

Machine Learning for Time Series

(MLTS or MLTS-Deluxe Lectures)

Dr. Dario Zanca

Machine Learning and Data Analytics (MaD) Lab
Friedrich-Alexander-Universität Erlangen-Nürnberg
07.12.2023

- Time series fundamentals and definitions (2 lectures)
- Bayesian Inference (1 lecture)
- Gaussian processes (2 lectures)
- State space models (2 lectures)
- Autoregressive models (1 lecture) ←
- Data mining on time series (1 lecture)
- Deep learning on time series (4 lectures)
- Domain adaptation (1 lecture)

Review concept: Stochastic process

*Non-deter.
↳ as manifestations
of S.P (set of random var)*

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

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

∞ seq = single realization. $S = \{\dots, s_{t-1}, s_t, s_{t+1}, \dots\} = \{s_t\}_{t=-\infty}^{+\infty}$

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

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

$$\{s_t^{(1)}, \dots, s_t^{(N)}\}$$

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

*vector
of $j + 1$
most recent.*

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

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

γ_{jt} 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:

linear predictability of X_t

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

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 $X_{t_1+h:t_2+h}$.

joint distribution has a lag.

The term h is called **lag**.

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

\hookrightarrow No time dependence

A process is said weakly stationary if it has:

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

*first moment constant
finite variation
2nd moment only
depends on $h = t - j$*

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

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:

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

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

↳ surjective operator

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

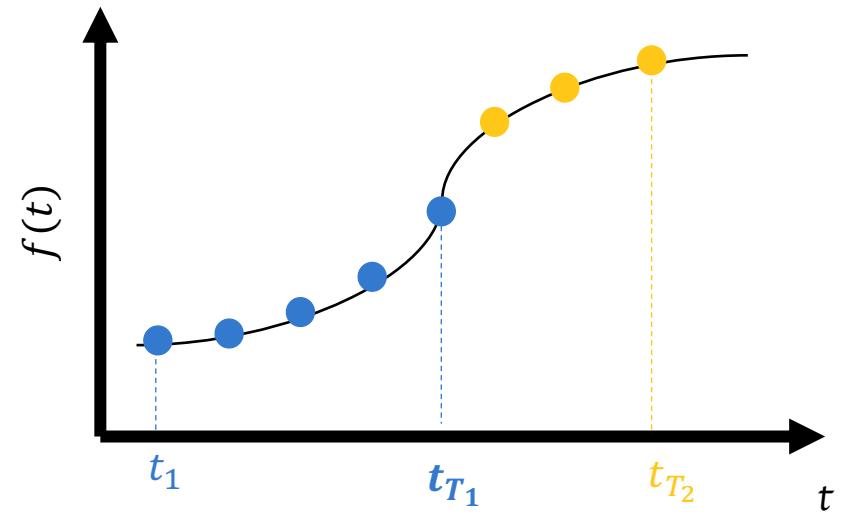
Review concept: Time series forecasting

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.,

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

forecasting.



In this lecture...

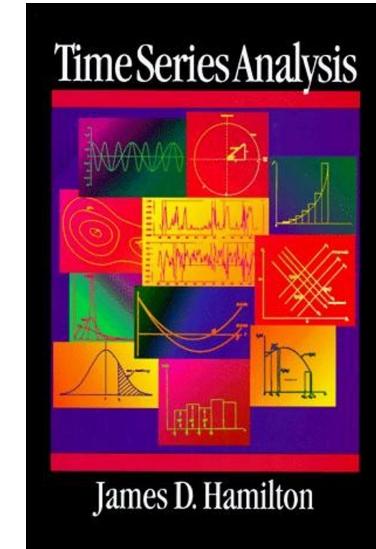
- Linear processes
 - Autoregressive processes (AR)
 - Moving average processes (MA)
- Combining AR and MA:
 - ARMA
 - ARIMA

→ Autoregressive processes (AR)
moving average processes (MA)

References

Time Series Analysis

by J. D. Hamilton (1994)





Autoregressive models

AR and 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)

(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.

random steps.

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

Equiv., for $t \geq 1$:

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

random white noise step

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

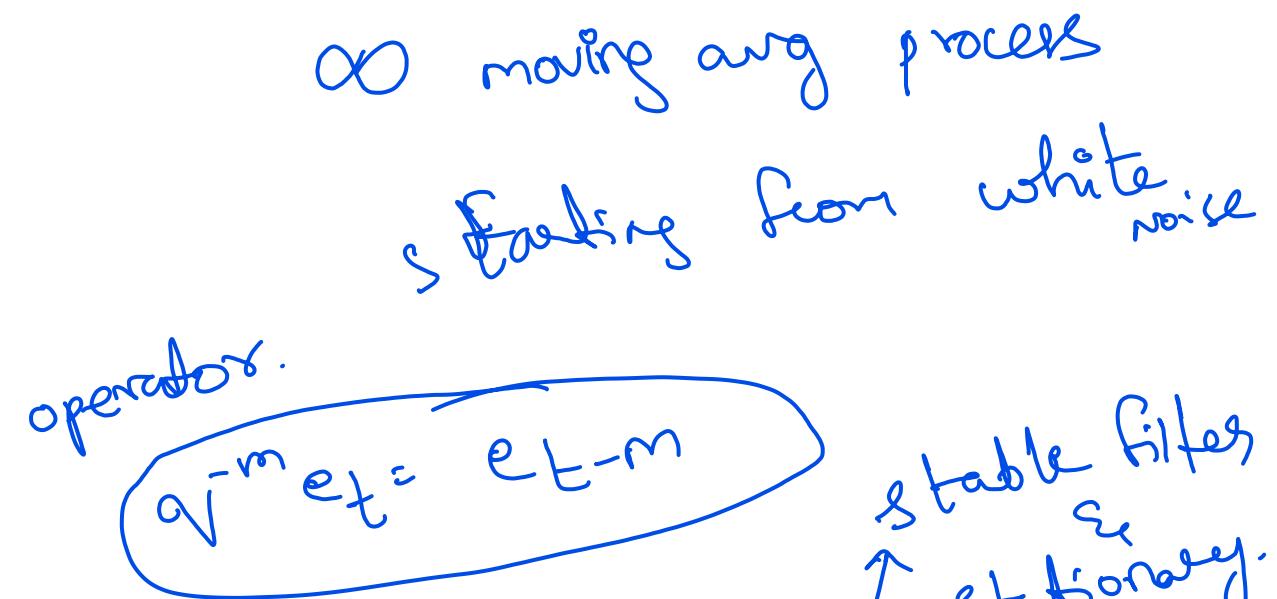
Linear process representation

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

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

where,

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



If the sequence ψ_1, ψ_2, \dots , has finite sum $\sum_{i=1}^{\infty} \psi_i < \infty$, 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:

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

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

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

where, similarly,

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

Linear process representation

The previous two formulations are algebraically equivalent, in fact:

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

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

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

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

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

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

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

$\text{AP} = \infty \text{ MA process}$

Invertible

$$\frac{\mu}{\Pi(q^{-1})}$$

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:

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

linear combination of n past values
up to error.

Where:

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

model params

(How far should I look)

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:

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

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

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

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

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

↑ error at time instance t

$$y_{t-1} = q^{-1}(y_t)$$

$$q^m e_t = e_{t-m}$$

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.

↳ white noise process.

Example: AR(1)

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

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

Notice that:

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

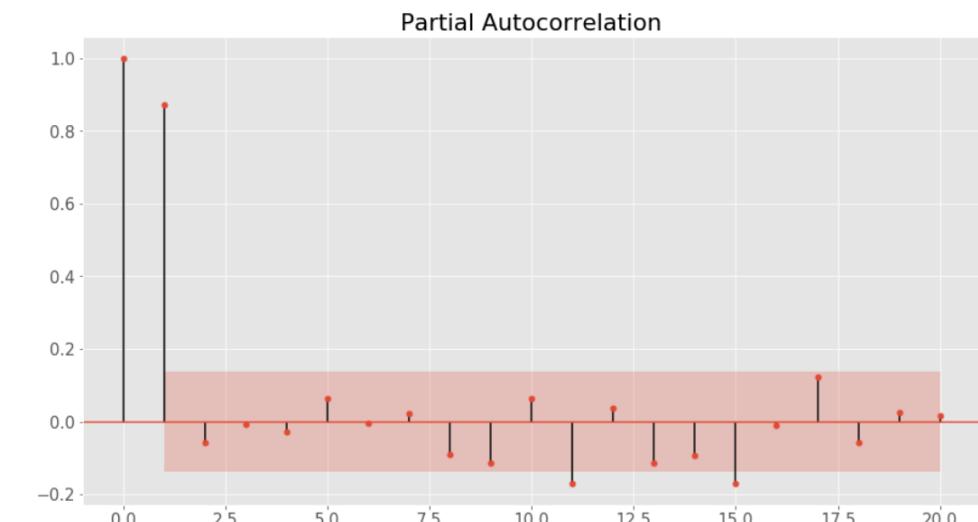
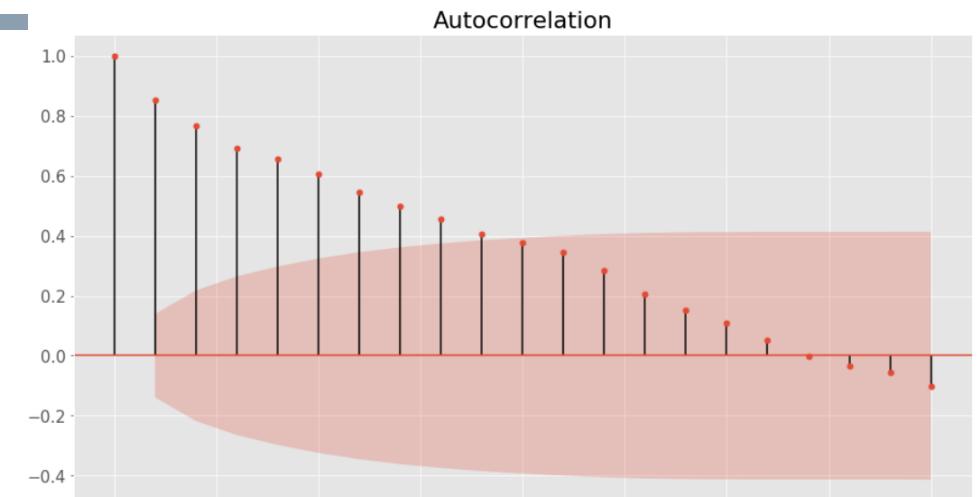
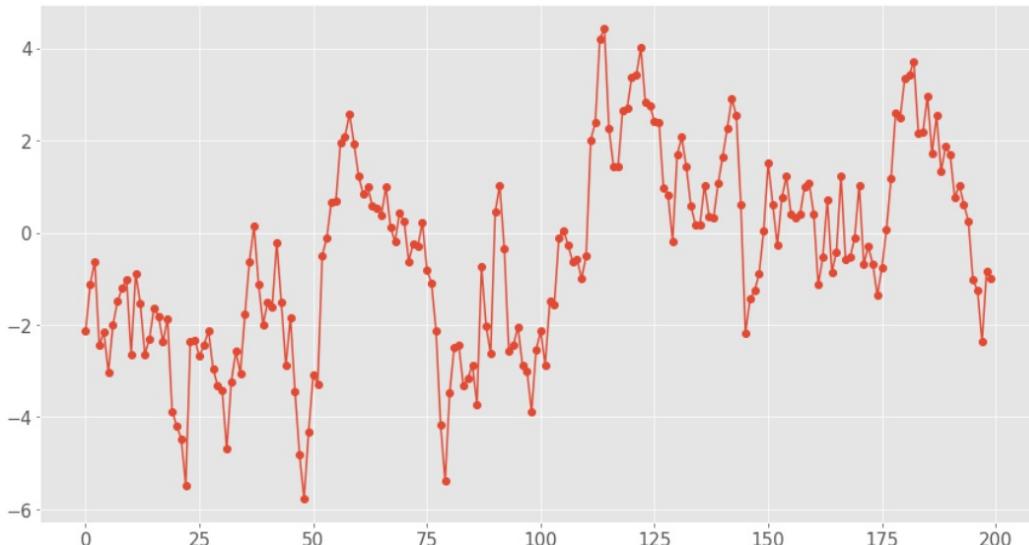
Example: AR(1)

A numerical example could be given by

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

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

random walk.



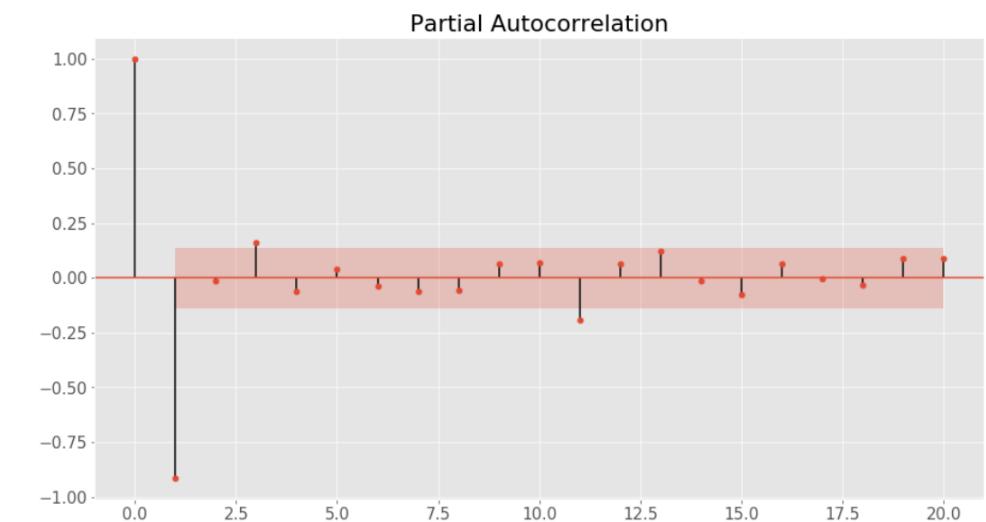
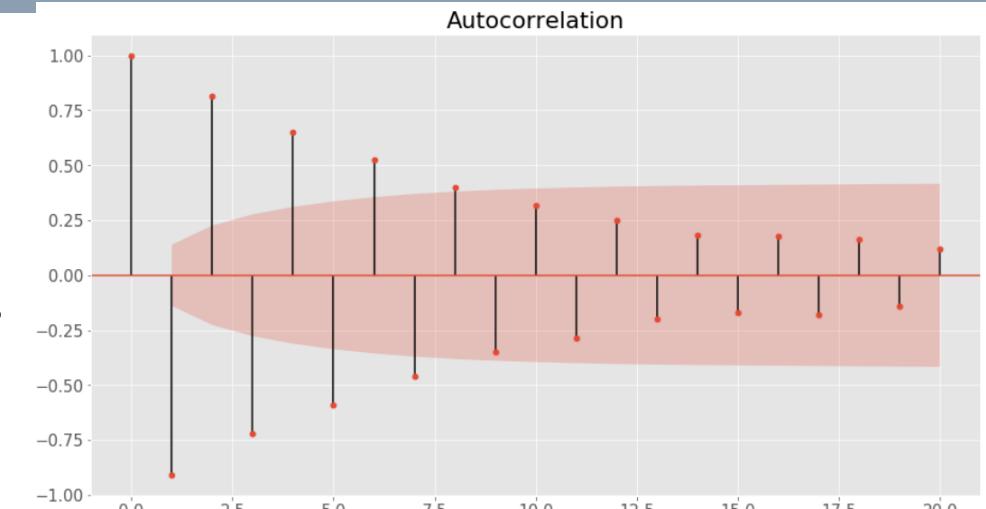
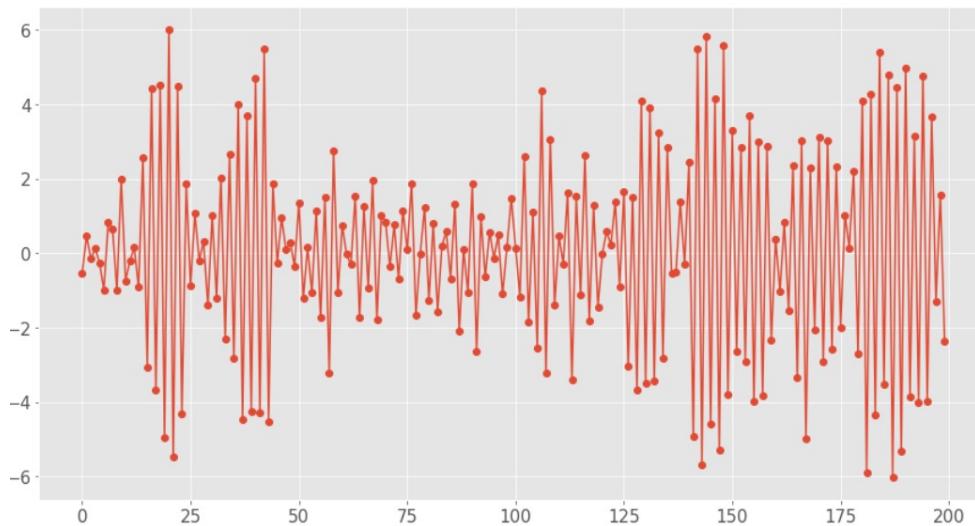
Example: AR(1)

A numerical example could be given by

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

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

Oscillates around zero



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.

(practice - 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

(zero) $h > n$

(not zero) $h = n$

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, $b_m \neq 0$

$\leftarrow n'$ past ifp random
shocks.
(How far to look back)

In other words, the hyper-parameter m , again, represents how far back to look for dependencies 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:

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

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

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

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

C.S.

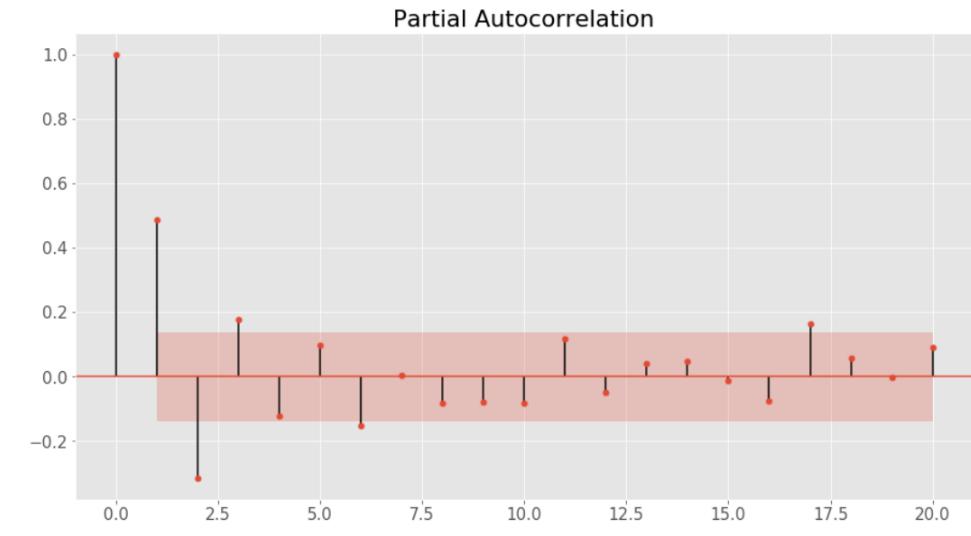
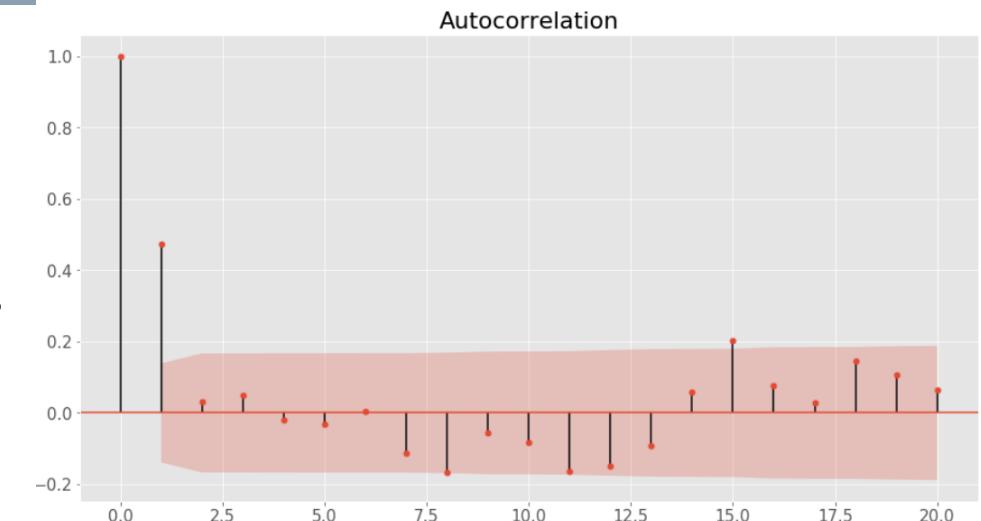
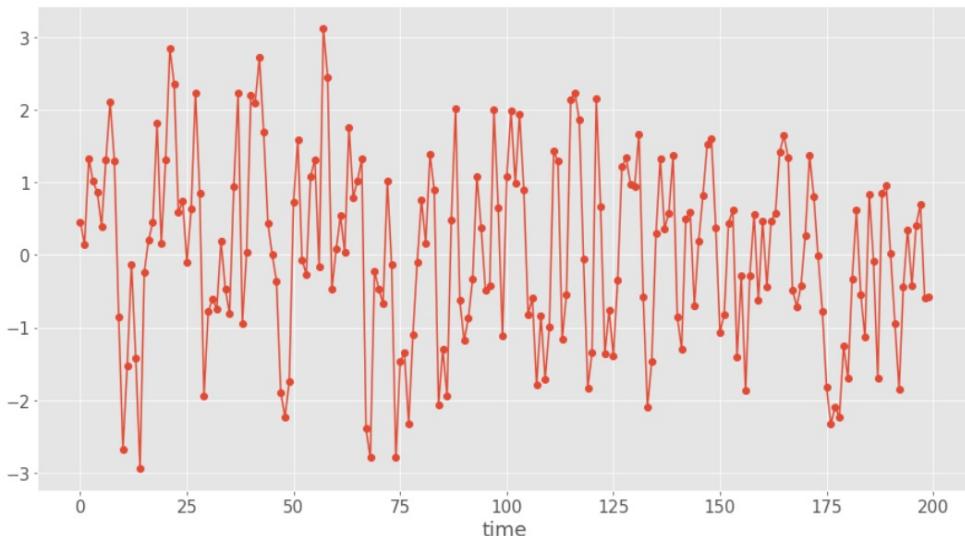
Example: MA(1)

A numerical example could be given by

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

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

Sum of two Gaussians



Chosing an MA(m)

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

(possible from ACF)

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.

$h > m$ (zero)

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

(not zero)

Critical comparison

- 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.

AR ignore correlated noise structures

↳ MA → stationary.

All AR & MA

MA difficult than AR models.



Autoregressive models

ARMA and ARIMA models



ARMA models

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

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

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

$$ARMA(n, m): A(q^{-1})y_t = B(q^{-1})e_t$$

Where $A(q^{-1})$ is the autoregressive operator and $B(q^{-1})$ 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 .

	$AR(n)$	$MA(m)$	$ARMA(n, m)$
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.

not fixed!

How to deal with Nonstationary time series?

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

Many times, we can assume the time series to composed by a non-stationary trend and a zero-mean stationary time series, i.e.,

$$y_t = \mu_t + \phi_t$$

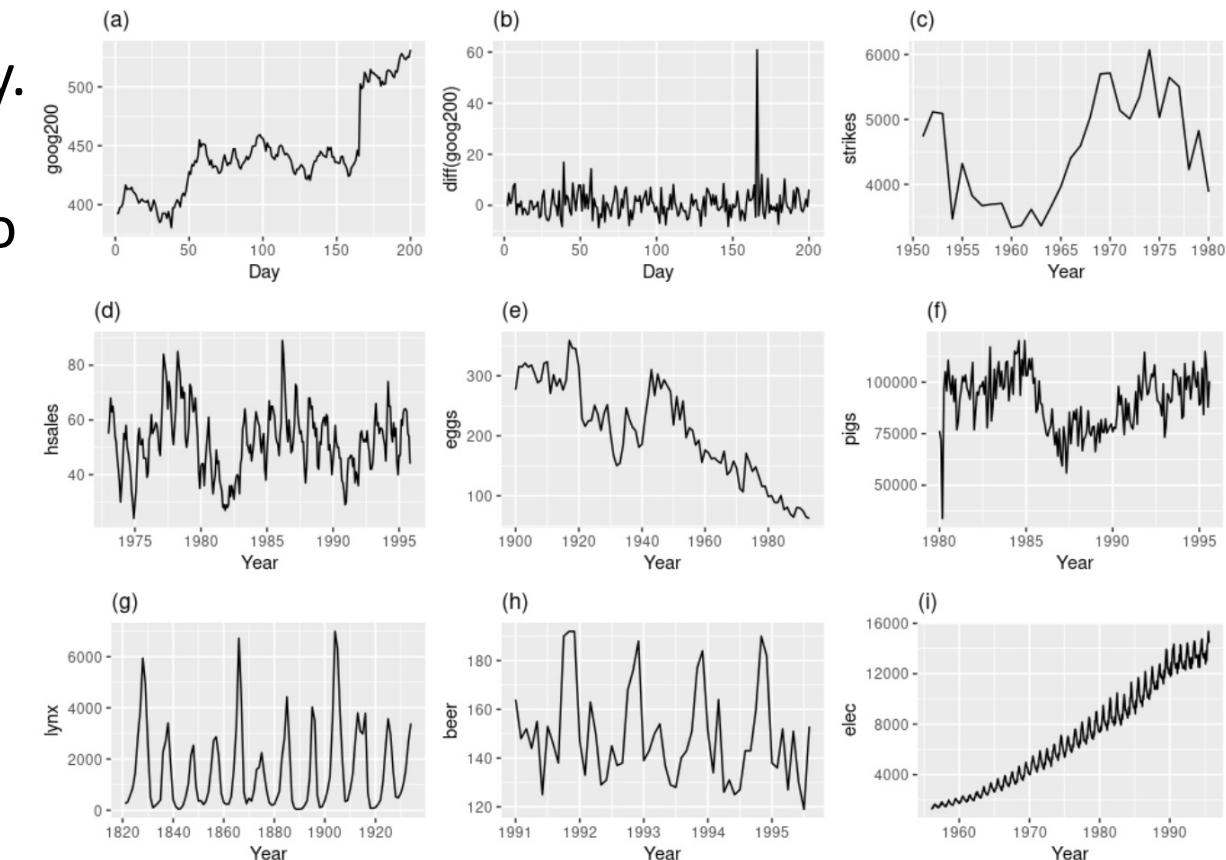
→ We can „stationarize“ time series.

↳ stationarize

We can stationarize in two ways:

- Detrending
- Differencing

stationary(t-s)



Stationarization: Detrending

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

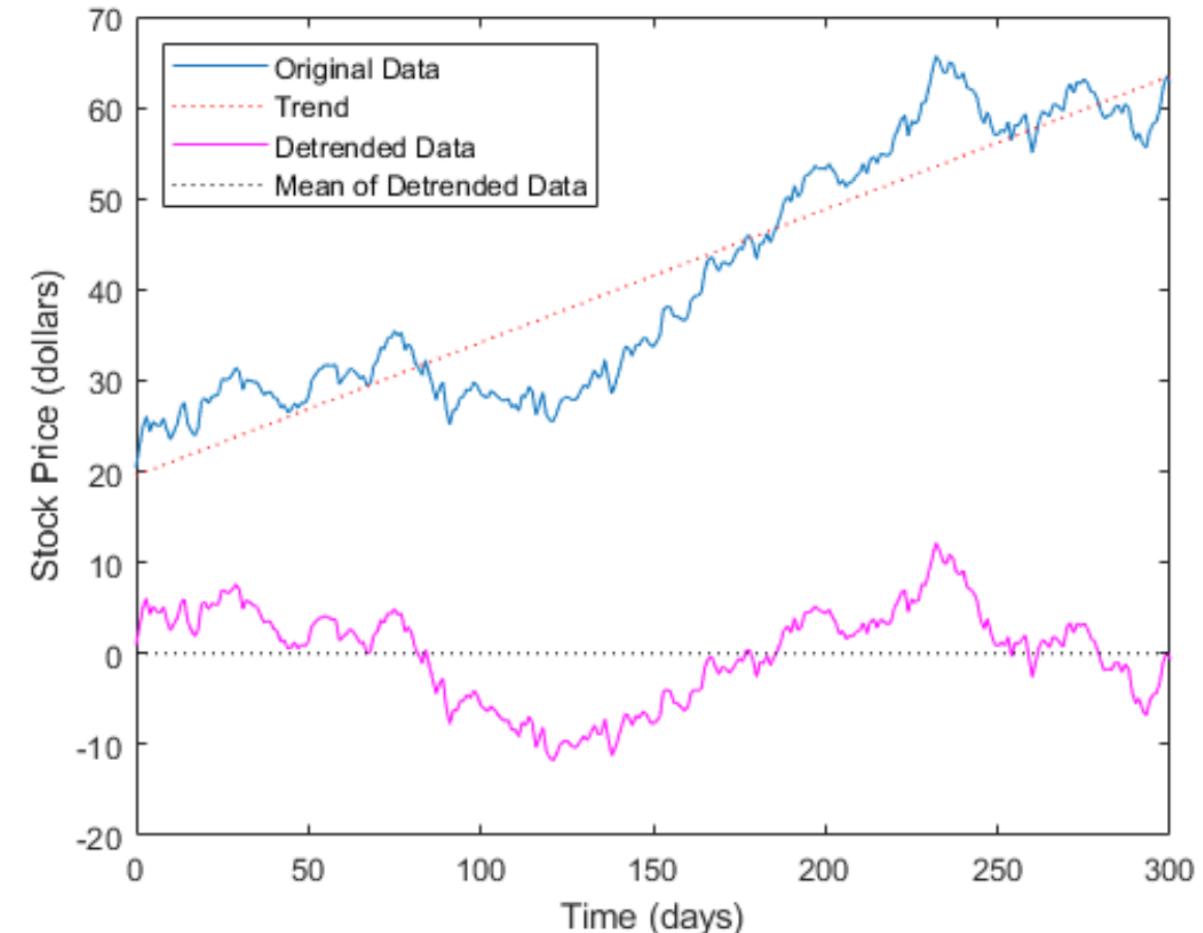
In formulas,

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

(param estimation)

Detrending needs parameters estimation.

Subtract
the
trend
to
get
residuals



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

No need of param estimation.

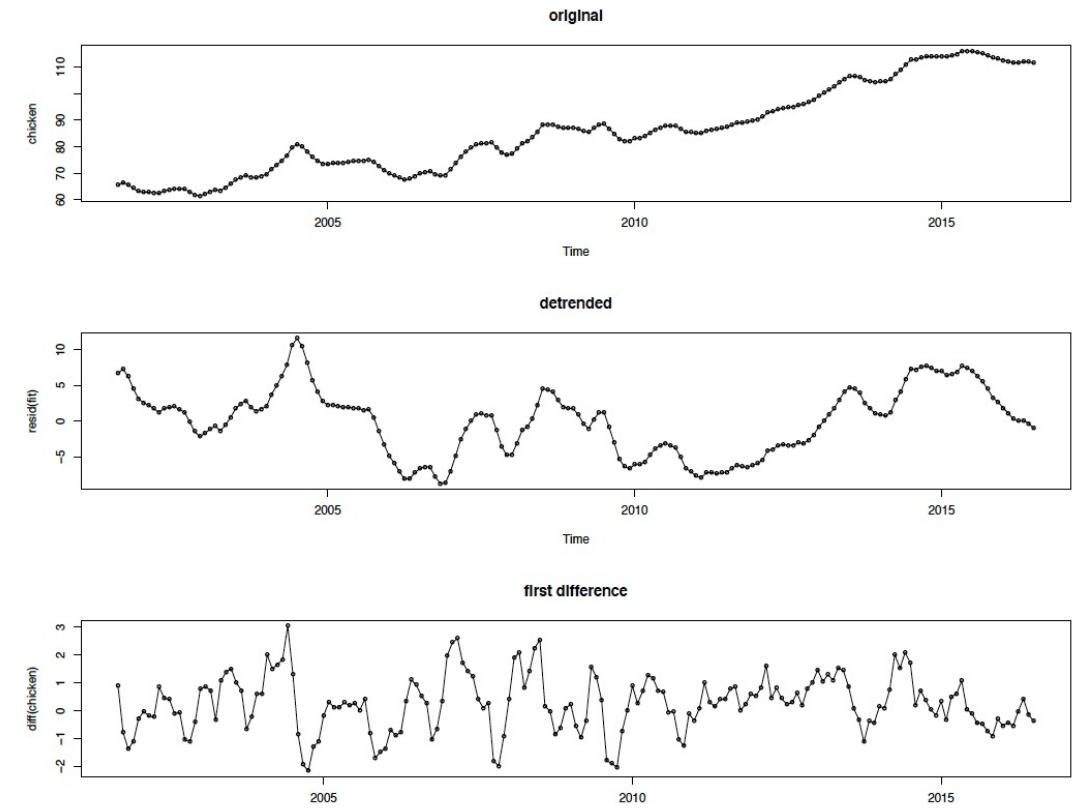
FAU

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.

stabilize mean; remove trends & seasonality.



ARIMA models

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

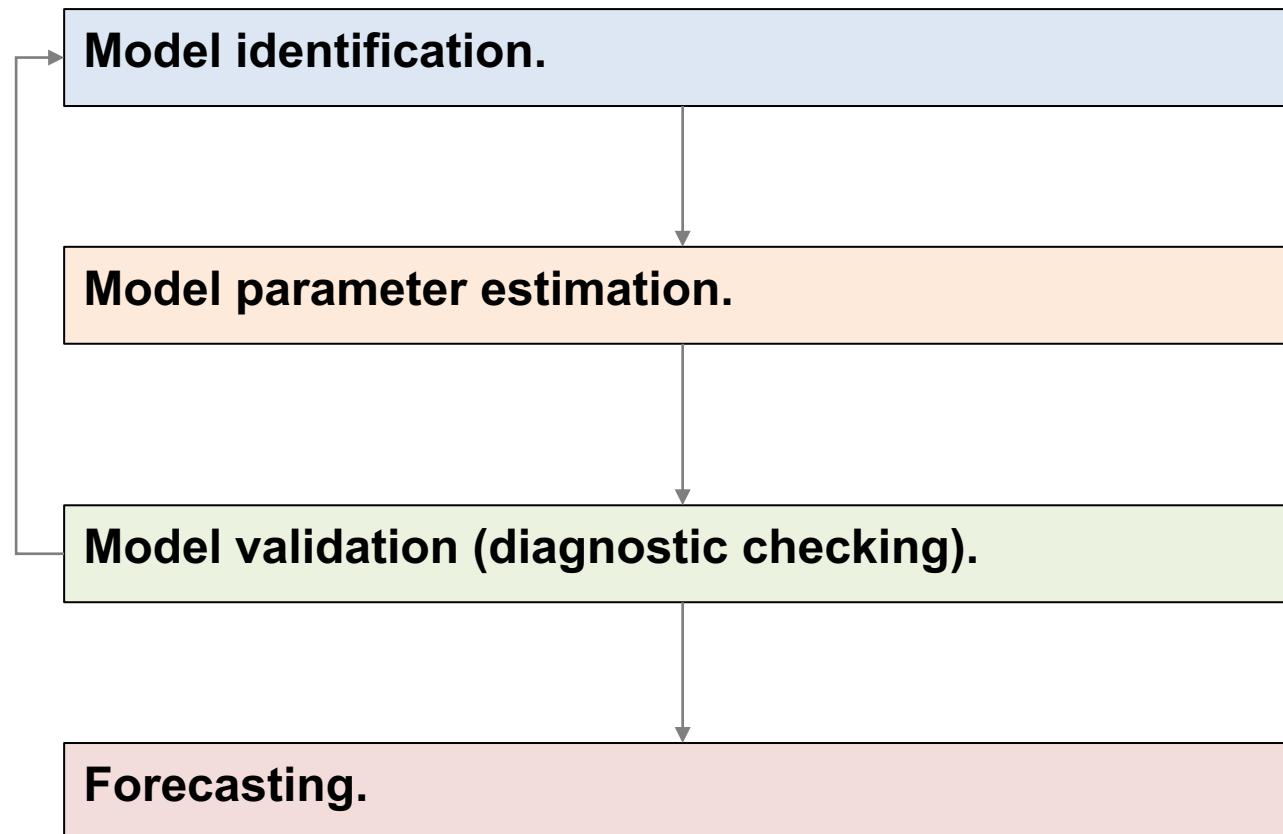
Then, the ARIMA model can be written as

(C-5)

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

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

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.



Lecture title

Recap



In this lecture...

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

ARIMA: Pros and Cons

- Pros:

- Effective in short-term series forecasting.
 - E.g., short-run inflation forecasts.
- It is a parametric model and it works better with relatively small number of observations.

parametric model.
(small observations)

- Cons:

- Techniques for identifying the correct model are difficult to understand and usually computationally expensive.
- ARIMA models performance are poor at predicting series with turning points.

Techniques are difficult. Computationally expensive.

(Any turning point? - WASEP)

