Applied Time Series Analysis
SS 2014 – Week 04

Marcel Dettling

Institute for Data Analysis and Process Design

Zurich University of Applied Sciences

marcel.dettling@zhaw.ch

http://stat.ethz.ch/~dettling

ETH Zürich, March 10, 2014

Applied Time Series Analysis
SS 2014 – Week 04

Basics of Modeling

Simulation & Generation

(Time Series) Model → Data

Estimation, Inference & Residual Analysis

Data → (Time Series) Model

We will first discuss the theoretical properties of the most important time series processes and then mainly focus on how to successfully fit models to data.

A Simple Model: White Noise

A time series $(W_1, W_2, ..., W_n)$ is a **White Noise series** if the random variables $W_1, W_2, ...$ are *independent and identically* distributed with *mean zero*.

This imples that all variables W_t have the same variance σ_W^2, and

$$Cov(W_i, W_j) = 0 \text{ for all } i \neq j.$$

Thus, there are no autocorrelations either: $\rho_k = 0$ for all $k \neq 0$.

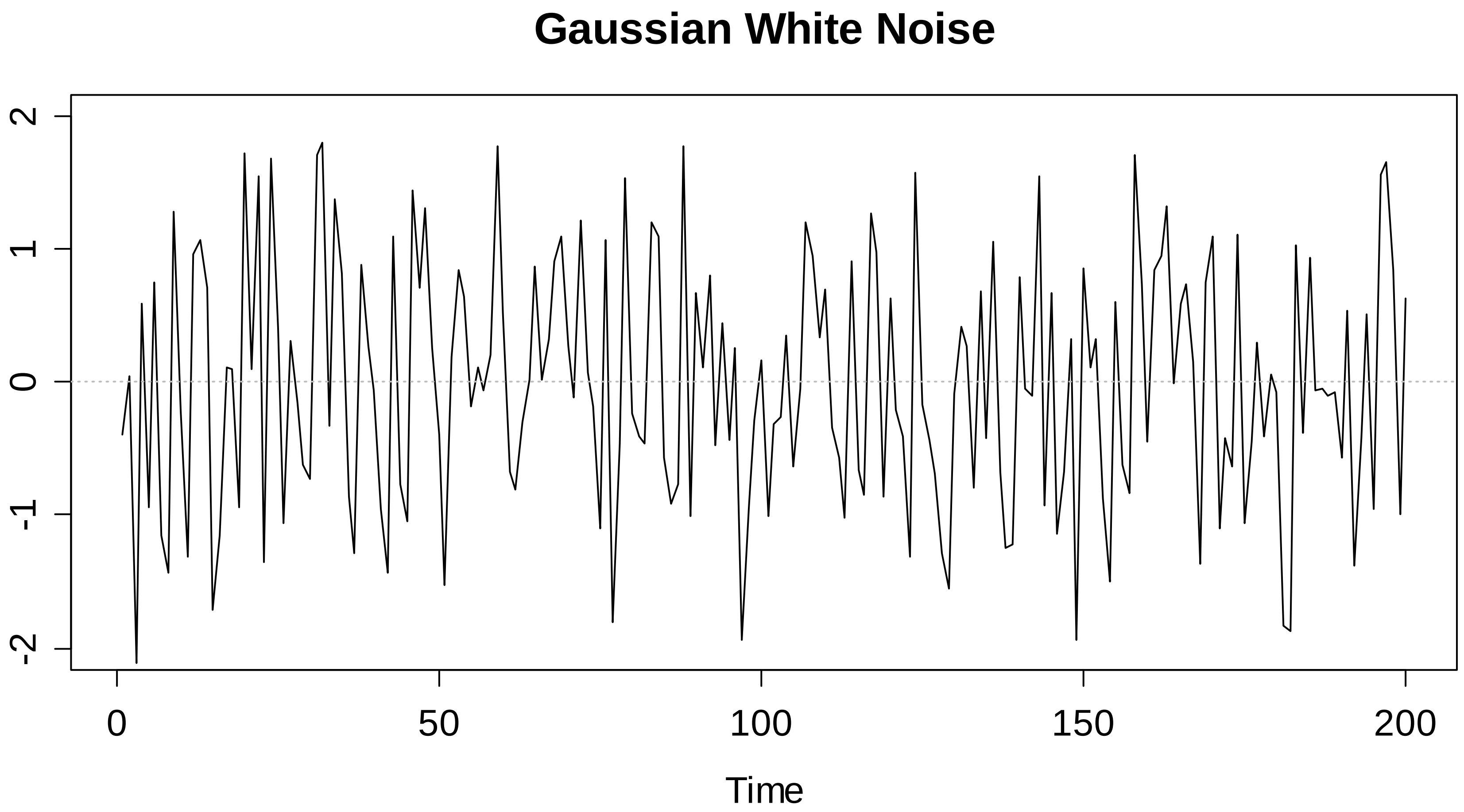
If in addition, the variables also follow a *Gaussian distribution*, i.e. $W_t \sim N(0, \sigma_W^2)$, the series is called **Gaussian White Noise**.

The term White Noise is due to the analogy to white light.

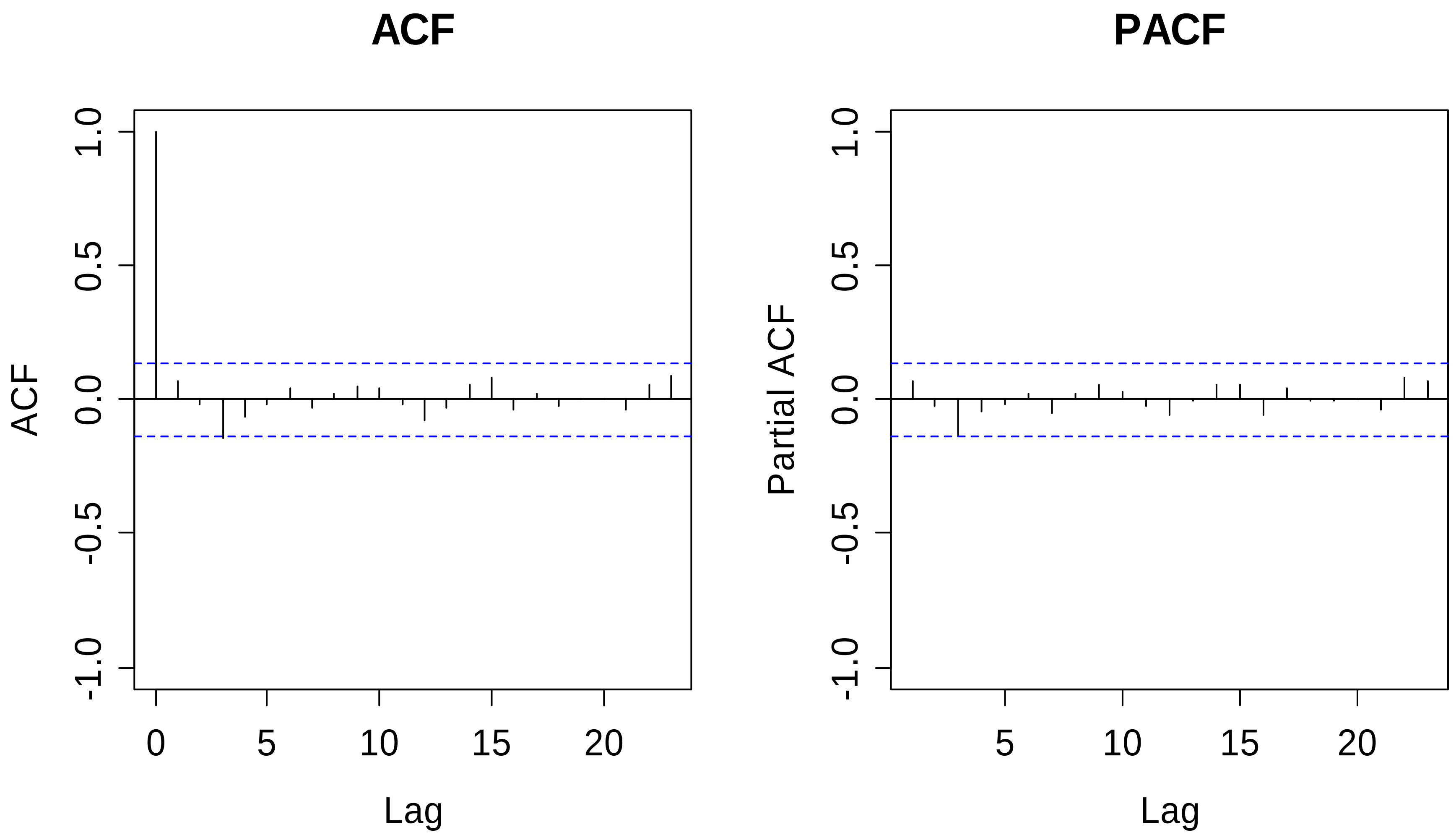
Applied Time Series Analysis
SS 2014 – Week 04

Example: Gaussian White Noise

```
> plot(ts(rnorm(200, mean=0, sd=1)))
```

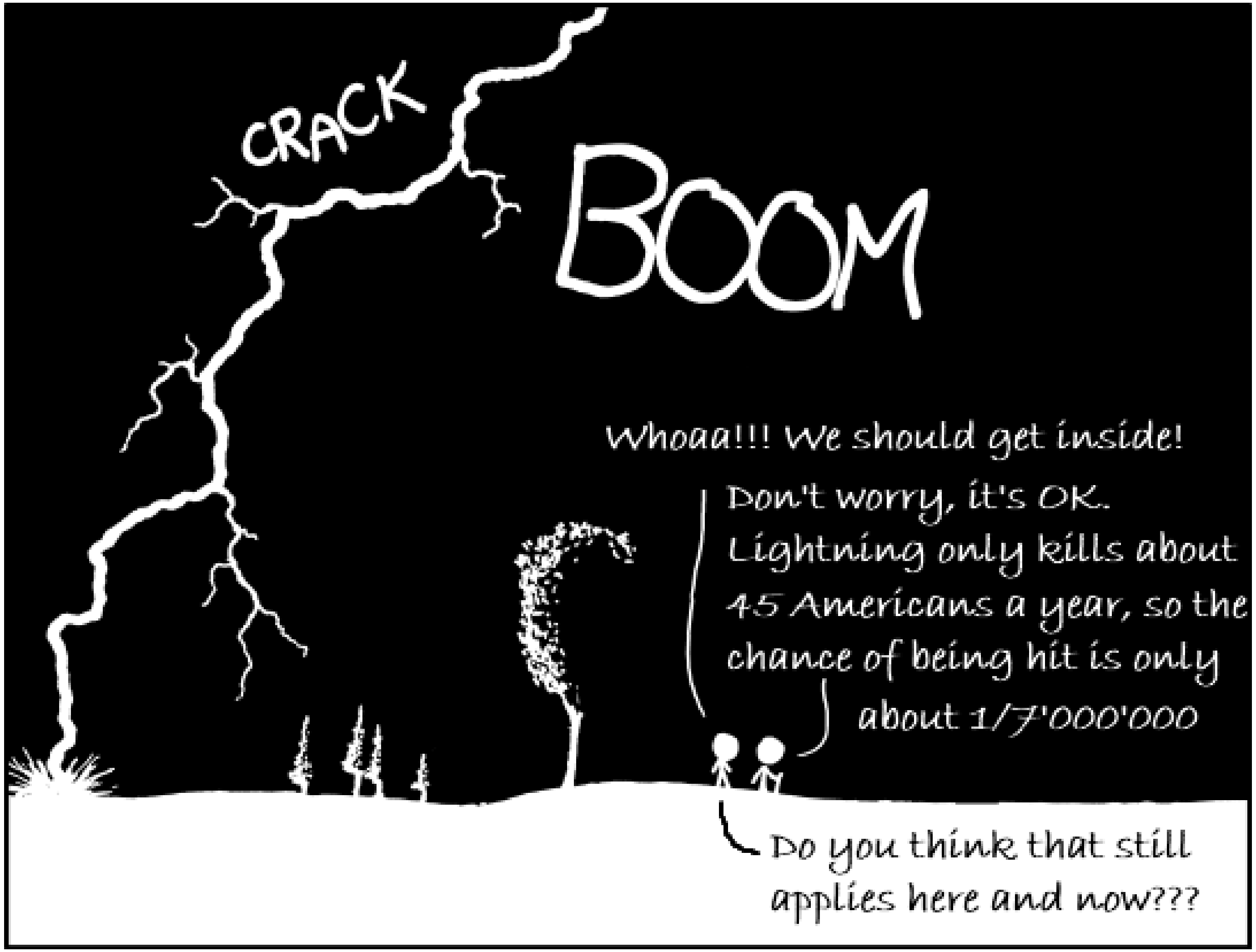


Example: Gaussian White Noise



Applied Time Series Analysis
SS 2014 – Week 04

Estimating the Conditional Mean



→ **see blackboard…**

Time Series Modeling

There is a wealth of time series models

- AR	autoregressive model
- MA	moving average model
- ARMA	combination of AR & MA
- ARIMA	non-stationary ARMAs
- SARIMA	seasonal ARIMAs
- …	

We start by discussing autoregressive models. They are perhaps the simplest and most intuitive time series models that exist.

Basic Idea for AR(p)-Models

We have a process where the random variable X_t depends on an auto-regressive linear combination of the preceding $X_{t-1},\ldots,X_{t-p}$, plus a „completely independent" term called innovation E_t.

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + E_t$$

Here, *p* is called the order of the autoregressive model. Hence, we abbreviate by AR(p). An alternative notation is with the backshift operator B:

$$(1-\alpha_1 B - \alpha_2 B^2 - \ldots - \alpha_p B^p) X_t = E_t \text{ or short, } \Phi(B) X_t = E_t$$

Here, $\Phi(B)$ is called the characteristic polynomial of the AR(p). It determines most of the relevant properties of the process.

AR(1)-Model

The simplest model is the AR(1)-model

$$X_t = \alpha_1 X_{t-1} + E_t$$

where

E_t is i.i.d with $E[E_t] = 0$ and $Var(E_t) = \sigma_E^2$
We also require that E_t is independent of $X_s, s < t$

Under these conditions, E_t is a **causal White Noise** process, or an **innovation**. Be aware that this is stronger than the iid requirement: not every iid process is an innovation and that property is central to AR(p)-modelling.

AR(p)-Models and Stationarity

The following is absolutely essential:

AR(p) models must only be fitted to stationary time series. Any potential trends and/or seasonal effects need to be removed first. We will also make sure that the processes are stationary.

Under which circumstances is an AR(p) stationary?

→ **see blackboard…**

Stationarity of AR(p)-Processes

As we have seen, any stationary AR(p) meets:

1) $E[X_t] = \mu = 0$

2) The condition on $(\alpha_1, ..., \alpha_p)$:

All (complex) roots of the characteristic polynom

$$1 - \alpha_1 z - \alpha_2 z^2 - \alpha_p z^p = 0$$

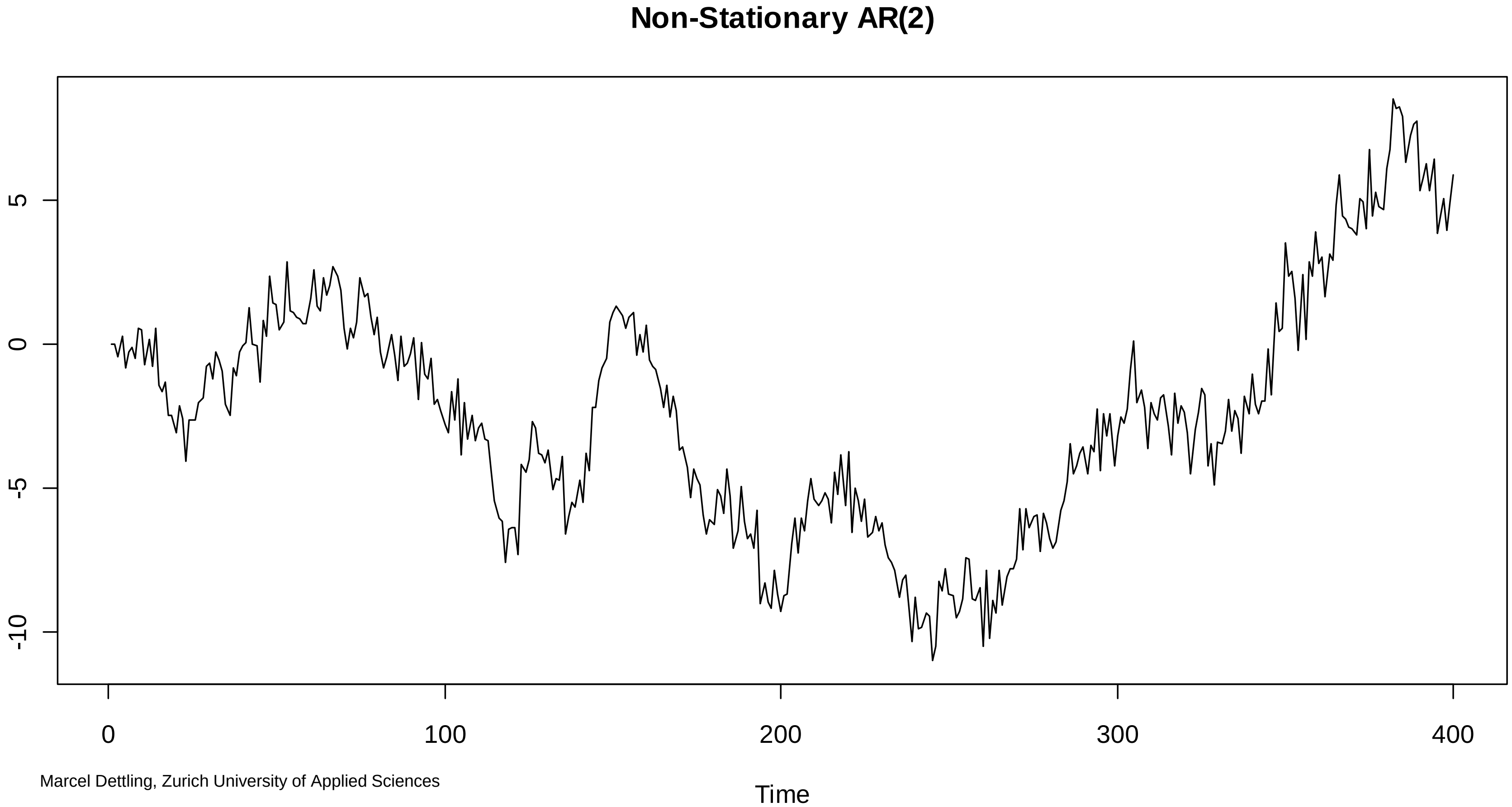
lie outside of the unit circle (can be verified with `polyroot()`)

We can always shift a stationary AR(p) process: $Y_t = m + X_t$
The resulting process is still stationary and allows for greater flexibility in modelling. It is a **shifted AR(p) process**.

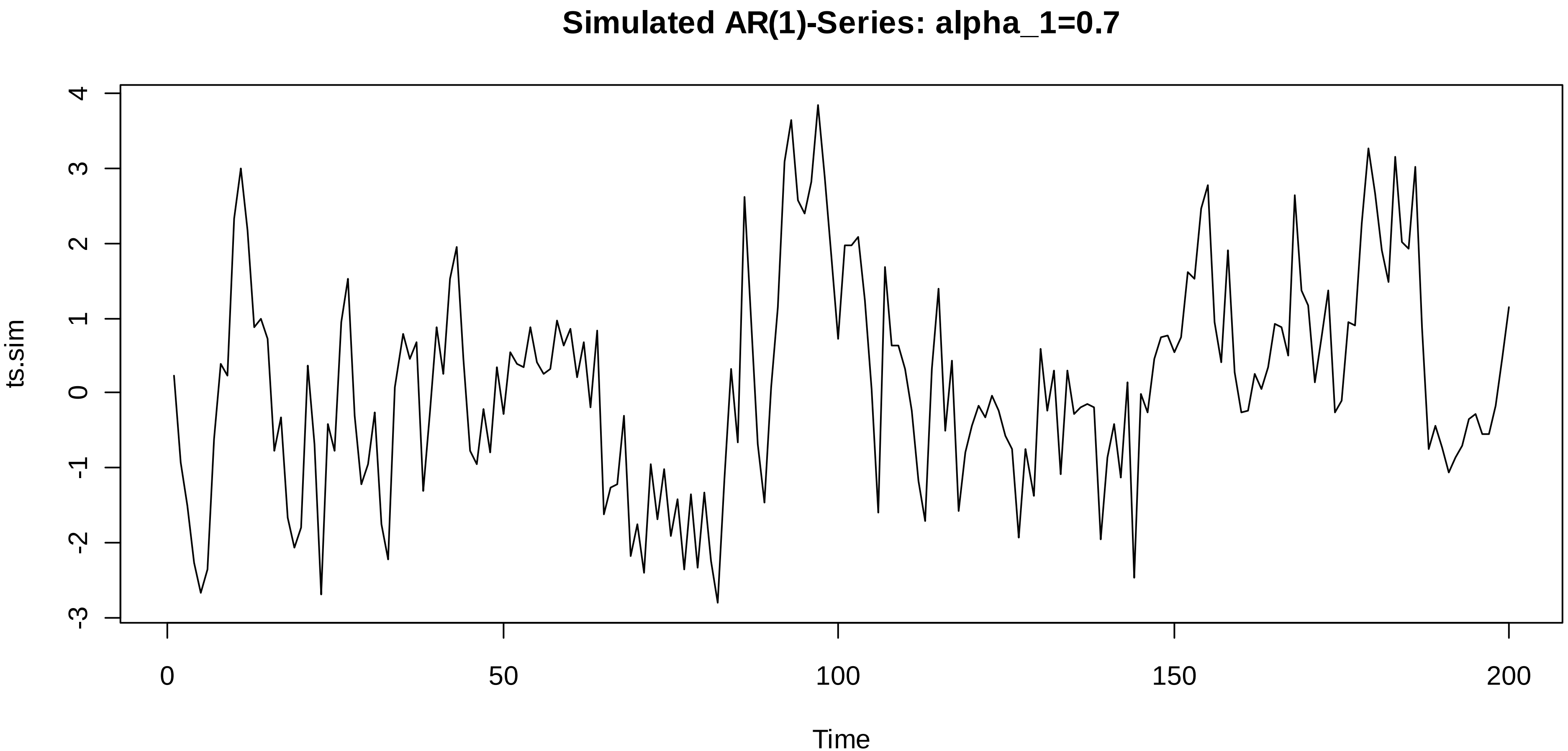
Applied Time Series Analysis
SS 2014 – Week 04

A Non-Stationary AR(2)-Process

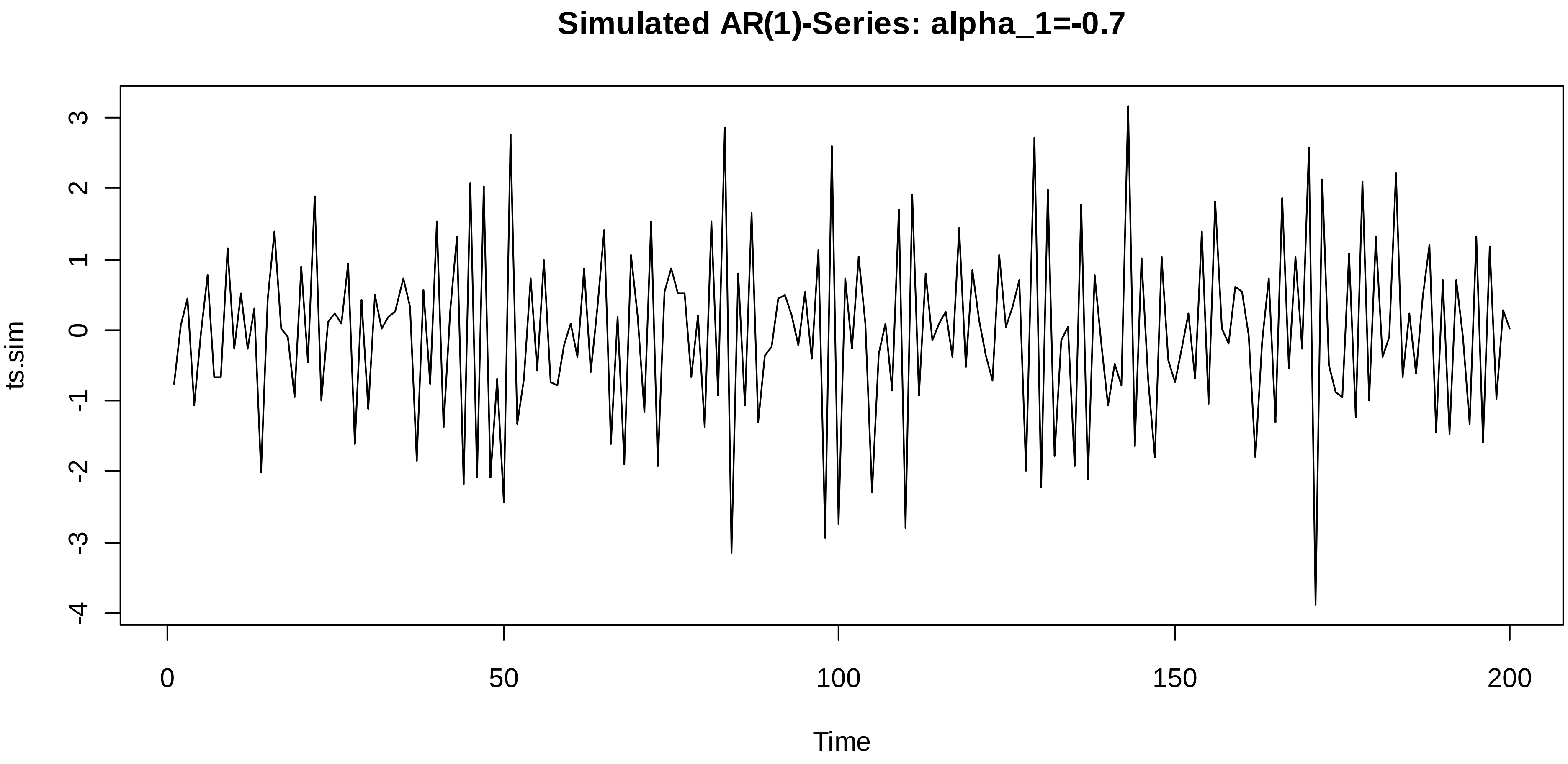
$X_t = \frac{1}{2} X_{t-1} + \frac{1}{2} X_{t-2} + E_t$ is not stationary…



Simulated AR(1)-Series

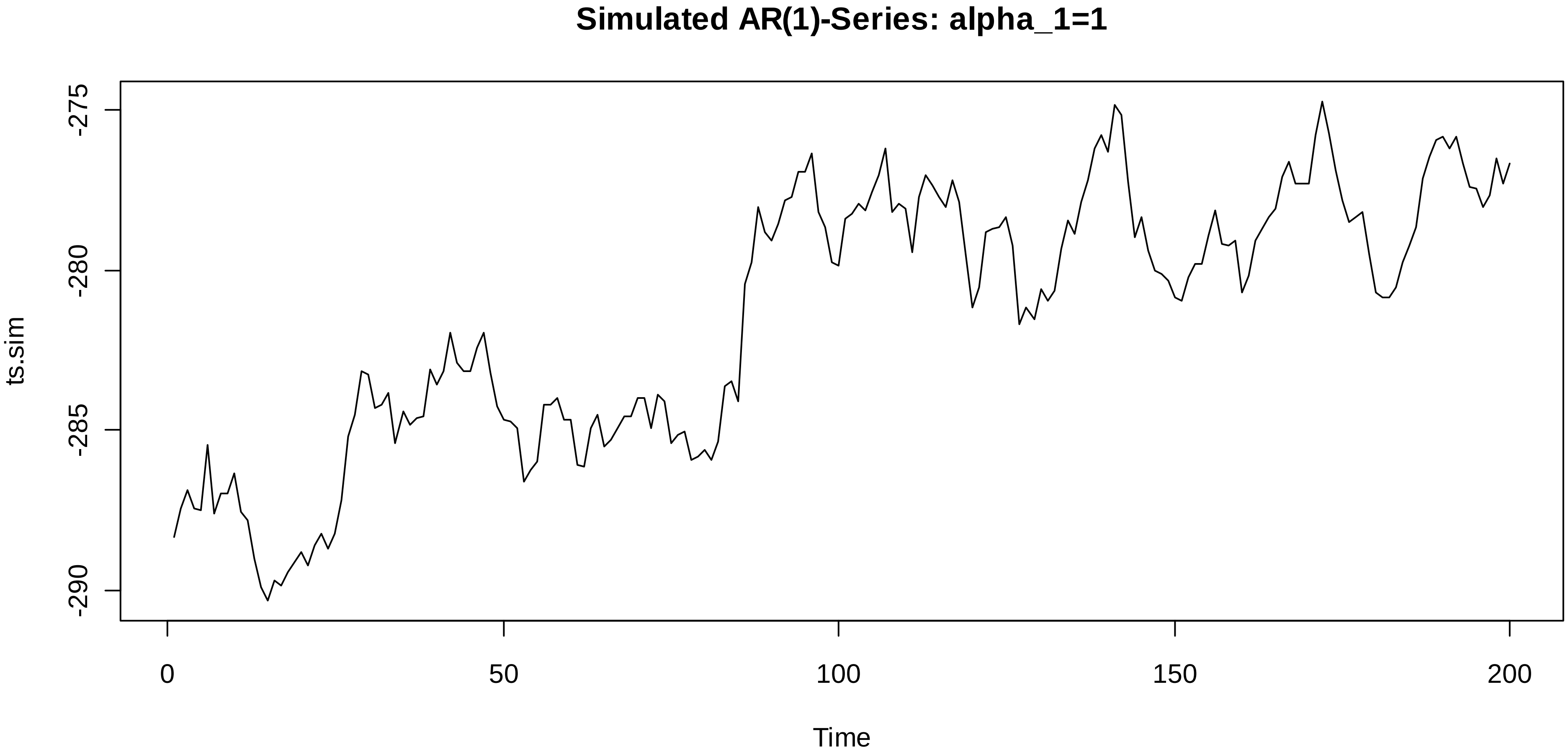


Simulated AR(1)-Series



Applied Time Series Analysis
SS 2014 – Week 04

Simulated AR(1)-Series



Autocorrelation of AR(p) Processes

On the blackboard…

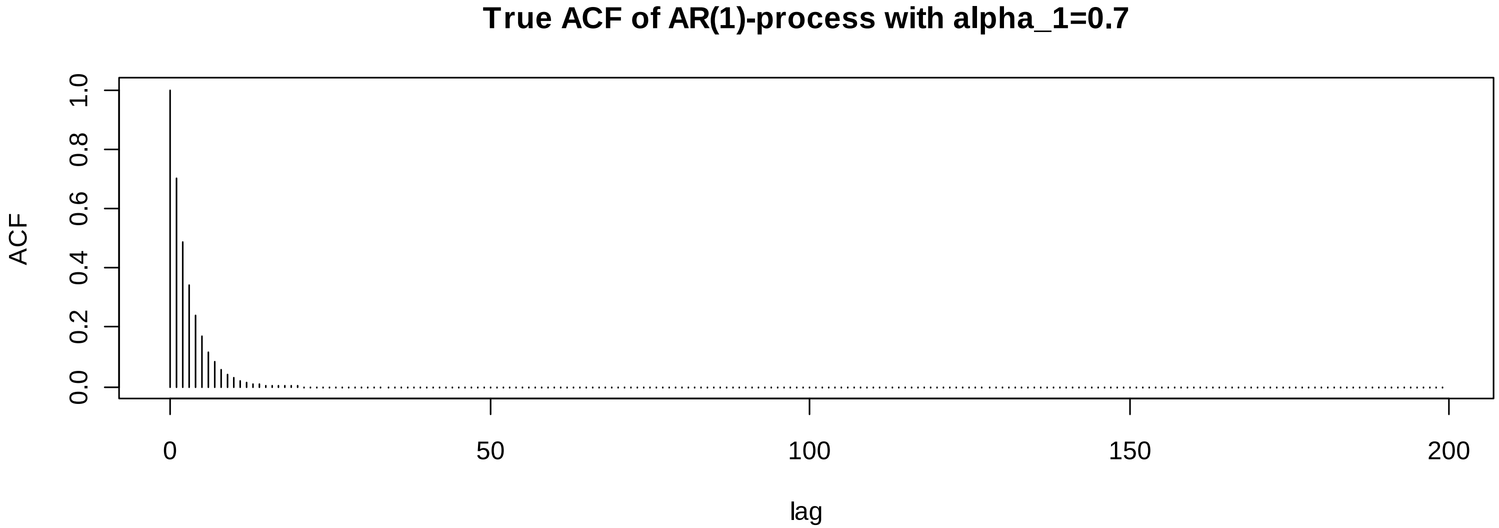
Yule-Walker Equations

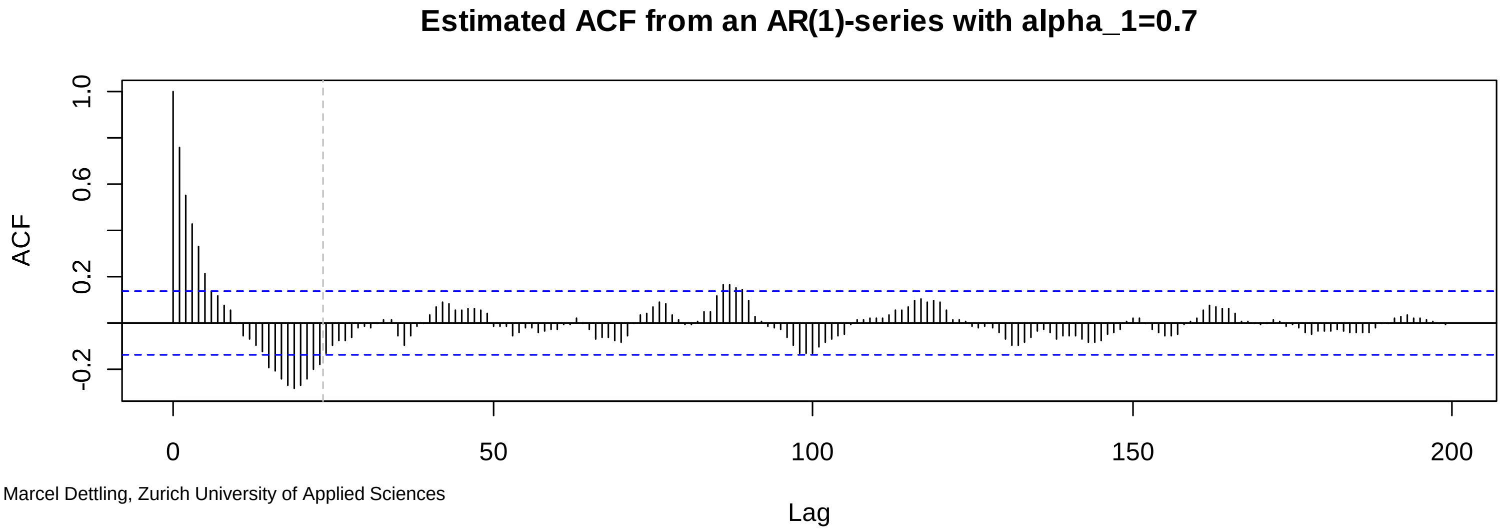
We observe that there exists a linear equation system built up from the AR(p)-coefficients and the ACF-coefficients of up to lag p. These are called Yule-Walker-Equations.

We can use these equations for fitting an AR(p)-model:

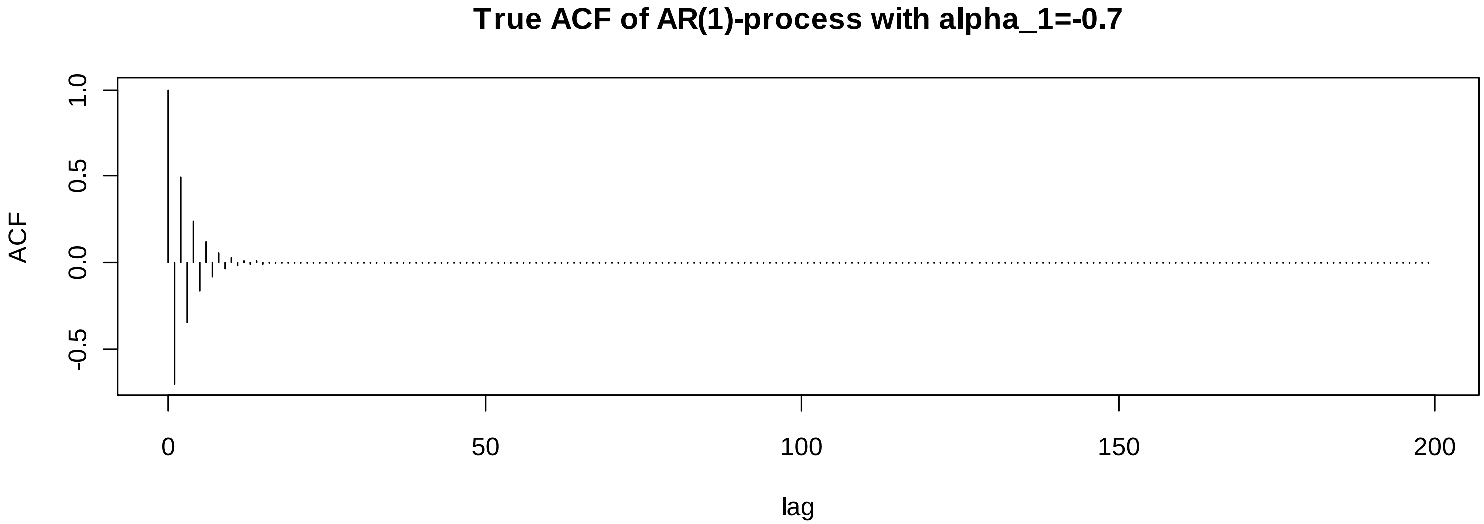
1) *Estimate the ACF from a time series*
2) *Plug-in the estimates into the Yule-Walker-Equations*
3) *The solution are the AR(p)-coefficients*

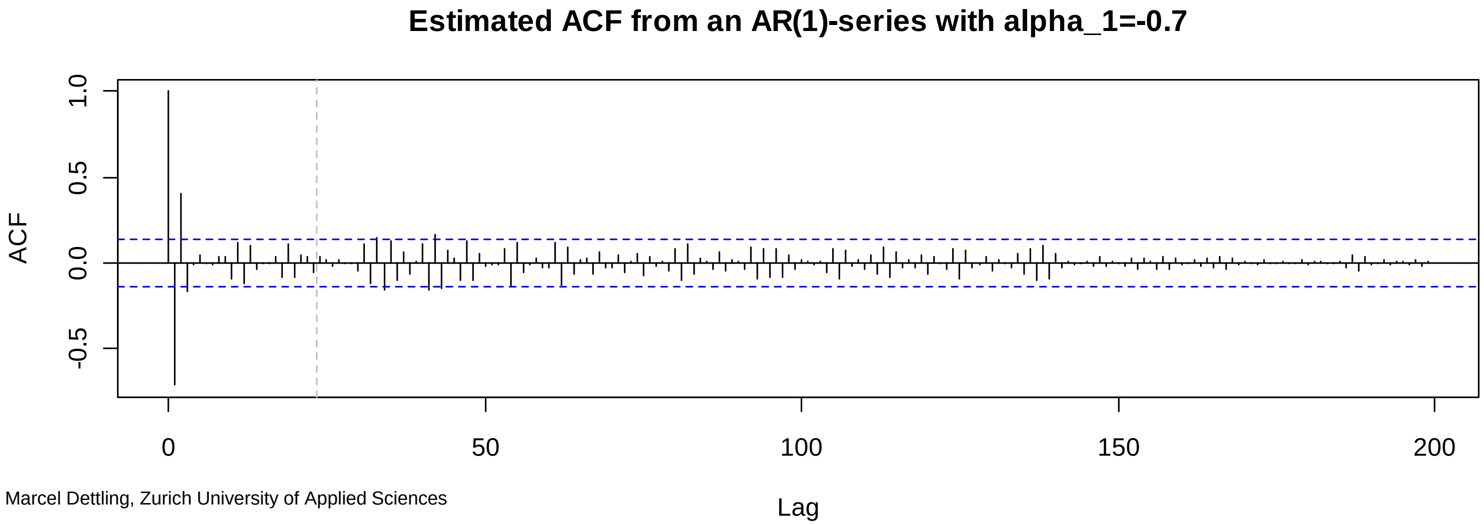
Theoretical vs. Estimated ACF





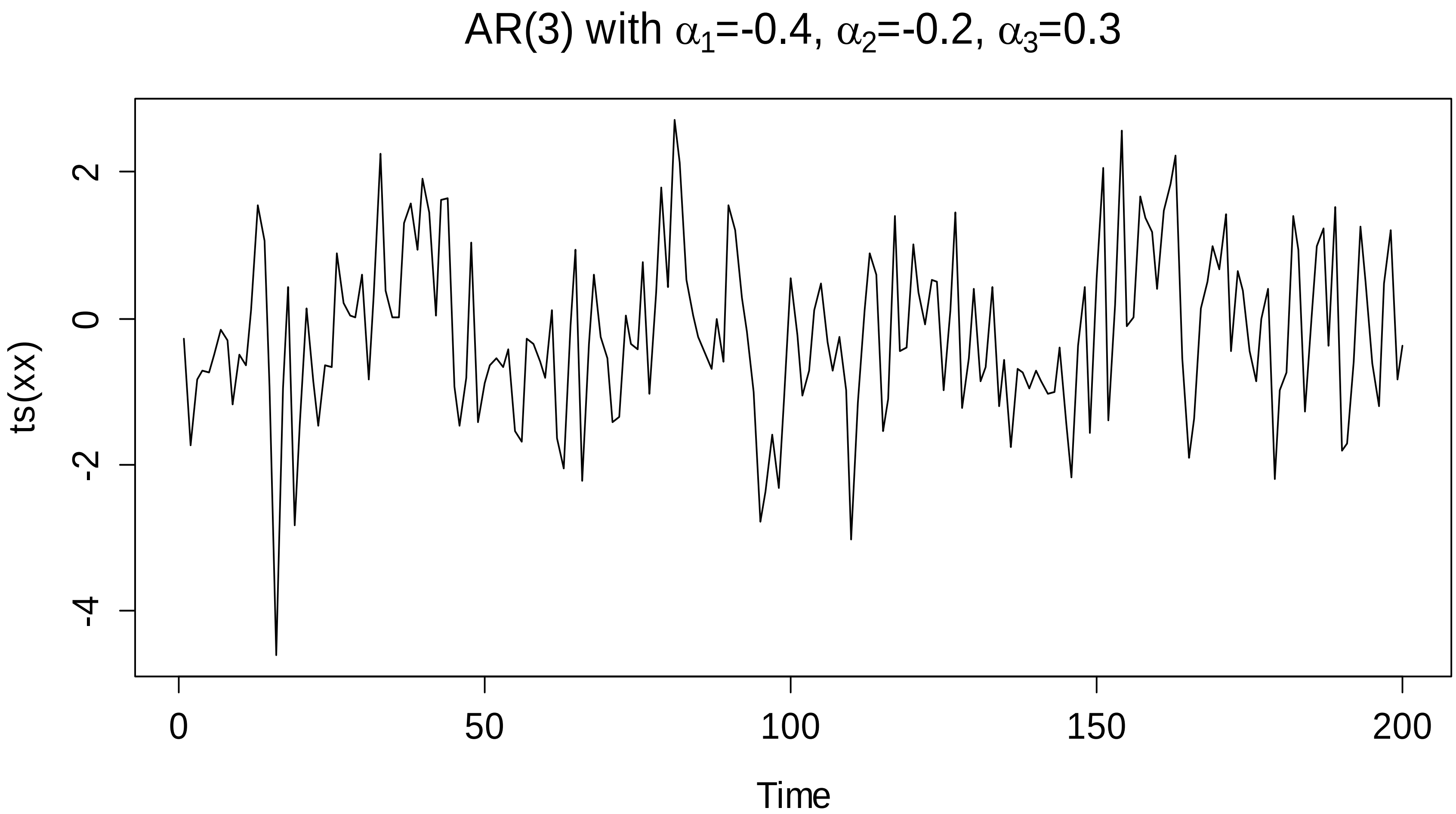
Theoretical vs. Estimated ACF





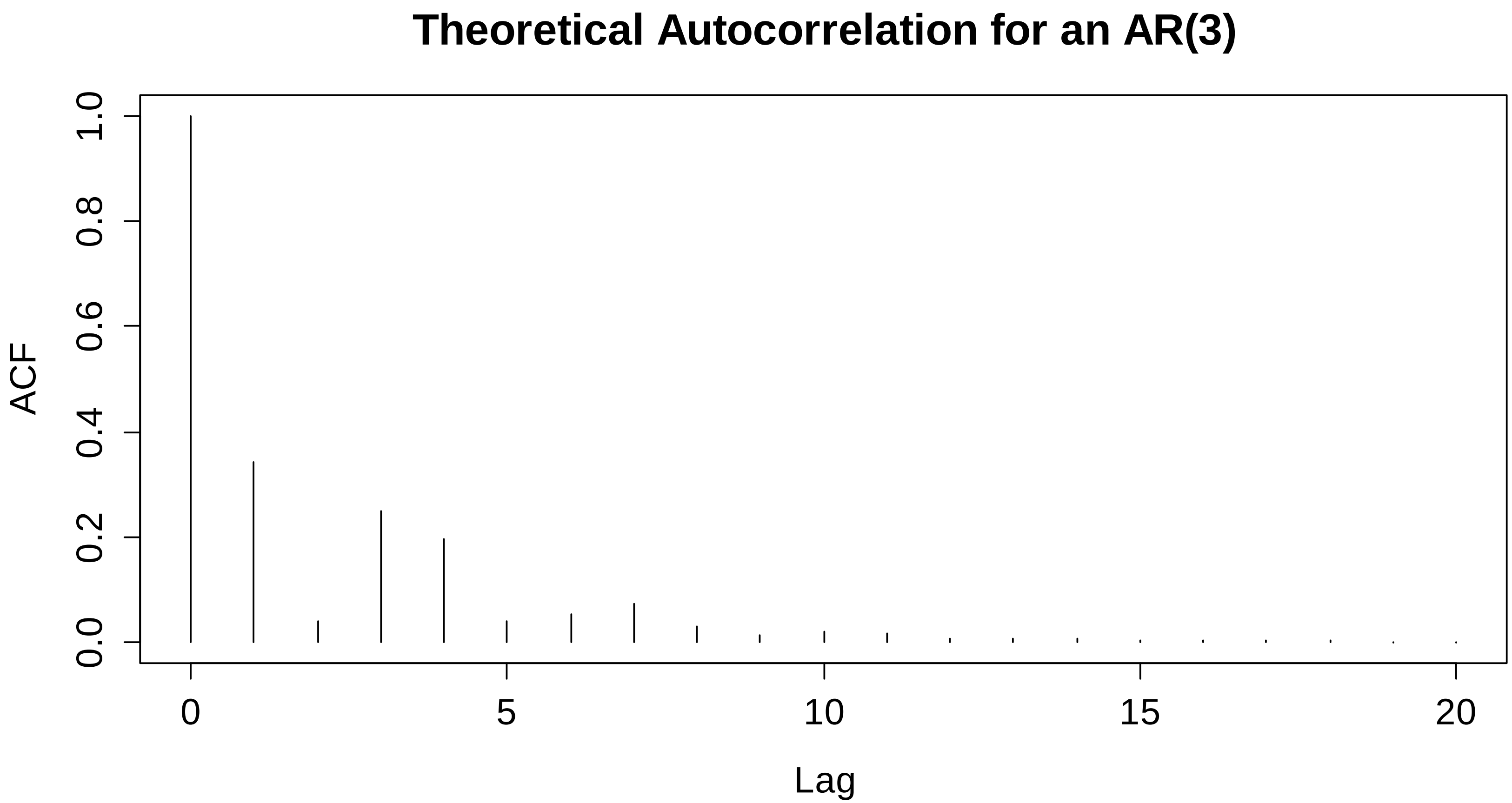
AR(3): Simulation and Properties

```
> xx <- arima.sim(list(ar=c(0.4, -0.2, 0.3)),
```



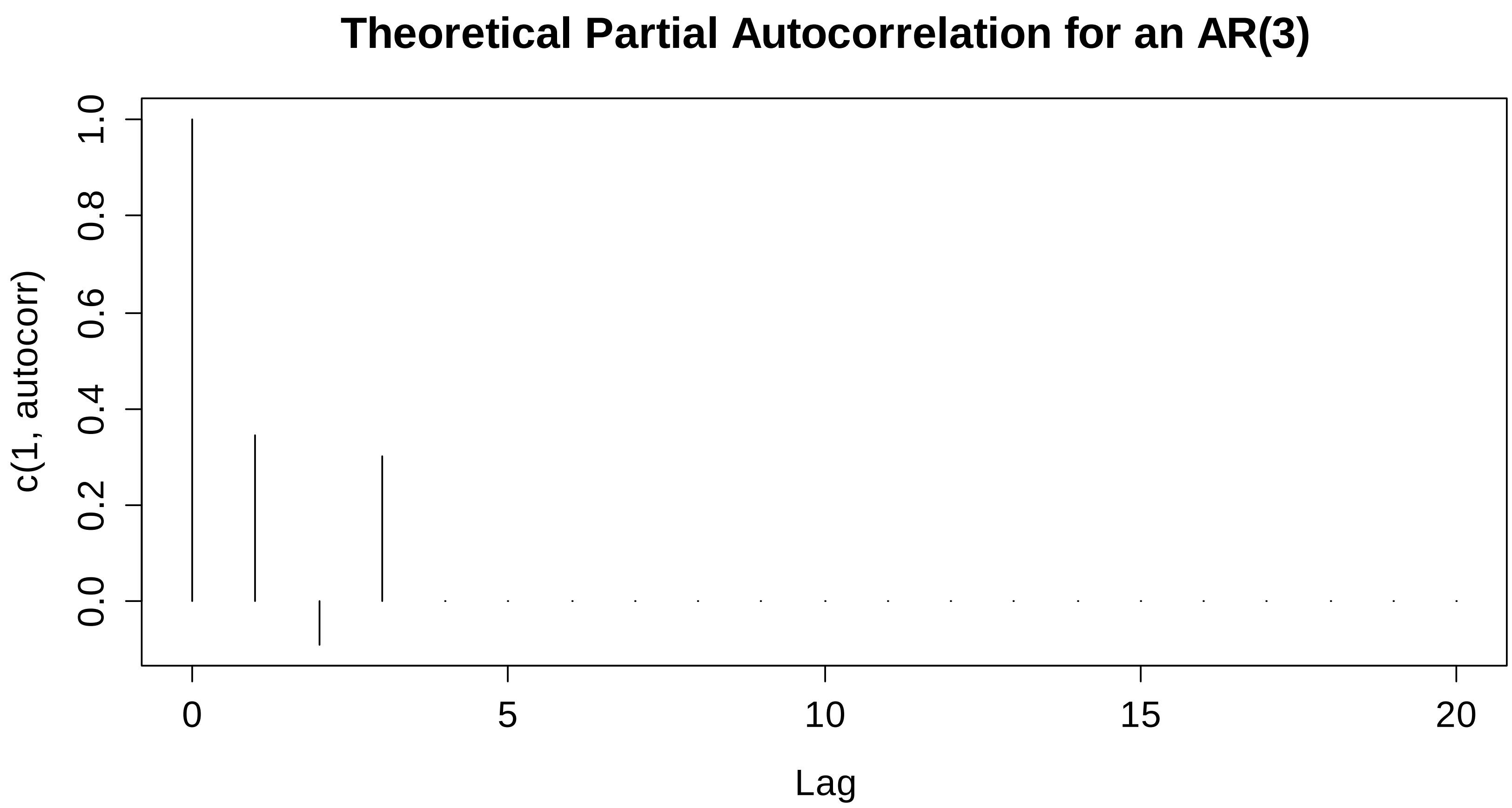
AR(3): Simulation and Properties

```
> autocorr <- ARMAacf(ar=c(0.4, -0.2, 0.3),...)
> plot(0:20, autocorr, type="h", xlab="Lag")
```



AR(3): Simulation and Properties

```
> autocorr <- ARMAacf(ar=..., pacf=TRUE, ...)
> plot(0:20, autocorr, type="h", xlab="Lag")
```



Fitting AR(p)-Models

This involves 3 crucial steps:

1) **Is an AR(p) suitable, and what is p?**

- will be based on ACF/PACF-Analysis

2) **Estimation of the AR(p)-coefficients**

- Regression approach
- Yule-Walker-Equations
- and more (MLE, Burg-Algorithm)

3) **Residual Analysis**

- to be discussed

AR-Modelling

1

Identification of the Order p

- ACF/PACF
- AIC/BIC

2

Parameter Estimation

- Regression
- Yule-Walker
- MLE
- Burg

3

Model Diagnostics

- Residual Analysis
- Simulation

Is an AR(p) suitable, and what is p?

- For all AR(p)-models, the **ACF** decays exponentially quickly, or is an exponentially damped sinusoid.

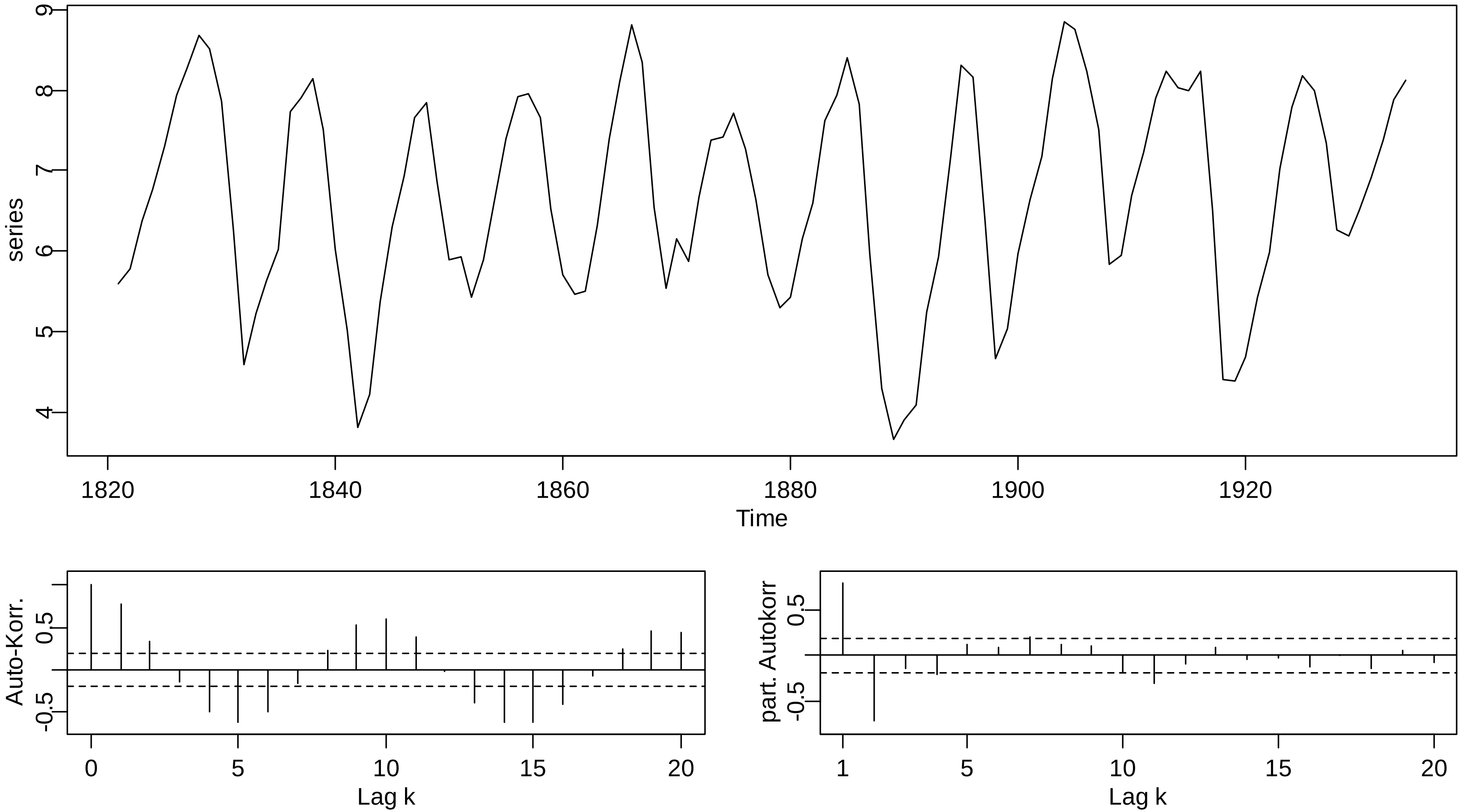
- For all AR(p)-models, the **PACF** is equal to zero for all lags k>p. The behavior before lag p can be anything.

If what we observe is fundamentally different from the above, it is unlikely that the series was generated from an AR(p)-process. We thus need other models, maybe more sophisticated ones.

Remember that the sample ACF has a few peculiarities (bias, variability, compensation issue) and is tricky to interpret!!!

Applied Time Series Analysis
SS 2014 – Week 04

Model Order for log(lynx)



Parameter Estimation for AR(p)

Observed time series are rarely centered. Then, it is inappropriate to fit a pure AR(p) process. All R routines by default assume the shifted process $Y_t = m + X_t$. Thus, we face the problem:

$$(Y_t - m) = \alpha_1 (Y_{t-1} - m) + ... + \alpha_p (Y_{t-p} - m) + E_t$$

The goal is to estimate the *global mean* m, the *AR-coefficients* $\alpha_1, ..., \alpha_p$, and some parameters defining the distribution of the innovation E_t. We usually assume a Gaussian, hence this is σ_E^2.

We will discuss 4 methods for estimating the parameters:

OLS, Burg's algorithm, Yule-Walker, MLE

Applied Time Series Analysis
SS 2014 – Week 04

OLS Estimation

If we rethink the previously stated problem:

$$(Y_t - m) = \alpha_1 (Y_{t-1} - m) + ... + \alpha_p (Y_{t-p} - m) + E_t$$

we recognize a multiple linear regression problem without intercept on the centered observations. What we need to do is:

1) Estimate $\hat{m} = \bar{y} = \sum_{t=1}^{n} y_t$ and determine $x_t = y_t - \hat{m}$

2) Run a regression w/o intercept on x_t to obtain $\hat{\alpha}_1, ..., \hat{\alpha}_p$

3) For $\hat{\sigma}_E^2$, take the residual standard error from the output.

This all works without any time series software, but is a bit cumbersome to implement. Dedicated procedures exist...

OLS Estimation

```
> f.ols <- ar.ols(llynx, aic=F, inter=F, order=2)
> f.ols
Coefficients:
      1        2
 1.3844  -0.7479

Order selected 2  sigma^2 estimated as  0.2738

> f.ols$x.mean
[1] 6.685933

> sum(na.omit(f.ols$resid)^2)/112
[1] 0.2737594
```

Applied Time Series Analysis
SS 2014 – Week 04

Burg‘s Algorithm

While OLS works, the first p instances are never evaluated as responses. This is cured by Burg’s algorithm, which uses the property of time-reversal in stochastic processes. We thus evaluate the RSS of forward and backward prediction errors:

$$\sum_{t=p+1}^{n} \left\{ \left(X_t - \sum_{k=1}^{p} \alpha_k X_{t-k} \right)^2 + \left(X_{t-p} - \sum_{k=1}^{p} \alpha_k X_{t-p+k} \right)^2 \right\}$$

In contrast to OLS, there is no explicit solution and numerical optimization is required. This is done with a recursive method called the Durbin-Levison algorithm (implemented in R).

Burg's Algorithm

```
> f.burg <- ar.burg(llynx, aic=F, order.max=2)
> f.burg

Coefficients:
      1        2
 1.3831  -0.7461

Order selected 2  sigma^2 estimated as  0.2707

> f.ar.burg$x.mean
[1] 6.685933
```

Note: The innovation variance is estimated from the Durbin-Levinson updates and not from the residuals using the MLE!

Yule-Walker Equations

The Yule-Walker-Equations yield a LES that connects the true ACF with the true AR-model parameters. We plug-in the estimated ACF coefficients

$$\hat{\rho}(k) = \hat{\alpha}_1 \hat{\rho}(k-1) + ... + \hat{\alpha}_p \hat{\rho}(k-p) \text{ for k=1,...,p}$$

and can solve the LES to obtain the AR-parameter estimates.

$\hat{m}$ is the arithmetic mean of the time series

$\hat{\sigma}_E^2$ is obtained from the fitted coefficients via the autocovariance of the series and takes a different value than before!

There is an implementation in R with function `ar.yw()`.

Yule-Walker Equations

```
> f.ar.yw

Call: ar.yw.default(x = log(lynx), aic = FALSE,
order.max = 2)

Coefficients:
      1        2
 1.3504  -0.7200

Order selected 2  sigma^2 estimated as  0.3109
```

While the Yule-Walker method is asymptotically equivalent to OLS and Burg's algorithm, it generally yields a solution with worse Gaussian likelihood on finite samples

Applied Time Series Analysis
SS 2014 – Week 04

Maximum-Likelihood-Estimation

Idea: Determine the parameters such that, given the observed time series $(y_1,...,y_n)$, the resulting model is the most plausible (i.e. the most likely) one.

This requires the choice of a probability model for the time series. By assuming Gaussian innovations, $E_t \sim N(0,\sigma_E^2)$, any AR(p) process has a multivariate normal distribution:

$$Y = (Y_1,...,Y_n) \sim N(m \cdot \underline{1}, V) \text{, with } V \text{ depending on } \underline{\alpha}, \sigma_E^2$$

MLE then provides simultaneous estimates by optimizing:

$$L(\alpha, m, \sigma_E^2) \propto \exp\left(\sum_{t=1}^{n} (x_t - \hat{x}_t)^2 \right)$$

Maximum-Likelihood Estimation

```
> f.ar.mle

Call: arima(x = log(lynx), order = c(2, 0, 0))

Coefficients:
         ar1      ar2  intercept
      1.3776  -0.7399     6.6863
s.e.  0.0614   0.0612     0.1349

sigma^2=0.2708; log likelihood=-88.58; aic=185.15
```

While MLE by default assumes Gaussian innovations, it still performs resonably for other distributions as long as they are not extremly skewed or have very precarious outliers.

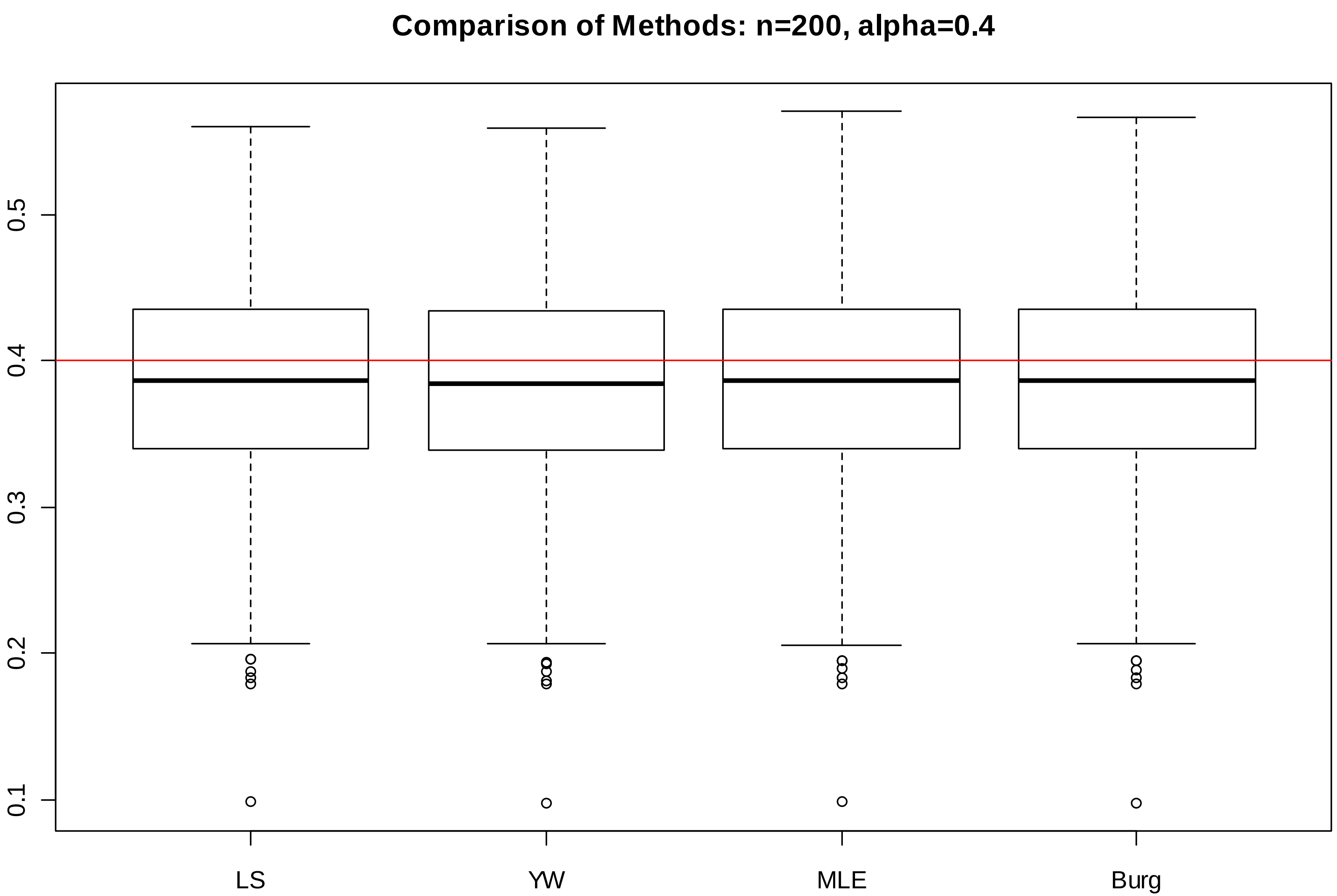
Practical Aspects

- All 4 estimation methods are asymptotically equivalent.
- Even on finite samples, the differences are usually small.
- Under Gaussian distribution, OLS and MLE coincide.
- OLS/YW: explicit solution; Burg/MLE: numerical solution.
- Functions `ar.xx()` provide easy AIC estimation of p.
- Function `arima()` provides standard errors for all parameters.

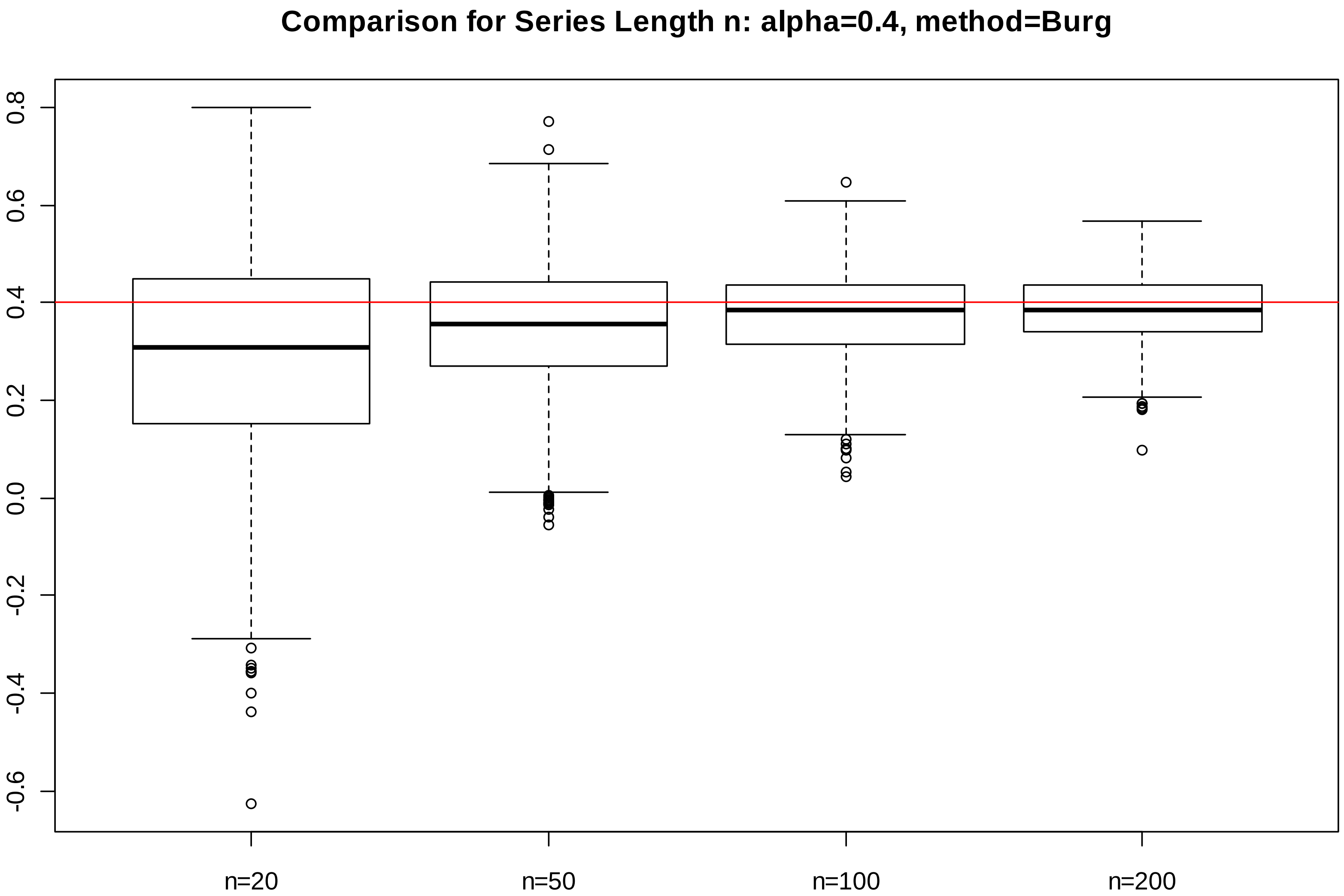
-> Either work with `ar.burg()` or with `arima()`, depending on whether you want AIC or standard errors. Watch out for warnings if the numerical solution do not converge.

Applied Time Series Analysis
SS 2014 – Week 04

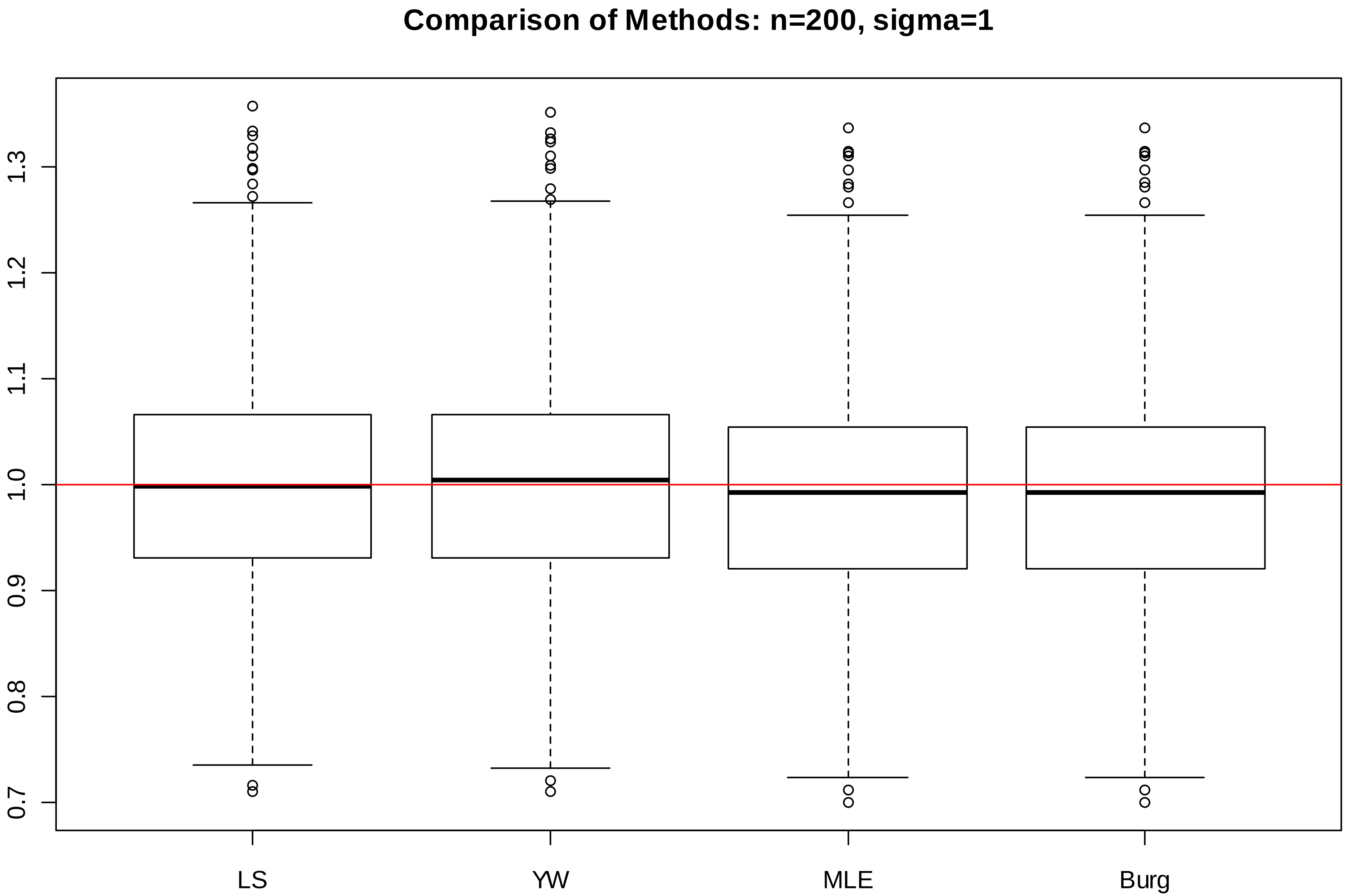
Comparison: Alpha Estimation vs. Method



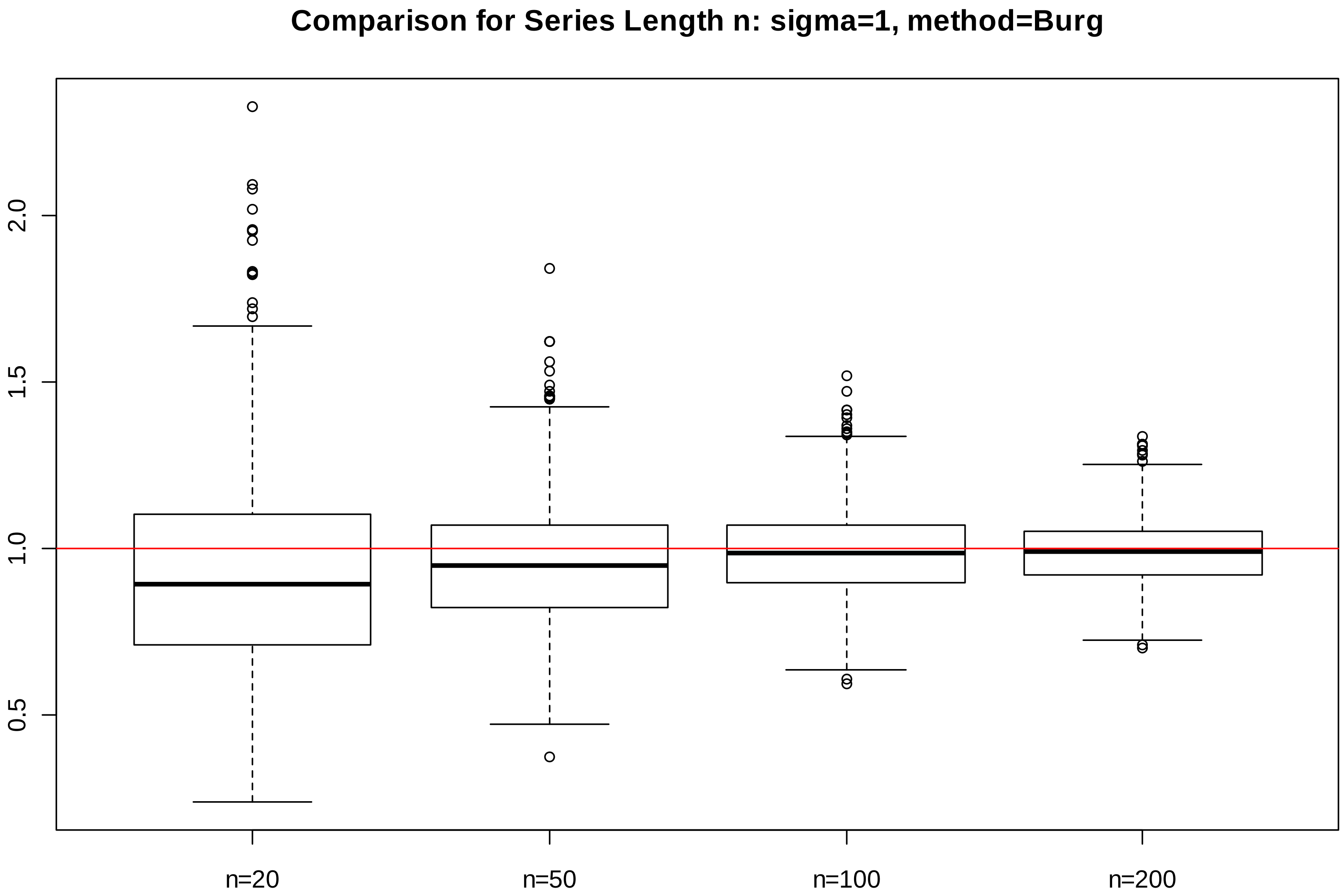
Comparison: Alpha Estimation vs. n



Comparison: Sigma Estimation vs. Method



Comparison: Sigma Estimation vs. n



Applied Time Series Analysis
SS 2014 – Week 04

Model Diagnostics

What we do here is Residual Analysis:

„residuals“ = „estimated innovations“

$$= \hat{E}_t$$

$$= (x_t - \hat{m}) - \left(\hat{\alpha}_1 (x_{t-1} - \hat{m}) - \ldots - \hat{\alpha}_p (x_{t-p} - \hat{m})\right)$$

Remember the assumptions we made:

E_t i.i.d, $E[E_t] = 0$, $Var(E_t) = \sigma_E^2$

and probably

$E_t \sim N(0, \sigma_E^2)$

Model Diagnostics

We check the assumptions we made with the following means:

a) Time series plot of $\hat{E}_t$

b) ACF/PACF plot of $\hat{E}_t$

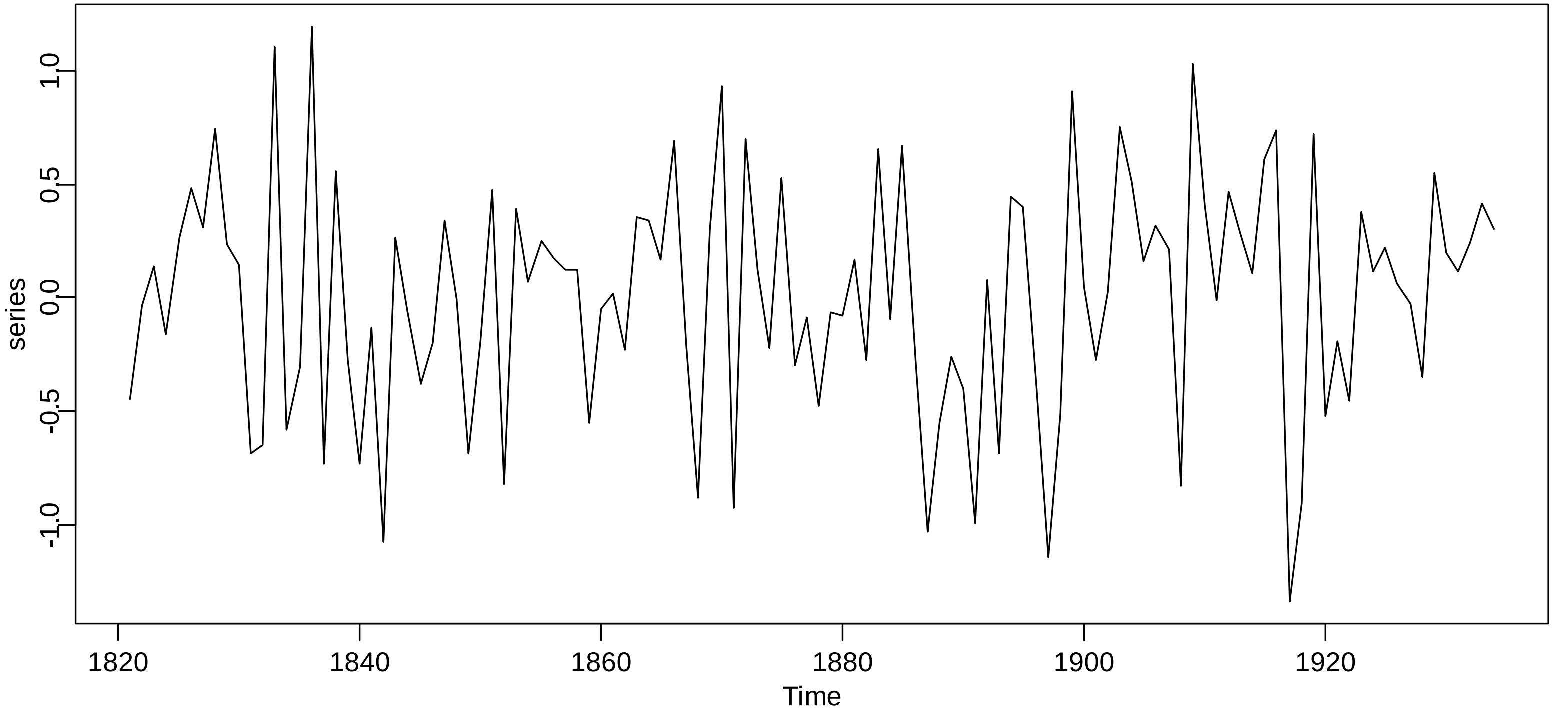
c) QQ-plot of $\hat{E}_t$

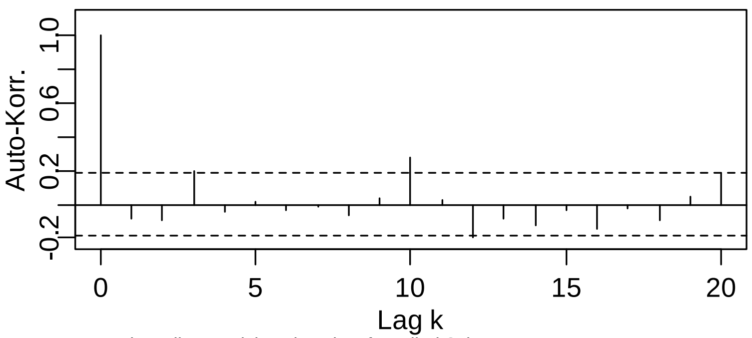
→ **The innovation time series $\hat{E}_t$ should look like white noise**

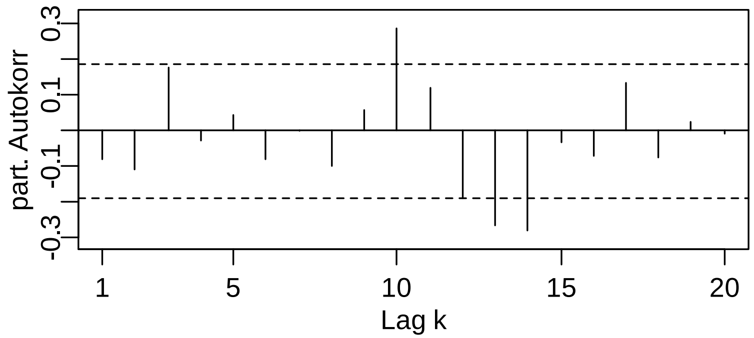
Lynx example:

```
fit <- arima(log(lynx), order=c(2,0,0))
acf(resid(fit)); pacf(resid(fit))
```

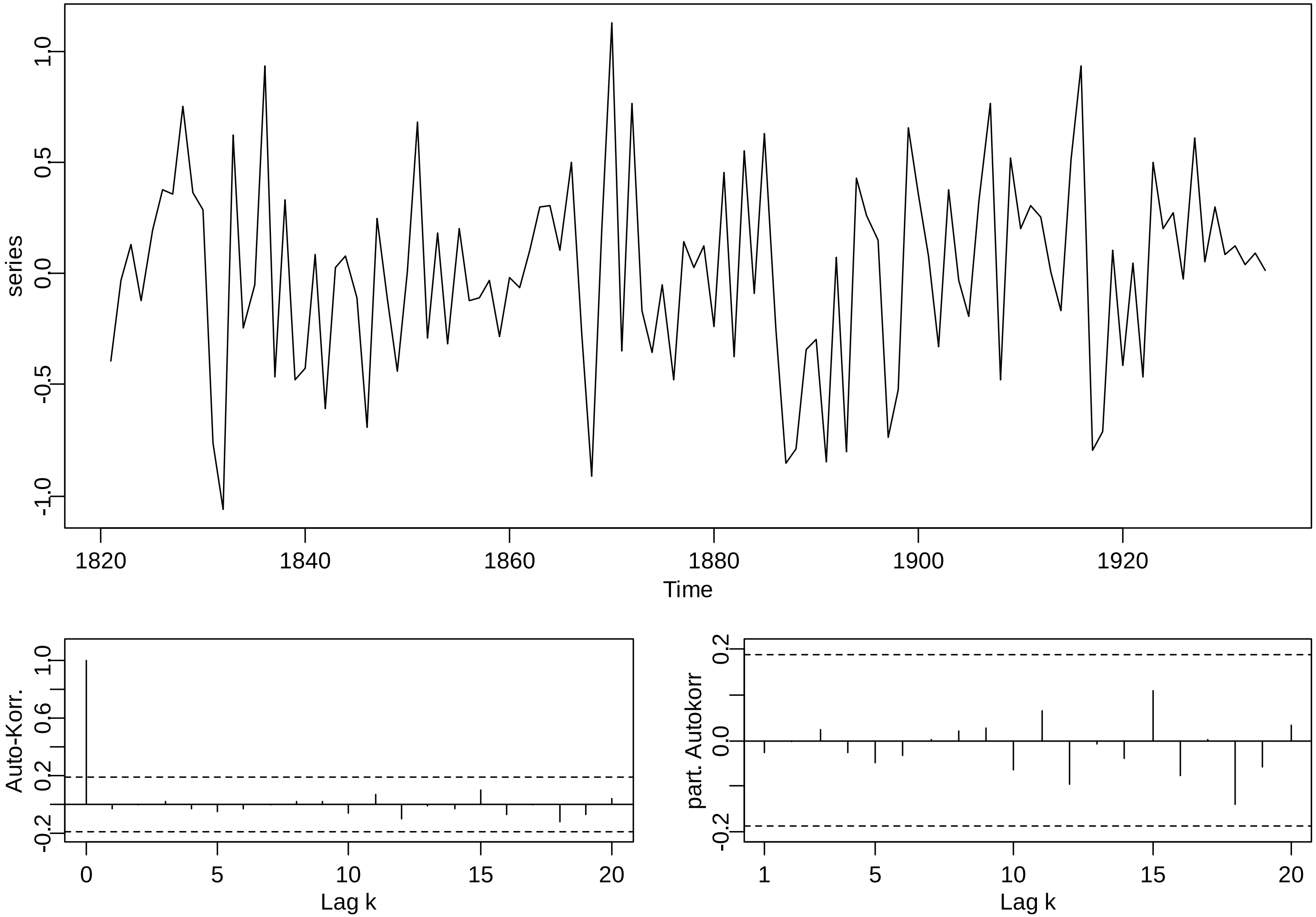
Model Diagnostics: log(lynx) data, AR(2)





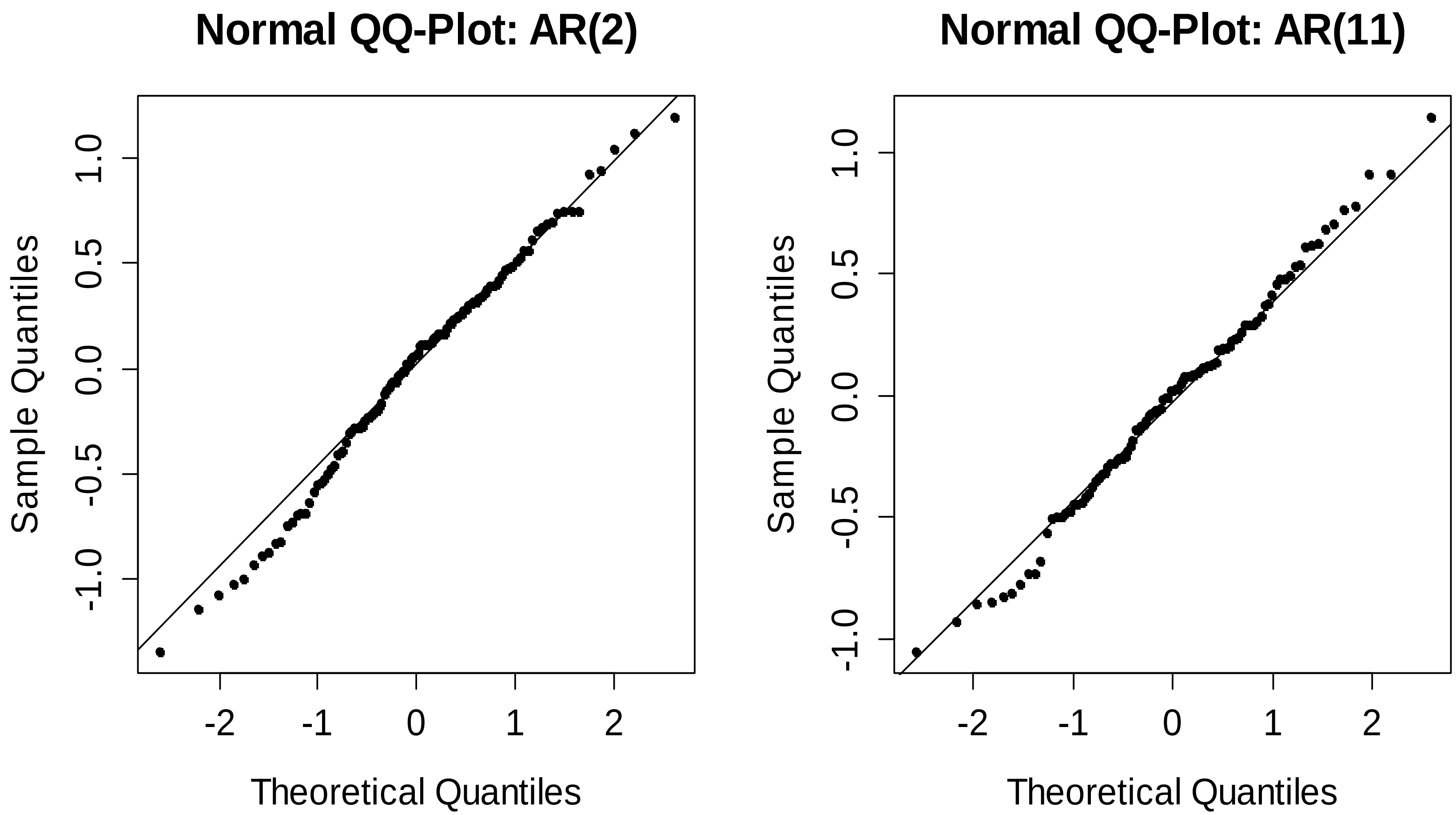


Model Diagnostics: log(lynx) data, AR(11)



Applied Time Series Analysis
SS 2014 – Week 04

Model Diagnostics: Normal Plots



AIC/BIC

If several alternative models show satisfactory residuals, using the information criteria AIC and/or BIC can help to choose the most suitable one:

AIC = $-2\log(L) + 2p$
BIC = $-2\log(L) + 2\log(n)p$

where

$L(\alpha, \mu, \sigma^2) = f(x, \alpha, \mu, \sigma^2)$ = „Likelihood Function“
p is the number of parameters and equals p or p+1
n is the time series length

Goal: **Minimization of AIC and/or BIC**

AIC/BIC

We need (again) a distribution assumption in order to compute the AIC and/or BIC criteria. Mostly, one relies again on i.i.d. normally distributed innovations. Then, the criteria simplify to:

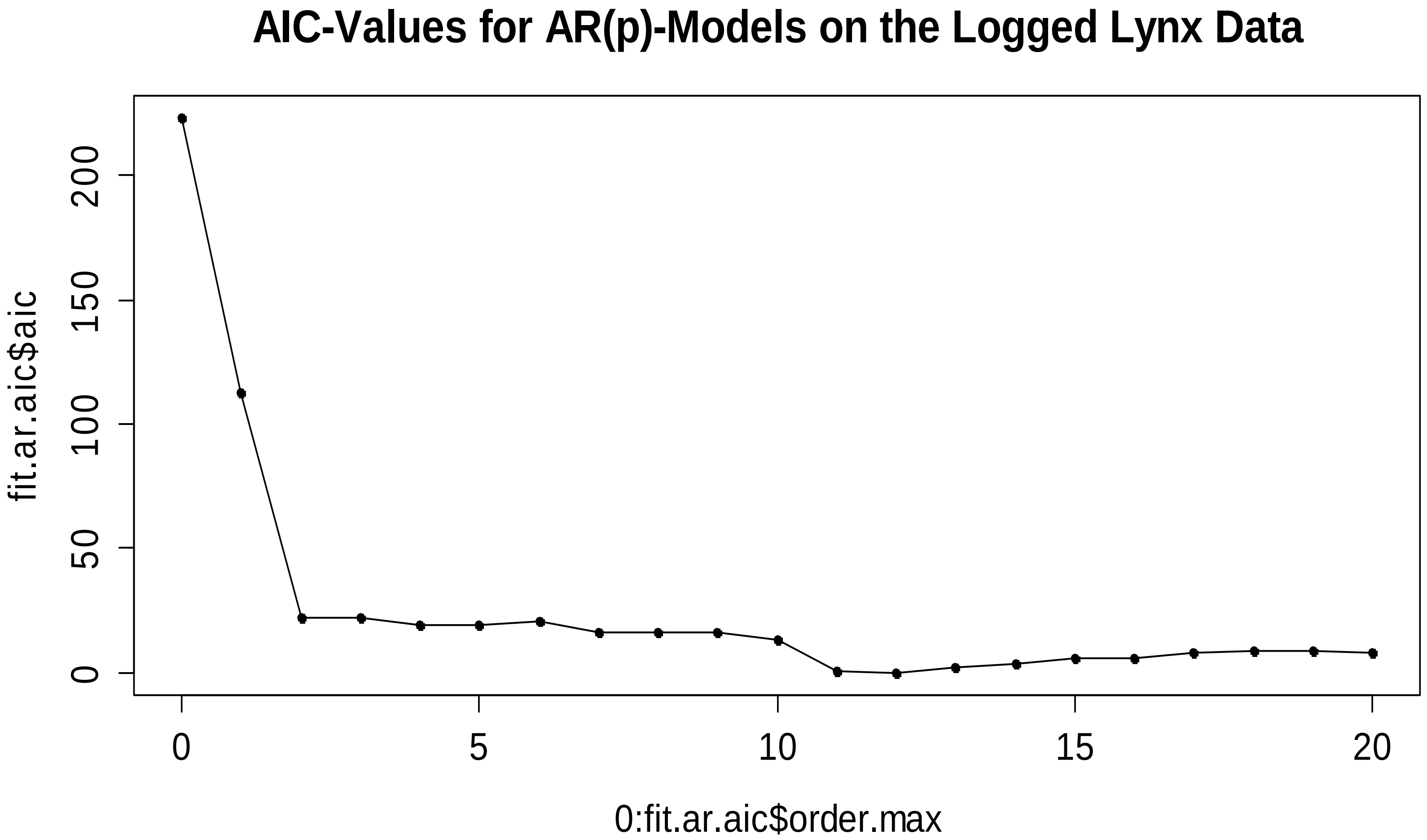
$$\text{AIC} = n\log(\hat{\sigma}_E^2) + 2p$$
$$\text{BIC} = n\log(\hat{\sigma}_E^2) + 2\log(n)p$$

Remarks:

→ AIC tends to over-, BIC to underestimate the true p

→ Plotting AIC/BIC values against p can give further insight. One then usually chooses the model where the last significant decrease of AIC/BIC was observed

AIC/BIC


AIC-Values for AR(p)-Models on the Logged Lynx Data
fit.ar.aic$aic
0
50
100
150
200
0
5
10
15
20
0:fit.ar.aic$order.max

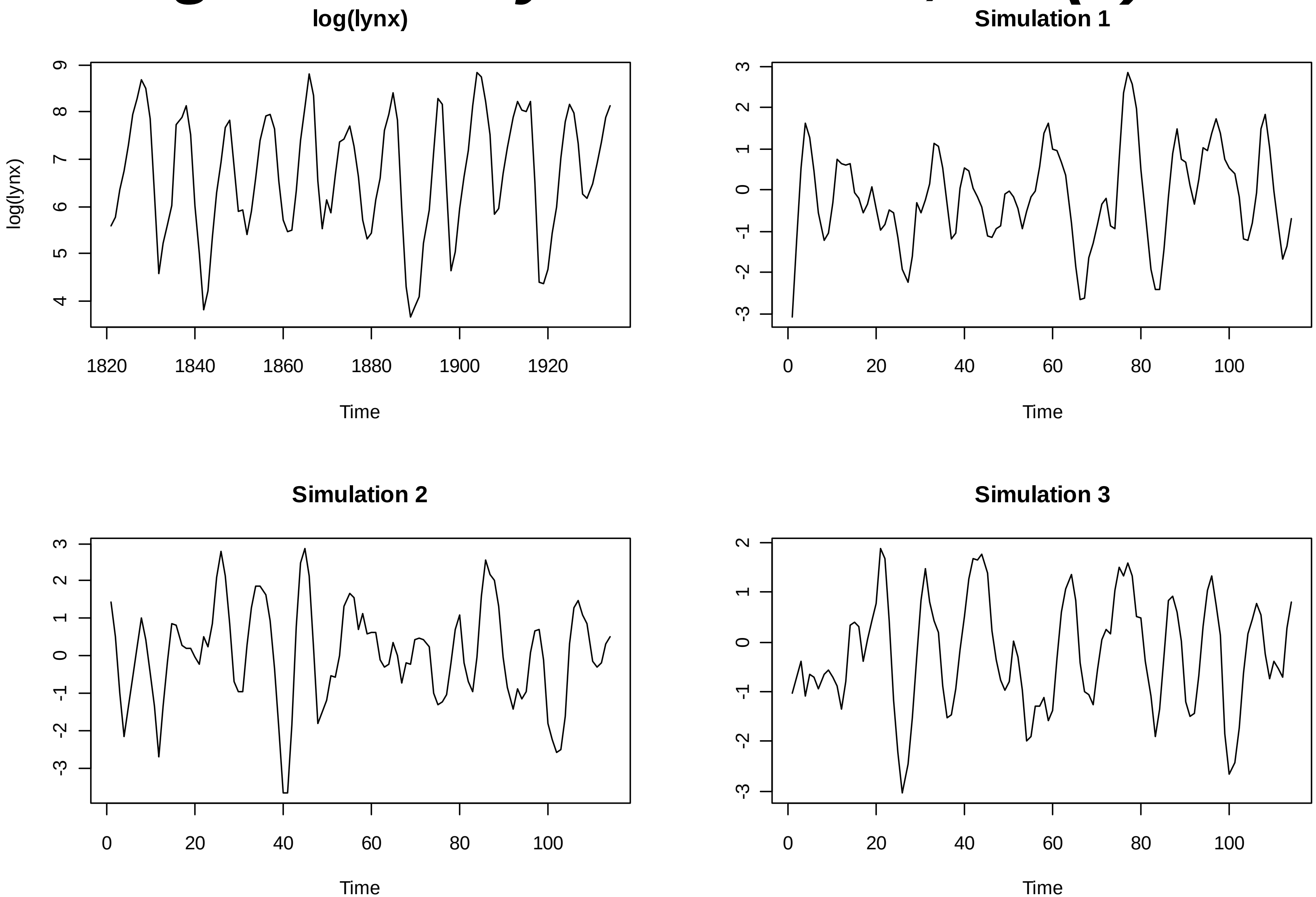
Diagnostics by Simulation

As a last check before a model is called appropriate, simulating from the estimated coefficients and visually inspecting the resulting series (without any prejudices) to the original can be done.

→ **The simulated series should „look like" the original. If this is not the case, the model failed to capture (some of) the properties of the original data.**

Applied Time Series Analysis
SS 2014 – Week 04

Diagnostics by Simulation, AR(2)



Applied Time Series Analysis
SS 2014 – Week 04

Diagnostics by Simulation, AR(11)