - L + L^2 - L^3)y_t = -y_t + y_{t-1} - y_{t-2} + y_{t-3}$
- z_{3t}: $-(1 - L^2)y_t = -y_t + y_{t-2}$
- z_{4t}: $-(L - L^3)y_t = -y_{t-1} + y_{t-3}$

Hypotheses

- $H_0 : \pi_1 = 0$: unit root at frequency 0
- $H_0 : \pi_2 = 0$: unit root at frequency π
- $H_0 : \pi_3 = \pi_4 = 0$: unit root at frequency $\pm\pi/2$

HEGY Test: Decision Rules with Examples

HEGY critical values (5%, n=100, with constant)

Test	Statistic	Critical value	If NOT rejected...
t_1 ($\pi_1 = 0$)	t-stat	-2.88	Requires $d = 1$
t_2 ($\pi_2 = 0$)	t-stat	-2.88	Requires $D = 1$
F_{34} ($\pi_3 = \pi_4 = 0$)	F-stat	6.57	Requires $D = 1$

Example: Quarterly GDP

- **HEGY results**: $t_1 = -1.52$, $t_2 = -4.21$, $F_{34} = 2.15$
- $t_1 = -1.52 > -2.88$: Cannot reject $\Rightarrow$ **requires** $d = 1$
- $t_2 = -4.21 < -2.88$: Reject $\Rightarrow$ no unit root at π
- $F_{34} = 2.15 < 6.57$: Cannot reject $\Rightarrow$ **requires** $D = 1$
- **Conclusion**: Use SARIMA with $d = 1, D = 1$

Canova-Hansen Test: The Opposite of HEGY

HEGY vs Canova-Hansen: Different null hypotheses!

	HEGY	**Canova-Hansen**
H_0	Seasonal unit root	**No** seasonal unit root
H_1	No seasonal unit root	Seasonal unit root
Reject H_0	Use seasonal dummies	Use differencing $(1 - L^s)$
Do not reject	Use differencing $(1 - L^s)$	Use seasonal dummies

Why does it matter?

- HEGY: "Prove there is NO unit root" (conservative towards differencing)
- CH: "Prove there IS a unit root" (conservative towards dummies)
- Use **both** tests for robust conclusions!

Canova-Hansen Test: Formula

Testing procedure

- **Step 1**: Regress y_t on seasonal dummies: $y_t = \sum_{j=1}^{s} \gamma_j D_{jt} + u_t$
- **Step 2**: Compute partial sums at seasonal frequency λ_i:
  - $S_{it}^{(c)} = \sum_{j=1}^{t} \hat{u}_j \cos(\lambda_i j), \quad S_{it}^{(s)} = \sum_{j=1}^{t} \hat{u}_j \sin(\lambda_i j)$

LM test statistic

- $LM_i = \frac{1}{T^2 \hat{\omega}_i} \left[\sum_{t=1}^{T} (S_{it}^{(c)})^2 + \sum_{t=1}^{T} (S_{it}^{(s)})^2 \right]$
- where $\hat{\omega}_i$ = consistent estimator of the spectral density at frequency λ_i

Decision

- **Rule**: reject H_0 (stationarity) if $LM >$ critical value $\Rightarrow$ seasonal differencing is needed

Summary: Choosing the Right Seasonality Test

Test	H_0	If rejected	Best for
F-test	No seasonality	Seasonality exists	Normal data
Kruskal-Wallis	No difference across seasons	Seasonality exists	Non-normal, outliers
HEGY	Unit root exists	Use dummies	Determining d, D
Canova-Hansen	No unit root	Use $(1 - L^s)$	Confirming stability

Key idea

- **F-test / Kruskal-Wallis**: "*Does seasonality exist?*"
- **HEGY / Canova-Hansen**: "*What type?*" (deterministic vs stochastic)

Box-Cox Transformation: Variance Stabilization

Box-Cox Family of Transformations

- **Formula**: $Y_t^{(\lambda)} = \begin{cases} \frac{Y_t^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \ln(Y_t) & \text{if } \lambda = 0 \end{cases}$
- **Special cases**: $\lambda = 1$ (no transformation), $\lambda = 0$ (logarithm), $\lambda = 0.5$ (square root)

Automatic Selection of λ

- **Profile likelihood**: maximizes the log-likelihood as a function of λ
- **Guerrero method (1993)**: minimizes the coefficient of variation of seasonal sub-series
- **Python**: `boxcox(y)` from `scipy.stats` or `BoxCox.lambda_(y)` in R

Why not just logarithm?

- Log ($\lambda = 0$) assumes variance proportional to level — not always the case
- Box-Cox chooses the optimal transformation based on data, not assumptions

Box-Cox on the Airline Data: Complete Example

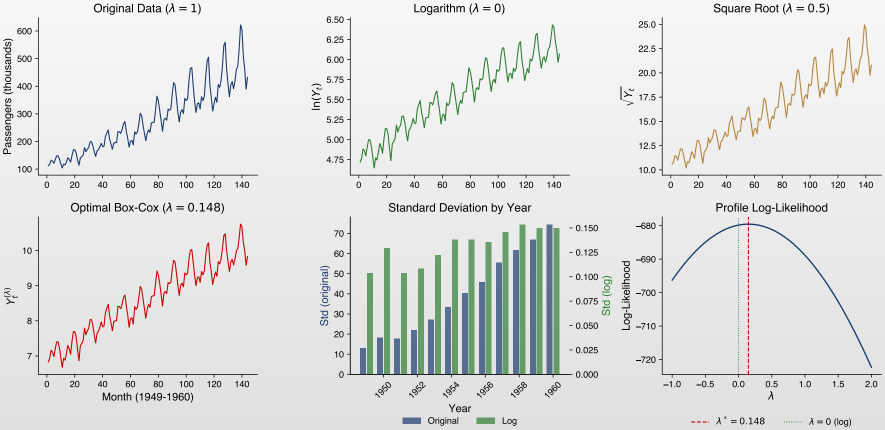
Result for Airline Passengers

- $\hat{\lambda} = 0.148 \approx 0 \Rightarrow$ log is nearly optimal
- Standard deviation per year: from increasing (original) to stable (log)

Bias Correction in Back-Transformation

- On log scale: $\hat{y}_{T+h}$ is the **median**, not the mean
- Correction: $\hat{Y}_{T+h} = \exp\left(\hat{y}_{T+h} + \frac{\sigma_h^2}{2}\right)$
- Without correction: systematically under-estimated forecasts!

Box-Cox on the Airline Data: Complete Example



TSA_ch4_boxcox_airline

STL Decomposition: Modern Alternatives

STL: Seasonal-Trend Decomposition using Loess (Cleveland et al., 1990)

- **Advantages**: time-varying seasonality, robust to outliers, any period s
- **Algorithm**: iterative locally weighted regression (loess)

Key Parameters

- **Seasonal window** (`seasonal`): controls how quickly seasonality changes
- **Trend window** (`trend`): smoothing of the trend component
- **Robustness** (`robust=True`): reduces influence of outliers

Practical Usage

- STL for exploration and preprocessing; SARIMA for modeling and forecasting
- Python: `STL(y, period=12).fit()` from `statsmodels`

The Seasonal Difference Operator

Definition 2 (Seasonal Difference)

The **seasonal difference operator** Δ_s is defined as:

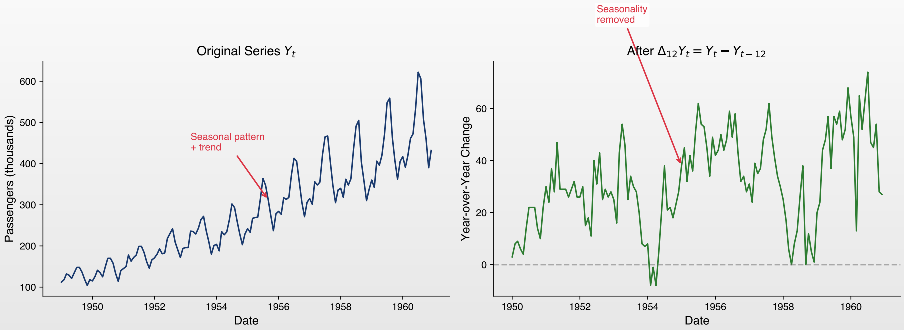
$$\Delta_s Y_t = (1 - L^s)Y_t = Y_t - Y_{t-s}$$

where $L^s Y_t = Y_{t-s}$ is the seasonal lag operator.

Examples

- Monthly data ($s = 12$): $\Delta_{12} Y_t = Y_t - Y_{t-12}$
  Compares each month to the same month last year
- Quarterly data ($s = 4$): $\Delta_4 Y_t = Y_t - Y_{t-4}$
  Compares each quarter to the same quarter last year

Seasonal Difference: Visual Illustration



- **Left**: Original series with clear seasonal pattern
- **Right**: After $\Delta_{12} = (1 - L^{12})$, seasonal pattern is removed
  - Year-over-year comparison eliminates seasonal effects

TSA_ch4_seasonal_diff

Proof: Seasonal Differencing Removes Deterministic Seasonality

Claim: If $Y_t = \mu_t + \varepsilon_t$ where $\mu_t = \mu_{t-s}$ (periodic mean), then $\Delta_s Y_t$ removes the seasonal mean.

Proof: Let $Y_t = \mu_t + \varepsilon_t$ where μ_t has period s. Apply seasonal difference:

$$\begin{aligned}\Delta_s Y_t &= Y_t - Y_{t-s} = (\mu_t + \varepsilon_t) - (\mu_{t-s} + \varepsilon_{t-s}) \\ &= \mu_t - \mu_{t-s} + \varepsilon_t - \varepsilon_{t-s} \\ &= 0 + \varepsilon_t - \varepsilon_{t-s} \quad (\text{since } \mu_t = \mu_{t-s})\end{aligned}$$

Properties of $\Delta_s Y_t = \varepsilon_t - \varepsilon_{t-s}$:

- $\mathbb{E}[\Delta_s Y_t] = 0$ (constant mean)
- $\text{Var}(\Delta_s Y_t) = 2\sigma^2$ (constant variance)
- Autocovariance: $\gamma(s) = -\sigma^2$, $\gamma(k) = 0$ for $k \neq 0, s$

Result

Seasonal differencing transforms periodic seasonal pattern into MA(1) at seasonal lag.

Combining Regular and Seasonal Differencing

Full Differencing

For series with both trend and seasonality:

$$\Delta\Delta_s Y_t = (1 - L)(1 - L^s)Y_t$$

Expansion

$(1 - L)(1 - L^s)Y_t = Y_t - Y_{t-1} - Y_{t-s} + Y_{t-s-1}$. For monthly: $\Delta\Delta_{12} Y_t = Y_t - Y_{t-1} - Y_{t-12} + Y_{t-13}$

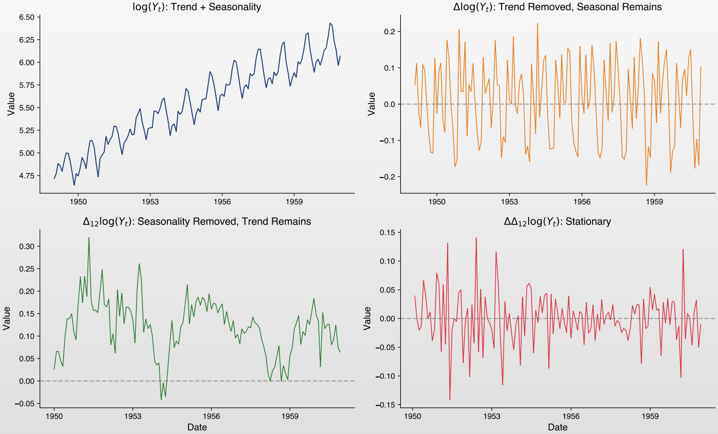
Order of Differencing

d: regular differences (trend removal); D: seasonal differences (seasonal trend removal)

Effect of Differencing Operations

- Regular differencing removes trend but seasonal pattern remains
- Seasonal differencing removes seasonality but trend pattern remains
- **Both differences** needed to achieve stationarity

Effect of Differencing Operations

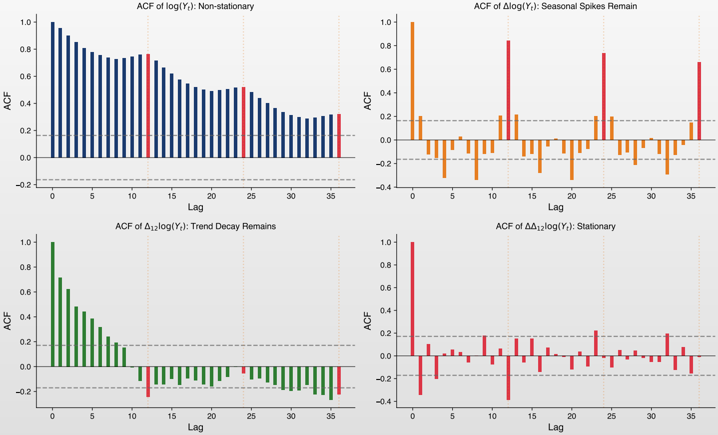


TSA_ch4_differencing_effect

ACF Before and After Differencing

- Original ACF: slow decay indicates non-stationarity
- After Δ: seasonal spikes remain at lags 12, 24, 36
- After Δ_{12}: trend decay remains at early lags
- After $\Delta\Delta_{12}$: ACF cuts off $\Rightarrow$ **stationary**

ACF Before and After Differencing



TSA_ch4_acf_differencing

Seasonal Integration

Definition 3 (Seasonally Integrated Process)

A series Y_t is **seasonally integrated** of order $(d, D)_s$, written $Y_t \sim I(d, D)_s$, if:

$$(1 - L)^d (1 - L^s)^D Y_t$$

is stationary.

Common Cases

- $I(1, 0)_{12}$: Regular unit root only (monthly)
- $I(0, 1)_{12}$: Seasonal unit root only
- $I(1, 1)_{12}$:
  - Both regular and seasonal unit roots

SARIMA Model Definition

Definition 4 (SARIMA$(p, d, q) \times (P, D, Q)_s$)

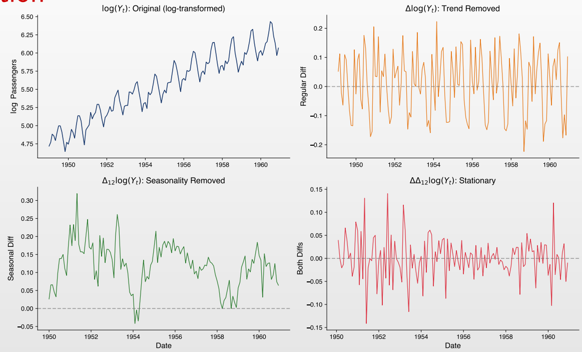
The **Seasonal ARIMA** model is:

$$\phi(L)\Phi(L^s)(1-L)^d(1-L^s)^D Y_t = c + \theta(L)\Theta(L^s)\varepsilon_t$$

Components

- $\phi(L) = 1 - \phi_1 L - \cdots - \phi_p L^p$: Non-seasonal AR
- $\Phi(L^s) = 1 - \Phi_1 L^s - \cdots - \Phi_P L^{Ps}$: Seasonal AR
- $\theta(L) = 1 + \theta_1 L + \cdots + \theta_q L^q$: Non-seasonal MA
- $\Theta(L^s) = 1 + \Theta_1 L^s + \cdots + \Theta_Q L^{Qs}$: Seasonal MA
- $(1-L)^d$:
  - Regular differencing; $(1-L^s)^D$: Seasonal differencing

SARIMA: Visual Illustration



- Original $\Rightarrow$ regular difference (removes trend) $\Rightarrow$ seasonal difference (removes seasonality)
- Apply minimum differencing needed to achieve stationarity

TSA_ch4_sarima_model

Proof: Multiplicative Seasonal Structure

Why multiplicative? Consider SARIMA$(1,0,0) \times (1,0,0)_s$:

$$(1 - \phi L)(1 - \Phi L^s)Y_t = \varepsilon_t$$

Expand: $(1 - \phi L)(1 - \Phi L^s)Y_t = Y_t - \phi Y_{t-\mathbf{1}} - \Phi Y_{t-s} + \phi\Phi Y_{t-s-\mathbf{1}}$

Interpretation (Monthly, $s = 12$)

Y_t depends on: $Y_{t-\mathbf{1}}$ (last month), $Y_{t-\mathbf{12}}$ (same month last year), $Y_{t-\mathbf{13}}$ (interaction).
Parsimony: Multiplicative form uses 2 parameters (ϕ, Φ); additive would need 3+.

SARIMA Notation

Full Specification

SARIMA$(p, d, q) \times (P, D, Q)_s$ has 7 parameters to specify:

Parameter	**Meaning**
p	Non-seasonal AR order
d	Non-seasonal differencing order
q	Non-seasonal MA order
P	Seasonal AR order
D	Seasonal differencing order
Q	Seasonal MA order
s	Seasonal period

Example

SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$: Monthly data with AR(1), MA(1), seasonal AR(1), seasonal MA(1), and both regular and seasonal differencing.

Common SARIMA Models

Airline Model: SARIMA$(0,1,1) \times (0,1,1)_s$

$(1-L)(1-L^s)Y_t = (1+\theta L)(1+\Theta L^s)\varepsilon_t$ – Classic model (Box & Jenkins, 1970)

SARIMA$(1,0,0) \times (1,0,0)_s$

$(1-\phi L)(1-\Phi L^s)Y_t = \varepsilon_t$ – Pure seasonal and non-seasonal AR

SARIMA$(0,1,1) \times (0,1,0)_s$

$(1-L)(1-L^s)Y_t = (1+\theta L)\varepsilon_t$ – Random walk + seasonal diff + MA(1)

ACF/PACF for Seasonal Models

Key Insight

Seasonal models show patterns at both:

- Non-seasonal lags: $1, 2, 3, \ldots$
- Seasonal lags: $s, 2s, 3s, \ldots$

Model	**ACF**	**PACF**
SAR(P)	Decays at $s, 2s, \ldots$	Cuts off after Ps
SMA(Q)	Cuts off after Qs	Decays at $s, 2s, \ldots$
SARMA	Decays at seasonal lags	Decays at seasonal lags

Example: Airline Model ACF/PACF

ACF: $\Delta\Delta_{12}\log(Y_t)$

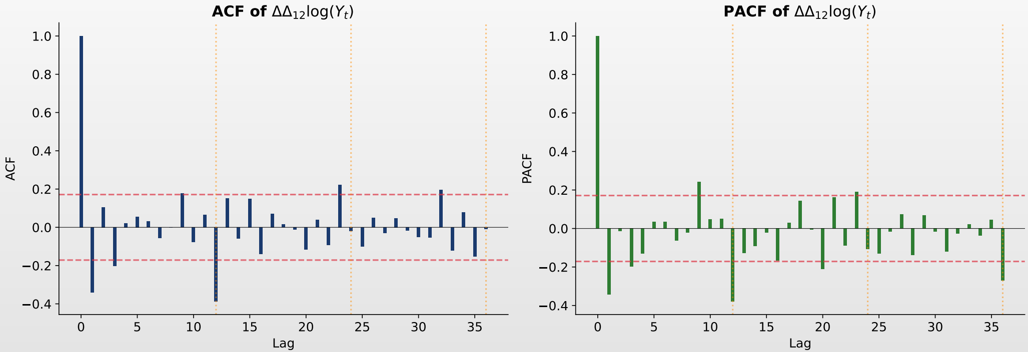
- Spike at lag 1 $\leftarrow$ MA(1), θ
- Spike at lag 12 $\leftarrow$ SMA(1), Θ
- Rest $\approx$ zero

PACF: exponential decay

- Decays at lags $1, 2, 3, \ldots$
- Decays at lags $12, 24, 36$
- $\Rightarrow$ **MA, not AR**

- **Conclusion**: ACF cuts off $\Rightarrow$ MA; PACF decays $\Rightarrow$ not AR. Model: $(0, 1, 1) \times (0, 1, 1)_{12}$

Example: Airline Model ACF/PACF



Model Identification Guidelines

Step-by-Step Process

1. Examine ACF for slow decay at seasonal lags $\Rightarrow$ seasonal differencing
2. After differencing, check ACF/PACF patterns
3. Non-seasonal behavior at lags $1, 2, \ldots, s-1$
4. Seasonal behavior at lags $s, 2s, 3s, \ldots$

Practical Tips

- Start with $d \leq 1$ and $D \leq 1$
- Usually $P, Q \leq 2$ is sufficient
- Use information criteria (AIC, BIC) for final selection
- Auto-SARIMA algorithms can help

Estimation Methods

Maximum Likelihood Estimation

Standard approach for SARIMA:

- Conditional MLE (conditional on initial values)
- Exact MLE (via Kalman filter)

Computational Considerations

- More parameters than ARIMA $\Rightarrow$ more data needed
- Seasonal parameters estimated from lags $s, 2s, \ldots$
- Need sufficient seasonal cycles (at least 3-4 years of monthly data)

Exact Likelihood: Prediction Error Decomposition

Why the Kalman Filter?

- **SARIMA**: has the structure of a state-space model
- **Kalman filter**: recursively computes prediction errors v_t and their variances f_t, without conditioning on initial values

Exact Log-Likelihood (Prediction Error Decomposition)

- **Formula**: $\ell(\theta) = -\frac{T}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\left[\ln(f_t) + \frac{v_t^2}{f_t}\right]$
- v_t: $Y_t - \hat{Y}_{t|t-1}$ (innovation); f_t: $\mathrm{Var}(v_t)$ (innovation variance)

Advantages over Conditional MLE

- Does not require choosing initial values
- Each term $\ln(f_t)$ weights observations differently (variable variance at start)
- Essential for short series where initial values matter
- Implemented by default in `statsmodels.tsa.SARIMAX()` with `method='mle'`

Stationarity and Invertibility

Stationarity Conditions

Both non-seasonal and seasonal AR polynomials must have roots outside the unit circle:

- $\phi(z) = 0 \Rightarrow |z| > 1$
- $\Phi(z^s) = 0 \Rightarrow |z| > 1$

Invertibility Conditions

Both non-seasonal and seasonal MA polynomials must have roots outside the unit circle:

- $\theta(z) = 0 \Rightarrow |z| > 1$
- $\Theta(z^s) = 0 \Rightarrow |z| > 1$

Diagnostic Checking

Residual Analysis

After fitting SARIMA, check that residuals are white noise:

1. Plot residuals over time (no patterns)
2. ACF of residuals (no significant spikes)
3. Ljung-Box test at multiple lags including seasonal
4. Normality tests (Q-Q plot, Jarque-Bera)

Important

Check ACF at **both** non-seasonal and seasonal lags!
Significant ACF at lag 12 suggests inadequate seasonal modeling.

Model Selection Criteria

Information Criteria

Compare competing SARIMA models using:

- $\text{AIC} = -2\ln(L) + 2k$
- $\text{BIC} = -2\ln(L) + k\ln(n)$
- $\text{AICc} = \text{AIC} + \frac{2k(k+1)}{n-k-1}$ (corrected for small samples)

where $k = p + q + P + Q + 1$ (plus 1 for variance).

Auto-SARIMA

Python's `pmdarima.auto_arima()` with `seasonal=True` automatically searches for optimal $(p, d, q) \times (P, D, Q)_s$.

Hyndman-Khandakar Algorithm (`auto_arima`)

How does automatic selection work? (Hyndman & Khandakar, 2008)

1. d: successive KPSS tests ($d = 0, 1, 2$); D: OCSB or Canova-Hansen test ($D = 0, 1$)
2. **Stepwise search**: starts from initial model, explores neighboring models
3. **Criterion**: AICc (correct for small samples)

Search Strategy

- **Initial model**: SARIMA$(2, d, 2)(1, D, 1)_s$ or SARIMA$(0, d, 0)(0, D, 0)_s$
- **Variations tested**: ± 1 for each order (p, q, P, Q); stops when no neighbor improves AICc
- **Complexity**: $O(20\text{-}30)$ models evaluated (vs. $O(k^4)$ for grid search)

Python: `pm.auto_arima(y, seasonal=True, m=12, stepwise=True, trace=True)`

- Set `stepwise=False` for exhaustive search (slower, sometimes better)

Point Forecasts

Forecast Computation

SARIMA forecasts are computed recursively:

- Replace future ε_{T+h} with 0
- Replace future Y_{T+h} with forecasts $\hat{Y}_{T+h|T}$
- Use known past values $Y_T, Y_{T-1}, \ldots$

Seasonal Pattern in Forecasts

SARIMA forecasts naturally capture seasonality:

- Short-term: influenced by recent values
- Long-term: revert to seasonal pattern

Forecast Intervals

Uncertainty Quantification

$(1-\alpha)\%$ prediction interval:

$$\hat{Y}_{T+h|T} \pm z_{\alpha/2}\sqrt{\text{Var}(e_{T+h})}$$

Variance computed from MA(∞) representation.

Key Properties

- Intervals widen with forecast horizon
- For $I(1,1)_s$ series: intervals grow without bound
- Seasonal pattern visible in point forecasts
- Uncertainty captures both trend and seasonal variation

Long-Horizon Forecasts

Behavior as $h \to \infty$

- Point forecasts converge to deterministic seasonal pattern
- If drift present: linear trend + seasonal pattern
- Forecast intervals continue to widen

Practical Implication

- Short-term: SARIMA captures both short-term dynamics and season
- Medium-term: Good seasonal forecasts, growing uncertainty
- Long-term: Mainly reflects seasonal pattern, wide intervals

The Seasonal Naive Benchmark

Definition: Seasonal Naive Forecast

- **Formula**: $\hat{Y}_{T+h} = Y_{T+h-s}$ (last observed season)
- **Monthly example**: Forecast for March 2025 = value from March 2024
- **Interpretation**: "The simplest model that respects seasonality"

Why is it essential?

- Any SARIMA model **must** outperform the seasonal naive benchmark
- If it doesn't $\Rightarrow$ the model complexity is not justified
- Surprisingly effective for many series with stable seasonality

Golden Rule

- **Always** report SARIMA performance relative to seasonal naive
- This is the **first thing** a reviewer or manager checks

The MASE Metric: Proper Evaluation for Seasonal Series

MASE — Mean Absolute Scaled Error (Hyndman & Koehler, 2006)

- **Formula**: $\text{MASE} = \frac{\frac{1}{H}\sum_{h=1}^{H}|e_{T+h}|}{\frac{1}{T-s}\sum_{t=s+1}^{T}|Y_t - Y_{t-s}|}$
- **Numerator**: mean absolute error of the model
- **Denominator**: mean absolute error of seasonal naive (on training data)

Interpretation

- $\text{MASE} < 1$: Model is **better** than seasonal naive
- $\text{MASE} = 1$: Model is **equivalent** to seasonal naive
- $\text{MASE} > 1$: Model is **worse** — abandon it!

Why MASE and not MAPE?

- MAPE: undefined for $Y_t = 0$; asymmetric; scale-dependent
- MASE: works with any data; symmetric; comparable across different series

Forecast Evaluation: Rolling Forecast Origin

Cross-Validation for Seasonal Time Series

- **Principle**: re-estimate model $\rightarrow$ forecast h steps $\rightarrow$ advance 1 step $\rightarrow$ repeat
- **Fixed window**: training on last w observations (constant size)
- **Expanding window**: training from beginning to $T + i$ (grows)

Step-by-step procedure

1. Train SARIMA on $Y_1, \ldots, Y_T$; forecast $\hat{Y}_{T+1}, \ldots, \hat{Y}_{T+h}$
2. Train SARIMA on $Y_1, \ldots, Y_{T+1}$; forecast $\hat{Y}_{T+2}, \ldots, \hat{Y}_{T+h+1}$
3. ... repeat N times; compute RMSE, MAE, MASE on all N forecasts

Important

- Minimum $N \geq 2s$ origins (2 complete seasonal cycles) for reliable results
- Never "look ahead" — test data is strictly after training data

SARIMA vs Holt-Winters/ETS: When to Use Which?

Comparison

Criterion	SARIMA	ETS / Holt-Winters
Approach	Box-Jenkins (ACF/PACF)	Exponential smoothing
Seasonality	Stochastic (differencing)	Additive or multiplicative
Interpretation	AR/MA coefficients	Smoothing weights α, β, γ
Flexibility	Very flexible (7 params.)	Less flexible
Automation	`auto_arima`	`ets()` / `ExponentialSmoothing`

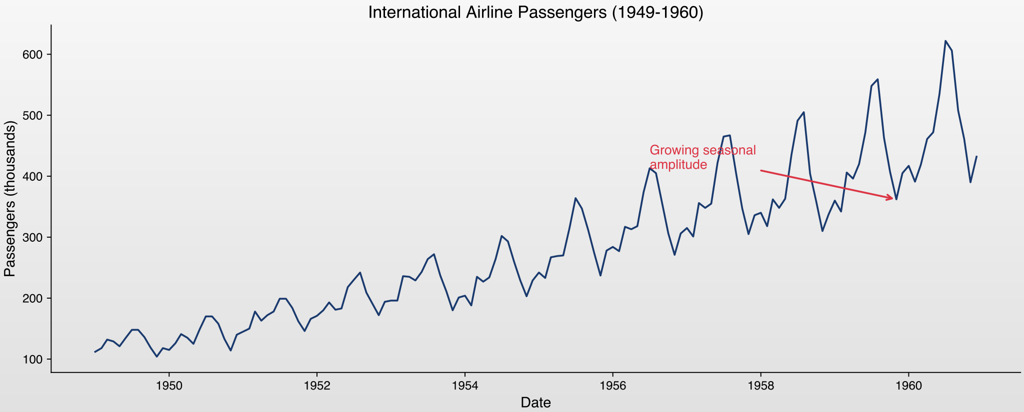
Practical Selection Guide

- **SARIMA preferred**: series with complex autocorrelation, stochastic seasonality, ARMA components
- **ETS preferred**: short series, stable seasonality, quick forecasts without diagnostics
- **Best**: compare both on out-of-sample data and choose the winner

Case Study: Airline Passengers Data

- Classic Box-Jenkins dataset: monthly airline passengers (1949-1960)
- Clear upward trend and increasing seasonal amplitude
- Multiplicative seasonality suggests log transformation

Case Study: Airline Passengers Data

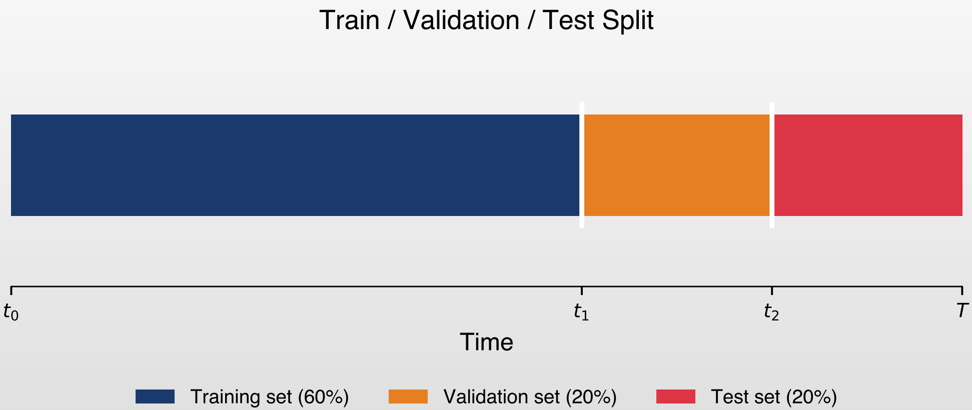


TSA_ch4_case_raw_data

Data Splitting Strategy

- **Training set (70%)** — Fit model parameters
  - Estimate SARIMA coefficients (ϕ, θ, Φ, Θ)
  - Largest portion ensures reliable parameter estimates
- **Validation set (15%)** — Select best model
  - Compare candidate models (different orders)
  - Choose model with lowest validation error
- **Test set (15%)** — Final evaluation
  - Unbiased out-of-sample performance; never used during development

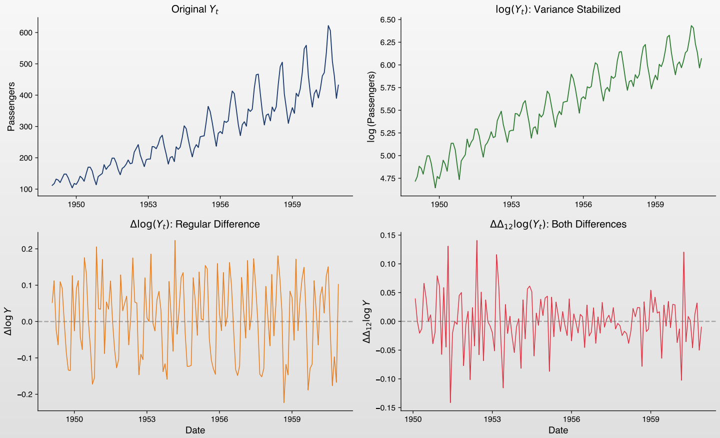
Data Splitting Strategy



Step 1: Transformations

- Log transform stabilizes variance (multiplicative $\rightarrow$ additive)
- First difference removes trend; seasonal difference removes seasonality
- Double-differenced series appears stationary

Step 1: Transformations

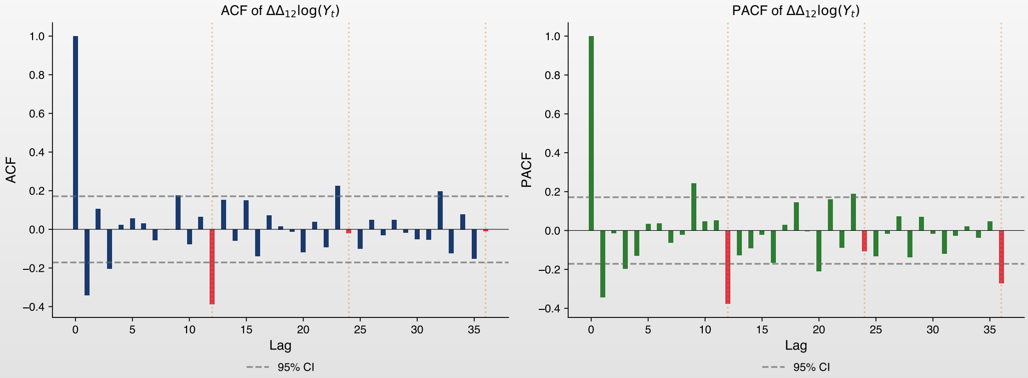


TSA_ch4_case_transformations

Step 2: ACF/PACF Analysis

- ACF: Significant spike at lag 1 and lag 12 $\Rightarrow$ MA(1), SMA(1)
- PACF: Exponential decay pattern confirms MA structure
- Suggests SARIMA$(0, 1, 1) \times (0, 1, 1)_{12}$ (airline model)

Step 2: ACF/PACF Analysis

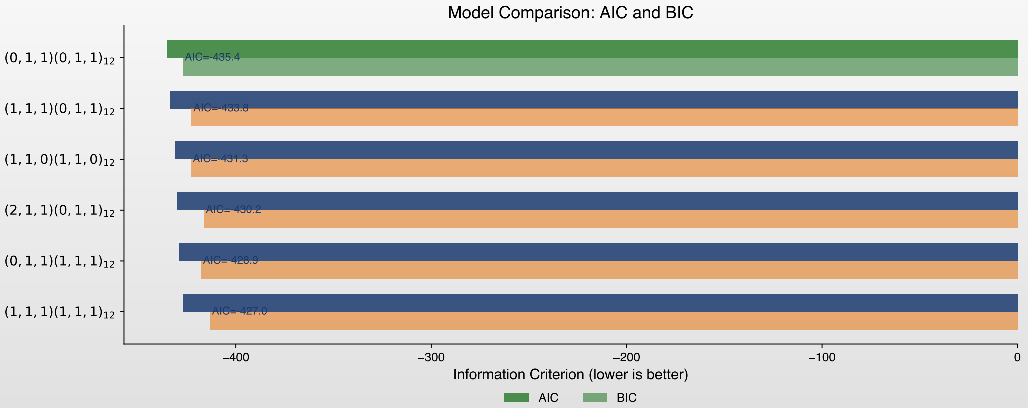


TSA_ch4_case_acf_pacf

Step 3: Model Comparison

- Compare candidate SARIMA models using AIC criterion
- SARIMA$(0,1,1) \times (0,1,1)_{12}$ provides best fit (lowest AIC)
- This is the famous "airline model" identified by Box & Jenkins

Step 3: Model Comparison

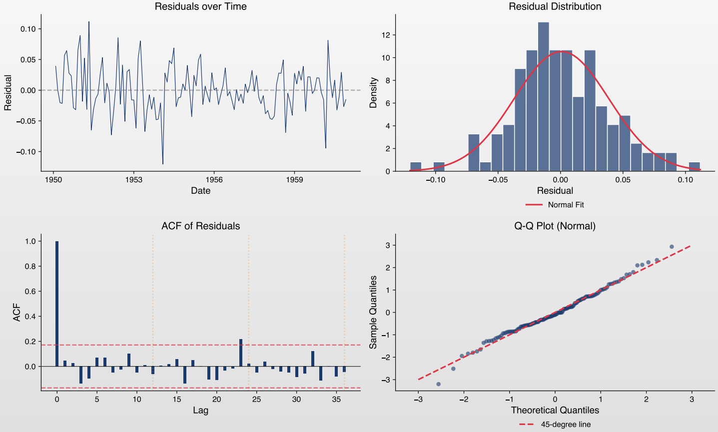


TSA_ch4_case_model_comparison

Step 4: Residual Diagnostics

- Residuals appear random with no remaining autocorrelation
- Q-Q plot shows approximate normality
- Model adequately captures both trend and seasonal structure

Step 4: Residual Diagnostics

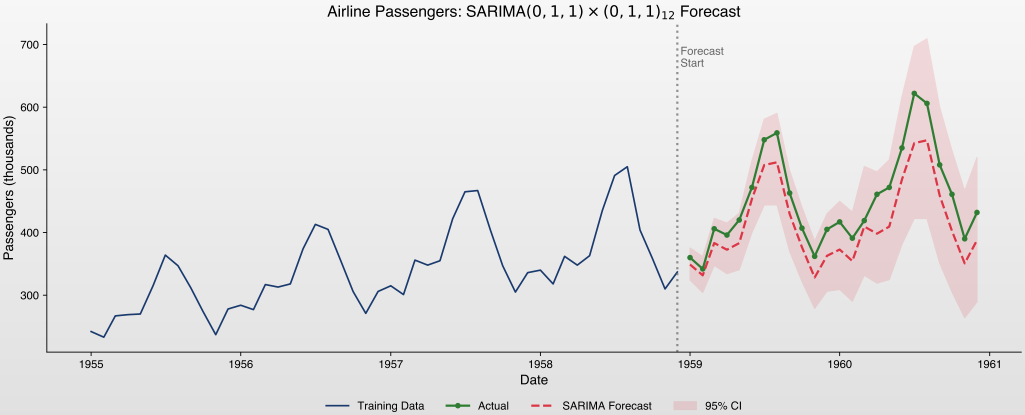


TSA_ch4_case_diagnostics

Step 5: Forecasting

- 24-month forecast with 95% confidence interval
- Model captures seasonal pattern and upward trend
- Prediction intervals widen appropriately with forecast horizon

Step 5: Forecasting



TSA_ch4_case_forecast

Practical Pitfalls in SARIMA Modeling

1. Over-differencing

- **Symptom**: ACF at lag $1 \approx -0.5$ (regular) or at lag $s \approx -0.5$ (seasonal)
- **Cause**: applying $(1 - L)$ or $(1 - L^s)$ too many times
- **Solution**: reduce d or D by 1 and re-examine ACF/PACF

2. Insufficient Data

- **Minimum**: 3–4 complete seasonal cycles (36–48 monthly obs.); **recommended**: 5+ cycles
- Seasonal parameters Φ, Θ are estimated from lags $s, 2s, 3s, \ldots$

3. Other Common Pitfalls

- **Root cancellation**: $\phi \approx \theta$ suggests over-parameterization
- **Parameters at invertibility boundary**: $|\theta| \approx 1$ or $|\Theta| \approx 1$ indicates problems
- **Forgetting inverse transformation**: forecasts on log scale must be back-transformed!

X-13ARIMA-SEATS: Official Seasonal Adjustment

What is seasonal adjustment?

- **Goal**: remove the seasonal component to reveal the true trend
- **Users**: Eurostat, US Census Bureau, central banks, national statistical offices
- **Example**: "GDP grew 0.3% compared to previous quarter" (seasonally adjusted data)

X-13ARIMA-SEATS (US Census Bureau)

- **Step 1**: Identify and estimate a regARIMA model (SARIMA + calendar effects)
- **Step 2**: Extract the seasonal component via SEATS or X-11 filters
- **Step 3**: $Y_t^{adjusted} = Y_t - \hat{S}_t$ (additive) or $Y_t^{adjusted} = Y_t / \hat{S}_t$ (multiplicative)

Why does it matter for economists?

- Published macroeconomic data is almost always seasonally adjusted
- Misinterpreting unadjusted data can lead to erroneous conclusions

Python Implementation: Complete SARIMA

SARIMA Workflow in Python

```
import numpy as np
from scipy.stats import boxcox
from statsmodels.tsa.statespace.sarimax import SARIMAX
import pmdarima as pm
# 1. Box-Cox Transformation
y_bc, lam = boxcox(y)
# 2. Automatic Selection
auto = pm.auto_arima(y_bc, seasonal=True, m=12,
                      d=1, D=1, stepwise=True, trace=True)
# 3. Manual Fit (if preferred)
model = SARIMAX(y_bc, order=(0,1,1),
                seasonal_order=(0,1,1,12))
res = model.fit()
# 4. Diagnostics
res.plot_diagnostics(figsize=(12,8))
# 5. Forecast + Inverse Transformation
fc = res.get_forecast(24)
from scipy.special import inv_boxcox
y_hat = inv_boxcox(fc.predicted_mean, lam)
```

AI Exercise: Critical Thinking

Prompt to test in ChatGPT / Claude / Copilot

"I have the `AirPassengers` dataset from statsmodels (monthly data, international airline passengers, 1949–1960, 144 obs.). Identify seasonality, apply Box-Cox transform if needed, estimate a SARIMA model, and forecast 12 months. Give me complete Python code with plots."

Exercise:

1. Run the prompt in an LLM of your choice and critically analyze the response.
2. Does it check seasonality with ACF at lags $s, 2s, 3s$? Does it use STL decomposition?
3. Does it apply Box-Cox *before* differencing? Does it justify the choice of λ?
4. How does it choose orders $(p, d, q) \times (P, D, Q)_s$? Only auto_arima or also ACF/PACF?
5. Does it evaluate with MASE relative to seasonal naïve? Does it use rolling forecast?

Warning: AI-generated code may run without errors and look professional. *That does not mean it is correct.*

Summary

What We Learned in This Chapter

- Seasonality in time series
  - Repetitive patterns at regular intervals; additive vs multiplicative
- Seasonal differencing and Box-Cox transformation
  - $(1 - L^s)$ removes stochastic seasonality; Box-Cox stabilizes variance
- SARIMA$(p, d, q) \times (P, D, Q)_s$ models
  - Extend ARIMA with seasonal components; automatic selection via `auto_arima`
- Forecasting and evaluation
  - Benchmark: MASE relative to seasonal naive; rolling forecast out-of-sample

Key Idea

- **Parsimony principle**: The Airline Model $(0, 1, 1) \times (0, 1, 1)_{12}$ with only 2 parameters is remarkably effective for many seasonal economic series.

Quick Quiz

Check Your Knowledge

1. What is the difference between deterministic and stochastic seasonality?
2. Why do we apply seasonal differencing $(1 - L^{12})$ and not just regular differencing $(1 - L)$?
3. What ACF pattern indicates the need for seasonal differencing?
4. How do we identify seasonal MA components from ACF?
5. Why do we use MASE instead of RMSE as the main metric for seasonal data?

Quiz Answers

Answers

1. **Types**: Deterministic (dummy variables/Fourier); stochastic (seasonal unit root $\rightarrow$ differencing)
2. **Differencing**: $(1 - L)$ removes trend; $(1 - L^{12})$ removes seasonality. They are complementary, not substitutable.
3. **ACF**: Persistent spikes at lags $s, 2s, 3s$ that decay slowly = stochastic seasonality
4. **SMA**: ACF spike at lag s that cuts off sharply = SMA(1); spikes at s and $2s$ = SMA(2)
5. **MASE**: Scale-free, relative to seasonal naive; RMSE depends on data scale

What's Next?

Chapter 5: Volatility Modeling — GARCH

- **Volatility**: conditional variation of financial returns
- **ARCH/GARCH**: models for conditional variance
- **Asymmetric extensions**: GJR-GARCH, EGARCH (leverage effect)
- **VaR**: Value-at-Risk based on GARCH models
- **Case study**: S&P 500 returns volatility

Questions?

Quiz Question 1

Question

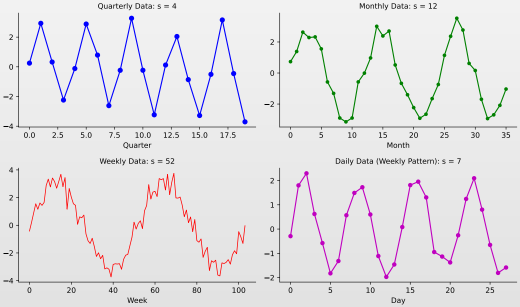
For monthly data with annual seasonality, what is the seasonal period s?

(A) $s = 4$
(B) $s = 7$
(C) $s = 12$
(D) $s = 52$

Quiz Question 1: Answer

Correct Answer: (C) $s = 12$ (12 months per year)

Common periods: Quarterly=4, Monthly=12, Weekly=52, Daily=7, Hourly=24



TSA_ch4_quiz1_seasonal_periods

Quiz Question 2

Question

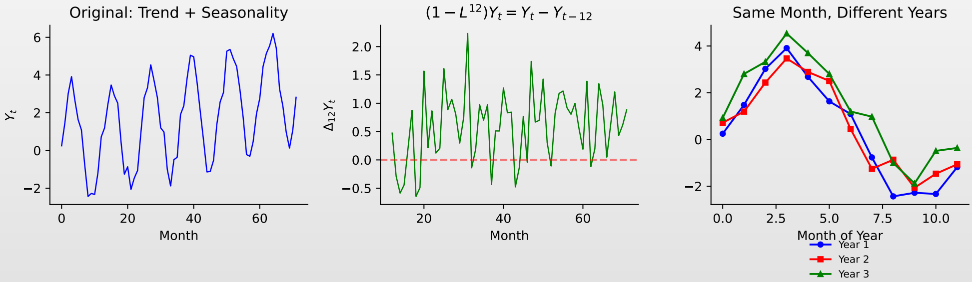
What does the seasonal difference operator $(1 - L^{12})$ do to a monthly series?

(A) Computes $Y_t - Y_{t-1}$ (month-to-month change)
(B) Computes $Y_t - Y_{t-12}$ (year-over-year change)
(C) Computes the 12-month moving average
(D) Removes the trend component only

Quiz Question 2: Answer

Correct Answer: (B) Year-over-year change

$(1 - L^{12})Y_t = Y_t - Y_{t-12}$ removes the seasonal pattern by comparing same months.



TSA_ch4_quiz2_seasonal_diff

Quiz Question 3

Question

In SARIMA$(1,1,1) \times (1,1,1)_{12}$ notation, what does the $(1,1,1)_{12}$ part represent?

(A) AR(1), differencing once, MA(1) at the regular level
(B) Seasonal AR(1), seasonal differencing once, seasonal MA(1)
(C) 12 AR terms, 12 differences, 12 MA terms
(D) The model has 12 parameters in total

Quiz Question 3: Answer

Correct Answer: (B)

Seasonal AR(1), seasonal differencing once, seasonal MA(1)

SARIMA Notation Breakdown

SARIMA$(p, d, q) \times (P, D, Q)_s$:

(p, d, q)	Non-seasonal: AR(p), d differences, MA(q)
$(P, D, Q)_s$	Seasonal: SAR(P), D seasonal diffs, SMA(Q)

For $(1, 1, 1) \times (1, 1, 1)_{12}$:

- Non-seasonal: AR(1), one regular difference, MA(1)
- Seasonal: SAR(1) at lag 12, one Δ_{12}, SMA(1) at lag 12

Quiz Question 4

Question

The "Airline Model" is SARIMA$(0, 1, 1) \times (0, 1, 1)_{12}$. How many parameters need to be estimated (excluding variance)?

(A) 1
(B) 2
(C) 4
(D) 12

Quiz Question 4: Answer

Correct Answer: (B) — 2 parameters

SARIMA$(0,1,1) \times (0,1,1)_{12}$: $(1-L)(1-L^{12})Y_t = (1+\theta_1 L)(1+\Theta_1 L^{12})\varepsilon_t$
Parameters: θ_1 (non-seasonal MA) and Θ_1 (seasonal MA), plus σ^2.

Why "Airline Model"?

Box & Jenkins (1970) used this model to forecast international airline passengers. Remarkably effective for many seasonal economic series!

Quiz Question 5

Question

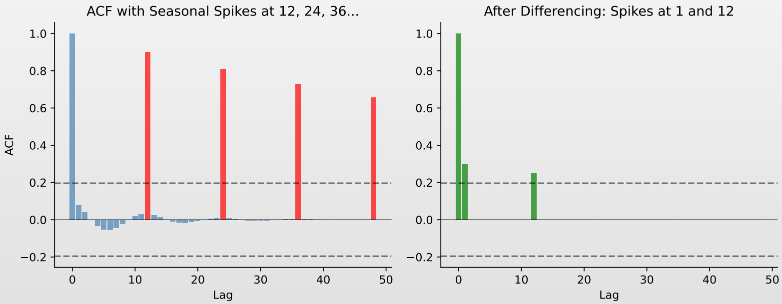
You observe significant ACF spikes at lags 12, 24, and 36 in a monthly series. What does this suggest?

(A) The series has a unit root
(B) The series has annual seasonality that needs seasonal differencing
(C) The series follows an AR(36) process
(D) The series is already stationary

Quiz Question 5: Answer

Correct Answer: (B) Needs seasonal differencing

ACF spikes at 12, 24, 36 = stochastic seasonality. Apply $(1 - L^{12})$ to remove it.



TSA_ch4_quiz5_seasonal_acf

Quiz Question 6

Question

After applying $(1-L)(1-L^{12})$ to a monthly series, the ACF shows a significant spike only at lag 1 and lag 12. What SARIMA model is suggested?

(A) $\text{SARIMA}(1,1,0)\times(1,1,0)_{12}$
(B) $\text{SARIMA}(0,1,1)\times(0,1,1)_{12}$
(C) $\text{SARIMA}(1,1,1)\times(1,1,1)_{12}$
(D) $\text{SARIMA}(0,1,0)\times(0,1,0)_{12}$

Quiz Question 6: Answer

Correct Answer: (B)

- **Model**: SARIMA$(0,1,1) \times (0,1,1)_{12}$ (The Airline Model)

ACF/PACF Identification Rules

- **Rule**: for MA processes, ACF cuts off after lag q
- **ACF spike at lag 1**: MA(1) for non-seasonal part
- **ACF spike at lag 12**: SMA(1) for seasonal part
- **Combined**: MA(1) $\times$ SMA(1) $= (0, d, 1) \times (0, D, 1)_{12}$
- **With** $d = 1$**,** $D = 1$: $(0,1,1) \times (0,1,1)_{12}$

References I

Seasonal Models – Foundational Works

- Box, G.E.P., Jenkins, G.M., Reinsel, G.C., & Ljung, G.M. (2015). *Time Series Analysis: Forecasting and Control*, 5th ed., Wiley.
- Hylleberg, S., Engle, R.F., Granger, C.W.J., & Yoo, B.S. (1990). Seasonal Integration and Cointegration, *Journal of Econometrics*, 44(1-2), 215–238.
- Canova, F., & Hansen, B.E. (1995). Are Seasonal Patterns Constant Over Time?, *Journal of Business & Economic Statistics*, 13(3), 237–252.

Seasonal Decomposition and Diagnostics

- Cleveland, R.B., Cleveland, W.S., McRae, J.E., & Terpenning, I. (1990). STL: A Seasonal-Trend Decomposition Procedure Based on Loess, *Journal of Official Statistics*, 6(1), 3–33.
- Hyndman, R.J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*, 3rd ed., OTexts.

References II

Textbooks and Additional References

- Shumway, R.H., & Stoffer, D.S. (2017). *Time Series Analysis and Its Applications*, 4th ed., Springer.
- Brockwell, P.J., & Davis, R.A. (2016). *Introduction to Time Series and Forecasting*, 3rd ed., Springer.
- Hyndman, R.J., & Khandakar, Y. (2008). Automatic Time Series Forecasting: The `forecast` Package for R, *Journal of Statistical Software*, 27(3), 1–22.

Online Resources and Code

- **Quantlet**: `https://quantlet.com` → Code platform for statistics
- **Quantinar**: `https://quantinar.com` → Learning platform for quantitative methods
- **GitHub TSA**: `https://github.com/QuantLet/TSA/tree/main/TSA_ch4` → Python code for this chapter

Thank You!

Questions?

Course materials available at: `https://danpele.github.io/Time-Series-Analysis/`

Quantlet Quantinar