Python For Finance

M1 - Economie & Finance - Introduction to time serie analysis

Université Paris Dauphine

November 2025

Why Time Series Analysis Matters in Finance

Key Role in Financial Modeling

- Financial data (prices, returns, trading volumes) are naturally ordered over time.
- Understanding their temporal structure is essential for decision-making.

Main Applications:

- Risk Management: modeling and simulating portfolio risk over time.
- Asset Pricing: estimating expected returns, beta coefficients, and factor models.

Challenges:

- Financial series often exhibit non-stationarity, volatility clustering, jumps, and fat tails.
- Requires specialized models (e.g., ARIMA, GARCH, VAR, deep learning).

Time Series Analysis: Stationarity

What is a stationary series?

- A time series is **stationary** if its statistical properties (mean, variance, autocorrelations) are **constant over time**.
- More formally: the distribution of the series does not change over time.

Why is stationarity important?

- Most time series models (ARIMA, VAR, etc.) **assume stationarity**.
- Stationarity ensures that relationships between variables are **stable and predictable**.
- Helps avoid *spurious correlations* that arise from non-stationary data.

In practice:

- Testing for stationarity is a crucial step before estimating models or making forecasts.

Stationarity in Finance: Prices vs Returns

Why are prices often non-stationary?

- Financial prices (stocks, indices, exchange rates, . . .) typically show:
  - Long-term trends (due to economic growth, inflation, structural changes),
  - Level shifts (market regimes or policy changes),
  - Time-varying volatility (heteroscedasticity).

$\Rightarrow$ Mean and variance are not constant: **prices are generally non-stationary**.

Stationarity in Finance: Prices vs Returns

Why are returns often closer to stationarity?

- **Returns** measure the relative change in price between two periods:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$$

- Taking returns removes price levels and many long-term trends.
- Returns tend to be:
  - Centered around a stable mean (often close to 0),
  - With more stable variance (apart from crises).

$\Rightarrow$ Returns are often modeled as stationary.

Testing Stationarity: Augmented Dickey-Fuller (ADF) Test

What is the ADF Test?

- A statistical test used to determine if a time series is **stationary**.
- Tests for the presence of a unit root in the series (behave like a random walk meaning that all shock have a permanent effect).

Hypotheses:

- H_0: The series has a unit root (**non-stationary**).
- H_1: The series is **stationary**.

How it Works:

- Runs a regression with lagged terms to account for autocorrelation.
- Produces a test statistic and a critical value to compare.
- Reject H_0 if the test statistic is less than the critical value.

What is an AR Model?

- A time series model where the current value depends on its past values.
- AR(p) model formula:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

- ϵ_t is white noise (zero mean, constant variance).

Key Properties

- **Linearity:** Linear combination of past values.
- **Stationarity:** Coefficients must satisfy certain constraints, e.g. $|\phi_1| < 1$ for AR(1).
- **Memory:** Remembers p previous observations.
- **PACF:** The Partial Autocorrelation Function is used to determine the lag for the model. It "cuts off" direct time dependency after lag p.

Step 1: Choose the Order p (Model Selection)

Goal: Decide how many lags p to include in the AR(p) model.

1. **Estimate** AR models with different orders $p = 0, 1, 2, \ldots, p_{\text{max}}$.
2. **Compute** the information criteria for each model:
   - AIC (Akaike Information Criterion)
   - BIC (Bayesian Information Criterion)
3. **Select** the model with:
   - The **smallest** AIC or BIC.
   - BIC usually penalizes complexity more than AIC.
4. **Use PACF plot** as a guide:
   - Look where the partial autocorrelations **cut off**.
   - If PACF is significant up to lag k, this suggests AR(k).

Step 2: Estimate the AR(p) Model

Once p is chosen:

1. Specify the model:

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \epsilon_t$$

2. **Estimate** parameters $(c, \phi_1, \ldots, \phi_p)$ using, e.g.:
   - Ordinary Least Squares (OLS)
   - Maximum Likelihood Estimation (MLE)
3. **Save** the residuals:

$$\hat{\epsilon}_t = Y_t - \hat{c} - \hat{\phi}_1 Y_{t-1} - \cdots - \hat{\phi}_p Y_{t-p}$$

4. These residuals will be used for diagnostics.

Step 3: Diagnostics – Are Residuals White Noise?

Goal: Check if the fitted model has captured all time dependence.

1. **Definition (White Noise):**
   - A series ϵ_t is **white noise** if it has:
     - Zero mean: $\mathbb{E}[\epsilon_t] = 0$
     - Constant variance: $\text{Var}(\epsilon_t) = \sigma^2$
     - No autocorrelation: $\mathbb{E}[\epsilon_t \epsilon_{t-k}] = 0$ for $k \neq 0$
2. **Visual checks:**
   - Plot residuals over time: look for patterns or changes in variance.
   - Plot ACF of residuals: should be close to zero at all lags.
3. **Formal test for autocorrelation:**
   - Use the Ljung–Box test on residuals.
   - H_0: residuals are white noise (no autocorrelation).
   - If p-value is **large** $\Rightarrow$ do not reject H_0.
4. **If diagnostics fail:**
   - Consider changing p, adding other terms, or using ARMA/ARIMA.

What is a Moving Average (MA) Model?

Definition:
A Moving Average model expresses the current value of a time series as a **linear combination of past error terms**.

$$Y_t = \mu + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t$$

- Y_t: observed value at time t
- μ: constant mean
- ϵ_t: white noise shock at time t
- q: order of the MA model (number of lags of shocks)

Key Properties of MA(q)

- **Linearity:** Linear in past shocks, not past values of Y_t.
- **Dependence length:** Only q past shocks affect Y_t.
- **Invertibility:** Can be rewritten as infinite AR model if invertibility conditions hold.
- **ACF behavior: Cuts off** after lag q.
- **PACF behavior:** Decays gradually.
- **No stationarity constraint** (MA models are always stationary).

Model Selection and Diagnostics (MA)

Step 1: Choose order q

- Estimate MA(q) models for different q.
- Select via AIC, BIC.
- Use **ACF cutoff** rule: significant autocorrelation up to lag q only.

Step 2: Estimate the model

- Use Maximum Likelihood or numerical optimization.
- Collect residuals $\hat{\epsilon}_t$.

Step 3: Diagnostics

- Check if residuals are white noise:
  - ACF of residuals $\rightarrow$ no significant lags.
  - Ljung–Box test (success if we fail to reject H_0).
- If diagnostics fail: increase q or move to ARMA/ARIMA.

What is an ARMA Model?

Definition:
An ARMA model combines autoregressive (AR) and moving average (MA) components to capture time dependence in both past values *and* past shocks.

$$Y_t = c + \phi_1 Y_{t-1} + \cdots + \phi_p Y_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t$$

- p: AR order (lags of Y_t)
- q: MA order (lags of shocks)
- ϵ_t: white noise error

Key Properties of ARMA(p, q)

- **Hybrid structure:** AR part explains persistence; MA part explains shock propagation.
- **Stationarity:** Requires AR roots outside the unit circle.
- Models with small (p, q) often outperform high-order AR or MA models.

Model Selection and Diagnostics (ARMA)

Step 1: Choose (p, q)

- Estimate candidate models, e.g. $(1, 1)$, $(2, 1)$, $(1, 2)$.
- Select using AIC or BIC.
- Use ACF/PACF behavior.

Step 2: Estimate parameters

- Typically by Maximum Likelihood.
- Extract residuals $\hat{\epsilon}_t$.

Step 3: Diagnostics

- Residuals should behave as white noise:
  - ACF of residuals shows no significant autocorrelation.
  - Ljung–Box test: do *not* reject H_0.
- If diagnostics fail: adjust (p, q) or move to ARIMA.

GARCH: Definition

GARCH = Generalized Autoregressive Conditional Heteroskedasticity

- A model for **time-varying volatility**.
- Often used for **financial returns**.
- Idea: today's volatility depends on
  - a baseline level of volatility,
  - recent shocks (big surprises),
  - and yesterday's volatility.

Intuition: Volatility Clustering

Empirical fact: Financial returns show **volatility clustering**.

- Periods of **calm**: small ups and downs.
- Periods of **turmoil**: large swings up and down.
- Big moves tend to be followed by big moves (of either sign).

GARCH intuition:

- If yesterday was very volatile $\Rightarrow$ today is likely volatile.
- If yesterday was calm $\Rightarrow$ today is likely calm.
- The model lets the **variance** change over time instead of being constant.

What is a GARCH Model?

Definition:
A GARCH model explains time-varying *volatility* rather than the mean of a time series. It is widely used for financial returns where volatility clusters over time.

$$Y_t = \mu + \epsilon_t, \qquad \epsilon_t = \sigma_t z_t$$
$$\sigma_t^2 = \omega + \alpha\epsilon_{t-1}^2 + \beta\sigma_{t-1}^2$$

- μ and ϵ_t ,are two independent process.
- z_t is the standardized shock or innovation, $z_t \sim$ i.i.d., typically normal or t-distributed
- Variance σ_t^2 changes dynamically over time
- GARCH$(1, 1)$ is the most commonly used specification

Key Properties of GARCH

- **Volatility clustering:** Large shocks are followed by large shocks. α measures how much new shocks (yesterday's squared residuals) affect today's volatility. Also called the short-run volatility persistence.
- **Persistence:** β measures how much past volatility carries over into today's volatility. Represents the persistence of volatility over time.
- **Stationarity:** Requires $\alpha + \beta < 1$.
- **Leverage effects:** Captured by variants (EGARCH, GJR-GARCH).
- **Always models variance, not mean.**

Model Selection and Diagnostics (GARCH)

Step 1: Check if GARCH is needed

- Look for volatility clusters in returns plot.
- ARCH test (Engle): reject H_0 = constant variance.

Step 2: Estimate GARCH(p, q)

- Most common: GARCH(1,1)
- Choose distribution distribution of the standardized residuals (Normal, Student-t)

Step 3: Diagnostics

- Check standardized residuals:
  - No autocorrelation in ϵ_t
  - No ARCH structure in ϵ_t^2
- Ljung–Box test on squared residuals.
- If fails: increase (p, q) or use EGARCH / GJR-GARCH.

Introduction to `yfinance`

What is `yfinance`?

- A Python library that provides easy access to financial market data.
- Retrieves stock price histories, financial statements, dividends, splits, and more.
- Designed for analysis, backtesting, machine learning, and portfolio modeling.

Why use it?

- Access live and historical data from Yahoo Finance.
- Simple syntax and integrates seamlessly with `pandas`.
- Fetch hundreds of tickers in one line of code.

Example: Using `yfinance` in 4 Steps

- Install the package: pip install yfinance
- Import the module: import yfinance as yf
- Define a list of tickers: tickers = ["AAPL", "GOOGL", "MSFT", "AMZN"]
- Download the data: data = yf.download(tickers, start="2020-01-01", end="2024-01-01")

Output: a Pandas DataFrame with prices and volumes for each ticker.

Tool Tip: Plotting Time Series with `matplotlib`

Steps to Plot a Time Series:

- Create a figure with appropriate size:
  plt.figure(figsize=(width, height))
- Plot the desired time series:
  plt.plot(dataframe['column'])
- Add a title and axis labels:
  plt.title('Plot Title')
  plt.xlabel('X-axis Label')
  plt.ylabel('Y-axis Label')
- Display the plot:
  plt.show()

Tip: Customize figures to help visualize different financial metrics (e.g., prices, returns, volatility).