Applied Time Series Analysis
SS 2014 – Week 05 & Week 06

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 17, 2014

Applied Time Series Analysis
SS 2014 – Week 05 & Week 06

Looking Back & Outlook

We did consider shifted **AR(p)-models** $Y_t = m + X_t$ with:

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

where the correlation structure was as follows:

ACF: „exponential decay"
PACF: = 0 for all lags k>p

Now, in practice we could well observe a time series whose autocorrelation differs from the above structure.

We will thus discuss **ARMA(p,q) models**, a class that is suitable for modeling a wider spectrum of dependency structures.

Moving Average Models

Whereas for AR(p) models, the current observation of a time series is written as a linear combination of its own past, **MA(q) models** can be seen as an extension of the „pure“ process

$X_t = E_t$, where E_t is a white noise process,

in the sense that past innovation terms $E_{t-1}, E_{t-2}, \ldots$ are included, too. We call this a **moving average** model:

$$X_t = E_t + \beta_1 E_{t-1} + \beta_2 E_{t-2} + \ldots + \beta_q E_{t-q}$$

This is a time series process that is stationary, but not iid. In many respects, MA(q) models are complementary to AR(p).

Notation for MA(q)-models

The backshift operator, and the characteristic polynom, allow for convenient notation:

MA(q): $X_t = E_t + \beta_1 E_{t-1} + \beta_2 E_{t-2} + \ldots + \beta_q E_{t-q}$

MA(q) with BS: $X_t = \left(1 + \beta_1 B + \beta_2 B^2 + \ldots + \beta_q B^q\right) E_t$

MA(q) with BS+CP: $X_t = \Theta(B) E_t$

where

$$\Theta(z) = 1 + \beta_1 z + \beta_2 z^2 + \ldots + \beta_q z^q$$

is the characteristic polynom

Stationarity of MA(1)-Models

We first restrict ourselves to the simple MA(1)-model

$X_t = E_t + \beta_1 E_{t-1}$, where E_t is a White Noise innovation

The series X_t is weakly stationary, no matter what the choice of the parameter β_1 is.

Remember that for proving this, we have to show that:

- the expected value is 0
- the variance is constant and finite
- the autocovariance only depends on the lag k

→ **see the blackboard for the proof**

ACF of the MA(1)-Process

We can deduct the ACF for the MA(1)-process:

$$\rho(1) = \frac{\gamma(1)}{\gamma(0)} = \frac{\beta_1}{(1+\beta_1^2)} < 0.5$$

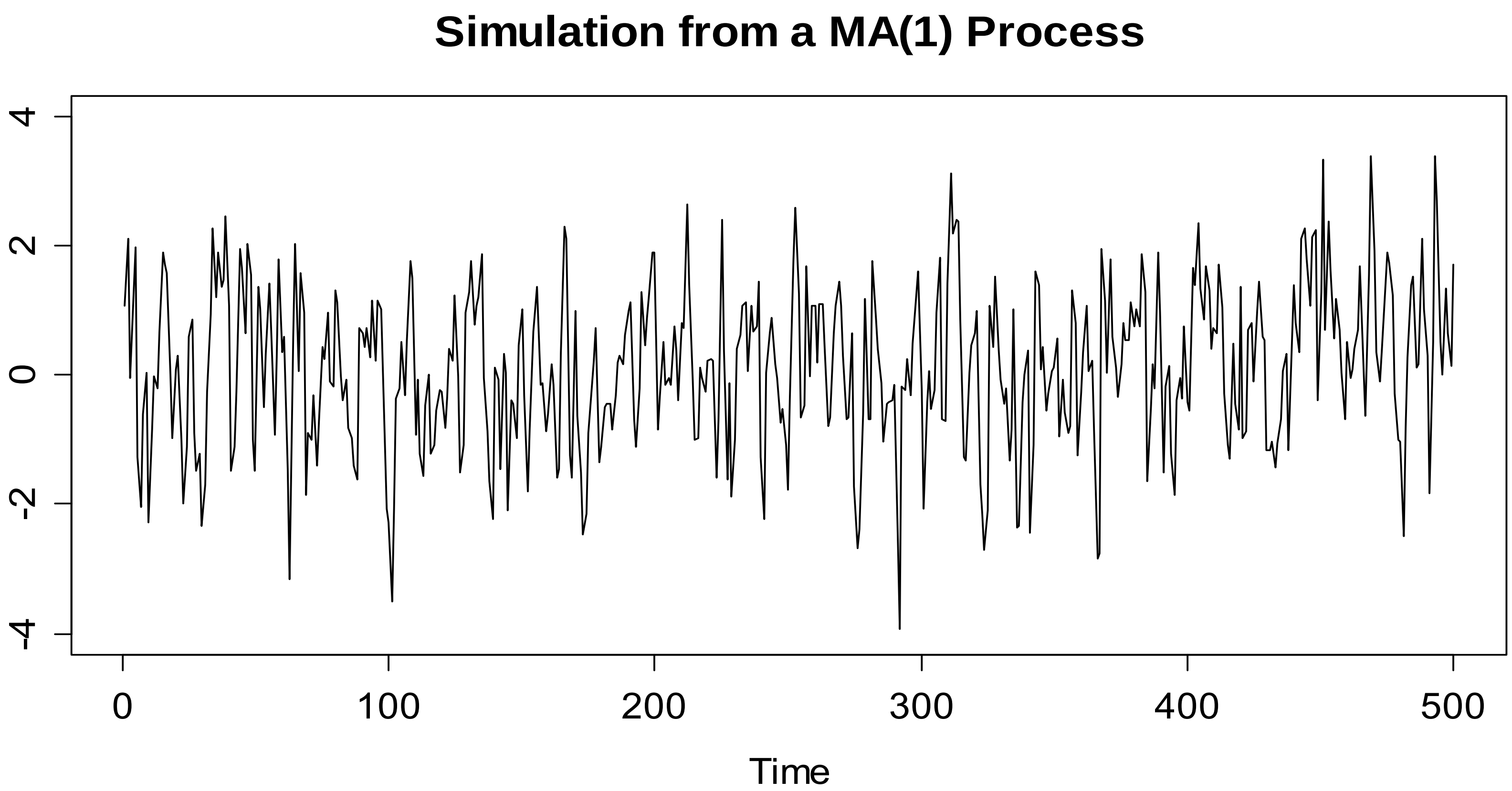
and

$$\rho(k) = 0 \quad \text{for all } k>1.$$

Thus, we have a "cut-off" situation, i.e. a similar behavior to the one of the PACF in an AR(1) process. This is why and how AR(1) and MA(1) are complementary.

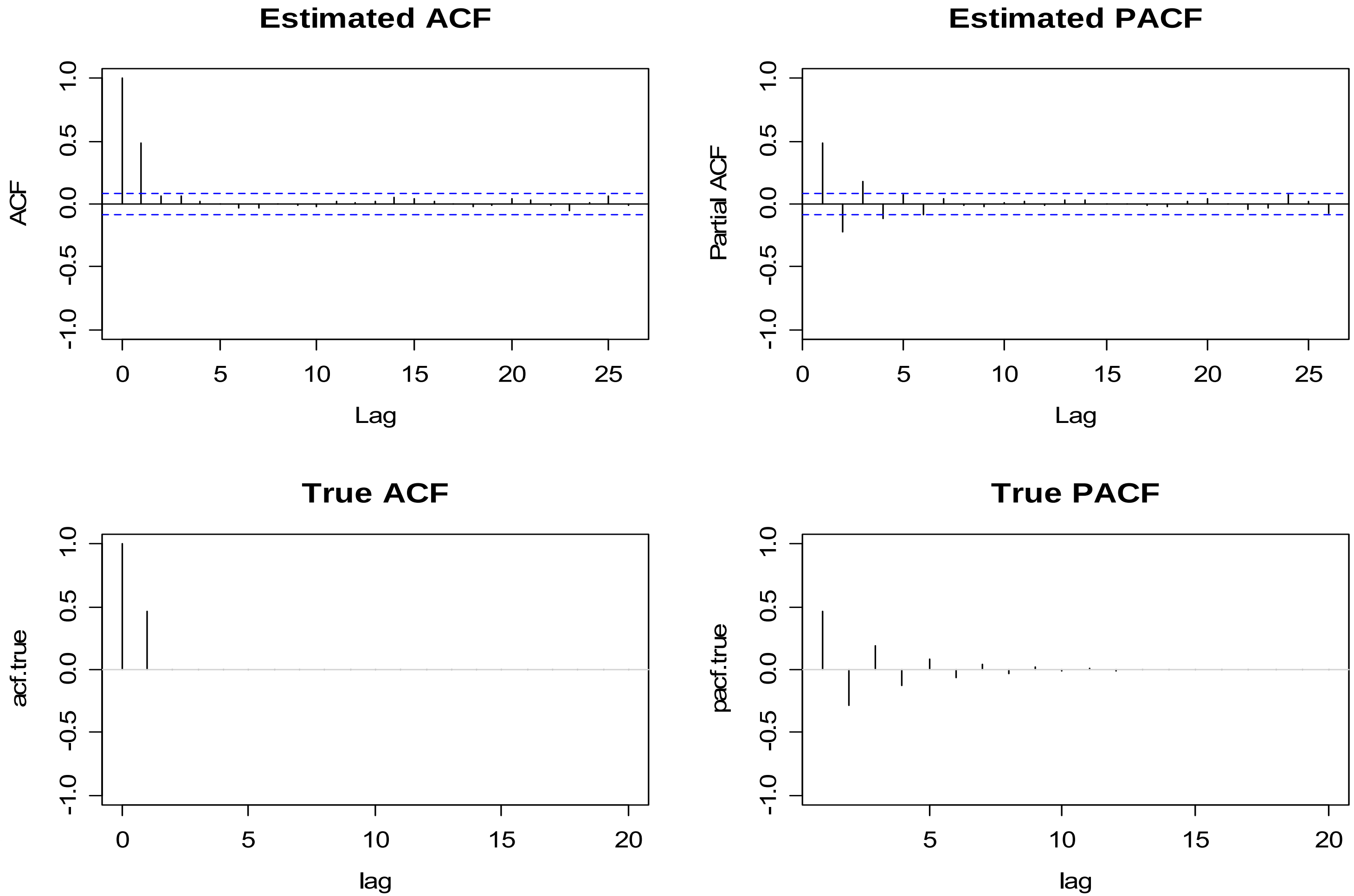
Simulated Process with β_1=0.7

```
> ts.ma1 <- arima.sim(list(ma=0.7), n=500)
> plot(ts.ma1, ylab="", ylim=c(-4,4))
```



ACF and PACF of MA(1)

```
> acf.true  <- ARMAacf(ma=0.7, lag.max=20)
> pacf.true <- ARMAacf(ma=0.7, pacf=T, lag.m=20)
```



MA(1): Remarks

Without additional assumptions, the ACF of an MA(1) doesn‘t allow identification of the generating model.

In particular, the two processes

$$X_t = E_t + 0.5 \cdot E_{t-1}$$

$$U_t = E_t + 2 \cdot E_{t-1}$$

have identical ACF:

$$\rho(1) = \frac{\beta_1}{1+\beta_1^2} = \frac{1/\beta_1}{1+(1/\beta_1^2)}$$

MA(1): Invertibilty

- An MA(1)-, or in general an MA(q)-process is said to be invertible if the roots of the characteristic polynomial $\Theta(B)$ lie outside of the unit circle.

- Under this condition, there exists only one MA(q)-process for any given ACF. But please note that any MA(q) is stationary, no matter if it is invertible or not.

- The condition on the characteristic polynomial translates to restrictions on the coefficients. For any MA(1)-model, $|\beta_1| < 1$ is required.

- R function `polyroot()` can be used for finding the roots.

Practical Importance of Invertibility

The condition of invertibility is not only a technical issue, but has important practical meaning. Invertible MA(1)-processes can be written as an AR(∞):

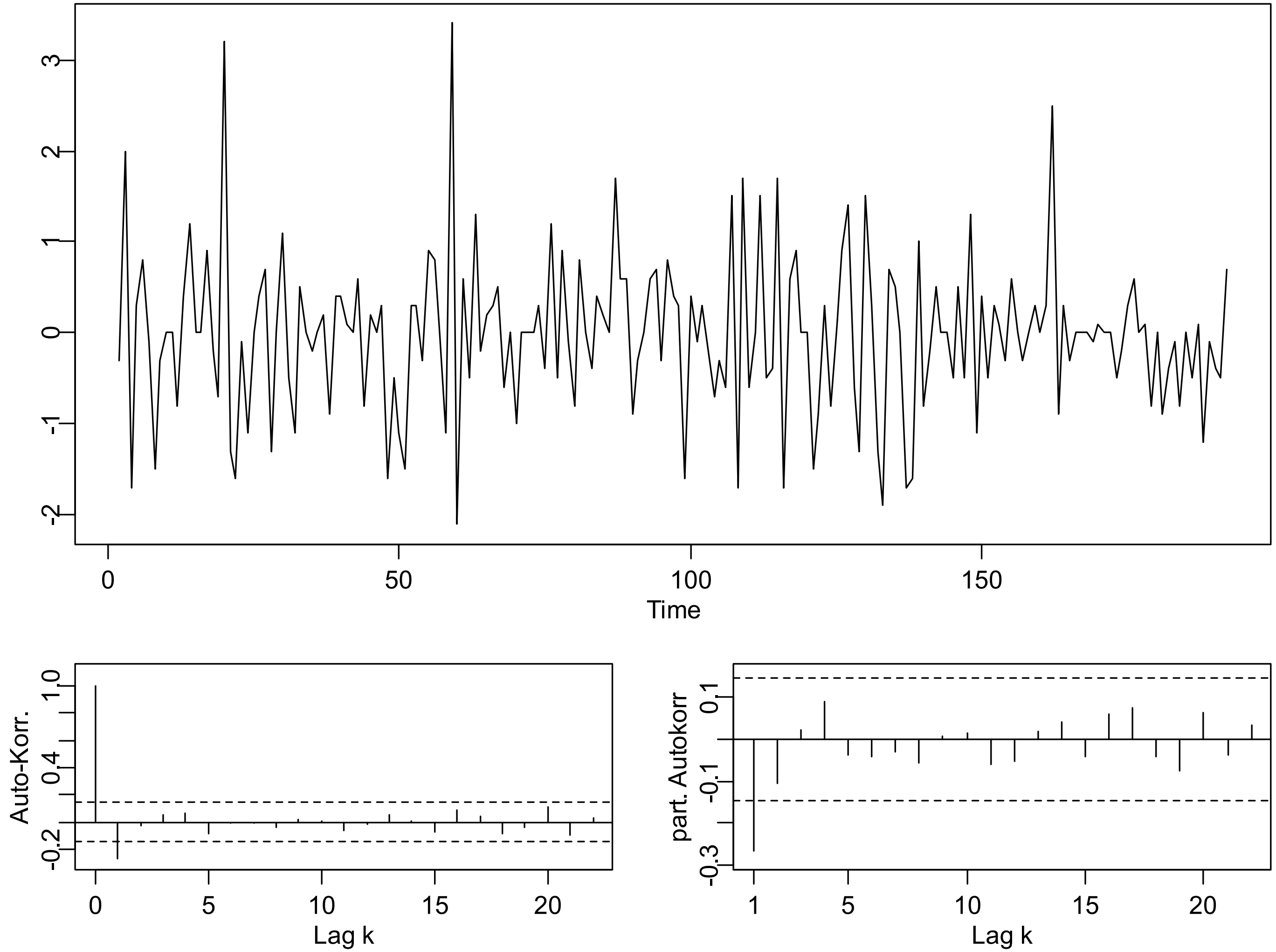
$$
\begin{aligned}
X_t &= E_t + \beta_1 E_{t-1} \\
&= E_t + \beta_1 (X_{t-1} - \beta_1 E_{t-2}) \\
&= \ldots \\
&= E_t + \beta_1 X_{t-1} - \beta_1^2 X_{t-2} + \beta_1^3 X_{t-3} + \ldots \\
&= E_t + \sum_{i=1}^{\infty} \psi_i X_{t-i}
\end{aligned}
$$

Invertibility is practically relevant for model fitting!

MA(1): Example

- daily return of an AT&T bond from 04/1975 to 12/1975
- the time series has 192 observations
- we are looking at the first-order differences
- an MA(1) model seems to fit the data (→ next slide)
- since we are looking at a differenced series, this is in fact an ARIMA(0,1,1) model (→ will be discussed later…)

MA(1): Example



3
2
1
0
-1
-2
0
50
100
150
Time
Auto-Korr.
1.0
0.4
-0.2
0
5
10
15
20
Lag k
part. Autokorr
0.1
-0.1
-0.3
1
5
10
15
20
Lag k

MA(q)-Models

The MA(q)-model is defined as follows:

$$X_t = E_t + \beta_1 E_{t-1} + \beta_2 E_{t-2} + \ldots + \beta_q E_{t-q} \ ,$$

where E_t are i.i.d. innovations (=a white noise process).

The ACF of this process can be computed from the coefficients:

$$\rho(k) = \frac{\sum_{i=0}^{q-k} \beta_i \beta_{i+k}}{\sum_{i=0}^{q} \beta_i^2} \ ,$$ for all k=1,…, q with $\beta_0 = 1$

$$\rho(k) = 0 \ ,$$ for all k>q

ACF/PACF of MA(q)

ACF

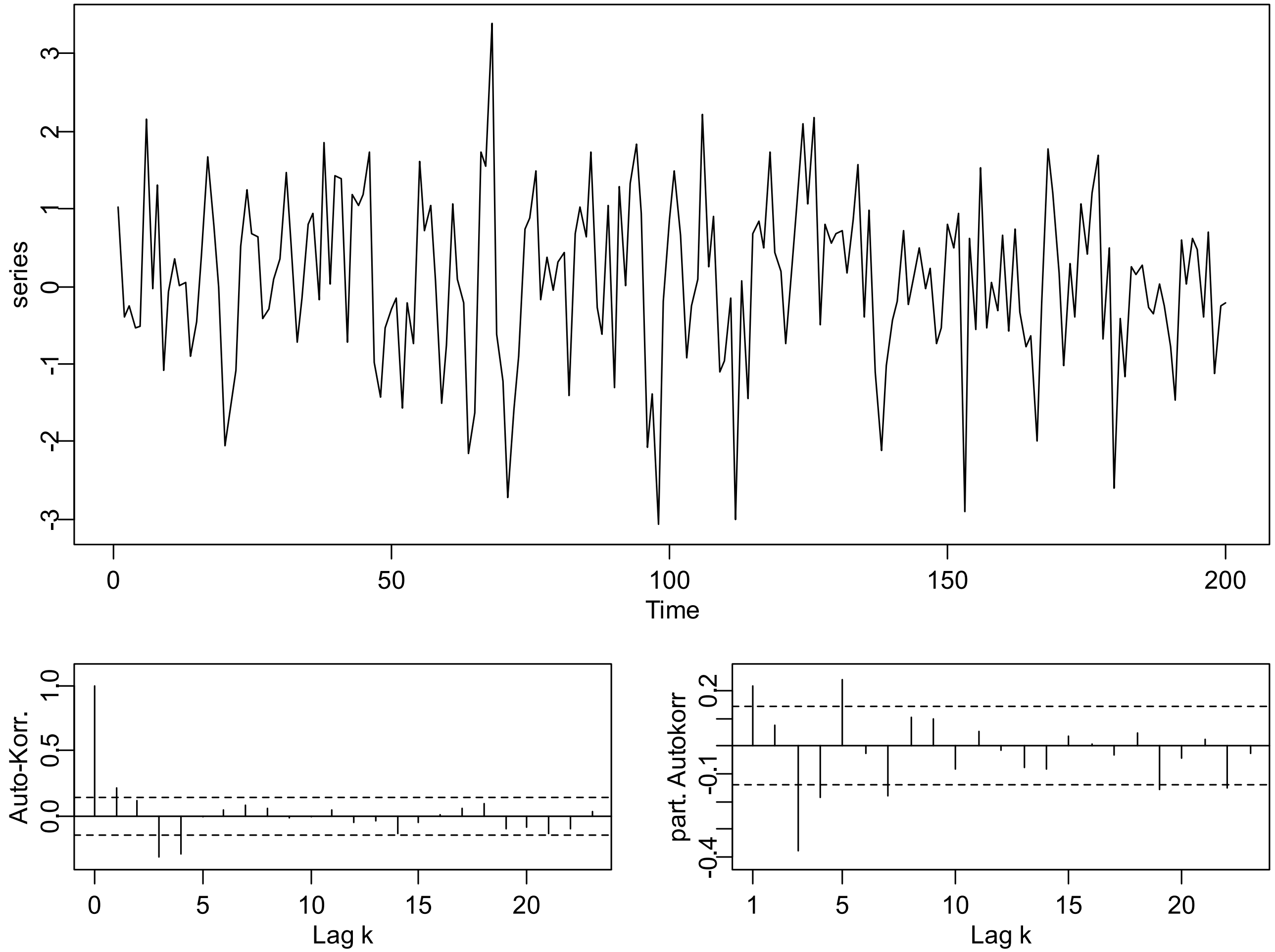
- the ACF of an MA(q) has a cut-off at lag k=q
- it behaves thus like the PACF of an AR(q)-model

PACF

- the PACF is (again) complicated to determine, but:
- the PACF of an MA(q) has an „exponential decay“
- it behaves thus like the ACF of an AR-model

MA(4): Example

$$X_t = E_t + 0.3 \cdot E_{t-1} + 0.3 \cdot E_{t-2} - 0.2 \cdot E_{t-3} - 0.2 \cdot E_{t-4}, \quad E_t \sim N(0,1)$$



Applied Time Series Analysis
SS 2014 – Week 05 & Week 06

ARMA(p,q)-Models

An ARMA(p,q)-model combines AR(p) and MA(q):

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + E_t + \beta_1 E_{t-1} + \ldots + \beta_q E_{t-q}$$

where E_t are i.i.d. innovations (=a white noise process).

It‘s easier to write an ARMA(p,q) with the characteristic polynom:

$\Phi(B) X_t = \Theta(B) E_t$, where

$\Phi(z) = 1 - \alpha_1 z - \ldots \alpha_p z^p$ is the cP of the AR-part, and

$\Theta(z) = 1 + \beta_1 z + \ldots + \beta_q z^q$ is the cP of the MA-part

Stationarity/Invertibility of ARMA(p,q)

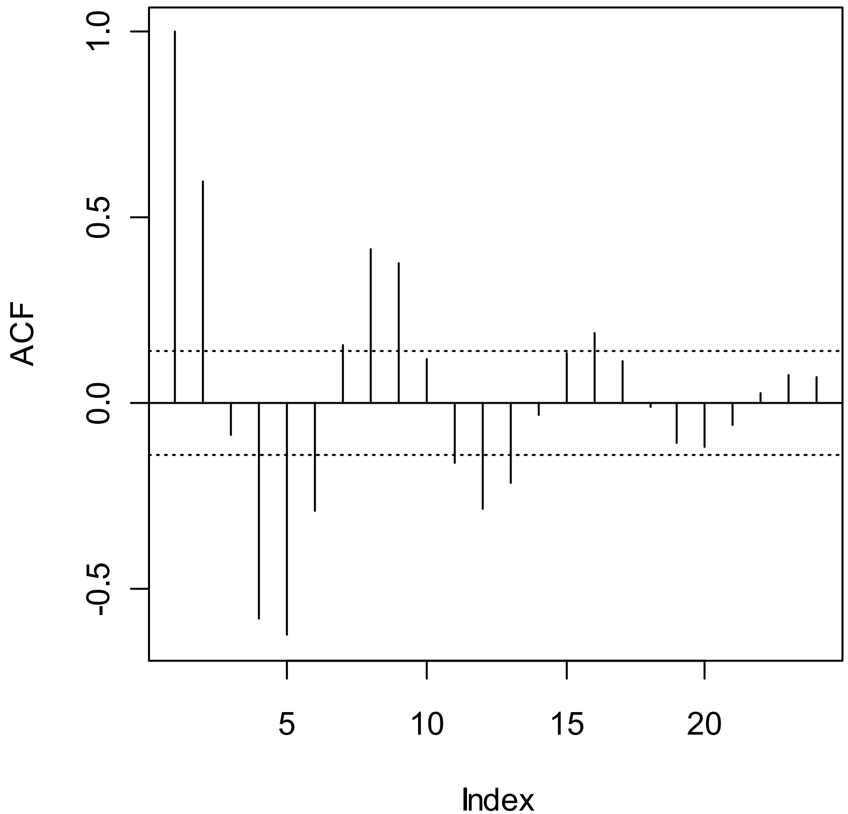
- both properties are determined by the cP
- the AR-cP determines stationarity
- the MA-cP determines invertibility
- condition: roots of the cP outside of the unit circle

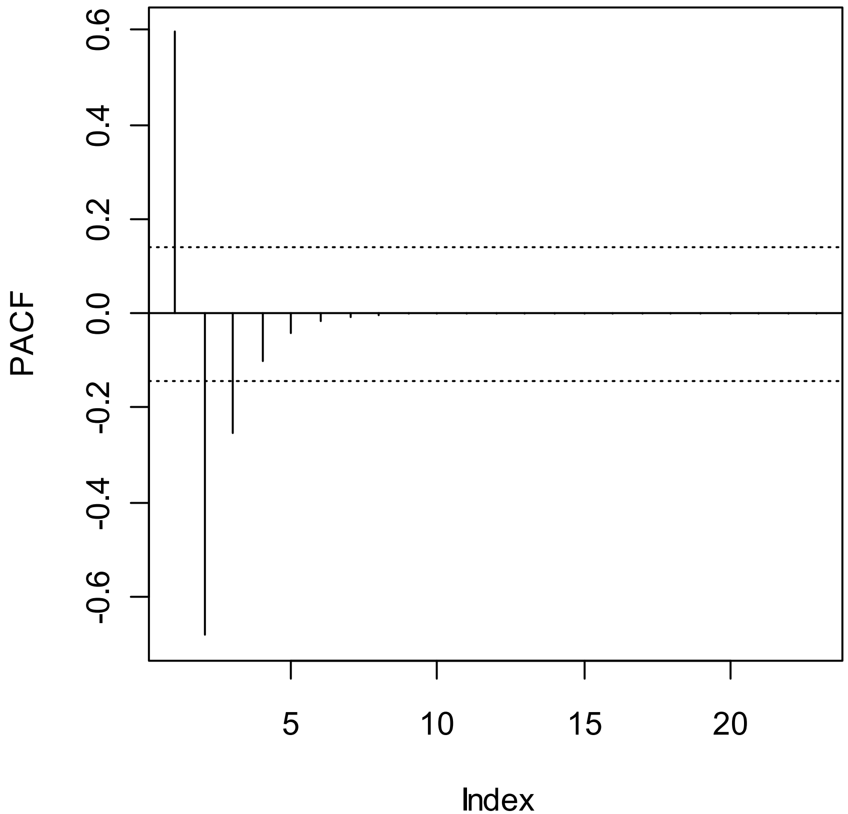
- stationarity: model can be written as a MA(∞)
- invertibility: model can be written as an AR(∞)

Applied Time Series Analysis
SS 2014 – Week 05 & Week 06

True ACF/PACF of an ARMA(2,1)

$$X_t = 1.2 \cdot X_{t-1} - 0.8 \cdot X_{t-2} + E_t + 0.4 \cdot E_{t-1}, \;\; E_t \sim N(0,1)$$

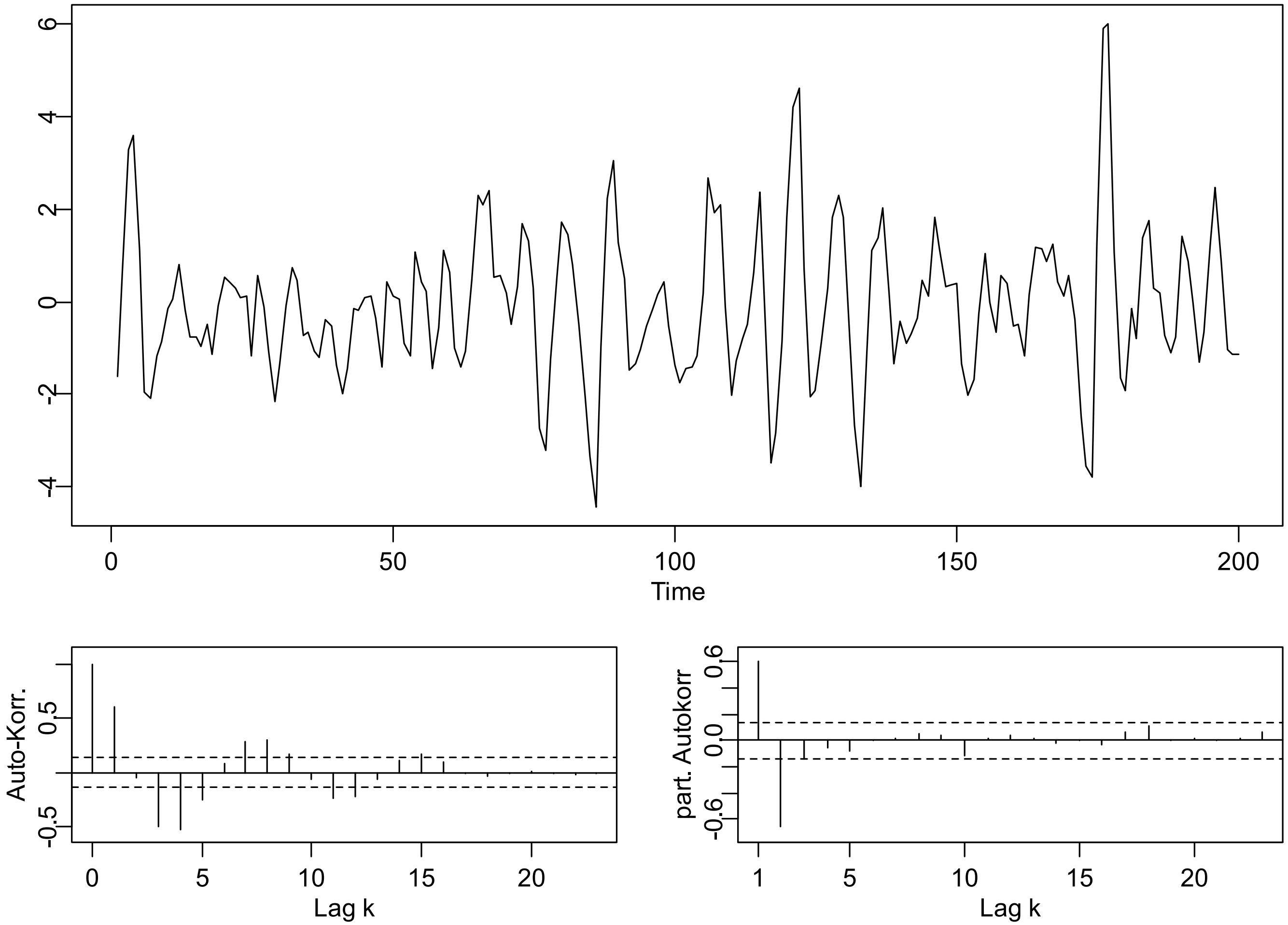




Applied Time Series Analysis
SS 2014 – Week 05 & Week 06

Simulated ACF/PACF of an ARMA(2,1)

$$X_t = 1.2 \cdot X_{t-1} - 0.8 \cdot X_{t-2} + E_t + 0.4 \cdot E_{t-1},\ E_t \sim N(0,1)$$



Properties of ACF/PACF in ARMA(p,q)

	ACF	PACF
AR(p)	exponential decay	cut-off at lag p
MA(q)	cut-off at lag q	exponential decay
ARMA(p,q)	mix decay/cut-off	mix decay/cut-off

→ all linear time series processes can be approximated by an ARMA(p,q) with possibly large p,q. They are thus are very rich class of models.

Applied Time Series Analysis
SS 2014 – Week 05 & Week 06

Fitting ARMA(p,q)

What needs to be done?

1) **Achieve stationarity**
   → transformations, differencing, modeling, …

2) **Choice of the order**
   → determining (p,q)

3) **Parameter estimation**
   → Estimation of $\alpha, \beta, m, \sigma_E^2$

4) **Residual analysis**
   → if necessary, repeat 1), and/or 2)-4)

Identification of the Order (p,q)

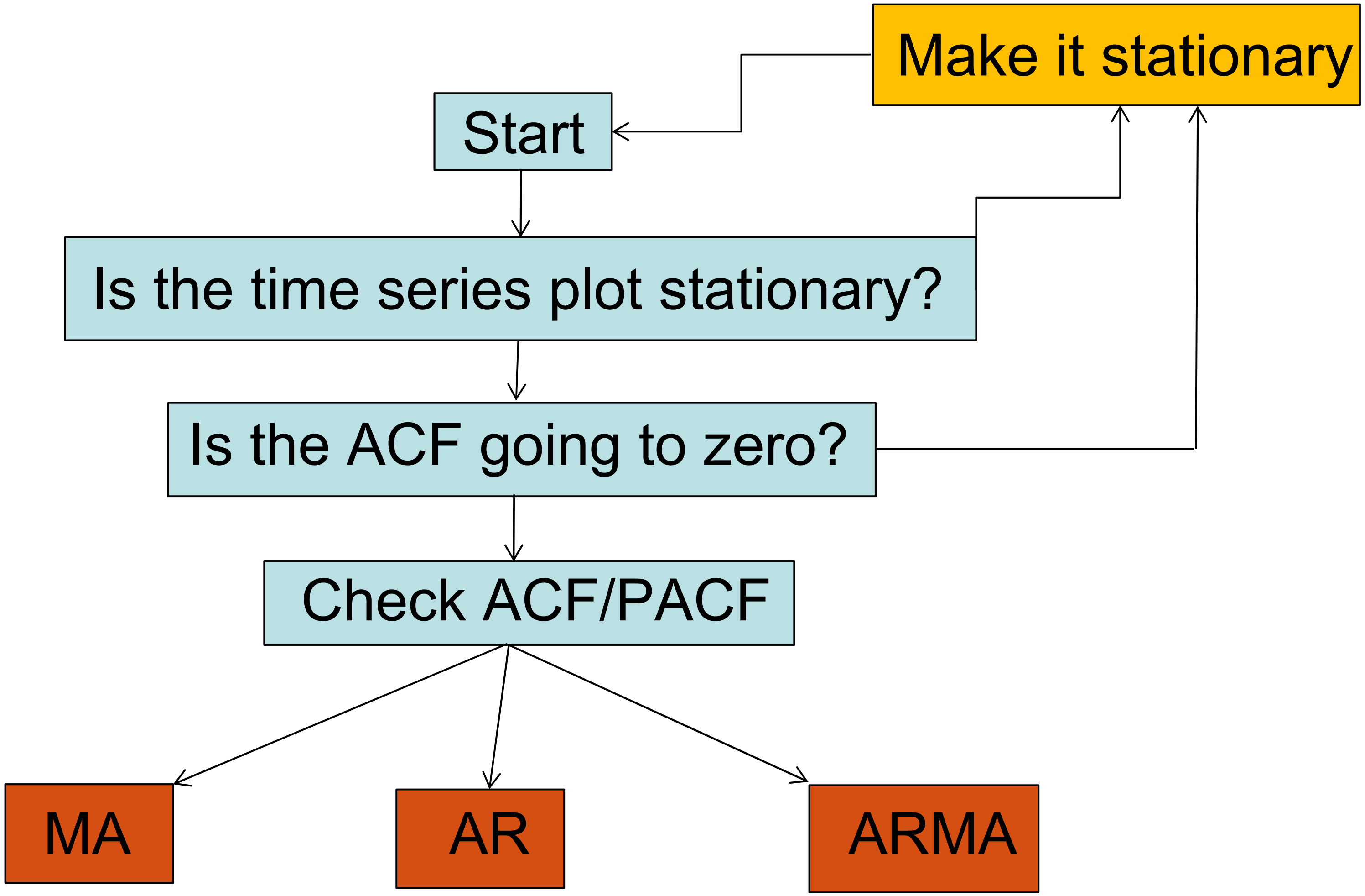
Please note:

- We only have one single realization of the time series with finite length.
- The plots (etc.) we base the order choice on are not „facts“, but are estimations with uncertainty.
- This holds especially for the ACF/PACF plots.
- Every ARMA(p,q) can be written as AR(∞) or MA(∞)

→ **There is usually >1 model that describes the data well.**

Applied Time Series Analysis
SS 2014 – Week 05 & Week 06

ARMA(p,q)-Modeling



Parameter Estimation

For parameter estimation with AR(p) models, we had 4 choices:

a) Regression
b) Yule-Walker
c) Maximum-Likelihood
d) Burg‘s Algorithm

For ARMA(p,q) models, only two options are remaining, and both of them require numerical optimization:

1) Conditional Sum of Squares
2) Maximum-Likelihood

Conditional Sum of Squares

Idea: This is an iterative approach where the parameters are determined such that the sum of squared errors (between observations and fitted values) are minimal.

$$S(\hat{\beta}_1,\ldots,\hat{\beta}_q)=\sum_{t=1}^{n}\hat{E}_t^2=\sum_{t=1}^{n}(X_t-(\hat{\beta}_1\hat{E}_{t-1}-\ldots-\hat{\beta}_1\hat{E}_{t-q})^2$$

This requires starting values which are chosen as:

$$\hat{E}_0=0,\ \hat{E}_{-1}=0,\ \ldots,\ \hat{E}_{1-q}=0$$

A numerical search is used to find the parameter values that minimize the entire conditional sum of squares. They also serve as starting values for MLE.

Maximum-Likelihood-Estimation

Idea: Determine the parameters such that, given the observed time series $x_1,\ldots,x_n$, the resulting model is the most plausible (i.e. the most likely) one.

→ **This requires the choice of a probability distribution for the time series $X = (X_1, \ldots, X_n)$**

Maximum-Likelihood-Estimation

If we assume the ARMA(p,q)-model

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + E_t + \beta_1 E_{t-1} + \ldots + \beta_q E_{t-q}$$

and i.i.d. normally distributed innovations

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

the time series vector has a multivariate normal distribution

$$X = (X_1, \ldots, X_n) \sim N(m \cdot \underline{1}, V)$$

with covariance matrix V that depends on the model parameters α, β and σ_E^2.

Maximum-Likelihood-Estimation

We then maximize the density of the multivariate normal distribution with respect to the parameters

$$\alpha, \beta, m \text{ and } \sigma_E^2 .$$

The observed x-values are hereby regarded as fixed values.

→ **This is a highly complex non-linear optimization problem that requires sophisticated algorithms and starting values which are usually provided by CSS (at least that's the default in R's `arima()`).**

Maximum-Likelihood-Estimation

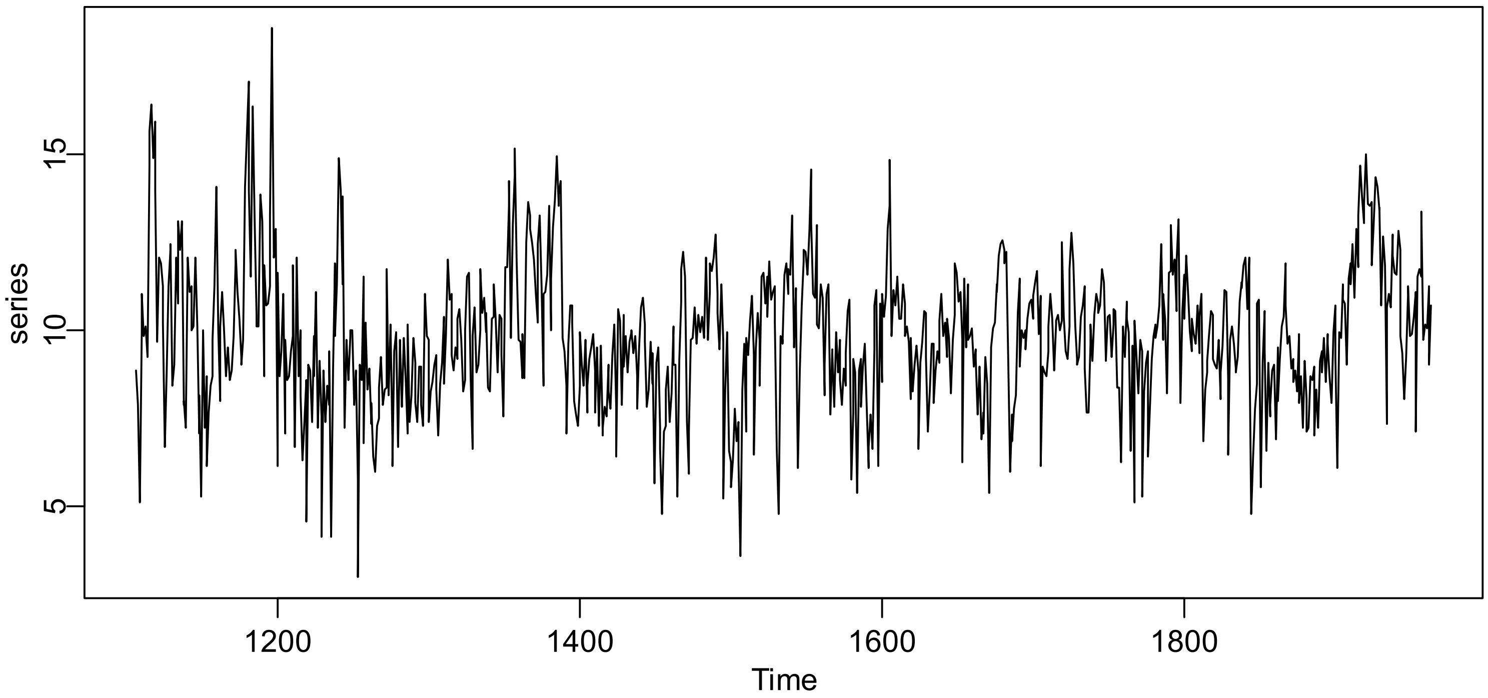
```
> r.Pmle <- arima(d.Psqrt,order=c(2,0,0),include.mean=T)
> r.Pmle

Call: arima(x=d.Psqrt, order=c(2,0,0), include.mean=T)

Coefficients:

        ar1    ar2  intercept
      0.275  0.395      3.554
s.e.  0.107  0.109      0.267

sigma^2 = 0.6:  log likelihood = -82.9,  aic = 173.8
```

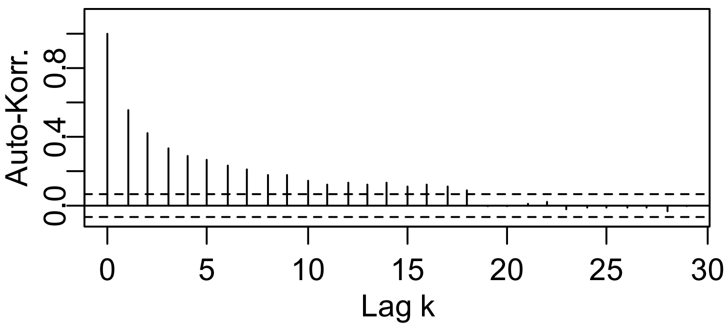
MLE: Remarks

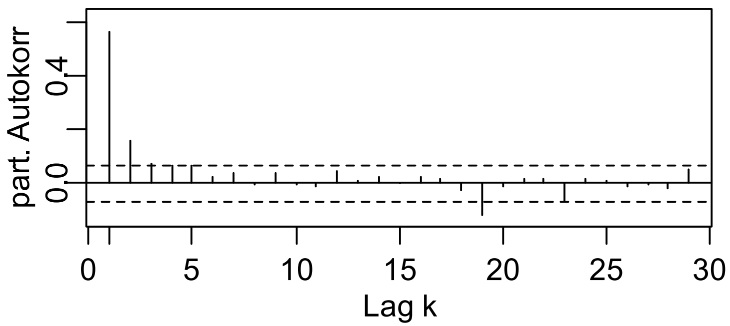
- The MLE approach would work for any distribution. However, for innovation distributions other than Gaussian, the joint distribution might be „difficult“.
- For „reasonable“ deviations from the normality assumption, MLE still yields „good“ results.
- Besides the parameter estimates, we also obtain an estimate of their standard error
- Other software packages such as for example SAS don't rely on MLE, but use CSS, which is in spirit similar to Burg's algorithm.

Applied Time Series Analysis
SS 2014 – Week 05 & Week 06

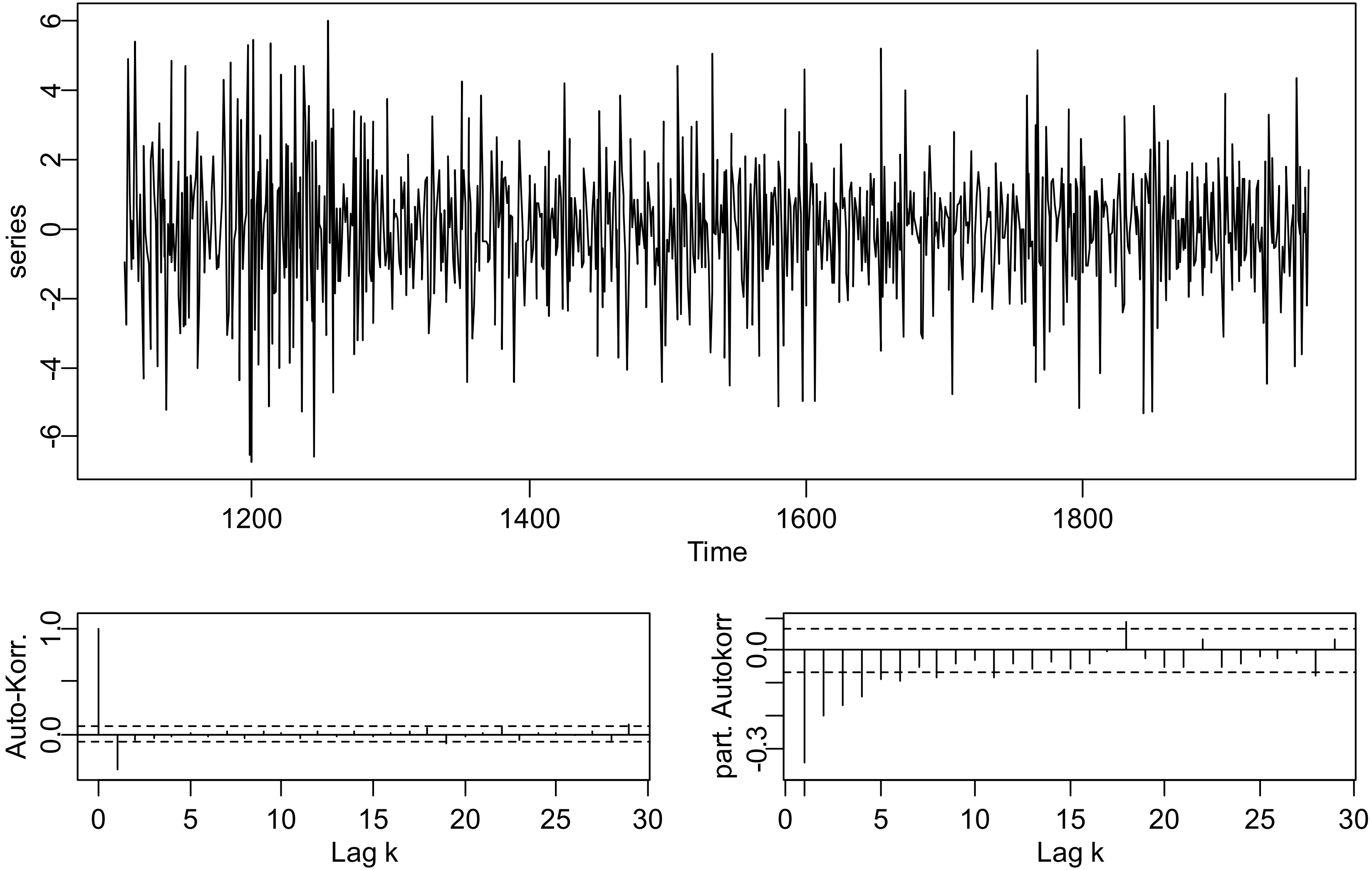
Douglas Fir: Original Data



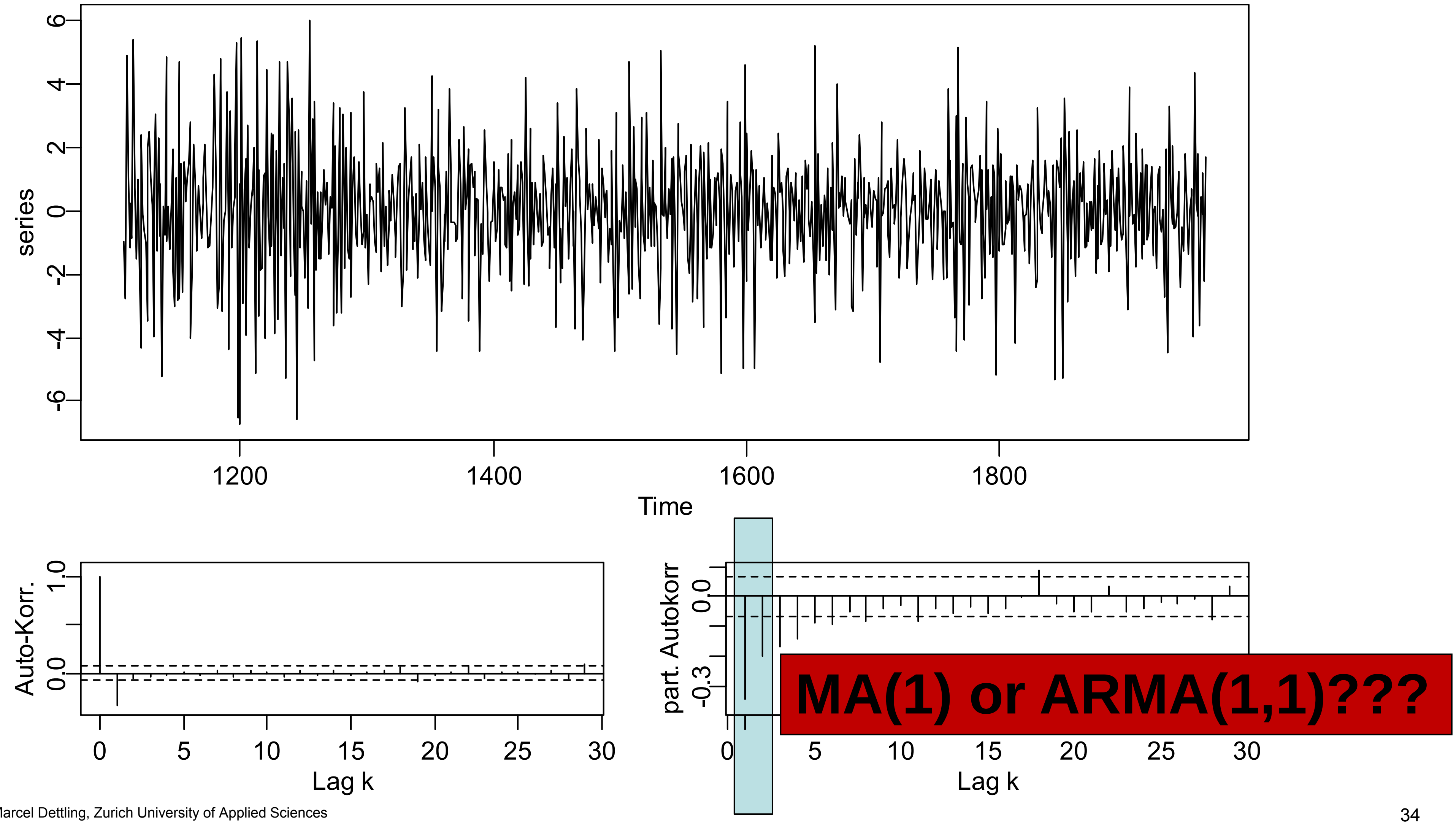




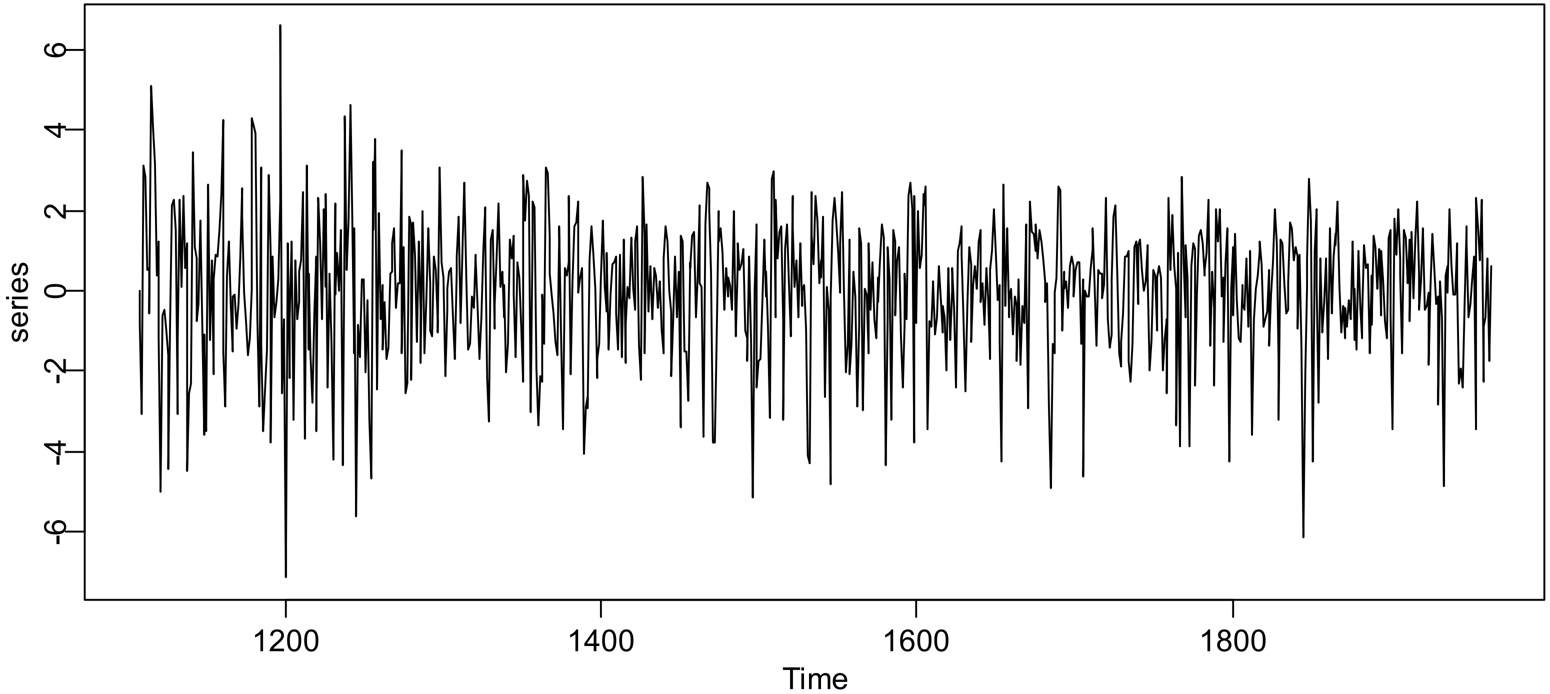
Douglas Fir: Differenced Series

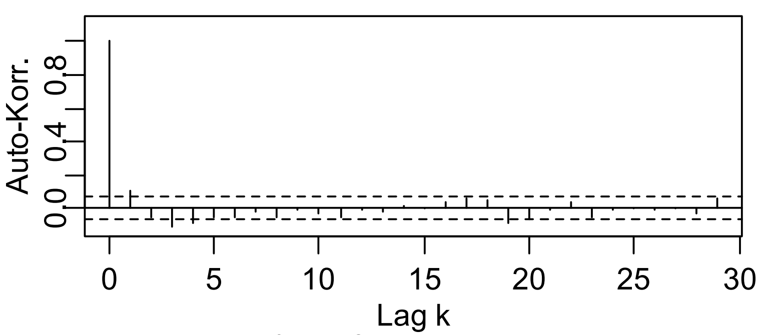


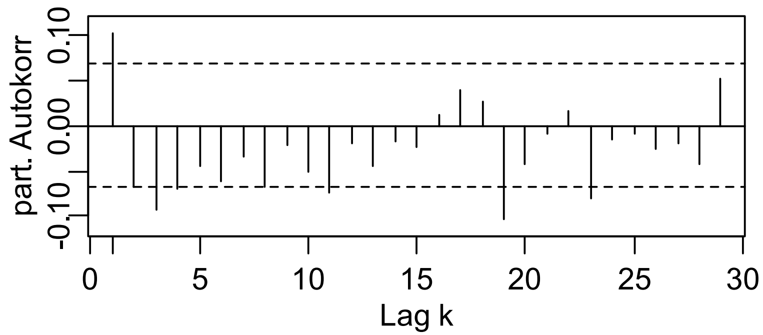
Douglas Fir: Differenced Series



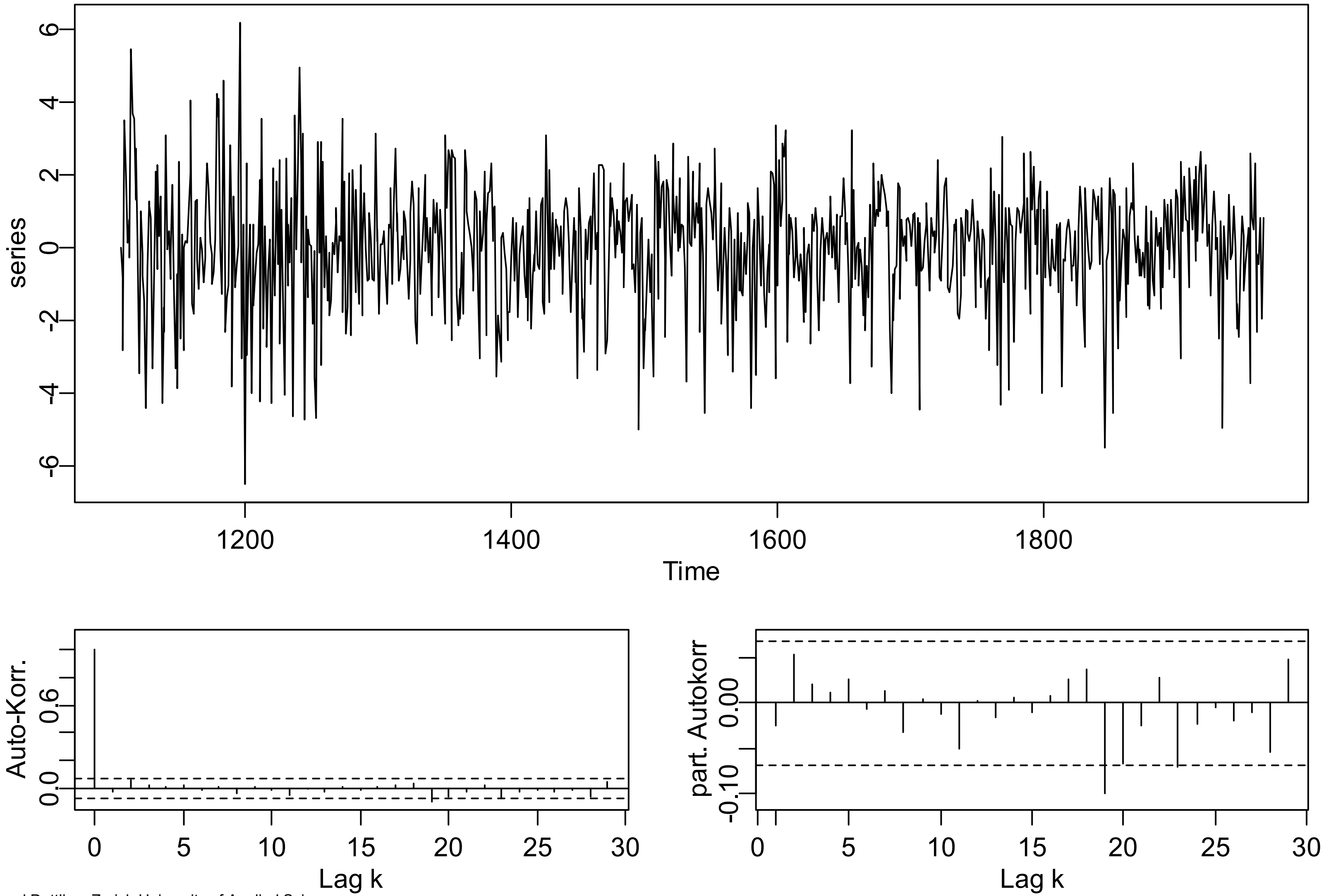
Residuals of MA(1)







Residuals of ARMA(1,1)



Another Example: Fitting ARMA(p,q)

What needs to be done?

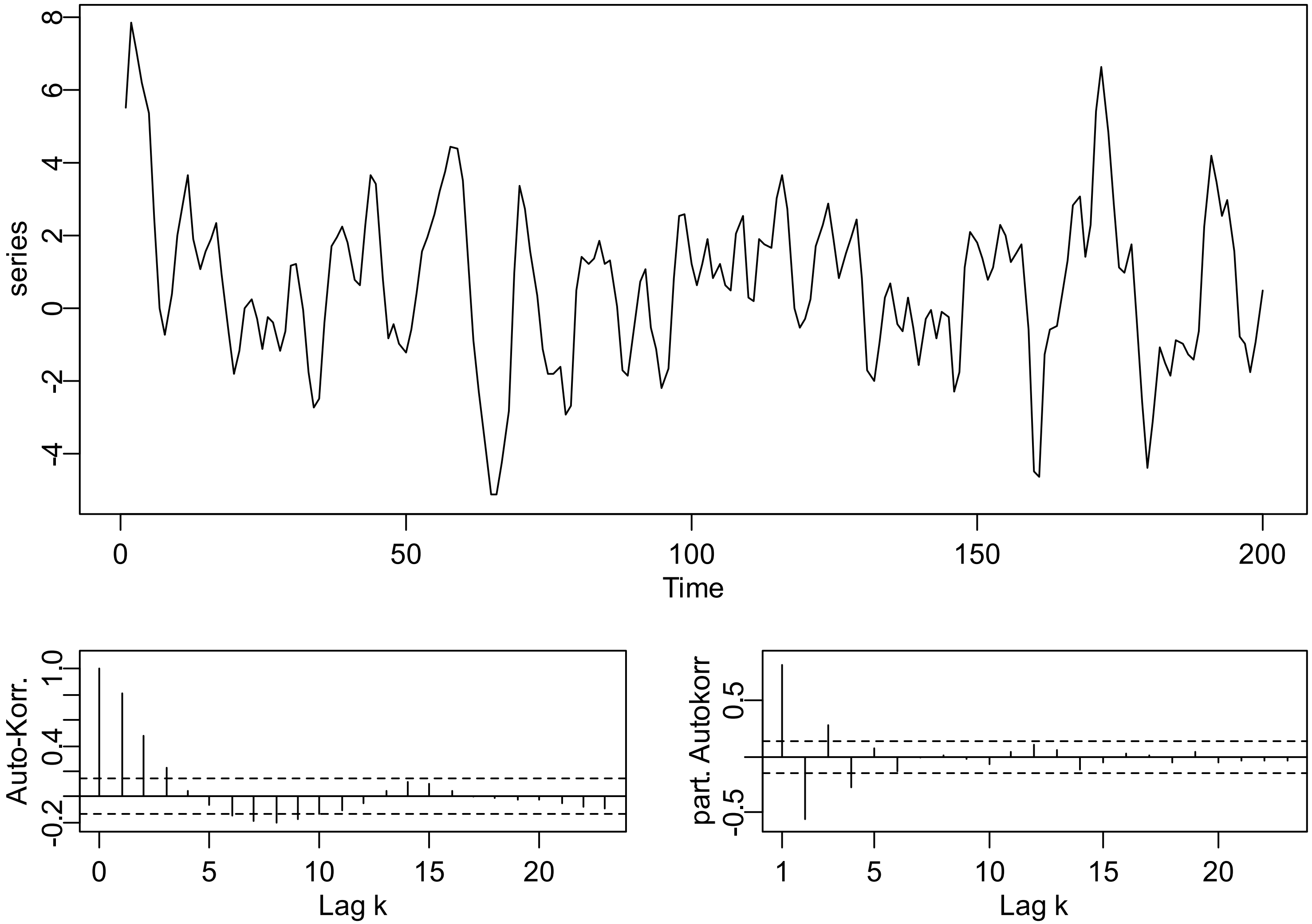
1) Achieve stationarity
   → transformations, differencing, modeling, …

2) Choice of the order
   → determining (p,q), plus integration order d for ARIMA

3) Parameter estimation
   → ML-estimation of $\alpha, \beta, \mu, \sigma_E^2$

4) Residual analysis
   → if necessary, repeat 1), and/or 2)-4)

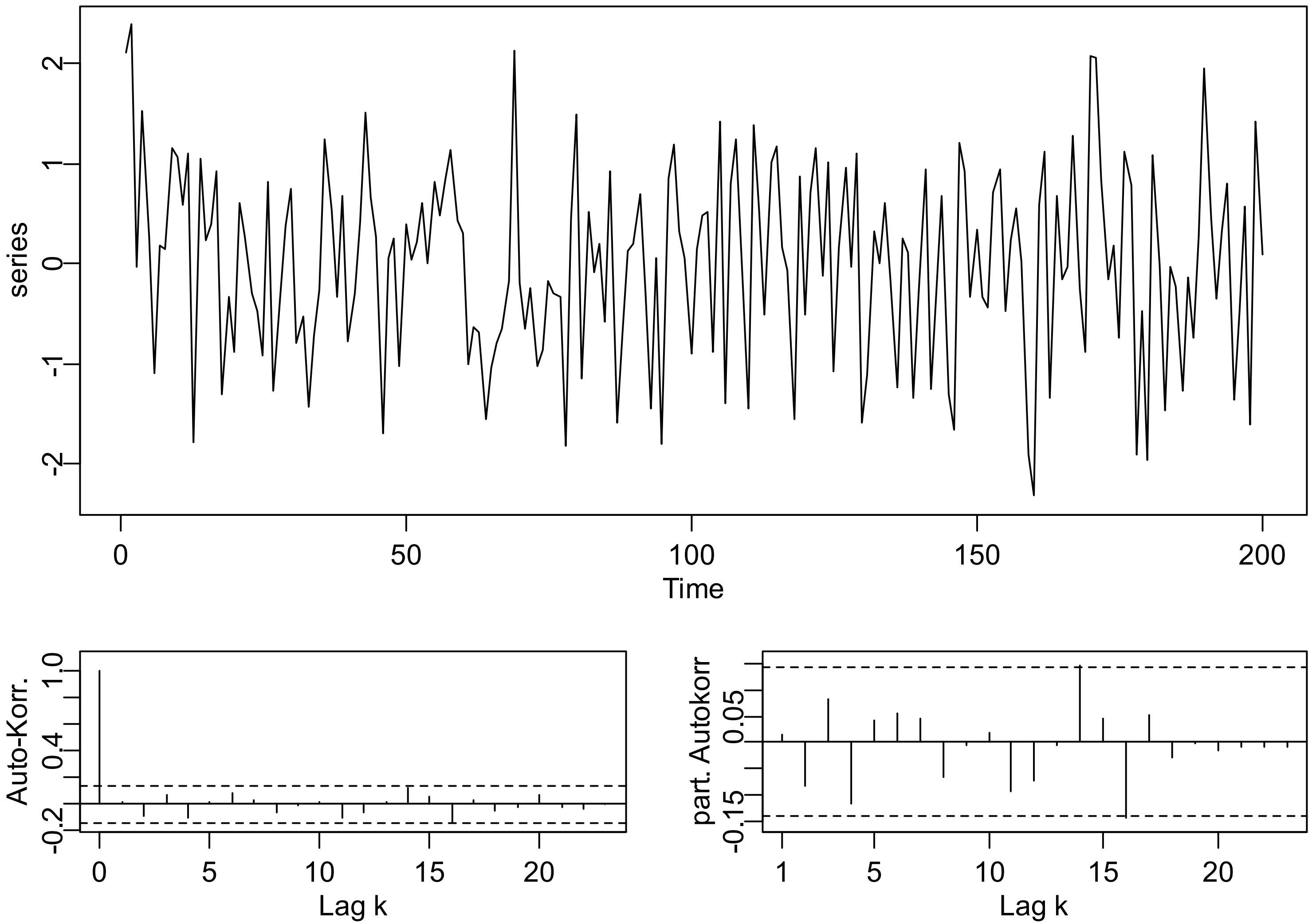
The Series, ACF and PACF



Model 1: AR(4)

```
> fit1

Call: arima(x = my.ts, order = c(4, 0, 0))

Coefficients:

         ar1      ar2     ar3      ar4  intercept
      1.5430  -1.2310  0.7284  -0.3000     0.6197
s.e.  0.0676   0.1189  0.1189   0.0697     0.2573

sigma^2=0.8923,  log likelihood=-273.67,  aic=559.33
```

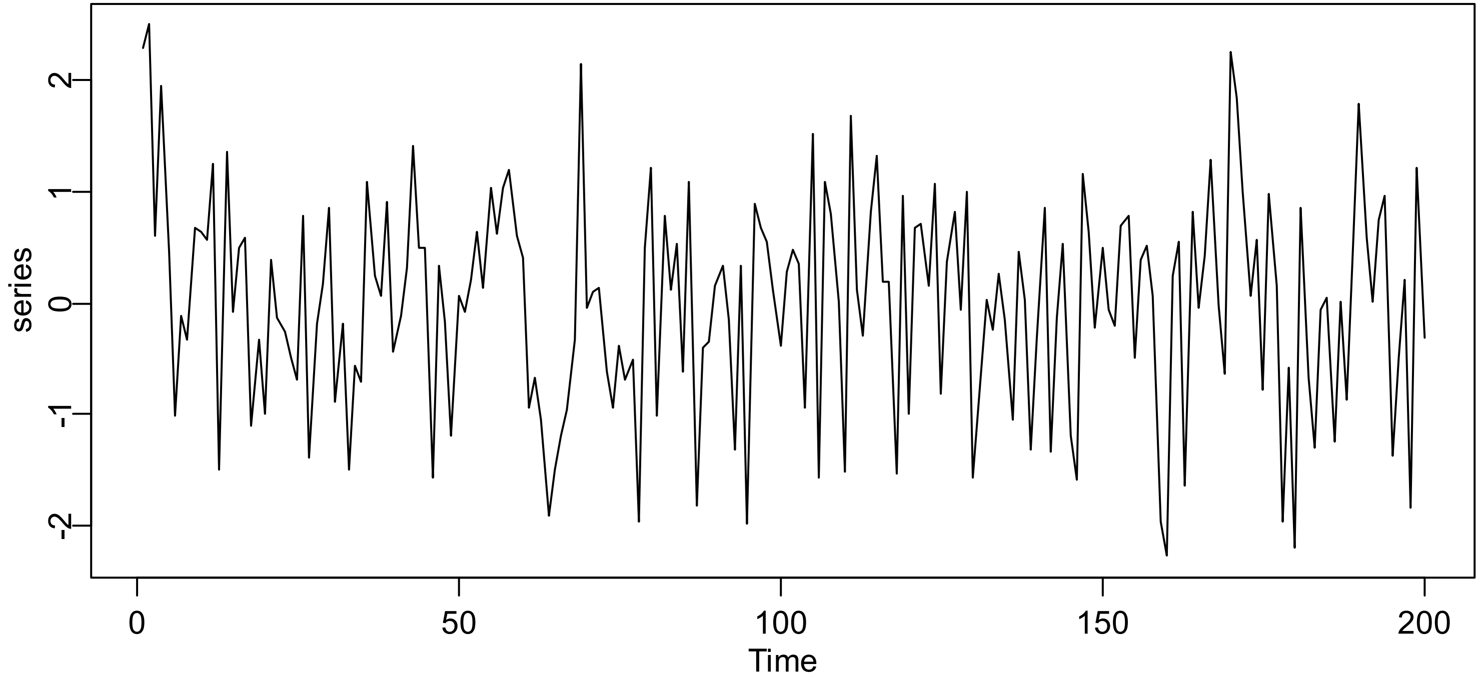
Residuals of Model 1: AR(4)

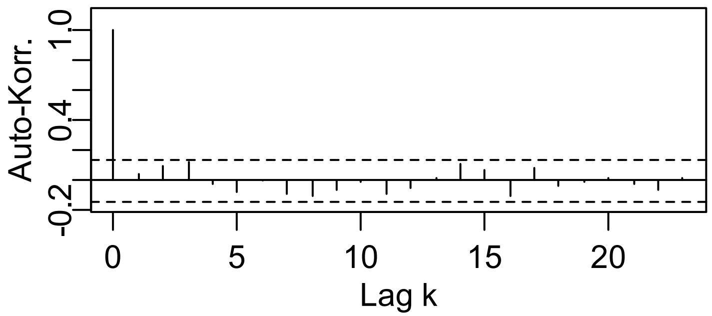


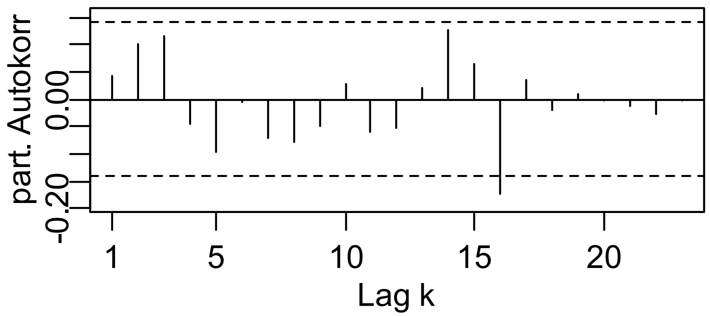
Model 2: MA(3)

```
> fit2

Call: arima(x = my.ts, order = c(0, 0, 3))

Coefficients:

         ma1     ma2     ma3  intercept
      1.5711  1.0056  0.3057     0.6359
s.e.  0.0662  0.0966  0.0615     0.2604

sigma^2=0.9098,  log likelihood=-275.64,  aic=561.29
```

Residuals of Model 2: MA(3)



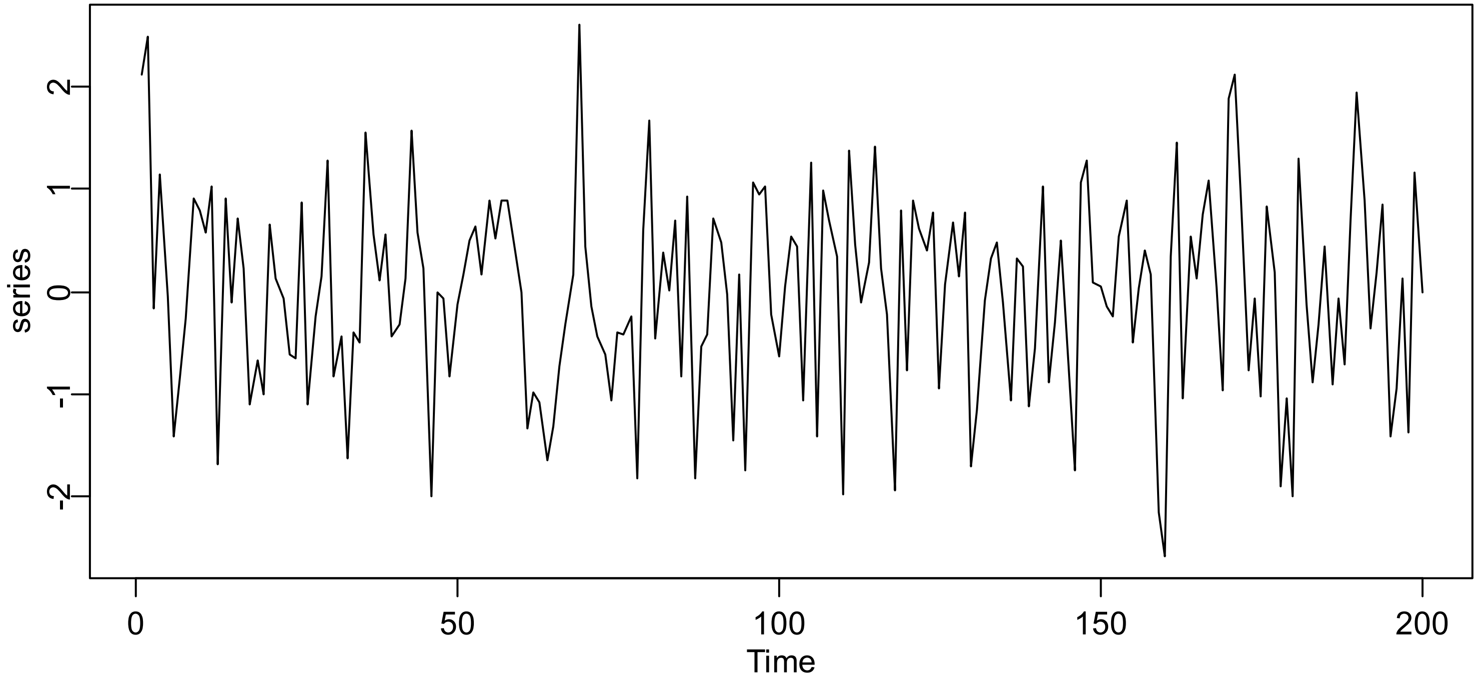


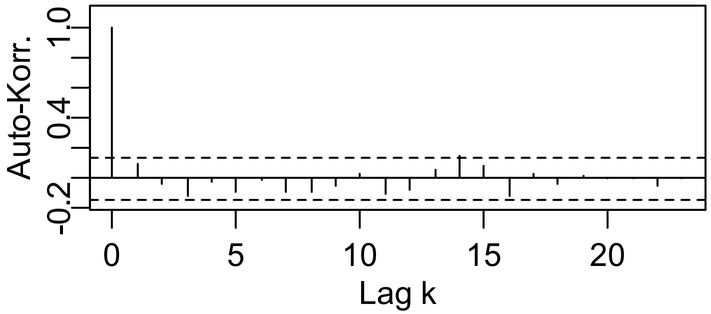


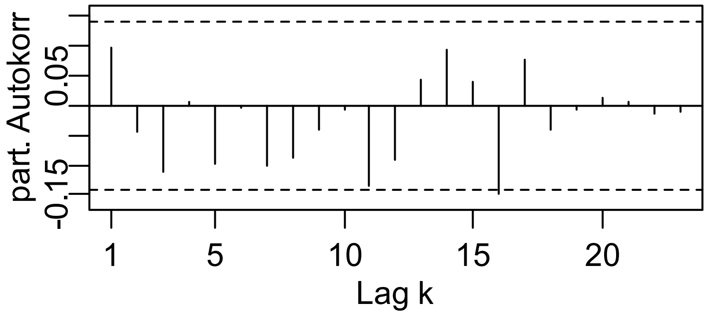
Model 3: ARMA(1,1)

```
> fit3

Call: arima(x = my.ts, order = c(1, 0, 1))

Coefficients:

         ar1     ma1  intercept
      0.6965  0.7981     0.6674
s.e.  0.0521  0.0400     0.3945

sigma^2=0.9107, log likelihood=-275.72,  aic=559.43
```

Residuals of Model 3: ARMA(1,1)



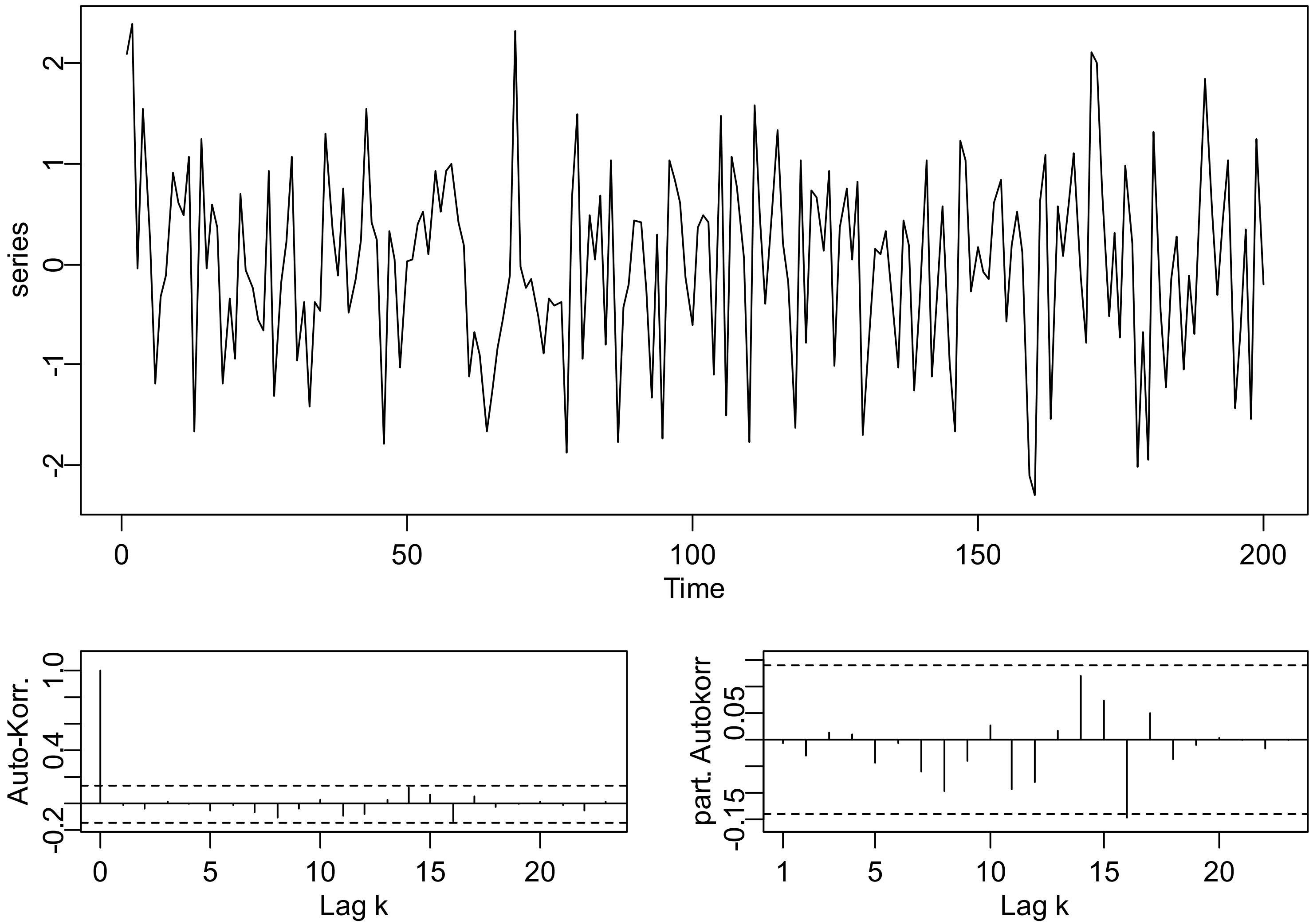




Model 4: ARMA(2,1)

```
> fit4

Call: arima(x = my.ts, order = c(2, 0, 1))

Coefficients:

         ar1      ar2     ma1  intercept
      0.8915  -0.2411  0.7061     0.6420
s.e.  0.0855   0.0856  0.0625     0.3208

sigma^2=0.8772, log likelihood=-272.01, aic=554.02
```

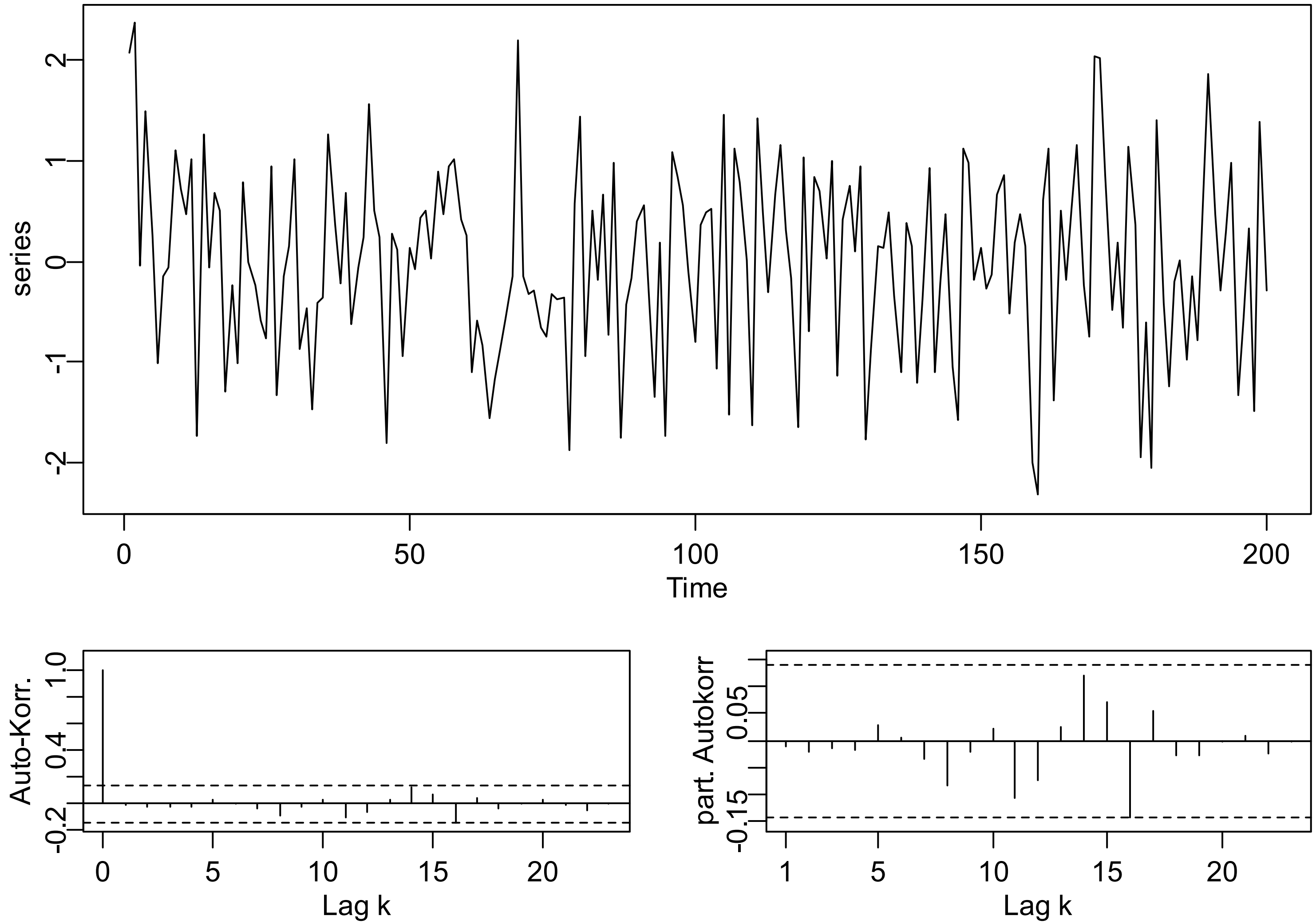
Residuals of Model 4: ARMA(2,1)



Model 5: ARMA(4,1)

```
> fit5

Call: arima(x = my.ts, order = c(4, 0, 1))

Coefficients:
         ar1      ar2     ar3      ar4     ma1  intercept
      1.0253  -0.4693  0.2190  -0.1280  0.5733     0.6312
s.e.  0.1725   0.2658  0.2124   0.1062  0.1653     0.2930

sigma^2=0.8708, log likelihood=-271.3, aic = 556.59
```

Residuals of Model 5: ARMA(4,1)



Summary of the Order Choice Problem

- Regarding ACF/PACF, all 5 models are plausible
  → ARMA(2,1) would be my favorite

- The residuals look fine (i.e. independent) for all 5 models
  → no further evidence for a particular model

- Regarding AIC, the ARMA models do better
  → ARMA(2,1) would be my favorite

- Significance of the coefficients
  → excludes the ARMA(4,1) as the last contender

Best choice: ARMA (2,1)