Friedrich-Alexander-Universität
Technische Fakultät

Machine Learning for Time Series (MLTS)

Lecture 6: Autoregressive models

Dr. Dario Zanca

Machine Learning and Data Analytics (MaD) Lab
Friedrich-Alexander-Universität Erlangen-Nürnberg

21.11.2024

Topics overview

1. Time series fundamentals and definitions (Part 1)
2. Time series fundamentals and definitions (Part 2)
3. Bayesian Inference and Gaussian Processes
4. State space models (Kalman Filters)
5. State space models (Particle Filters)
6. Autoregressive models
7. Data mining on time series
8. Deep Learning (DL) for Time Series (Introduction to DL)
9. DL – Convolutional models (CNNs)
10. DL – Recurrent models (RNNs and LSTMs)
11. DL – Attention-based models (Transformers)
12. DL – From BERT to ChatGPT
13. DL – New Trends in Time Series processing
14. Time series in the real world

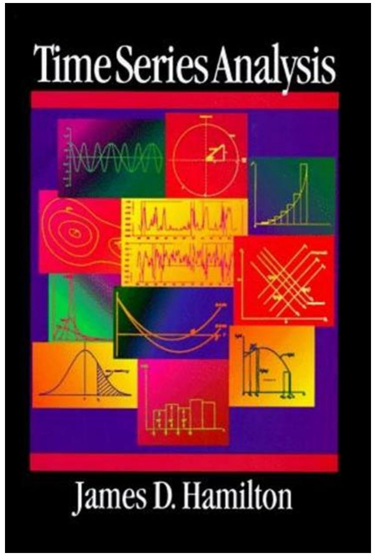
Topics overview

1. Time series fundamentals and definitions (Part 1)
2. Time series fundamentals and definitions (Part 2)
3. Bayesian Inference and Gaussian Processes
4. State space models (Kalman Filters)
5. State space models (Particle Filters)
6. Autoregressive models
7. Data mining on time series
8. Deep Learning (DL) for Time Series (Introduction to DL)
9. DL – Convolutional models (CNNs)
10. DL – Recurrent models (RNNs and LSTMs)
11. DL – Attention-based models (Transformers)
12. DL – From BERT to ChatGPT
13. DL – New Trends in Time Series processing
14. Time series in the real world

References

Time Series Analysis

by J. D. Hamilton (1994)



In this lecture…

1. *Concepts recap*
2. *Autoregressive (AR) and Moving Average (MA) models*
3. *ARMA and ARIMA*
4. *Recap*

Autoregressive models

Concepts recap

Review concept: Stochastic process

Non-deterministic time series can be regarded as manifestations (equiv., realization) of a **stochastic process**, which is in turn defined as a set of random variables $\{X_t\}_{t\in\{1,\ldots,\mathrm{T}\}}$

Even if we were to imagine having observed the process for an infinite period T of time, the infinite sequence

$$S = \{\ldots, s_{t-1}, s_t, s_{t+1,\ldots}\} = \{s_t\}_{t=-\infty}^{+\infty}$$

would still be a single **realization** from that process.

If we had a battery of N computers generating series $S^{(1)}$, …, $S^{(N)}$, and considering selecting the observation at time t from each series,

$$\left\{s_t^{(1)}, \ldots, s_t^{(N)}\right\}$$

this would be described as a sample of **N realizations** of the random variable X_t

Review concept: Autocovariance

Given any particular realization $S^{(i)}$ of a stochastic process (i.e., a time series), we can define the vector of the $j+1$ most recent observations

$$x_t^{\mathrm{i}} = [s_{t-j}^{(i)}, \dots, s_t^{(i)}]$$

We want to know the probability distribution of this vector x_t^{i} across realizations. We can calculate the ***j*-th autocovariance**

$$\gamma_{jt} = E(X_t - \mu_t)(X_{t-j} - \mu_{t-j})$$

Review concept: Autocorrelation function (ACF)

We can express the linear predictability of X_t from an adjacent value X_s, using the **autocorrelation function**:

$$\rho(s,t) = \frac{\gamma_{st}}{\sqrt{\gamma_{ss}\gamma_{tt}}}$$

where γ is the autocovariance defined previously.

Review concept: Stationarity

There are two types of stationarity.

A process is said **strictly stationary** if the joint distribution of $X_{t_1:t_2}$ is the same as that of $X_{t_1+h:t_2+h}$.

The term h is called **lag**.

For stricly stationary time series, all statistics do not depend on time.

A process is said **weakly stationary** if it has:

- $\mu = const.$
- $\sigma^2 < \infty$
- $\gamma_{jt} = \gamma_{j+h,t+h}$

A weakly stationary time series has finite variation, constant first moment, and that the second moment only depends on $h = t - j$.

Review concept: Stationarity

- Strict stationarity $\nRightarrow$ Weakly stationary
    - In fact, strict stationarity does not assume finite variance.

- Property: A nonlinear function of a strictly stationary time series is still strictly stationary
    - But this is not true for weakly stationary!

- Weak stationarity $\nRightarrow$ Strict stationarity
    - Higher moments of the process may depend on time t.
    - What happens if the process is Gaussian?

Review concept: Partial autocorrelation function (PACF)

For stationary time series, the **partial autocorrelation function** expresses the correlation between X_t and an adjacent value X_s, but “removes” the effect of all values in between:

$$\phi_{11} = corr(X_{t+1}, X_t) = \rho_1$$

$$\phi_{hh} = corr\left(X_{t+h} - P_{t,h}(X_{t+h}), X_t - P_{t,h}(X_t)\right) = \rho_h$$

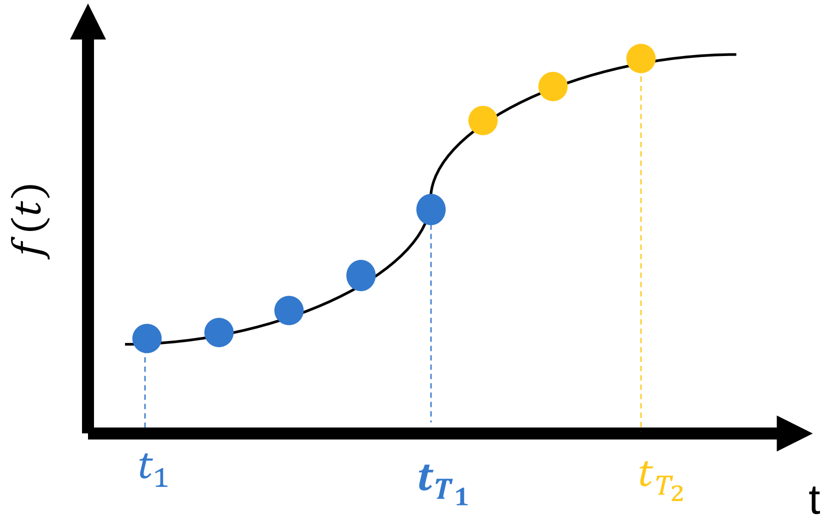
for $h \geq 2$, where $P_{t,h}$ is the surjective operator of orthogonal projection onto the linear subspace spanned by the intermediate values $X_{t+1}, \dots, X_{t+h-1}$.

Review concept: Time series forecasting

Let $S = \{s_1, \dots, s_{T_1}, s_{T_1+1}, \dots, s_{T_2}\}$ be a time series, with s_i being the i-th observation collected at time t_i, and $t_i < t_j, \forall j$.

Then, a time series forecasting task is about predicting future values of a time series given some past data, i.e.,

$$f(s_1, \dots, s_{T_1}) = (s_{T_1+1}, \dots, s_{T_2})$$



Autoregressive models

Autoregressive (AR) and Moving Average (MA) models

White noise

The basic building block for all the processes considered in this lecture is the **white noise**, defined as a sequence

$$\{e_t\}_{t=-\infty}^{+\infty}$$

whose elements have zero mean and variance σ^2, and are *uncorrelated* across time, i.e.,

- $\mathbb{E}(e_t) = 0$ **(zero mean)**
- $\mathbb{E}(e_t^2) = \sigma^2$ **(variance)**
- $\mathbb{E}(e_t e_\tau) = 0$ **(zero autocovariance, i.e., uncorrelated)**

If $e_t \sim \mathcal{N}(0, \sigma^2)$, then we have a so-called **Gaussian white noise**.

Random walk

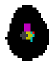
A random walk is a stochastic random process that describes a path started at y_0 and consisting of random steps.

$$y_t = y_{t-1} + e_t$$

Equiv., for $t \geq 1$:

$$y_t = y_0 + \sum_{i=1}^{t} e_i$$

where e_i can be regarded as random variables of a white noise process.



https://www.mit.edu/~kardar/teaching/projects/chemotaxis(AndreaSchmidt)/random.htm

Linear process representation

A linear process can be represented as an **infinite moving average process,** starting from a white noise $\{e_t\}_{t=-\infty}^{+\infty}$, as

$$\begin{aligned} y_{\mathrm{t}} &= \mu + e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \cdots \\ &= \mu + e_t + \sum_{j=1}^{+\infty} \psi_j e_{t-j} \\ &= \mu + \Psi(q^{-1}) e_t \end{aligned}$$

where,

- ψ_i are constant values
- q^{-m} is the *backshift operator*, such that $q^{-m} e_t = e_{t-m}$
- $\Psi(\mathrm{q}^{-1}) = 1 + \psi_1 q^{-1} + \psi_2 q^{-2} + \cdots = \sum_{j=0}^{+\infty} \psi_j q^{-j}$ is a **linear filter**.

Property: If the sequence $\psi_1, \psi_2, \dots,$ has finite sum $\sum_i^{\infty} \psi_i < \infty$, then the filter is stable and the process y_{t} is stationary.

Linear process representation

Alternatively, a linear process can be represented with respect to its previous values as an **infinite autoregressive process**:

$$y_t = \mu + e_t + \pi_1 y_{t-1} + \pi_2 y_{t-2} + \cdots$$

$$y_t = \mu + e_t + \sum_{j=1}^{+\infty} \pi_j y_{t-j}$$

$$\Pi(q^{-1}) y_t = \mu + e_t$$

where, similarly,

- π_i are constant values
- q^{-m} is the *backshift operator*, such that $q^{-m} e_t = e_{t-m}$
- $\Pi(\mathrm{q}^{-1}) = 1 + \pi_1 q^{-1} + \pi_2 q^{-2} + \cdots = \sum_{j=0}^{+\infty} \pi_j q^{-j}$ is a **linear filter**.

Linear process representation

The previous two formulations are algebrically equivalent, in fact:

$$y_t = \mu + e_t + \sum_{j=1}^{+\infty} \pi_j y_{t-j}$$ (infinite autoregressive process)

$$y_t = \mu + e_t + \pi_1 q^{-1} y_t + \cdots$$

$$y_t - \pi_1 \, q^{-1} y_t - \cdots = \mu + e_t$$

$$(1 - \pi_1 \, q^{-1} - \cdots) \, y_t = \mu + e_t$$

$$\Pi(q^{-1}) y_t = \mu + e_t$$

If the linear filter $\Pi(q^{-1})$ is **invertible**, then:

$$y_t = \bar{\mu} + \frac{1}{\Pi(q^{-1})} e_t$$ (infinite moving average process)

Autoregressive models (AR)

Autoregressive models are based on the idea that the value of a time series at time t can be expressed as a linear combination of n past values, up to a random error:

$$AR(n): y_t = a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_n y_{t-n} + e_t$$

Where:

- n is the model's order
- $a_1, \dots, a_n$ are the model's parameters, $a_n \neq 0$

In other words, the hyper-parameter n represents how far back to look for dependences with previous values in the time series.

Autoregressive models (AR)

We can simplify the notation for $AR(n)$ using the backshift operator:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \cdots + a_n y_{t-n} + e_t$$

$$y_t - a_1 y_{t-1} - \ldots - a_n y_{t-n} = e_t$$

$$(1 - a_1 q^{-1} - \cdots - a_n q^{-n}) y_t = e_t$$

$$\boldsymbol{A(q^{-1}) y_t = e_t}$$

where $\boldsymbol{A(q^{-1})}$ is called **autoregressive operator**.

Example: AR(0)

The simplest autoregressive model is **AR(0)**, which has no dependences between values in the time series.

$$AR(0): y_t = e_t$$

→ AR(0) is equivalent to a white noise process.

Example: AR(1)

The first order autoregressive model AR(1) can be written as:

$$AR(1)\colon\ y_t = a_1 y_{t-1} + e_t$$

Notice that:

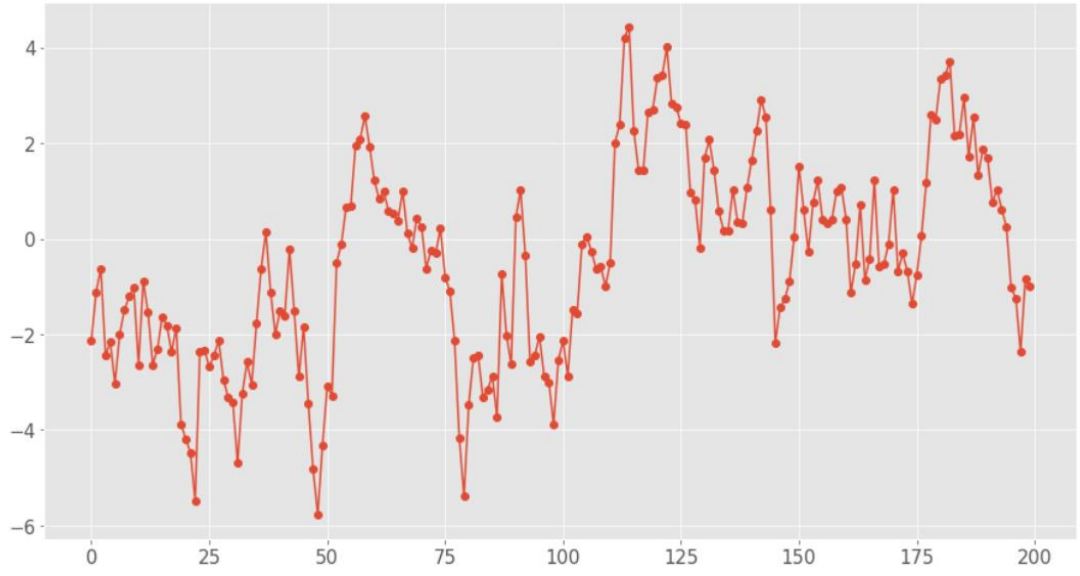
- Only the previous term y_{t-1} and the current noise e_t contribute to the output.
- As $|a_1| \to 0$, the process looks like white noise.
- When $a_1 < 0$, the process oscillates around zero.
- When $a_1 = 1$, the process is equivalent to a *random walk*.

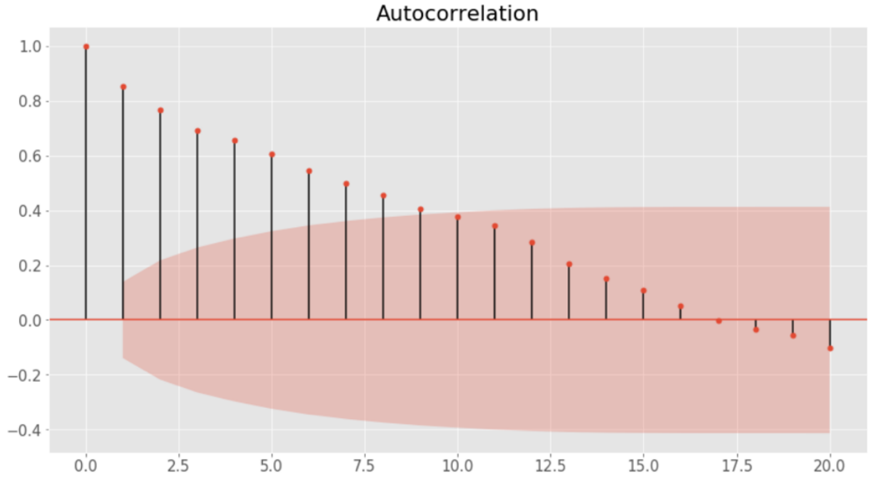
Example: AR(1)

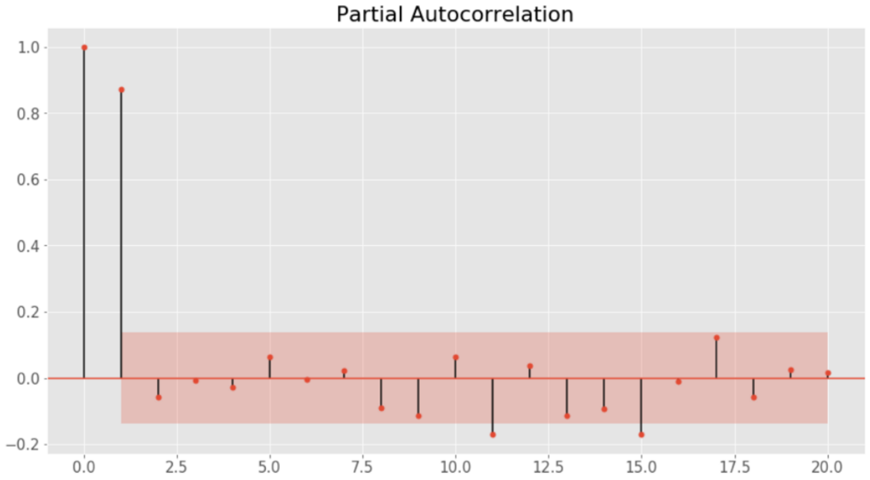
A numerical example could be given by

$$y_t = \mathbf{0.9}\, y_{t-1} + e_t$$

Where, e.g., the Gaussian white noise $e_t = \mathcal{N}(0,1)$.





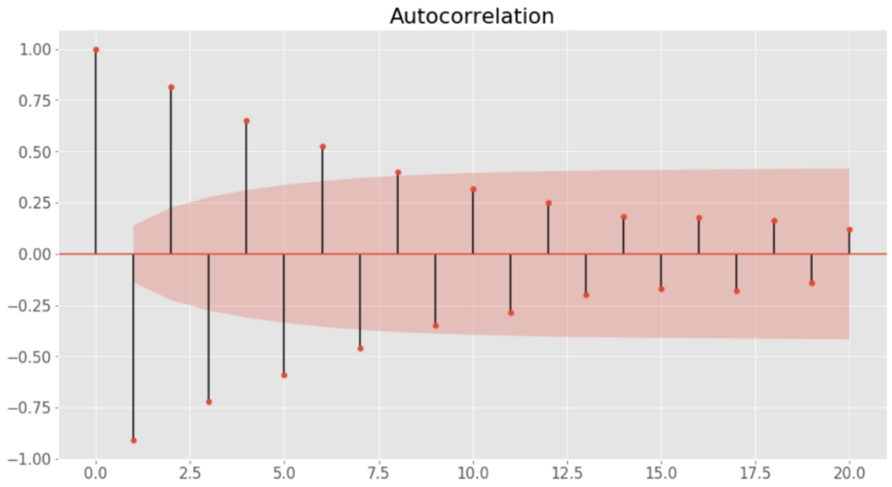


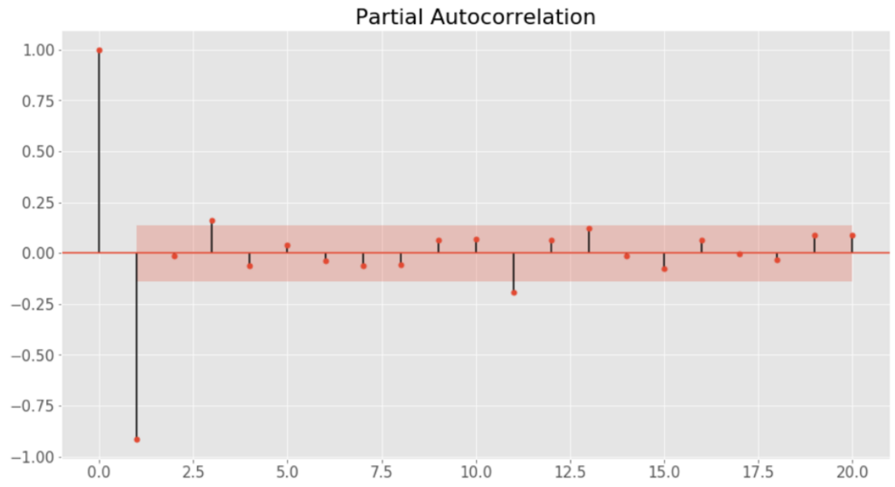
Example: AR(1)

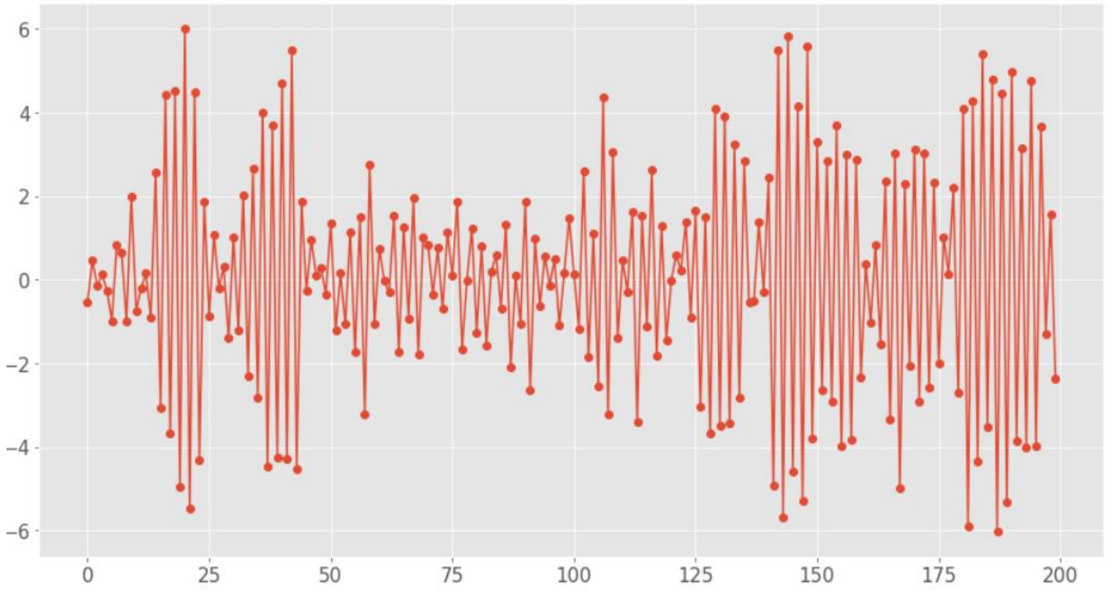
Another numerical example could be given by

$$y_t = \mathbf{-0.9}\, y_{t-1} + e_t$$

Where, e.g., the Gaussian white noise $e_t = \mathcal{N}(0,1)$.







Chosing an AR(n)

In concrete applications, the value of n is an hyper-parameter to optimize.

It is possible to identify the $AR(n)$ model by looking at the partial autocorrelation function (PACF). In fact:

- The theoretical partial autocorrelation for lags $h > n$ is zero.
  → For concrete experimental data, it might be small but non-zero.

- For $h = n$ the partial autocorrelation ϕ_n is not zero.
  → For all lag values in between, it is not necessarily zero

Moving Average models (MA)

Moving average models (MA) are based on the idea that the value of a time series at time t can be expressed as a linear combination of n past input random shock (or white noise).

$$MA(m): y_t = e_t + b_1 e_{t-1} + \cdots + b_m e_{t-m}$$

Where:

- m is the model‘s order
- $b_1, \dots, b_m$ are the model‘s parameters, $b_m \neq 0$

In other words, the hyper-parameter m, again, represents how far back to look for dependecies with previous noise values.

Moving Average models (MA)

Similarly to the autoregressive case, the moving average model $MA(m)$ can be expressed with a more synthetic notation by using the backshift operator:

$$y_t = e_t + b_1 e_{t-1} + \cdots + b_m e_{t-m}$$

$$y_t = (1 + b_1 q^{-1} + \cdots + b_m q^{-m}) e_t$$

$$\boldsymbol{y_t = B(q^{-1}) e_t}$$

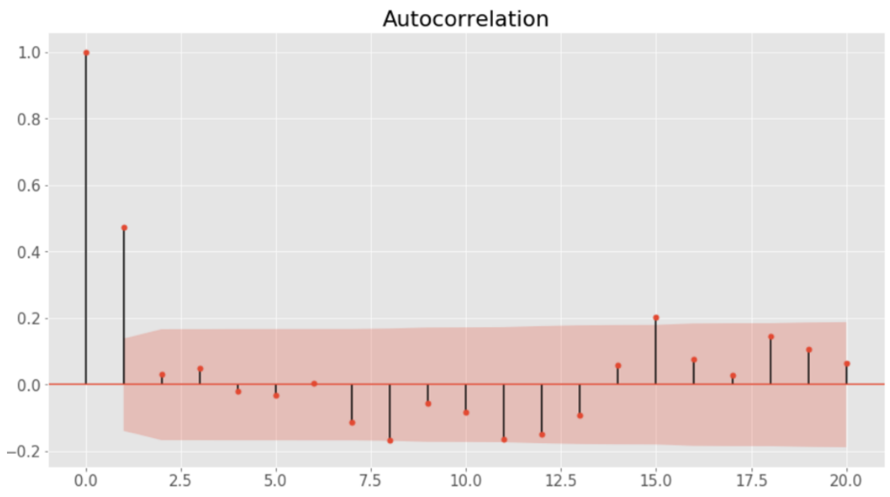
where $\boldsymbol{B\left(q^{-1}\right)}$ is called **moving average operator.**

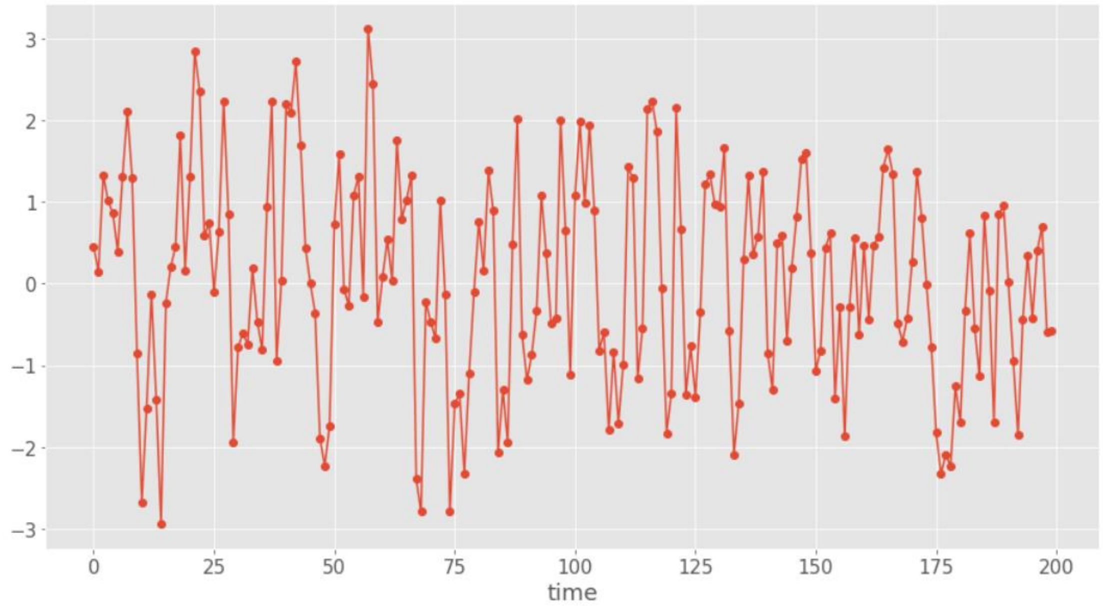
Example: MA(1)

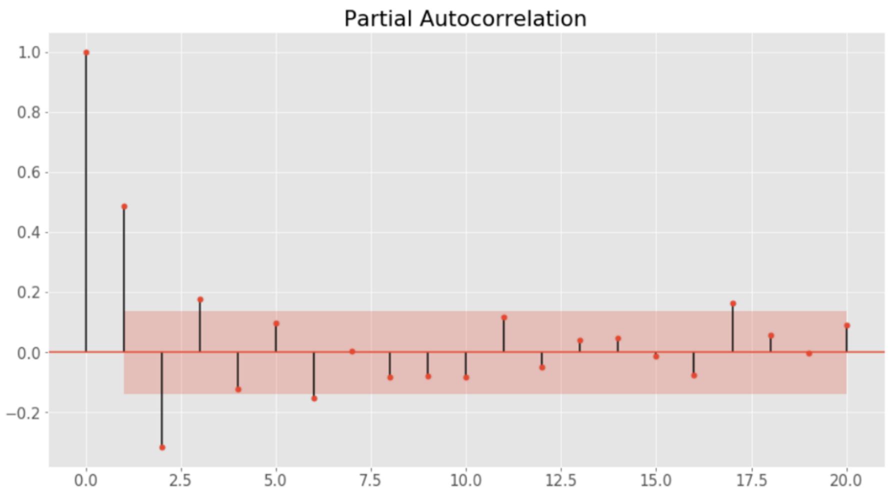
A numerical example could be given by

$$y_t = e_t + \mathbf{0.8}\, e_{t-1}$$

Where, e.g., the Gaussian white noise $e_t = \mathcal{N}(0, 1)$.







Chosing an MA(m)

In concrete applications, the value of m is an hyper-parameter to optimize.

It is possible to identify the $MA(m)$ model by looking at the autocorrelation function (ACF). In fact:

- The theoretical autocorrelation for lags $h > m$ is zero.
  → For concrete experimental data, it might be small but non-zero.

- For $h = m$ the autocorrelation ρ_m is not zero.
  → For all lag values in between, it is not necessarily zero

Properties

- Autoregressive models (AR) ignore correlated noise structures in the time series.
- Differently by AR models, finite moving average models (MA) are always stationary.
- It can be proved that:
  - All finite autoregressive processes $AR(n)$ are infinite moving average processes
  - All finite and invertible moving average $MA(m)$ processes are infinite autoregressive processes
- In practice, parameter estimation for MA models is generally more difficult than for AR models.

Critical analysis: AR vs MA

We saw that AR(p) and MA(q) are equivalent for infinite processes (i.e., AR → MA(∞) and vice versa). **However:**

- **Finite models (e.g., AR(2) vs. MA(2)) have distinct structures:**
  - **AR(p)**: Uses lagged values of the series → better for gradual autocorrelation decay.
  - **MA(q)**: Uses lagged noise terms → better for short-term dependencies.
- **Applications:**
  - AR models: Simpler to estimate, generally suited for forecasting.
  - MA models: Harder to estimate, better for sharp autocorrelation cut-offs.

→ Finite AR and MA models approximate processes differently, leading to distinct results in real-world usage.

Autoregressive models

ARMA and ARIMA models

ARMA models

ARMA model is a combination of **autoregressive (AR)** and **moving average (MA)** models.

$$ARMA(n,m): y_t = \boldsymbol{a_1 y_{t-1}} + \boldsymbol{a_2 y_{t-2}} + \cdots + \boldsymbol{a_n y_{t-n}} + \boldsymbol{b_1 e_{t-1}} + \cdots + \boldsymbol{b_m e_{t-m}} + e_t$$

Which can be re-written using the backshift notation as:

$$ARMA(n,m):\ \boldsymbol{A}(\boldsymbol{q}^{-1})\boldsymbol{y}_t = \boldsymbol{B}(\boldsymbol{q}^{-1})\boldsymbol{e}_t$$

Where $\boldsymbol{A}(\boldsymbol{q}^{-1})$ is the autoregressive operator and $\boldsymbol{B}(\boldsymbol{q}^{-1})$ is the moving average operator, as defined previously.

Chosing an ARMA(n,m)

We can observe the ACF and PACF to determine the suitable hyper-parameters n and m.

	$AR(n)$	$\boldsymbol{MA(m)}$	$AR\boldsymbol{MA}(n, \boldsymbol{m})$
ACF	Tails off	Cuts off after lag m	Tails off
PACF	Cuts off after lag n	Tails off	Tails off

The choice of n and m is not unique.

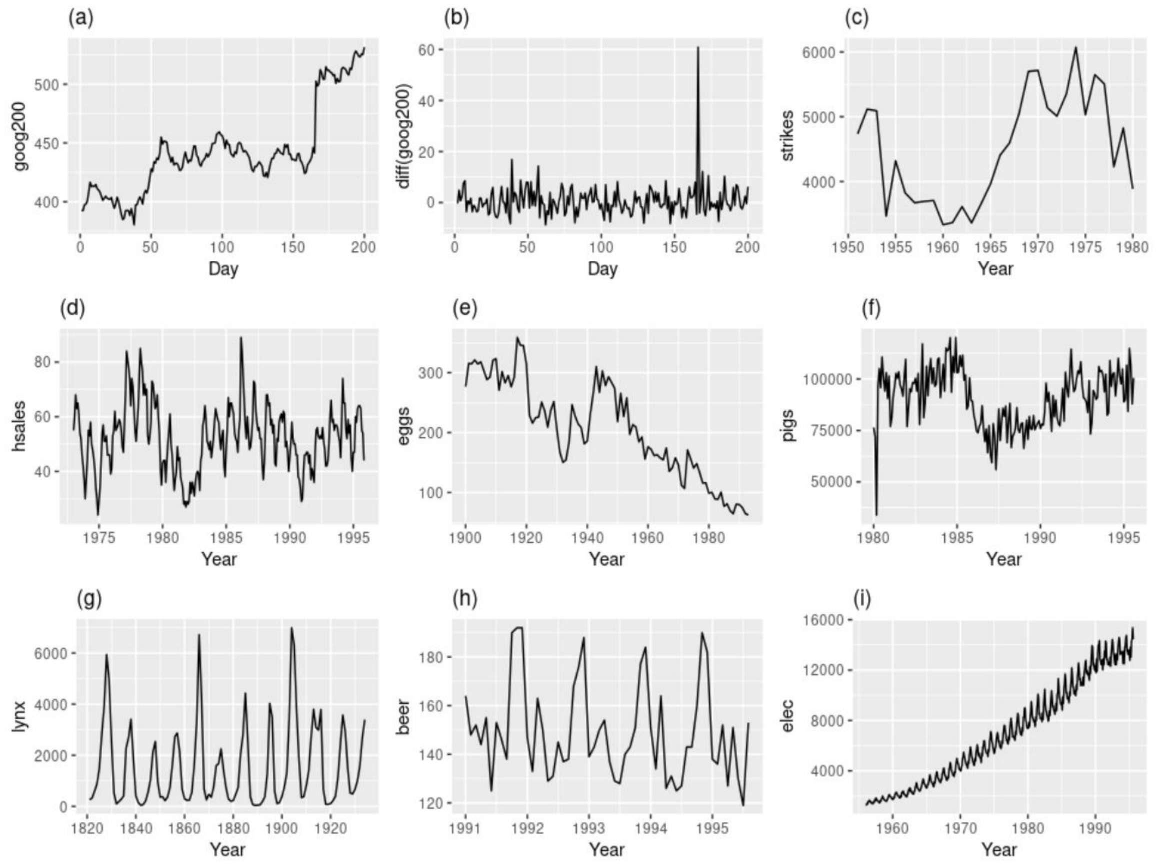
How to deal with Nonstationary time series?

A limitation of the ARMA models is the assumption of our time series to be stationary.

In practice, we can often assume the time series to be composed by a **non-stationary trend** and a **zero-mean stationary time series**, i.e.,

$$y_t = \mu_t + \phi_t$$

→ We can **„stationarize“** time series.



We can stationarize in two ways:

- Detrending
- Differencing

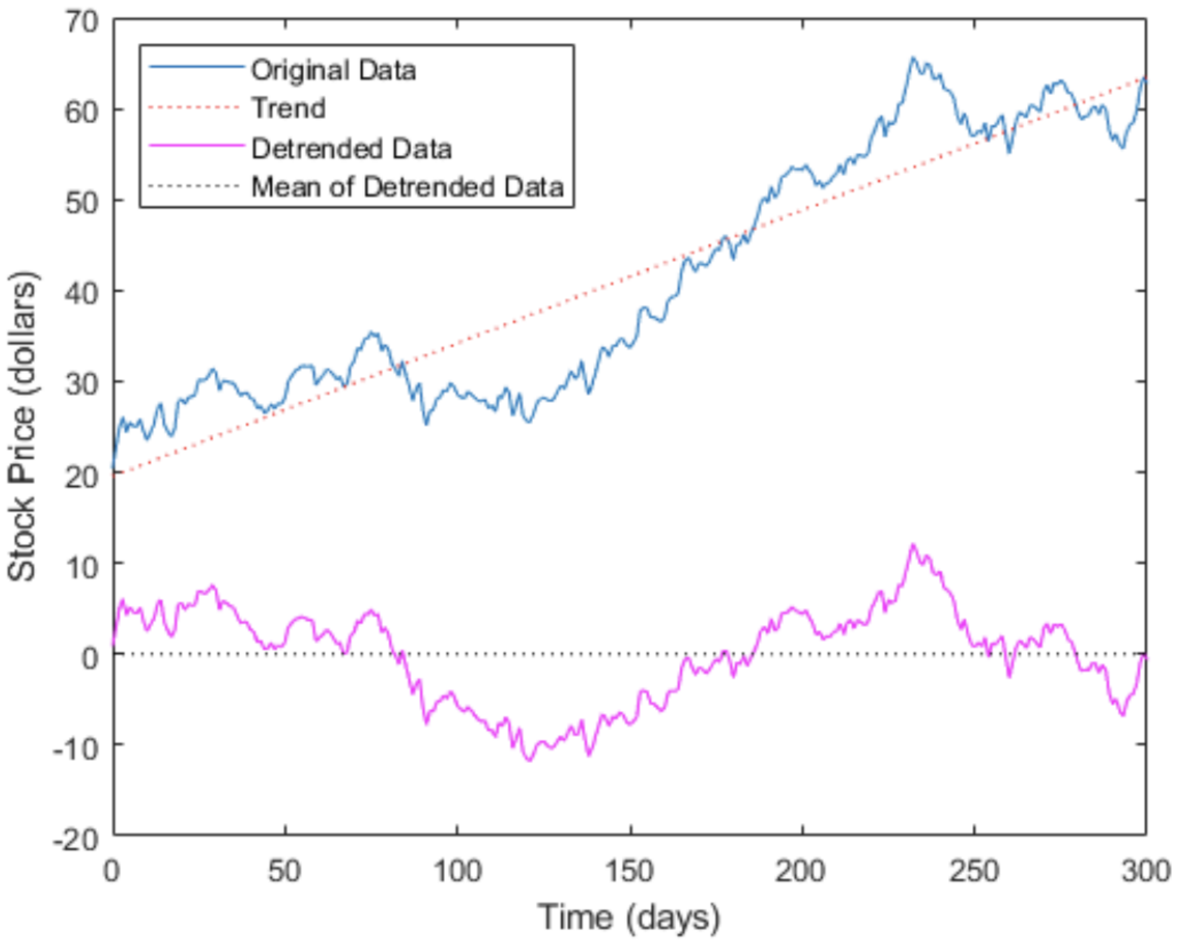
Stationarization: Detrending

By **detrending**, we can subtract an estimate for the time series‘ trend and deal with the remaining terms (i.e., residuals)

In formulas,

$$\hat{y}_t = \mathrm{y_t} - \hat{\mu}_t$$

Detrending needs parameters estimation.



https://www.mathworks.com/help/matlab/data_analysis/detrending-data.html

Stationarization: Differencing

The differencing operator is defined by

$$\nabla y_t = y_t - y_{t-1}$$

By using our backshift operator, the operator can be written as

$$\nabla = 1 - q^{-1}$$

Higher-order d differencing operations are given by

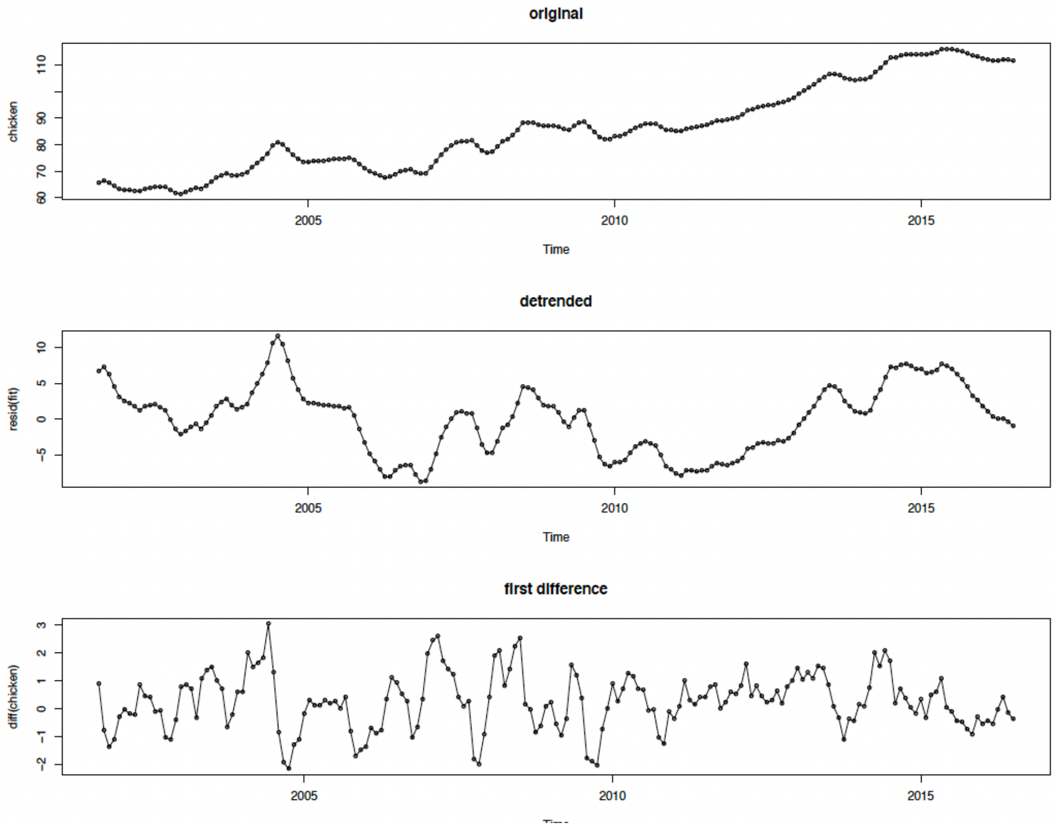
$$\nabla^d = (1 - q^{-1})^d$$

Stationarization: Differencing

By **differencing**, we compute the differences (or higher-order differences) of consecutive observations.

An advantage over detrending is that we do not need to estimate any parameters.

The differencing operation helps to stabilize the mean of a time series, by removing trends and seasonality.



Combine Stationarization with ARMA

Steps to combine our stationarization techniques, in combination with ARMA:

1. **Diagnose Non-Stationarity** (Use tests: Augmented Dickey-Fuller (ADF) or KPSS to confirm)
2. **Apply Stationarity Transformations** (Detrending or Differencing)
3. **Iterative Approach** (Check stationarity after each step)
4. **Combining with ARMA**
    1. **Once Stationary:** Fit ARMA model to the transformed data
    2. **Post-Modeling:** Reverse transformations (e.g., integrate differenced series) to return predictions to the original scale.

ARIMA models

A process y_t is said to be $ARIMA(n, d, m)$ if d-th order differenciation $\nabla^d y_t$ is $ARMA(n, m)$.

Then, the ARIMA model can be written as

$$\boldsymbol{ARIMA}(n, \mathbf{d}, m):\ A(q^{-1})\boldsymbol{\nabla}^{\mathbf{d}} y_t = B(q^{-1}) e_t$$

Notice that, $ARIMA(n, 0, m)$ is equivalent to $ARMA(n, m)$.

Considerations When Using ARIMA

1. Given its complexity, ARIMA can be prone to overfitting.

→ Avoid overly complex models. Adding unnecessary AR or MA terms can reduce predictive accuracy.

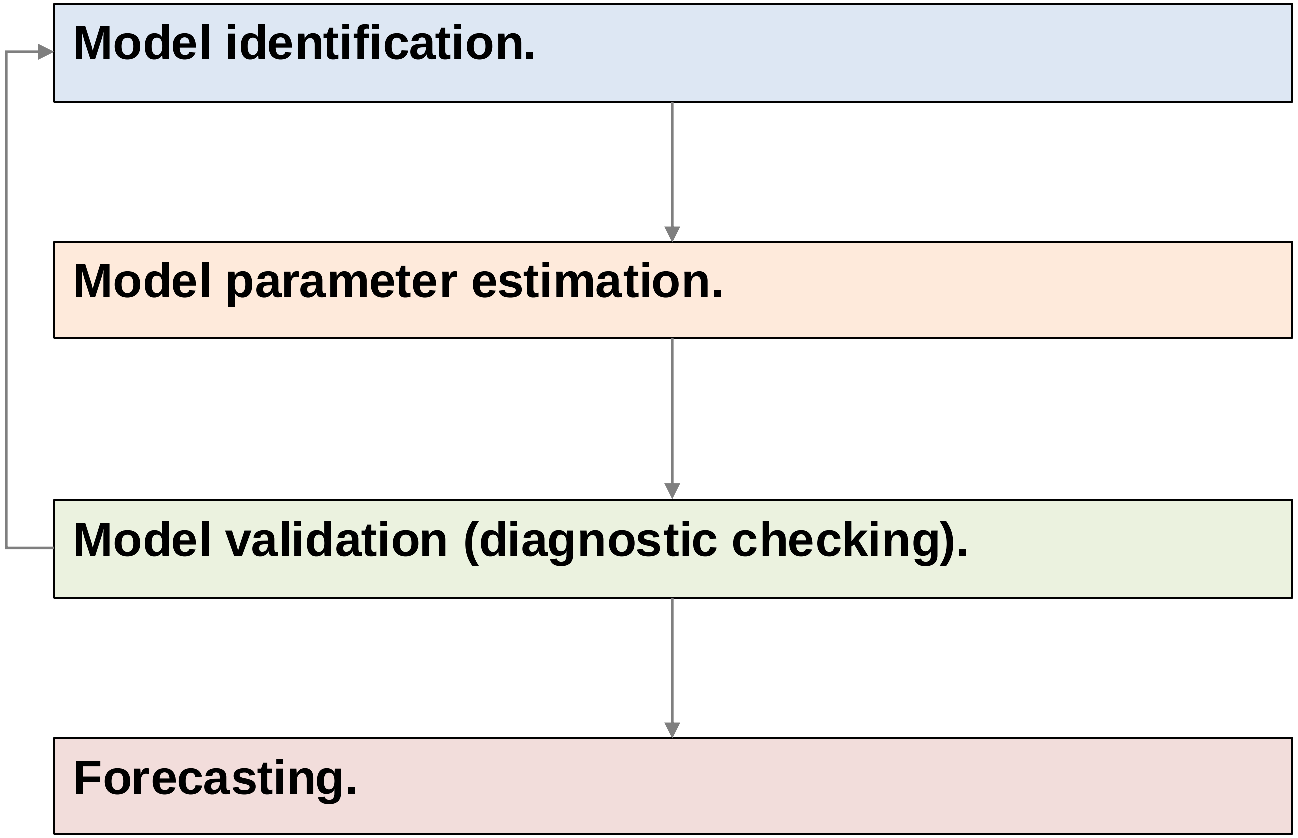
2. Basic ARIMA models cannot handle seasonal patterns directly.

→ Variations exists, such as SARIMA or seasonal differencing for seasonal data.

3. Residuals (errors) should be white noise (uncorrelated and normally distributed).

→ Check residual diagnostics post-model fitting.

General scheme



- Check stationarity and seasonality, perform differentiation if necessary, to chose ARIMA(n, d, m).
- Determine the model's parameters that produce the best fitting, e.g., by Least square (LS) or Maximum likelihood estimation (MLE) methods.
- Perform a diagnostic checking, for example, by residual series analysis.
- We use the selected model for forecasting.

ARIMA: Pros and Cons

Pros:

- Effective in short-term series forecasting.
- E.g., short-run inflation forecasts.
- Effective for a variety of time series (non-seasonal, stationary, and differenced data).
- It is a parametric model and it works better with relatively small number of observations.
- Well-established methodology with clear interpretability of components.

Cons:

- Selecting optimal (n, d, m) values can be challenging and time-consuming.
- Cannot capture nonlinear relationships or structural breaks in data.
- ARIMA models performance are poor at predicting series with turning points.
- Forecast uncertainty increases rapidly for long horizons.
- Seasonality Limitation:
  Requires extensions like SARIMA for seasonal patterns.

Lecture title

Recap

In this lecture…

- Linear processes
  - Autoregressive processes (AR)
  - Moving average processes (MA)
- Combining AR and MA:
  - ARMA
  - ARIMA