Time Series Analysis

Lecture Notes

Master M1 — 2025–2026

Yaë Ulrich Gaba

"Prediction is very difficult, especially about the future."
— Niels Bohr

March 25, 2026

Contents

Preface

Time-indexed data are ubiquitous: stock prices recorded every minute, daily temperatures, quarterly GDP, EEG signals, network traffic. **Time series analysis** provides the mathematical framework for modeling the *temporal dependence* inherent in such data, and exploiting this structure for *forecasting*, *anomaly detection*, and *understanding* the underlying mechanisms.

This course, aimed at Master students in statistics, economics, and data science, covers classical theory (ARMA processes, spectral analysis, Kalman filter) and modern developments (GARCH, cointegration, VAR models). Each concept is illustrated with real data and accompanied by Python implementations.

Prerequisites. Probability theory (convergences, conditional expectation), mathematical statistics (estimation, testing), real analysis, linear algebra.

References.

- Box, Jenkins, Reinsel, Ljung — *Time Series Analysis: Forecasting and Control*, Wiley.
- Brockwell, Davis — *Introduction to Time Series and Forecasting*, Springer.
- Hamilton — *Time Series Analysis*, Princeton University Press.
- Shumway, Stoffer — *Time Series Analysis and Its Applications*, Springer.
- Gouriéroux, Monfort — *Séries Temporelles et Modèles Dynamiques*, Economica.

Chapter 1

Introduction to Time Series

1.1 Motivating example

The daily closing price of the CAC 40 index from 2000 to 2024 exhibits a trend, cycles, and time-varying volatility. Unlike iid data, each observation depends on the preceding ones. Time series analysis exploits this dependence.

1.2 Fundamental definitions

Definition 1.1 (Time series)**.** A **time series** (or **discrete-time stochastic process**) is a sequence of random variables $(X_t)_{t\in\mathbb{Z}}$ defined on a probability space $(\Omega, \mathcal{F}, P)$. A **realisation** (or sample path) is an observed sequence $(x_1, \ldots, x_n)$.

Definition 1.2 (White noise)**.** A **white noise** (ε_t) is a sequence of random variables such that:

1. $\mathbb{E}[\varepsilon_t] = 0$ for all t,
2. $\mathrm{Var}(\varepsilon_t) = \sigma^2$ for all t,
3. $\mathrm{Cov}(\varepsilon_t, \varepsilon_s) = 0$ for $t \neq s$.

Notation: $\varepsilon_t \sim \mathrm{WN}(0, \sigma^2)$. If, in addition, the ε_t are Gaussian, it is called a **Gaussian white noise**.

Definition 1.3 (Strong white noise)**.** A **strong white noise** is an iid sequence of random variables with zero mean and finite variance. A (weak) white noise only requires uncorrelatedness, not independence.

1.3 Components of a time series

Definition 1.4 (Classical decomposition)**.** A time series is often decomposed as:

$$X_t = T_t + S_t + R_t,$$

where T_t is the **trend**, S_t the **seasonality** (periodic component), and R_t the **residual** (random component).

Multiplicative model: $X_t = T_t \cdot S_t \cdot R_t$ (common when the seasonal amplitude grows with the level).

1.4 Lag and difference operators

Definition 1.5 (Lag operator)**.** The **lag operator** B (or *backshift operator*) is defined by:

$$BX_t = X_{t-1}, \quad B^k X_t = X_{t-k}.$$

The **difference operator** is $\nabla = 1 - B$: $\nabla X_t = X_t - X_{t-1}$.

Remark 1.6. Polynomials in B are manipulated as ordinary polynomials. For example, $(1 - \phi B)X_t = \varepsilon_t$ can be written as $X_t - \phi X_{t-1} = \varepsilon_t$.

1.5 Examples of time series

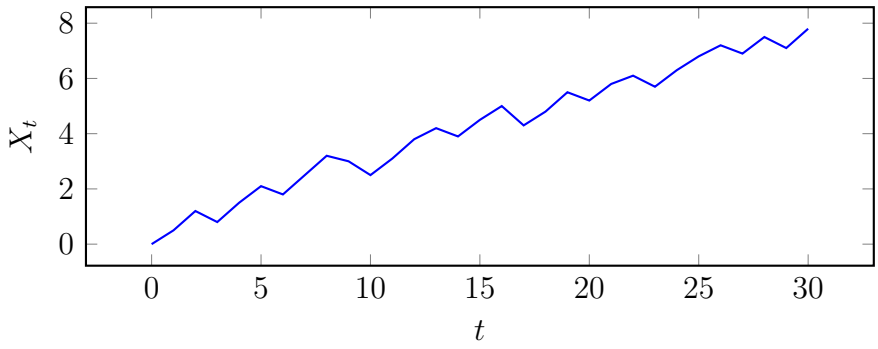
Example 1.7 (Random walk)**.** $X_t = X_{t-1} + \varepsilon_t$ with $\varepsilon_t \sim \mathrm{WN}(0, \sigma^2)$. Then $X_t = X_0 + \sum_{i=1}^{t} \varepsilon_i$ and $\mathrm{Var}(X_t) = t\sigma^2 \to \infty$: the random walk is **not stationary**.

Example 1.8 (Linear trend + noise)**.** $X_t = a + bt + \varepsilon_t$. Not stationary (the mean $\mathbb{E}[X_t] = a + bt$ depends on t).

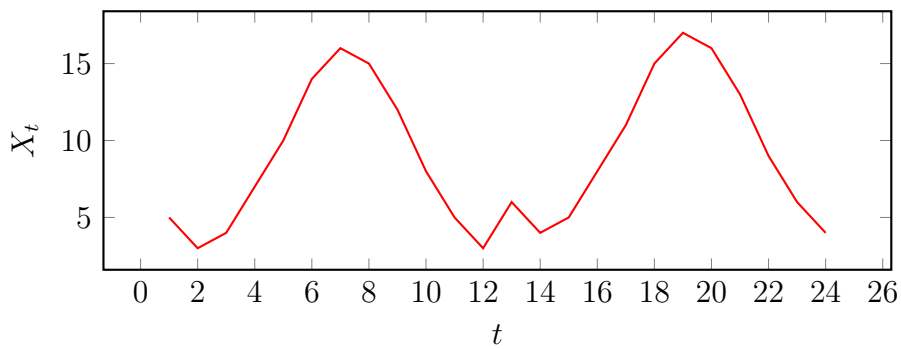
Example 1.9 (Seasonal series)**.** Monthly temperatures: period $s = 12$. The component S_t is modelled by trigonometric functions or monthly indicator variables.

1.6 Visualisation

Realisation of a random walk



Series with seasonality ($s = 12$)



1.7 Implementation

Python

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Simuler une marche aléatoire
np.random.seed(42)
n = 500
eps = np.random.normal(0, 1, n)
rw = np.cumsum(eps)

# Simuler un AR(1)
phi = 0.8
ar1 = np.zeros(n)
for t in range(1, n):
    ar1[t] = phi * ar1[t-1] + eps[t]

# Charger des données réelles (exemple : AirPassengers)
# url =
↪  'https://raw.githubusercontent.com/jbrownlee/Datasets/master/airline-passengers.csv'
# df = pd.read_csv(url, parse_dates=['Month'], index_col='Month')

fig, axes = plt.subplots(2, 1, figsize=(12, 6))
axes[0].plot(rw, lw=0.8)
axes[0].set_title('Marche aléatoire')
axes[0].set_xlabel('t')

axes[1].plot(ar1, lw=0.8, color='red')
axes[1].set_title('AR(1) avec $\\phi=0.8$')
axes[1].set_xlabel('t')

plt.tight_layout()
plt.savefig('ch01_intro.pdf')
plt.show()
```

1.8 Exercises

Exercise 1.1 ($\star$)**.** Simulate 1000 realisations of a random walk of length $n = 200$. Plot 10 sample paths on the same graph. Compute the empirical variance of X_{200}.

Exercise 1.2 ($\star$)**.** Load the `AirPassengers` series. Visually identify the trend, seasonality, and increasing volatility. Suggest an additive or multiplicative model.

Exercise 1.3 ($\star\star$)**.** Show that if $X_t = X_{t-1} + \varepsilon_t$ (random walk), then $\mathrm{Var}(X_t) = t\sigma^2$ and $\mathrm{Corr}(X_t, X_s) = \sqrt{\min(t,s)/\max(t,s)}$ for $t, s > 0$.

Exercise 1.4 ($\star\star\star$ – Project)**.** Download the daily prices of a stock (e.g. Apple) over 5

years. Decompose the series into trend, seasonality, and residual. Compute the log-returns $r_t = \ln(P_t/P_{t-1})$ and analyse their statistical properties.

Key Formulas

$$BX_t = X_{t-1}, \quad \nabla X_t = X_t - X_{t-1}$$

$$X_t = T_t + S_t + R_t \quad \text{(additive decomposition)}$$

$$\text{Random walk} : X_t = X_{t-1} + \varepsilon_t, \quad \text{Var}(X_t) = t\sigma^2$$

Chapter 2

Stationarity and Autocovariance

2.1 Motivating example

The daily temperature in Paris over 30 years exhibits a pronounced seasonality, but after deseasonalisation, the residuals appear to fluctuate around a constant mean with a time-invariant dependence structure. This is **stationarity**.

2.2 Strict and weak stationarity

Definition 2.1 (Strict stationarity)**.** (X_t) is **strictly stationary** if for all $k \geq 1$, all $(t_1, \ldots, t_k)$, and all $h \in \mathbb{Z}$:

$$(X_{t_1}, \ldots, X_{t_k}) \stackrel{d}{=} (X_{t_1+h}, \ldots, X_{t_k+h}).$$

Definition 2.2 (Weak stationarity (second-order))**.** (X_t) is **weakly stationary** (or wide-sense stationary) if:

1. $\mathbb{E}[X_t^2] < \infty$ for all t,
2. $\mu = \mathbb{E}[X_t]$ does not depend on t,
3. $\gamma(t, t+h) = \mathrm{Cov}(X_t, X_{t+h})$ depends only on h.

Remark 2.3. Strict stationarity + finite second moments $\Rightarrow$ weak stationarity. The converse is false in general, but holds for Gaussian processes.

2.3 Autocovariance function

Definition 2.4 (Autocovariance and autocorrelation)**.** For a weakly stationary process:

$$\gamma(h) = \mathrm{Cov}(X_t, X_{t+h}) = \mathbb{E}[(X_t - \mu)(X_{t+h} - \mu)], \tag{2.1}$$

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \mathrm{Corr}(X_t, X_{t+h}). \tag{2.2}$$

The function $\rho(\cdot)$ is the **autocorrelation function** (ACF).

Theorem 2.5 (Properties of γ)**.** *1.* $\gamma(0) \geq 0$ *(it is the variance).*

2. $\gamma(h) = \gamma(-h)$ *(symmetry).*

3. $|\gamma(h)| \le \gamma(0)$ *(by Cauchy–Schwarz).*

4. γ *is* ***positive semi-definite****: for all* n*, all* $(a_1, \dots, a_n) \in \mathbb{R}^n$*, and all* $(t_1, \dots, t_n)$*:*

$$\sum_{i,j} a_i a_j \gamma(t_i - t_j) \ge 0.$$

Proof. (4): $\sum_{i,j} a_i a_j \gamma(t_i - t_j) = \mathrm{Var}(\sum_i a_i X_{t_i}) \ge 0$. □

Theorem 2.6 (Herglotz)**.** $\gamma : \mathbb{Z} \to \mathbb{R}$ *is an autocovariance function if and only if it is positive semi-definite. Moreover, there exists a finite positive measure* F *on* $[-\pi, \pi]$ *(spectral measure) such that:*

$$\gamma(h) = \int_{-\pi}^{\pi} e^{i\omega h}\, dF(\omega).$$

2.4 Estimation of the autocovariance

Definition 2.7 (Sample autocovariance)**.**

$$\hat\gamma(h) = \frac{1}{n} \sum_{t=1}^{n-|h|} (X_t - \bar X)(X_{t+|h|} - \bar X), \quad \hat\rho(h) = \frac{\hat\gamma(h)}{\hat\gamma(0)}.$$

Dividing by n (rather than $n - |h|$) ensures that the matrix $(\hat\gamma(i-j))$ is positive semi-definite.

Theorem 2.8 (Bartlett)**.** *For a white noise of size* n*:* $\hat\rho(h) \overset{approx}{\sim} \mathcal{N}(0, 1/n)$ *for* $h \ge 1$*. The* ***Bartlett bands*** $\pm 1.96/\sqrt{n}$ *are used to test whether* $\rho(h) = 0$*.*

2.5 Partial autocorrelation

Definition 2.9 (PACF)**.** The **partial autocorrelation** at lag h is:

$$\alpha(h) = \mathrm{Corr}(X_t - \hat X_t,\ X_{t+h} - \hat X_{t+h}),$$

where $\hat X_t$ and $\hat X_{t+h}$ are the linear projections onto $(X_{t+1}, \dots, X_{t+h-1})$.

Equivalently, $\alpha(h)$ is the last coefficient ϕ_{hh} in the linear regression of X_t on $(X_{t-1}, \dots, X_{t-h})$.

Theorem 2.10. *For an AR(*p*):* $\alpha(h) = 0$ *for* $h > p$*. This is the main tool for identifying the order* p*.*

2.6 Stationarity tests

Definition 2.11 (Augmented Dickey–Fuller test (ADF))**.** To test for the presence of a unit root in $X_t = \phi X_{t-1} + \varepsilon_t$:

$$H_0 : \phi = 1 \quad \text{(non-stationary)} \quad \text{vs} \quad H_1 : |\phi| < 1 \quad \text{(stationary)}.$$

One estimates $\nabla X_t = (\phi - 1) X_{t-1} + \varepsilon_t$ and tests whether $\phi - 1 < 0$.

Warning

The distribution of the ADF statistic is **not** Gaussian under H_0. One must use the Dickey–Fuller tables (or critical values computed by simulation).

2.7 Implementation

Python

```python
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.stattools import acf, pacf, adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Simuler un AR(1)
np.random.seed(0)
n = 500
phi = 0.7
eps = np.random.normal(0, 1, n)
x = np.zeros(n)
for t in range(1, n):
    x[t] = phi * x[t-1] + eps[t]

# ACF et PACF
fig, axes = plt.subplots(1, 2, figsize=(12, 4))
plot_acf(x, lags=30, ax=axes[0], title='ACF')
plot_pacf(x, lags=30, ax=axes[1], title='PACF')
plt.tight_layout()
plt.savefig('ch02_acf_pacf.pdf')
plt.show()

# Test ADF
result = adfuller(x, regression='c')
print(f"ADF statistic: {result[0]:.4f}")
print(f"p-value: {result[1]:.4f}")
print(f"Stationnaire: {result[1] < 0.05}")

# Marche aléatoire (non stationnaire)
rw = np.cumsum(eps)
result_rw = adfuller(rw, regression='c')
print(f"\nMarche aléatoire - ADF: {result_rw[0]:.4f},
↪  p={result_rw[1]:.4f}")
```

2.8 Exercises

Exercise 2.1 ($\star$)**.** Compute the autocovariance function of a white noise $\varepsilon_t \sim \mathrm{WN}(0, \sigma^2)$.

Exercise 2.2 ($\star$)**.** Compute the ACF of $X_t = \varepsilon_t + \theta\varepsilon_{t-1}$ (MA(1)). Show that $\rho(h) = 0$ for $|h| \geq 2$.

Exercise 2.3 ($\star\star$)**.** Show that if (X_t) is strictly stationary with $\mathbb{E}[X_t^2] < \infty$, then (X_t) is weakly stationary.

Exercise 2.4 ($\star\star$)**.** Prove that the sample autocovariance $\hat{\gamma}(h)$ (divided by n) yields a positive semi-definite Toeplitz matrix.

Exercise 2.5 ($\star\star\star$ – Project)**.** Apply the ADF test and the KPSS test to 10 economic time series (GDP, unemployment, inflation, etc.). Compare the conclusions and discuss cases of disagreement.

Key Formulas

$$\gamma(h) = \mathrm{Cov}(X_t, X_{t+h}), \quad \rho(h) = \gamma(h)/\gamma(0)$$
$$\hat{\rho}(h) \overset{\text{approx}}{\sim} \mathcal{N}(0, 1/n) \quad \text{under } H_0 : \rho(h) = 0$$
$$\text{ADF} : \nabla X_t = (\phi - 1)X_{t-1} + \varepsilon_t, \quad H_0 : \phi = 1$$

Chapter 3

AR, MA, ARMA Models

3.1 Motivating example

The daily log-returns of the CAC 40 exhibit weak but significant correlation at the first few lags. An AR(1) or ARMA(1,1) model captures this linear dependence.

3.2 Autoregressive processes AR(p)

Definition 3.1 (AR(p))**.**

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t,$$

where $\varepsilon_t \sim \mathrm{WN}(0, \sigma^2)$. In operator notation: $\phi(B)X_t = \varepsilon_t$ with $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$.

Theorem 3.2 (Stationarity of an AR(p))**.** *The AR(p) admits a unique stationary solution if and only if all roots of $\phi(z) = 0$ lie* ***outside the unit disc****: $|z_i| > 1$ for all i.*

Sketch for AR(1). $X_t = \phi X_{t-1} + \varepsilon_t$. By recursion: $X_t = \sum_{j=0}^{\infty} \phi^j \varepsilon_{t-j}$. This series converges in L^2 if and only if $\sum \phi^{2j} < \infty$, i.e. $|\phi| < 1$. The root of $\phi(z) = 1 - \phi z$ is $z = 1/\phi$, which lies outside the disc if and only if $|\phi| < 1$. □

Theorem 3.3 (ACF of an AR(1))**.** *If $X_t = \phi X_{t-1} + \varepsilon_t$ with $|\phi| < 1$:*

$$\gamma(h) = \frac{\sigma^2 \phi^{|h|}}{1 - \phi^2}, \quad \rho(h) = \phi^{|h|}.$$

The ACF decays exponentially.

Proof. $\gamma(h) = \mathbb{E}[X_t X_{t+h}] = \phi\gamma(h-1)$ for $h \geq 1$ (by multiplying the AR equation by X_{t+h} and taking expectations). With $\gamma(0) = \sigma^2/(1 - \phi^2)$. □

Theorem 3.4 (PACF of an AR(p))**.** *For an AR(p): $\alpha(h) = 0$ for all $h > p$. The PACF truncates at order p.*

3.3 Yule–Walker equations

Definition 3.5. For a stationary AR(p), the relations:

$$\gamma(h) = \phi_1 \gamma(h-1) + \cdots + \phi_p \gamma(h-p) \quad (h \geq 1)$$

yield the **Yule–Walker** system:

$$\begin{pmatrix} \gamma(0) & \gamma(1) & \cdots & \gamma(p-1) \\ \gamma(1) & \gamma(0) & \cdots & \gamma(p-2) \\ \vdots & & \ddots & \vdots \\ \gamma(p-1) & \gamma(p-2) & \cdots & \gamma(0) \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix} = \begin{pmatrix} \gamma(1) \\ \gamma(2) \\ \vdots \\ \gamma(p) \end{pmatrix}.$$

3.4 Moving average processes MA(q)

Definition 3.6 (MA(q))**.**

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} = \theta(B)\varepsilon_t,$$

where $\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$.

Theorem 3.7 (Stationarity of an MA(q))**.** *Any MA(q) is **always stationary** (finite sum of white noise terms).*

Theorem 3.8 (ACF of an MA(q))**.**

$$\gamma(h) = \begin{cases} \sigma^2 \sum_{j=0}^{q-|h|} \theta_j \theta_{j+|h|} & \text{if } |h| \le q, \\ 0 & \text{if } |h| > q, \end{cases}$$

*with $\theta_0 = 1$. The ACF **truncates** at lag q.*

Definition 3.9 (Invertibility)**.** An MA(q) is **invertible** if all roots of $\theta(z) = 0$ lie outside the unit disc. In that case, X_t admits an AR(∞) representation: $\pi(B)X_t = \varepsilon_t$ with $\pi(z) = \theta(z)^{-1}$.

3.5 ARMA(p, q) processes

Definition 3.10 (ARMA(p, q))**.**

$$\phi(B)X_t = \theta(B)\varepsilon_t, \quad \text{i.e.} \quad X_t - \sum_{i=1}^{p} \phi_i X_{t-i} = \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j}.$$

Theorem 3.11 (Stationarity and invertibility conditions)**.**

- *Stationary $\iff$ roots of $\phi(z) = 0$ outside the unit disc.*
- *Invertible $\iff$ roots of $\theta(z) = 0$ outside the unit disc.*
- *One always requires that $\phi(z)$ and $\theta(z)$ share **no common root** (otherwise the model is over-parameterised).*

Theorem 3.12 (MA(∞) representation)**.** *If the ARMA(p, q) is stationary: $X_t = \psi(B)\varepsilon_t$ with $\psi(z) = \theta(z)/\phi(z) = \sum_{j=0}^{\infty} \psi_j z^j$. The ψ_j are the **impulse response weights**.*

3.6 ACF/PACF signatures

Model	ACF	PACF
AR(p)	Exponential/sinusoidal decay	Truncates after p
MA(q)	Truncates after q	Exponential/sinusoidal decay
ARMA(p, q)	Decay after q	Decay after p

3.7 Implementation

Python

```python
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_process import ArmaProcess
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Simuler un AR(2)
np.random.seed(42)
ar_params = np.array([1, -0.75, 0.25])  # 1 - 0.75*B + 0.25*B^2
ma_params = np.array([1])
ar2 = ArmaProcess(ar_params, ma_params)
x = ar2.generate_sample(nsample=500)

fig, axes = plt.subplots(1, 3, figsize=(15, 4))
axes[0].plot(x, lw=0.7)
axes[0].set_title('AR(2): $\\phi_1=0.75, \\phi_2=-0.25$')
plot_acf(x, lags=25, ax=axes[1], title='ACF')
plot_pacf(x, lags=25, ax=axes[2], title='PACF')
plt.tight_layout()
plt.savefig('ch03_arma.pdf')
plt.show()

# Simuler un MA(2)
ma_params = np.array([1, 0.6, -0.3])
ma2 = ArmaProcess(np.array([1]), ma_params)
y = ma2.generate_sample(nsample=500)

# Simuler un ARMA(1,1)
arma11 = ArmaProcess(np.array([1, -0.8]), np.array([1, 0.4]))
z = arma11.generate_sample(nsample=500)

# Yule-Walker
from statsmodels.regression.linear_model import yule_walker
rho, sigma = yule_walker(x, order=2)
print(f"Yule-Walker: phi = {rho}")
```

3.8 Exercises

Exercise 3.1 ($\star$)**.** Compute the ACF of an MA(1): $X_t = \varepsilon_t + 0.5\varepsilon_{t-1}$. Verify numerically.

Exercise 3.2 ($\star\star$)**.** Show that the AR(1) $X_t = \phi X_{t-1} + \varepsilon_t$ with $|\phi| < 1$ has an MA(∞) representation: $X_t = \sum_{j=0}^{\infty} \phi^j \varepsilon_{t-j}$.

Exercise 3.3 ($\star\star$)**.** Show that an ARMA(1,1) with $\phi = \theta$ reduces to a white noise (common root).

Exercise 3.4 ($\star\star$)**.** Derive the Yule–Walker equations for an AR(2) and solve the system.

Exercise 3.5 ($\star\star\star$ – Project)**.** Simulate AR(1), AR(2), MA(1), MA(2), ARMA(1,1), and ARMA(2,1) processes. For each, plot the theoretical and empirical ACF and PACF. Build a visual identification guide.

Key Formulas

$$\begin{aligned} \text{AR}(p) &: \phi(B)X_t = \varepsilon_t, \quad \text{stationary iff roots of } \phi(z) \text{ outside disc} \\ \text{MA}(q) &: X_t = \theta(B)\varepsilon_t, \quad \text{always stationary} \\ \text{ACF AR}(1) &: \rho(h) = \phi^{|h|} \\ \text{ACF MA}(q) &: \rho(h) = 0 \text{ for } |h| > q \end{aligned}$$

Chapter 4

Identification, Estimation, and Diagnostics

4.1 Motivating example

Given a real time series, how does one choose the orders p and q of an ARMA(p, q), estimate the parameters, and verify that the model is adequate? This is the **Box–Jenkins methodology**.

4.2 The Box–Jenkins methodology

Method

1. **Identification**: examine the ACF and PACF to propose orders (p, q).
2. **Estimation**: estimate $(\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q, \sigma^2)$ by maximum likelihood or conditional least squares.
3. **Diagnostics**: verify that the residuals behave as white noise.
4. **Forecasting**: use the validated model for prediction.

4.3 Identification

4.3.1 Examining the ACF and PACF

Practical rules (summarised in Chapter 3):

- ACF truncates at $q \Rightarrow$ MA(q).
- PACF truncates at $p \Rightarrow$ AR(p).
- Both decay $\Rightarrow$ ARMA(p, q).

4.3.2 Information criteria

Definition 4.1 (AIC and BIC)**.**

$$\mathrm{AIC}(p,q) = -2\ln L(\hat{\boldsymbol{\theta}}) + 2(p+q+1), \tag{4.1}$$

$$\mathrm{BIC}(p,q) = -2\ln L(\hat{\boldsymbol{\theta}}) + (p+q+1)\ln n. \tag{4.2}$$

One selects the model that **minimises** the criterion. The BIC penalises complexity more heavily.

Theorem 4.2 (Asymptotic properties)**.**

- *The AIC is asymptotically equivalent to leave-one-out cross-validation.*
- *The BIC is **consistent**: it selects the true model with probability* $\to 1$*.*
- *The AIC tends to over-parameterise.*

4.4 Estimation

4.4.1 Maximum likelihood

Definition 4.3. For a Gaussian ARMA(p,q), the exact log-likelihood is:

$$\ln L(\boldsymbol{\theta}) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\ln v_t - \frac{1}{2}\sum_{t=1}^{n}\frac{e_t^2}{v_t},$$

where $e_t = X_t - \hat{X}_{t|t-1}$ is the one-step-ahead prediction error and $v_t = \mathrm{Var}(e_t)$, computed recursively (innovations algorithm or Kalman filter).

Theorem 4.4 (Properties of the MLE)**.** *Under regularity conditions, the MLE* $\hat{\boldsymbol{\theta}}_n$ *is:*

- *Consistent:* $\hat{\boldsymbol{\theta}}_n \xrightarrow{P} \boldsymbol{\theta}_0$*.*
- *Asymptotically normal:* $\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \xrightarrow{d} \mathcal{N}(0, I(\boldsymbol{\theta}_0)^{-1})$*.*
- *Asymptotically efficient.*

4.4.2 Conditional least squares

Definition 4.5. Conditioning on the first observations, one minimises:

$$S(\boldsymbol{\theta}) = \sum_{t=p+1}^{n}\left(X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} - \theta_1\hat{\varepsilon}_{t-1} - \cdots - \theta_q\hat{\varepsilon}_{t-q}\right)^2.$$

4.5 Diagnostics

4.5.1 Residual analysis

Definition 4.6. The model residuals are $\hat{\varepsilon}_t = X_t - \hat{X}_{t|t-1}$. If the model is correct, the $\hat{\varepsilon}_t$ should behave as white noise.

Checks:

1. **ACF of residuals**: no significant autocorrelation.
2. **Ljung–Box test**:

$$Q(m) = n(n+2)\sum_{h=1}^{m} \frac{\hat{\rho}_{\hat{\varepsilon}}(h)^2}{n-h} \overset{H_0}{\sim} \chi^2(m-p-q).$$

3. **Normality**: QQ-plot, Jarque–Bera test.
4. **Heteroscedasticity**: ARCH effect (cf. Chapter 6).

Theorem 4.7 (Ljung–Box). *Under H_0 (residuals = white noise), $Q(m) \sim \chi^2(m-p-q)$ asymptotically. One rejects H_0 if $Q(m) > \chi^2_{1-\alpha}(m-p-q)$.*

4.6 Implementation

Python

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.stats.diagnostic import acorr_ljungbox
from statsmodels.graphics.tsaplots import plot_acf

# Simuler un ARMA(1,1) connu
np.random.seed(42)
from statsmodels.tsa.arima_process import ArmaProcess
true_ar = np.array([1, -0.7])
true_ma = np.array([1, 0.3])
process = ArmaProcess(true_ar, true_ma)
data = process.generate_sample(nsample=500)

# Étape 1 : Identification (ACF/PACF déjà examinée)

# Étape 2 : Estimation par MLE
model = ARIMA(data, order=(1, 0, 1))
results = model.fit()
print(results.summary())
print(f"\nphi_1 = {results.arparams[0]:.4f} (vrai: 0.7)")
print(f"theta_1 = {results.maparams[0]:.4f} (vrai: 0.3)")

# Étape 3 : Diagnostic
residuals = results.resid

fig, axes = plt.subplots(2, 2, figsize=(12, 8))
axes[0,0].plot(residuals, lw=0.5)
axes[0,0].set_title('Résidus')
plot_acf(residuals, lags=25, ax=axes[0,1], title='ACF des résidus')
```

```
axes[1,0].hist(residuals, bins=30, density=True, alpha=0.7)
axes[1,0].set_title('Distribution des résidus')
from scipy import stats
stats.probplot(residuals, dist='norm', plot=axes[1,1])
axes[1,1].set_title('QQ-plot')
plt.tight_layout()
plt.savefig('ch04_diagnostic.pdf')
plt.show()

# Test de Ljung-Box
lb_test = acorr_ljungbox(residuals, lags=20, return_df=True)
print(lb_test)

# Sélection par AIC/BIC
results_table = []
for p in range(4):
    for q in range(4):
        try:
            m = ARIMA(data, order=(p, 0, q)).fit()
            results_table.append({'p': p, 'q': q,
                                  'AIC': m.aic, 'BIC': m.bic})
        except:
            pass
df = pd.DataFrame(results_table).sort_values('BIC')
print(df.head(5))
```

4.7 Exercises

Exercise 4.1 ($\star$)**.** Simulate an AR(2) and apply the complete methodology: identification, estimation, diagnostics.

Exercise 4.2 ($\star\star$)**.** Show that the Ljung–Box test with m lags follows asymptotically a $\chi^2(m-p-q)$ distribution under H_0.

Exercise 4.3 ($\star\star$)**.** Compare AIC and BIC on 1000 simulations of an AR(1). Which one selects the correct order most often? Does the BIC over-penalise in small samples?

Exercise 4.4 ($\star\star\star$ – Project)**.** Apply the complete Box–Jenkins methodology to the French monthly unemployment rate series. Present each step with justification.

Key Formulas

$$\text{AIC} = -2\ln L + 2k, \quad \text{BIC} = -2\ln L + k\ln n$$

$$Q(m) = n(n+2)\sum_{h=1}^{m}\frac{\hat{\rho}_{\hat{\varepsilon}}(h)^2}{n-h} \sim \chi^2(m-p-q)$$

Chapter 5

ARIMA Models and Differencing

5.1 Motivating example

The French quarterly GDP exhibits an upward trend: it is not stationary. After differencing ($\nabla X_t = X_t - X_{t-1}$), the growth rate is approximately stationary and can be modelled by an ARMA process.

5.2 Integrated processes

Definition 5.1 (Integrated process of order d)**.** (X_t) is **integrated of order** d, denoted $X_t \sim I(d)$, if $\nabla^d X_t = (1-B)^d X_t$ is stationary but $\nabla^{d-1} X_t$ is not.

Example 5.2. The random walk $X_t = X_{t-1} + \varepsilon_t$ is $I(1)$ since $\nabla X_t = \varepsilon_t$ is stationary.

5.3 ARIMA(p, d, q) model

Definition 5.3 (ARIMA(p, d, q))**.**

$$\phi(B)(1-B)^d X_t = \theta(B)\varepsilon_t.$$

X_t follows an ARIMA(p, d, q) if $W_t = \nabla^d X_t$ follows an ARMA(p, q).

Warning

One differences to achieve stationarity, but **over-differencing** (d too large) introduces MA roots close to the unit circle, which degrades forecasts. In practice, $d \leq 2$.

5.4 SARIMA: seasonality

Definition 5.4 (SARIMA$(p, d, q) \times (P, D, Q)_s$)**.** For a series with period s:

$$\Phi(B^s)\phi(B)(1-B)^d(1-B^s)^D X_t = \Theta(B^s)\theta(B)\varepsilon_t,$$

where Φ, Θ are the seasonal polynomials of orders P, Q.

Example 5.5 (Airline model)**.** The Box–Jenkins model for AirPassengers is a SARIMA$(0,1,1)\times(0,1,1)_{12}$:

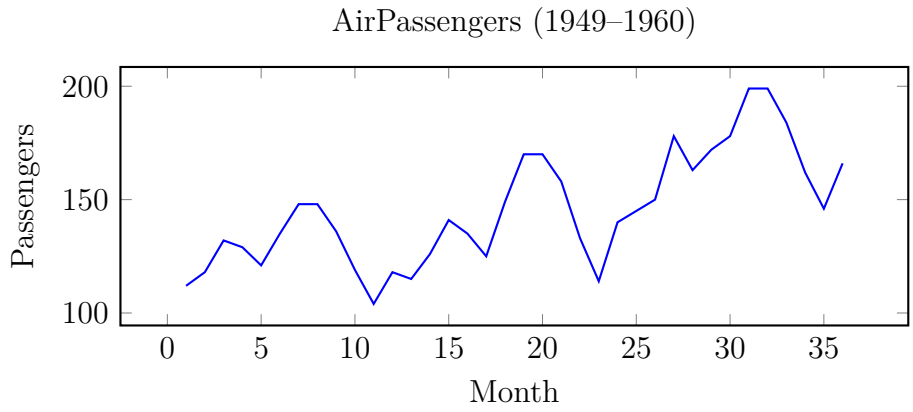
$$(1-B)(1-B^{12})X_t = (1+\theta_1 B)(1+\Theta_1 B^{12})\varepsilon_t.$$

5.5 Differencing strategy

Method

1. Plot the series and its ACF. If the ACF decays very slowly, difference.
2. Apply the ADF test (or KPSS). If H_0 (unit root) is not rejected, difference.
3. After differencing, verify that the resulting series is stationary.
4. For seasonality, difference at order s if the ACF shows periodic peaks.

5.6 Numerical example: AirPassengers



5.7 Implementation

Python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import pmdarima as pm

# Charger AirPassengers
url = ('https://raw.githubusercontent.com/jbrownlee/Datasets/'
       'master/airline-passengers.csv')
df = pd.read_csv(url, parse_dates=['Month'], index_col='Month')
y = df['Passengers']

# Log-transformation (stabilise la variance)
y_log = np.log(y)
```

```
# Test ADF
print("ADF sur log(y):", adfuller(y_log)[1])  # > 0.05 => non stat.
y_diff = y_log.diff().dropna()
print("ADF sur diff(log(y)):", adfuller(y_diff)[1])  # < 0.05 => stat.

# SARIMA automatique
auto_model = pm.auto_arima(y_log, seasonal=True, m=12,
                           stepwise=True, trace=True)
print(auto_model.summary())

# Ajustement SARIMA(0,1,1)(0,1,1)[12]
model = SARIMAX(y_log, order=(0,1,1),
                seasonal_order=(0,1,1,12))
results = model.fit(disp=False)
print(results.summary())

# Prévision
forecast = results.get_forecast(steps=24)
pred = np.exp(forecast.predicted_mean)
ci = np.exp(forecast.conf_int())

fig, ax = plt.subplots(figsize=(12, 5))
y.plot(ax=ax, label='Observé')
pred.plot(ax=ax, label='Prévision', color='red')
ax.fill_between(ci.index, ci.iloc[:,0], ci.iloc[:,1],
                alpha=0.2, color='red')
ax.legend()
ax.set_title('SARIMA - AirPassengers')
plt.savefig('ch05_arima.pdf')
plt.show()
```

5.8 Exercises

Exercise 5.1 ($\star$)**.** Show that $(1-B)^2 X_t = X_t - 2X_{t-1} + X_{t-2}$.

Exercise 5.2 ($\star$)**.** Apply `pm.auto_arima` to a series of your choice. Interpret the selected model.

Exercise 5.3 ($\star\star$)**.** Show that if $X_t \sim \text{ARIMA}(p, 1, q)$, then the long-term forecasts converge to a straight line (constant forecast in differenced levels).

Exercise 5.4 ($\star\star\star$ – Project)**.** Model the quarterly GDP of a country using a seasonal ARIMA. Compare out-of-sample forecasts with different orders.

Key Formulas

$$\text{ARIMA}(p,d,q): \phi(B)(1-B)^d X_t = \theta(B)\varepsilon_t$$
$$\text{SARIMA}: \Phi(B^s)\phi(B)(1-B)^d(1-B^s)^D X_t = \Theta(B^s)\theta(B)\varepsilon_t$$
$$I(d): \nabla^d X_t \text{ stationary}, \nabla^{d-1} X_t \text{ not}$$

Chapter 6

GARCH Models and Conditional Volatility

6.1 Motivating example

Financial returns are approximately uncorrelated, but their squares r_t^2 exhibit strong autocorrelation: periods of high volatility are clustered (**volatility clustering**). ARMA models do not capture this phenomenon.

6.2 Stylised facts of financial returns

1. Returns are approximately uncorrelated: $\rho_r(h) \approx 0$.
2. Squared returns are correlated: $\rho_{r^2}(h) > 0$ (volatility persistence).
3. Heavy tails (leptokurtosis): kurtosis > 3.
4. Volatility asymmetry (leverage effect): declines increase volatility more than rises.

6.3 ARCH model

Definition 6.1 (ARCH(q))**.**

$$r_t = \sigma_t z_t, \quad z_t \overset{\text{iid}}{\sim} (0,1), \quad \sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i r_{t-i}^2,$$

where $\omega > 0$, $\alpha_i \geq 0$, and $\sum \alpha_i < 1$ for stationarity.

Theorem 6.2 (Properties of an ARCH(q))**.** *1.* $\mathbb{E}[r_t] = 0$, $\mathrm{Var}(r_t) = \omega/(1 - \sum \alpha_i)$.

2. (r_t) *is a white noise (uncorrelated).*

3. (r_t^2) *follows an AR(q):* $r_t^2 = \omega + \sum \alpha_i r_{t-i}^2 + v_t$ *where* $v_t = r_t^2 - \sigma_t^2$ *is a white noise.*

4. Kurtosis > 3 *(heavy tails even if* z_t *is Gaussian).*

6.4 GARCH model

Definition 6.3 (GARCH(p, q))**.**

$$r_t = \sigma_t z_t, \quad \sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i r_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2.$$

Stationarity conditions: $\omega > 0$, $\alpha_i, \beta_j \geq 0$, $\sum_{i=1}^{q} \alpha_i + \sum_{j=1}^{p} \beta_j < 1$.

Theorem 6.4 (Unconditional variance of GARCH$(1, 1)$)**.**

$$\mathrm{Var}(r_t) = \frac{\omega}{1 - \alpha_1 - \beta_1}.$$

*The **persistence** of volatility is measured by $\alpha_1 + \beta_1$ (close to 1 in finance).*

Intuition

The GARCH$(1, 1)$ is the most widely used volatility model in practice. The one-step-ahead conditional variance forecast is:

$$\hat{\sigma}_{t+1}^2 = \omega + \alpha_1 r_t^2 + \beta_1 \sigma_t^2.$$

It is an exponentially weighted moving average of past squared values.

6.5 IGARCH

Definition 6.5. If $\alpha_1 + \beta_1 = 1$ (integrated GARCH), volatility shocks persist indefinitely. The process is not second-order stationary but remains **strictly stationary**.

6.6 Extensions

6.6.1 EGARCH

Definition 6.6. Nelson's EGARCH model:

$$\ln \sigma_t^2 = \omega + \alpha(|z_{t-1}| - \mathbb{E}[|z_{t-1}|]) + \gamma z_{t-1} + \beta \ln \sigma_{t-1}^2.$$

The term γz_{t-1} captures the **leverage effect**. No positivity constraint on the parameters is needed.

6.6.2 GJR-GARCH

Definition 6.7.

$$\sigma_t^2 = \omega + (\alpha + \gamma \mathbf{1}_{r_{t-1}<0}) r_{t-1}^2 + \beta \sigma_{t-1}^2.$$

$\gamma > 0$ implies that negative returns increase volatility more.

6.7 Estimation

Definition 6.8 (Quasi-maximum likelihood)**.** The Gaussian conditional log-likelihood:

$$\ell(\boldsymbol{\theta}) = -\frac{1}{2}\sum_{t=1}^{n}\left(\ln\sigma_t^2 + \frac{r_t^2}{\sigma_t^2}\right).$$

Even if z_t is not Gaussian, the QMLE is consistent under certain conditions.

6.8 Tests for ARCH effects

Definition 6.9 (Engle's ARCH-LM test)**.** Regress r_t^2 on $r_{t-1}^2, \ldots, r_{t-q}^2$ and test for joint significance via $nR^2 \sim \chi^2(q)$.

6.9 Implementation

Python

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from arch import arch_model

# Simuler un GARCH(1,1)
np.random.seed(42)
n = 1000
omega, alpha, beta = 0.05, 0.10, 0.85
sigma2 = np.zeros(n)
r = np.zeros(n)
sigma2[0] = omega / (1 - alpha - beta)
for t in range(1, n):
    sigma2[t] = omega + alpha * r[t-1]**2 + beta * sigma2[t-1]
    r[t] = np.sqrt(sigma2[t]) * np.random.normal()

fig, axes = plt.subplots(3, 1, figsize=(12, 8))
axes[0].plot(r, lw=0.5)
axes[0].set_title('Rendements simulés (GARCH(1,1))')
axes[1].plot(r**2, lw=0.5, color='orange')
axes[1].set_title('$r_t^2$ (volatilité réalisée)')
axes[2].plot(np.sqrt(sigma2), lw=0.5, color='red')
axes[2].set_title('$\\sigma_t$ (volatilité conditionnelle)')
plt.tight_layout()
plt.savefig('ch06_garch.pdf')
plt.show()

# Ajuster un GARCH(1,1)
am = arch_model(r, vol='Garch', p=1, q=1, dist='normal')
res = am.fit(disp='off')
print(res.summary())
```

```python
# Prévision de volatilité
forecasts = res.forecast(horizon=10)
print("Variance prévisionnelle:")
print(forecasts.variance.iloc[-1])
```

6.10 Exercises

Exercise 6.1 ($\star$)**.** Show that an ARCH(1) with $r_t = \sigma_t z_t$, $\sigma_t^2 = \omega + \alpha r_{t-1}^2$ implies $\mathbb{E}[r_t^4] = 3\omega^2(1+\alpha)/(1-3\alpha^2)$ (if $3\alpha^2 < 1$). Deduce the kurtosis.

Exercise 6.2 ($\star\star$)**.** Show that the GARCH$(1,1)$ can be written as an ARCH(∞).

Exercise 6.3 ($\star\star$)**.** Prove that the process r_t^2 of a GARCH$(1,1)$ follows an ARMA$(1,1)$.

Exercise 6.4 ($\star\star\star$ – Project)**.** Download the daily returns of the S&P 500 over 10 years. Estimate a GARCH$(1,1)$, an EGARCH$(1,1)$, and a GJR-GARCH$(1,1)$. Compare by AIC and analyse the leverage effect.

Key Formulas

$$\begin{aligned} \text{GARCH}(1,1): \sigma_t^2 &= \omega + \alpha r_{t-1}^2 + \beta \sigma_{t-1}^2, \quad \alpha + \beta < 1 \\ \text{Var}(r_t) &= \omega/(1-\alpha-\beta) \\ \text{EGARCH}: \ln \sigma_t^2 &= \omega + \alpha(|z_{t-1}| - \mathbb{E}|z_{t-1}|) + \gamma z_{t-1} + \beta \ln \sigma_{t-1}^2 \end{aligned}$$

Chapter 7

Spectral Analysis

7.1 Motivating Example

A climate signal contains cycles of different periods (daily, annual, El Niño). Spectral analysis decomposes the total variance by frequency, revealing hidden periodicities.

7.2 Spectral Density

Definition 7.1 (Power Spectral Density)**.** If $\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty$, the **spectral density** of (X_t) is:

$$f(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \gamma(h)\, e^{-i\omega h}, \quad \omega \in [-\pi, \pi].$$

This is the Fourier transform of the autocovariance function.

Theorem 7.2 (Properties of f)**.** *1. $f(\omega) \geq 0$ for all ω.*

2. f is even: $f(\omega) = f(-\omega)$.

3. Inversion formula: $\gamma(h) = \int_{-\pi}^{\pi} f(\omega)\, e^{i\omega h}\, d\omega$.

4. $\gamma(0) = \mathrm{Var}(X_t) = \int_{-\pi}^{\pi} f(\omega)\, d\omega$ (Parseval's theorem).

Sketch of (1). By Herglotz's theorem, γ is positive semi-definite. Hence $f(\omega) = \frac{1}{2\pi} \sum \gamma(h) e^{-i\omega h} \geq 0$ (as a limit of positive quadratic forms). □

7.3 Spectra of Classical Processes

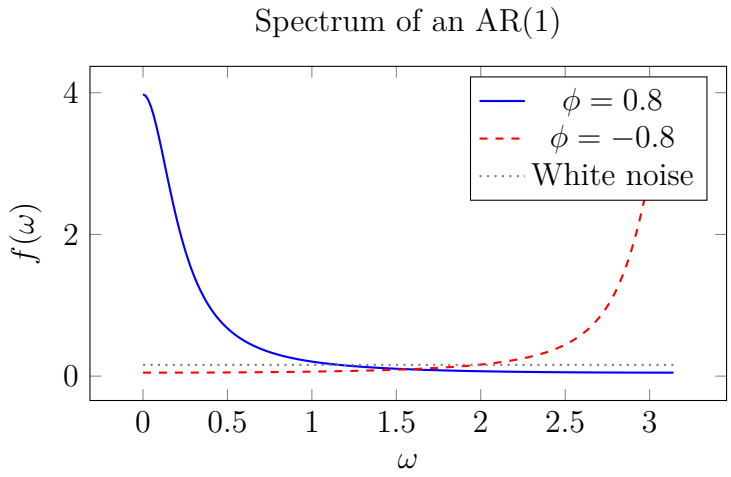
Theorem 7.3 (Spectrum of an ARMA(p, q))**.**

$$f(\omega) = \frac{\sigma^2}{2\pi} \frac{|\theta(e^{-i\omega})|^2}{|\phi(e^{-i\omega})|^2}.$$

Example 7.4 (Spectrum of an AR(1))**.** $\phi(z) = 1 - \phi z$:

$$f(\omega) = \frac{\sigma^2}{2\pi} \frac{1}{|1 - \phi e^{-i\omega}|^2} = \frac{\sigma^2}{2\pi(1 - 2\phi \cos\omega + \phi^2)}.$$

If $\phi > 0$: peak at $\omega = 0$ (dominant low frequencies). If $\phi < 0$: peak at $\omega = \pi$.



7.4 Periodogram

Definition 7.5 (Periodogram)**.** The naive estimator of $f(\omega)$:

$$I_n(\omega) = \frac{1}{2\pi n}\left|\sum_{t=1}^{n} X_t\, e^{-i\omega t}\right|^2.$$

Theorem 7.6 (Properties of the Periodogram)**.**

1. $\mathbb{E}[I_n(\omega_j)] \to f(\omega_j)$ *at the Fourier frequencies* $\omega_j = 2\pi j/n$.
2. $I_n(\omega_j)$ *is an* ***inconsistent*** *estimator:* $\mathrm{Var}(I_n(\omega_j)) \not\to 0$.
3. *At Fourier frequencies:* $I_n(\omega_j)/f(\omega_j) \overset{approx}{\sim} \chi^2(2)/2$.

Warning

The raw periodogram is very erratic. It must be smoothed to obtain a consistent estimator.

7.5 Spectral Estimation

7.5.1 Spectral Window (Smoothing)

Definition 7.7 (Smoothed Periodogram)**.**

$$\hat{f}(\omega) = \sum_{|h|\le M} w(h)\,\hat{\gamma}(h)\,e^{-i\omega h},$$

where w is a **window** (Bartlett, Parzen, Tukey, etc.) and M is the **bandwidth**.

7.5.2 Parametric Method

Estimate an AR(p) model by Yule–Walker or Burg, then use the theoretical spectrum $\hat{f}(\omega) = \frac{\hat{\sigma}^2}{2\pi|\hat{\phi}(e^{-i\omega})|^2}$.

7.6 Implementation

Python

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import periodogram, welch

# Simuler un AR(2) avec pic spectral
np.random.seed(42)
n = 1024
eps = np.random.normal(0, 1, n)
x = np.zeros(n)
phi1, phi2 = 1.5, -0.85  # racines complexes => pic
for t in range(2, n):
    x[t] = phi1*x[t-1] + phi2*x[t-2] + eps[t]

# Périodogramme brut
freq_p, Pxx_p = periodogram(x, fs=1, scaling='density')

# Périodogramme lissé (Welch)
freq_w, Pxx_w = welch(x, fs=1, nperseg=256, scaling='density')

fig, axes = plt.subplots(1, 2, figsize=(14, 5))
axes[0].semilogy(freq_p, Pxx_p, lw=0.5)
axes[0].set_title('Périodogramme brut')
axes[0].set_xlabel('Fréquence')

axes[1].semilogy(freq_w, Pxx_w, lw=1.5, color='red')
axes[1].set_title('Périodogramme lissé (Welch)')
axes[1].set_xlabel('Fréquence')

plt.tight_layout()
plt.savefig('ch07_spectral.pdf')
plt.show()

# Spectre théorique AR(2)
omega = np.linspace(0.01, np.pi, 500)
phi_z = 1 - phi1*np.exp(-1j*omega) - phi2*np.exp(-2j*omega)
f_theo = 1 / (2*np.pi * np.abs(phi_z)**2)

plt.figure(figsize=(8, 4))
plt.plot(omega, f_theo, 'b-', label='Théorique')
plt.title('Spectre théorique AR(2)')
plt.xlabel('$\\omega$'); plt.ylabel('$f(\\omega)$')
plt.legend()
plt.savefig('ch07_spectral_theo.pdf')
plt.show()
```

7.7 Exercises

Exercise 7.1 ($\star$)**.** Compute the spectral density of an MA(1): $X_t = \varepsilon_t + \theta\varepsilon_{t-1}$.

Exercise 7.2 ($\star\star$)**.** Show that the periodogram can be written as $I_n(\omega) = \sum_{|h|<n} \hat{\gamma}(h)e^{-i\omega h}/(2\pi)$.

Exercise 7.3 ($\star\star$)**.** Prove that for a Gaussian white noise, $2nI_n(\omega_j)/\sigma^2 \sim \chi^2(2)$ at nonzero Fourier frequencies.

Exercise 7.4 ($\star\star\star$ – Project)**.** Analyze the spectrum of daily temperatures in Paris over 50 years. Identify the peaks corresponding to the annual, semi-annual, and weekly cycles.

Key Formulas

$$f(\omega) = \frac{1}{2\pi}\sum_{h}\gamma(h)e^{-i\omega h}, \quad \gamma(h) = \int_{-\pi}^{\pi} f(\omega)e^{i\omega h}d\omega$$

$$f_{\text{ARMA}}(\omega) = \frac{\sigma^2}{2\pi}\frac{|\theta(e^{-i\omega})|^2}{|\phi(e^{-i\omega})|^2}$$

$$I_n(\omega) = \frac{1}{2\pi n}\left|\sum_{t=1}^{n} X_t e^{-i\omega t}\right|^2$$

Chapter 8

State Space Models and the Kalman Filter

8.1 Motivating Example

The "true" GDP of an economy is unobservable; one only observes noisy and revised estimates. A state space model separates the **latent state** from the **noisy observation**, and the Kalman filter provides the optimal estimate of the state.

8.2 Linear Gaussian State Space Model

Definition 8.1 (State Space Model)**.**

$$\boldsymbol{x}_{t+1} = F\boldsymbol{x}_t + G\boldsymbol{w}_t \qquad \text{(state equation)} \tag{8.1}$$

$$\boldsymbol{y}_t = H\boldsymbol{x}_t + \boldsymbol{v}_t \qquad \text{(observation equation)} \tag{8.2}$$

where $\boldsymbol{x}_t \in \mathbb{R}^p$ is the (latent) state, $\boldsymbol{y}_t \in \mathbb{R}^m$ the observation, $\boldsymbol{w}_t \sim \mathcal{N}(0, Q)$ and $\boldsymbol{v}_t \sim \mathcal{N}(0, R)$ are independent.

Example 8.2 (Local Level)**.** $x_{t+1} = x_t + w_t$, $y_t = x_t + v_t$. The state is the "true value" smoothed over time.

Example 8.3 (ARMA as State Space)**.** Any ARMA(p, q) can be cast in state space form. For instance, the AR(1) $X_t = \phi X_{t-1} + \varepsilon_t$ corresponds to $F = \phi$, $H = 1$, $G = 1$, $Q = \sigma^2$, $R = 0$.

8.3 The Kalman Filter

Definition 8.4 (Kalman Filter)**.** Notation: $\hat{\boldsymbol{x}}_{t|s} = \mathbb{E}[\boldsymbol{x}_t | \boldsymbol{y}_1, \ldots, \boldsymbol{y}_s]$ and $P_{t|s} = \mathrm{Cov}(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_{t|s})$.

Prediction:

$$\hat{\boldsymbol{x}}_{t+1|t} = F\hat{\boldsymbol{x}}_{t|t}, \tag{8.3}$$

$$P_{t+1|t} = FP_{t|t}F^\top + GQG^\top. \tag{8.4}$$

Update:

$$K_t = P_{t|t-1}H^\top (HP_{t|t-1}H^\top + R)^{-1} \quad \text{(Kalman gain)}, \tag{8.5}$$

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + K_t(\boldsymbol{y}_t - H\hat{\boldsymbol{x}}_{t|t-1}), \tag{8.6}$$

$$P_{t|t} = (I - K_t H)P_{t|t-1}. \tag{8.7}$$

Theorem 8.5 (Optimality of the Kalman Filter)**.** *In the linear Gaussian model, the Kalman filter yields the MMSE (minimum mean square error) estimator of* $\boldsymbol{x}_t$ *given* $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t$*. It is the* ***optimal*** *estimator among all estimators (not only linear ones).*

Sketch. By induction. At each step, the conditional distribution $\boldsymbol{x}_t|\boldsymbol{y}_{1:t}$ is Gaussian (by closure of the Gaussian distribution under linear conditioning). The conditional expectation is the MMSE estimator for Gaussian distributions. □

8.4 Kalman Smoother

Definition 8.6 (Rauch–Tung–Striebel Smoother)**.** To compute $\hat{\boldsymbol{x}}_{t|n}$ (estimate of the past state using *all* observations):

$$J_t = P_{t|t}F^\top P_{t+1|t}^{-1}, \tag{8.8}$$

$$\hat{\boldsymbol{x}}_{t|n} = \hat{\boldsymbol{x}}_{t|t} + J_t(\hat{\boldsymbol{x}}_{t+1|n} - \hat{\boldsymbol{x}}_{t+1|t}), \tag{8.9}$$

$$P_{t|n} = P_{t|t} + J_t(P_{t+1|n} - P_{t+1|t})J_t^\top. \tag{8.10}$$

One iterates backward from $t = n-1$ to $t = 1$.

8.5 Likelihood via the Kalman Filter

Theorem 8.7 (Innovation Decomposition)**.** *The log-likelihood decomposes as:*

$$\ln L = -\frac{nm}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{n}\left[\ln|S_t| + \boldsymbol{e}_t^\top S_t^{-1}\boldsymbol{e}_t\right],$$

where $\boldsymbol{e}_t = \boldsymbol{y}_t - H\hat{\boldsymbol{x}}_{t|t-1}$ *is the* ***innovation*** *and* $S_t = HP_{t|t-1}H^\top + R$*.*

8.6 Implementation

Python

```python
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.structural import UnobservedComponents

# Simuler un modèle local level
np.random.seed(42)
n = 200
sigma_w, sigma_v = 0.5, 1.0
x = np.zeros(n)
```

```
y = np.zeros(n)
for t in range(1, n):
    x[t] = x[t-1] + sigma_w * np.random.normal()
    y[t] = x[t] + sigma_v * np.random.normal()

# Filtre de Kalman manuel
F, H, Q, R = 1.0, 1.0, sigma_w**2, sigma_v**2
x_filt = np.zeros(n)
P_filt = np.zeros(n)
P_filt[0] = 1.0
for t in range(1, n):
    # Prédiction
    x_pred = F * x_filt[t-1]
    P_pred = F * P_filt[t-1] * F + Q
    # Mise à jour
    S = H * P_pred * H + R
    K = P_pred * H / S
    x_filt[t] = x_pred + K * (y[t] - H * x_pred)
    P_filt[t] = (1 - K * H) * P_pred

fig, ax = plt.subplots(figsize=(12, 5))
ax.plot(y, 'o', ms=2, alpha=0.5, label='Observations $y_t$')
ax.plot(x, 'g-', lw=1.5, label='État vrai $x_t$')
ax.plot(x_filt, 'r--', lw=1.5, label='Filtre de Kalman $\\hat{x}_{t|t}$')
ax.legend()
ax.set_title('Filtre de Kalman - Local Level Model')
plt.savefig('ch08_kalman.pdf')
plt.show()

# Avec statsmodels
model = UnobservedComponents(y, 'local level')
results = model.fit(disp=False)
print(results.summary())
```

8.7 Exercises

Exercise 8.1 ($\star$)**.** Write the AR(2) in state space form.

Exercise 8.2 ($\star\star$)**.** Prove that the Kalman gain K_t minimizes $\text{tr}(P_{t|t})$.

Exercise 8.3 ($\star\star$)**.** Show that in the stationary local level model (σ_w, σ_v constant), the Kalman gain converges to a constant K_∞.

Exercise 8.4 ($\star\star\star$ – Project)**.** Implement an extended Kalman filter (EKF) for a stochastic volatility model: $\ln \sigma_t^2 = \phi \ln \sigma_{t-1}^2 + w_t$, $r_t = \sigma_t z_t$.

Key Formulas

$$\hat{\boldsymbol{x}}_{t+1|t} = F\hat{\boldsymbol{x}}_{t|t}, \quad P_{t+1|t} = FP_{t|t}F^\top + GQG^\top$$
$$K_t = P_{t|t-1}H^\top(HP_{t|t-1}H^\top + R)^{-1}$$
$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + K_t(\boldsymbol{y}_t - H\hat{\boldsymbol{x}}_{t|t-1})$$
$$\ln L = -\tfrac{1}{2}\sum[\ln|S_t| + \boldsymbol{e}_t^\top S_t^{-1}\boldsymbol{e}_t]$$

Chapter 9

Multivariate Time Series — VAR and Cointegration

9.1 Motivating Example

Interest rates and inflation evolve jointly: a rise in one influences the other. A **VAR** model captures these dynamic interdependencies. **Cointegration** reveals long-run equilibrium relationships between nonstationary variables.

9.2 Stationary Vector Processes

Definition 9.1 (Stationary Vector Process)**.** $(\boldsymbol{Y}_t)_{t\in\mathbb{Z}}$ with $\boldsymbol{Y}_t \in \mathbb{R}^k$ is **weakly stationary** if:

1. $\mathbb{E}[\|\boldsymbol{Y}_t\|^2] < \infty$,
2. $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{Y}_t]$ does not depend on t,
3. $\Gamma(h) = \mathrm{Cov}(\boldsymbol{Y}_t, \boldsymbol{Y}_{t+h}) = \mathbb{E}[(\boldsymbol{Y}_t - \boldsymbol{\mu})(\boldsymbol{Y}_{t+h} - \boldsymbol{\mu})^\top]$ depends only on h.

Note: $\Gamma(h) = \Gamma(-h)^\top$ (but $\Gamma(h) \neq \Gamma(h)^\top$ in general).

9.3 VAR(p) Model

Definition 9.2 (VAR(p))**.**

$$\boldsymbol{Y}_t = \boldsymbol{c} + A_1\boldsymbol{Y}_{t-1} + A_2\boldsymbol{Y}_{t-2} + \cdots + A_p\boldsymbol{Y}_{t-p} + \boldsymbol{u}_t,$$

where $\boldsymbol{u}_t \sim \mathrm{WN}(\boldsymbol{0}, \Sigma_u)$ (k-dimensional) and $A_i \in \mathbb{R}^{k\times k}$.

Theorem 9.3 (Stationarity of a VAR(p))**.** *The VAR(p) is stationary if and only if all roots of*

$$\det(I_k - A_1 z - \cdots - A_p z^p) = 0$$

lie outside the unit disk.

9.3.1 Estimation

Theorem 9.4. *The parameters of a VAR(p) can be estimated equation by equation by OLS. The estimator is consistent and asymptotically normal.*

9.4 Granger Causality

Definition 9.5 (Granger Causality)**.** $Y_{2,t}$ **Granger-causes** $Y_{1,t}$ if the past values of Y_2 improve the forecast of Y_1, i.e. if in the VAR, the coefficients associated with the lags of Y_2 in the equation for Y_1 are jointly significant.

Test: $H_0 : A_1^{(12)} = \cdots = A_p^{(12)} = 0$ via a Wald test (F-test).

9.5 Impulse Response Functions

Definition 9.6 (IRF)**.** The **impulse response function** measures the effect of a unit shock to $u_{j,t}$ on $Y_{i,t+h}$:

$$\Psi_h = \frac{\partial \boldsymbol{Y}_{t+h}}{\partial \boldsymbol{u}_t^\top},$$

where $\boldsymbol{Y}_t = \sum_{h=0}^{\infty} \Psi_h \boldsymbol{u}_{t-h}$ is the MA(∞) representation.

9.6 Cointegration

Definition 9.7 (Cointegration)**.** $I(1)$ series $Y_{1,t}, \ldots, Y_{k,t}$ are **cointegrated** if there exists $\boldsymbol{\beta} \neq \mathbf{0}$ such that $\boldsymbol{\beta}^\top \boldsymbol{Y}_t \sim I(0)$. The vector $\boldsymbol{\beta}$ is a **cointegrating vector** and $\boldsymbol{\beta}^\top \boldsymbol{Y}_t$ represents a long-run equilibrium relationship.

Example 9.8. The spot and futures prices of a commodity are each $I(1)$, but their difference (the basis) is $I(0)$: they are cointegrated with $\boldsymbol{\beta} = (1, -1)^\top$.

Definition 9.9 (Vector Error Correction Model (VECM))**.**

$$\nabla \boldsymbol{Y}_t = \Pi \boldsymbol{Y}_{t-1} + \sum_{i=1}^{p-1} \Gamma_i \nabla \boldsymbol{Y}_{t-i} + \boldsymbol{u}_t,$$

where $\Pi = \alpha\beta^\top$ with $\alpha \in \mathbb{R}^{k \times r}$ (adjustment speeds) and $\beta \in \mathbb{R}^{k \times r}$ (cointegrating vectors), $r = \text{rank}(\Pi)$ is the number of cointegrating relationships.

Theorem 9.10 (Granger Representation Theorem)**.** *If* $\boldsymbol{Y}_t \sim I(1)$ *and cointegrated of rank r:*

- $0 < r < k$*: cointegration. The VECM is the appropriate specification.*
- $r = 0$*: no cointegration, VAR in differences.*
- $r = k$*: all series are* $I(0)$*, VAR in levels.*

9.6.1 Johansen Test

Definition 9.11. The Johansen test determines the cointegration rank r through sequential maximum eigenvalue or trace tests.

9.7 Implementation

Python

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.api import VAR
from statsmodels.tsa.vector_ar.vecm import coint_johansen

# Simuler un VAR(1) bivarié
np.random.seed(42)
n = 500
A = np.array([[0.5, 0.3], [-0.2, 0.6]])
Y = np.zeros((n, 2))
for t in range(1, n):
    Y[t] = A @ Y[t-1] + np.random.multivariate_normal([0,0],
                                        [[1, 0.3],[0.3, 1]])

df = pd.DataFrame(Y, columns=['Y1', 'Y2'])

# Ajuster VAR
model = VAR(df)
results = model.fit(maxlags=5, ic='bic')
print(results.summary())

# Causalité de Granger
granger = results.test_causality('Y1', 'Y2', kind='f')
print(f"\nGranger Y2 -> Y1: p-value = {granger.pvalue:.4f}")

# IRF
irf = results.irf(20)
irf.plot(orth=True)
plt.savefig('ch09_irf.pdf')
plt.show()

# Coïntégration (Johansen) sur données simulées I(1)
rw1 = np.cumsum(np.random.normal(0, 1, n))
rw2 = 0.5 * rw1 + np.cumsum(np.random.normal(0, 0.3, n))
data_coint = np.column_stack([rw1, rw2])
joh = coint_johansen(data_coint, det_order=0, k_ar_diff=1)
print("\nJohansen trace stats:", joh.lr1)
print("Critical values (90%, 95%, 99%):", joh.cvt)
```

9.8 Exercises

Exercise 9.1 ($\star$). Verify that the VAR(1) $\boldsymbol{Y}_t = A\boldsymbol{Y}_{t-1} + \boldsymbol{u}_t$ with $A = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.3 \end{pmatrix}$ is stationary. Compute $\Gamma(0)$.

Exercise 9.2 ($\star\star$). Show that in a VAR(1), $\Gamma(h) = A^h\Gamma(0)$ for $h \geq 0$.

Exercise 9.3 ($\star\star$)**.** Show that if $Y_{1,t}$ and $Y_{2,t}$ are independent random walks, they are not cointegrated.

Exercise 9.4 ($\star\star\star$ – Project)**.** Estimate a VAR for the pair (interest rate, inflation) of a country. Test Granger causality in both directions and interpret the impulse response functions.

Key Formulas

$$\text{VAR}(p): \boldsymbol{Y}_t = \boldsymbol{c} + A_1 \boldsymbol{Y}_{t-1} + \cdots + A_p \boldsymbol{Y}_{t-p} + \boldsymbol{u}_t$$

$$\text{VECM}: \nabla \boldsymbol{Y}_t = \alpha \beta^\top \boldsymbol{Y}_{t-1} + \sum \Gamma_i \nabla \boldsymbol{Y}_{t-i} + \boldsymbol{u}_t$$

$$\text{Cointegration}: \boldsymbol{\beta}^\top \boldsymbol{Y}_t \sim I(0) \text{ with } \boldsymbol{Y}_t \sim I(1)$$

Chapter 10

Forecasting and Confidence Intervals

10.1 Motivating Example

Forecasting electricity demand for the next day, together with a confidence interval, is essential for grid management. The quality of the forecast determines economic costs.

10.2 Optimal Forecast

Definition 10.1 (Optimal Forecast)**.** The **optimal forecast** of X_{n+h} given $X_1, \ldots, X_n$ in the least squares sense is:

$$\hat{X}_{n+h|n} = \mathbb{E}[X_{n+h}|X_1, \ldots, X_n].$$

The forecast error is $e_{n+h|n} = X_{n+h} - \hat{X}_{n+h|n}$.

Theorem 10.2 (Mean Squared Error)**.** *The conditional expectation minimizes the MSE:*

$$\hat{X}_{n+h|n} = \arg\min_{g(X_1,\ldots,X_n)} \mathbb{E}\left[(X_{n+h} - g)^2\right].$$

10.3 ARMA Forecasting

Theorem 10.3 (Forecast of an ARMA(p,q))**.** *Using the MA(∞) representation: $X_t = \sum_{j=0}^{\infty} \psi_j \varepsilon_{t-j}$, the h-step-ahead forecast is:*

$$\hat{X}_{n+h|n} = \sum_{j=h}^{\infty} \psi_j \varepsilon_{n+h-j}.$$

The forecast error is $e_{n+h|n} = \sum_{j=0}^{h-1} \psi_j \varepsilon_{n+h-j}$, with variance:

$$\sigma_h^2 = \sigma^2 \sum_{j=0}^{h-1} \psi_j^2.$$

Corollary 10.4. *For a stationary ARMA, $\sigma_h^2 \to \gamma(0) = \mathrm{Var}(X_t)$ as $h \to \infty$: the forecast converges to the unconditional mean and the confidence interval widens to the unconditional interval.*

10.4 Prediction Intervals

Definition 10.5. Under the Gaussian assumption, a $(1-\alpha)$ prediction interval for X_{n+h} is:

$$\hat{X}_{n+h|n} \pm z_{1-\alpha/2}\,\sigma_h,$$

where $z_{1-\alpha/2}$ is the quantile of the standard normal distribution.

Warning

Prediction intervals do not account for the uncertainty in estimated parameters. Bootstrap or simulation methods can be used to obtain more realistic intervals.

10.5 ARIMA Forecasting

Theorem 10.6. *For an ARIMA(p,d,q), the long-term forecast of the* level X_{n+h} *grows linearly (if $d=1$) or quadratically (if $d=2$): the prediction interval widens without bound.*

10.6 Evaluation Criteria

Definition 10.7 (Forecast Metrics)**.**

$$\text{MAE} = \frac{1}{H}\sum_{h=1}^{H}|e_{n+h|n}|, \tag{10.1}$$

$$\text{RMSE} = \sqrt{\frac{1}{H}\sum_{h=1}^{H}e_{n+h|n}^2}, \tag{10.2}$$

$$\text{MAPE} = \frac{100}{H}\sum_{h=1}^{H}\frac{|e_{n+h|n}|}{|X_{n+h}|}\%. \tag{10.3}$$

Definition 10.8 (Temporal Cross-Validation)**.** To evaluate predictive ability, one uses a **train/test** split that respects the temporal order (no shuffling). The **rolling window** or **expanding window** method is standard.

10.7 Diebold–Mariano Test

Definition 10.9. To compare two forecasting models, let $d_t = L(e_{1,t}) - L(e_{2,t})$ be the loss differential. Under H_0 (equality):

$$\text{DM} = \frac{\bar{d}}{\sqrt{\hat{\sigma}_d^2/H}} \xrightarrow{d} \mathcal{N}(0,1).$$

10.8 Implementation

Python

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Simuler un ARMA(1,1)
np.random.seed(42)
from statsmodels.tsa.arima_process import ArmaProcess
ar = np.array([1, -0.7])
ma = np.array([1, 0.3])
data = ArmaProcess(ar, ma).generate_sample(nsample=300)

# Train/test split
train, test = data[:250], data[250:]

# Ajuster et prévoir
model = ARIMA(train, order=(1, 0, 1))
results = model.fit()

forecast = results.get_forecast(steps=50)
pred = forecast.predicted_mean
ci = forecast.conf_int(alpha=0.05)

# Métriques
mae = mean_absolute_error(test, pred)
rmse = np.sqrt(mean_squared_error(test, pred))
print(f"MAE  = {mae:.4f}")
print(f"RMSE = {rmse:.4f}")

# Graphique
fig, ax = plt.subplots(figsize=(12, 5))
ax.plot(range(250), train, 'b-', lw=0.7, label='Train')
ax.plot(range(250, 300), test, 'g-', lw=1, label='Test')
ax.plot(range(250, 300), pred, 'r--', lw=1.5, label='Prévision')
ax.fill_between(range(250, 300), ci[:, 0], ci[:, 1],
                alpha=0.2, color='red')
ax.legend()
ax.set_title('Prévision ARMA(1,1)')
plt.savefig('ch10_forecast.pdf')
plt.show()

# Rolling forecast
rolling_preds = []
for i in range(len(test)):
    train_i = data[:250+i]
    model_i = ARIMA(train_i, order=(1,0,1)).fit()
```

```
    rolling_preds.append(model_i.forecast(steps=1)[0])
rmse_roll = np.sqrt(mean_squared_error(test, rolling_preds))
print(f"RMSE rolling: {rmse_roll:.4f}")
```

10.9 Exercises

Exercise 10.1 ($\star$). For an AR(1) with $\phi = 0.8$ and $\sigma^2 = 1$, compute $\hat{X}_{n+1|n}$, $\hat{X}_{n+2|n}$ and the corresponding forecast error variances.

Exercise 10.2 ($\star\star$). Show that the h-step-ahead forecast error variance of an AR(1) is $\sigma^2 \frac{1-\phi^{2h}}{1-\phi^2}$.

Exercise 10.3 ($\star\star$). Explain why the MAPE is problematic when X_t takes values close to zero.

Exercise 10.4 ($\star\star\star$ – Project). Compare out-of-sample forecasts of an ARIMA, exponential smoothing, and a naive model ($\hat{X}_{n+h} = X_n$) on 5 real-world series. Use the Diebold–Mariano test.

Key Formulas

$$\hat{X}_{n+h|n} = \sum_{j=h}^{\infty} \psi_j \varepsilon_{n+h-j}, \qquad \sigma_h^2 = \sigma^2 \sum_{j=0}^{h-1} \psi_j^2$$

$$\text{CI}_{95\%} : \hat{X}_{n+h|n} \pm 1.96\, \sigma_h$$

$$\text{RMSE} = \sqrt{\frac{1}{H} \sum e_{n+h}^2}$$

Chapter 11

Applications: Finance, Economics, Meteorology

11.1 Motivating Example

The tools developed in this course apply directly to concrete problems: portfolio management, macroeconomic forecasting, and climate modeling.

11.2 Finance: Volatility Modeling

11.2.1 Value at Risk (VaR)

Definition 11.1 (VaR)**.** The **Value at Risk** at level α is the α-quantile of the loss distribution:

$$P(L_t > \mathrm{VaR}_\alpha) = \alpha.$$

With a GARCH model: $\mathrm{VaR}_\alpha = -\hat{\mu}_t + \hat{\sigma}_t\, q_\alpha$, where q_α is the quantile of the innovation distribution.

11.2.2 Application

Python — VaR with GARCH

```python
import numpy as np
import pandas as pd
from arch import arch_model

# Simuler des rendements type financier
np.random.seed(42)
n = 2000
omega, alpha, beta = 0.01, 0.08, 0.90
sigma2 = np.zeros(n)
r = np.zeros(n)
sigma2[0] = omega / (1 - alpha - beta)
for t in range(1, n):
    sigma2[t] = omega + alpha * r[t-1]**2 + beta * sigma2[t-1]
    r[t] = np.sqrt(sigma2[t]) * np.random.standard_t(5)
```

```python
# Estimer GARCH(1,1) avec innovations t de Student
am = arch_model(r * 100, vol='Garch', p=1, q=1, dist='t')
res = am.fit(disp='off')

# VaR 1%
forecasts = res.forecast(horizon=1)
sigma_next = np.sqrt(forecasts.variance.values[-1, 0])
from scipy.stats import t as t_dist
nu = res.params['nu']
VaR_01 = sigma_next * t_dist.ppf(0.01, nu)
print(f"VaR 1% = {VaR_01:.4f}%")
```

11.3 Economics: Macroeconomic Forecasting

11.3.1 GDP and Leading Indicators

Method

1. Load quarterly series (GDP, consumption, investment).
2. Test for stationarity (ADF). Difference if necessary.
3. Estimate a VAR or a seasonal ARIMA.
4. Evaluate by out-of-sample validation.

Python — Macro Forecasting

```python
import pandas as pd
import numpy as np
from statsmodels.tsa.api import VAR
from statsmodels.tsa.stattools import adfuller

# Exemple avec données simulées (remplacer par FRED/Eurostat)
np.random.seed(0)
n = 200
gdp = np.cumsum(np.random.normal(0.5, 1, n))      # trend + bruit
cpi = 0.3 * gdp + np.cumsum(np.random.normal(0, 0.5, n))

df = pd.DataFrame({'GDP_growth': np.diff(gdp),
                   'Inflation': np.diff(cpi)})

# VAR
model = VAR(df)
results = model.fit(ic='bic')
print(f"Ordre sélectionné: VAR({results.k_ar})")
print(results.summary())
```

```
# Prévision
forecast = results.forecast(df.values[-results.k_ar:], steps=8)
print("Prévision PIB (8 trimestres):", forecast[:, 0])
```

11.4 Meteorology: Temperatures and Seasonality

Python — Temperature Series

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.statespace.sarimax import SARIMAX

# Simuler des températures mensuelles
np.random.seed(42)
n_years = 30
t = np.arange(n_years * 12)
seasonal = 10 * np.sin(2 * np.pi * t / 12)
trend = 0.01 * t  # réchauffement
noise = np.random.normal(0, 2, len(t))
temp = 15 + trend + seasonal + noise

df = pd.DataFrame({'temp': temp},
    index=pd.date_range('1990-01', periods=len(t), freq='ME'))

# Décomposition
decomp = seasonal_decompose(df['temp'], model='additive', period=12)
decomp.plot()
plt.savefig('ch11_decomp.pdf')
plt.show()

# SARIMA
train = df.iloc[:-24]
test = df.iloc[-24:]
model = SARIMAX(train, order=(1,0,1),
                seasonal_order=(1,1,1,12))
results = model.fit(disp=False)
pred = results.forecast(steps=24)

rmse = np.sqrt(np.mean((test['temp'] - pred.values)**2))
print(f"RMSE prévision 24 mois: {rmse:.2f}°C")
```

11.5 Methodological Summary

Domain	Typical Model	Main Challenge
Finance	GARCH, EGARCH	Volatility, heavy tails
Macroeconomics	VAR, VECM, ARIMA	Causality, cointegration
Meteorology	SARIMA, state space	Seasonality, trend
Epidemiology	SIR + state space	Nonlinearity
Signal processing	Spectral analysis	Hidden periodicities

11.6 Exercises

Exercise 11.1 ($\star$)**.** Compute the 5% VaR for a GARCH(1,1) with $\hat{\sigma}_{t+1} = 2\%$ and Gaussian innovations.

Exercise 11.2 ($\star\star$)**.** Compare SARIMA and Holt–Winters exponential smoothing forecasts on monthly temperatures. Which is more accurate?

Exercise 11.3 ($\star\star\star$ – Project)**.** Choose a real-world dataset in one of the three domains. Apply the complete methodology: exploration, modeling (at least 3 models), diagnostics, out-of-sample forecasting, comparison. Write a 10-page report.

Key Formulas

$$\text{VaR}_\alpha = -\hat{\mu}_t + \hat{\sigma}_t\, q_\alpha$$

$$\text{RMSE} = \sqrt{\frac{1}{H}\sum (X_{n+h} - \hat{X}_{n+h})^2}$$

$$\text{DM} = \frac{\bar{d}}{\sqrt{\hat{\sigma}_d^2/H}} \xrightarrow{d} \mathcal{N}(0,1)$$

Formula Sheet

.1 ARMA Processes

$$\text{AR}(p) : X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t \tag{1}$$
$$\text{MA}(q) : X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \tag{2}$$
$$\text{ARMA}(p,q) : \phi(B) X_t = \theta(B) \varepsilon_t \tag{3}$$
$$\text{ARIMA}(p,d,q) : \phi(B)(1-B)^d X_t = \theta(B) \varepsilon_t \tag{4}$$

.2 Autocovariance and Autocorrelation

$$\gamma(h) = \text{Cov}(X_t, X_{t+h}) \tag{5}$$
$$\rho(h) = \gamma(h)/\gamma(0) \tag{6}$$
$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} (X_t - \bar{X})(X_{t+h} - \bar{X}) \tag{7}$$

.3 Information Criteria

$$\text{AIC} = -2 \ln L + 2k \tag{8}$$
$$\text{BIC} = -2 \ln L + k \ln n \tag{9}$$

.4 GARCH

$$\text{GARCH}(1,1) : \sigma_t^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \tag{10}$$

.5 Kalman Filter

$$\text{State} : \boldsymbol{x}_{t+1} = F \boldsymbol{x}_t + \boldsymbol{w}_t \tag{11}$$
$$\text{Observation} : \boldsymbol{y}_t = H \boldsymbol{x}_t + \boldsymbol{v}_t \tag{12}$$

.6 VAR

$$\text{VAR}(p) : \boldsymbol{Y}_t = \boldsymbol{c} + A_1 \boldsymbol{Y}_{t-1} + \cdots + A_p \boldsymbol{Y}_{t-p} + \boldsymbol{u}_t \tag{13}$$

Bibliography

[1] Box, G.E.P., Jenkins, G.M., Reinsel, G.C. & Ljung, G.M. (2015). *Time Series Analysis: Forecasting and Control*. 5th ed., Wiley.

[2] Brockwell, P.J. & Davis, R.A. (2016). *Introduction to Time Series and Forecasting*. 3rd ed., Springer.

[3] Hamilton, J.D. (1994). *Time Series Analysis*. Princeton University Press.

[4] Shumway, R.H. & Stoffer, D.S. (2017). *Time Series Analysis and Its Applications*. 4th ed., Springer.

[5] Gouriéroux, C. & Monfort, A. (1995). *Séries Temporelles et Modèles Dynamiques*. 2nd ed., Economica.

[6] Tsay, R.S. (2010). *Analysis of Financial Time Series*. 3rd ed., Wiley.