

Gliederung der Vorlesung

- 1) Einführung und Grundlagen
- 2) Klassische Zeitreihenanalyse
 - a) Trendbestimmung
 - b) Glättung von Zeitreihen durch Filter (lokale Trendbestimmung)
 - c) Saisonbereinigung
 - d) Exponentielles Glätten, Prognosen
- 3) Modellierung von Reihen durch stochastische Prozesse
 - a) Autoregressive (AR) Modelle
 - b) Moving Average (MA) Modelle
 - c) ARMA Modelle
 - d) ARIMA Modelle

Erinnerung

Sei (ε_t) ein White Noise mit $\mathbb{E}[\varepsilon_t] = 0$ und $\text{var}(\varepsilon_t) = \sigma^2$. Der Prozess

$$X_t := \mu + \sum_{n=-\infty}^{\infty} \psi_n \varepsilon_{t-n} \quad \text{wobei} \quad \sum_{n=-\infty}^{\infty} |\psi_n| < \infty$$

heißt ein **linearer Prozess**.

Hängt ein linearer Prozess nicht von der Zukunft ab, d.h. gilt $\psi_n = 0$ für $n < 0$, so dass

$$X_t = \mu + \sum_{n=0}^{\infty} \psi_n \varepsilon_{t-n},$$

so heißt (X_t) **kausal**.

Satz 1.1 Jeder lineare Prozess ist **stationär** mit

$$\mathbb{E}[X_t] = \mu \quad \text{für alle } t \quad \text{und} \quad \gamma(\tau) = \sigma^2 \sum_{n=-\infty}^{\infty} \psi_{n+\tau} \psi_n \quad \text{für alle } \tau \geq 0.$$

AR(1)-Prozess

Ein stochastischer Prozess (X_t) heißt ein **autoregressiver Prozess der Ordnung 1** oder **AR(1)-Prozess**, wenn er sich in der Form

$$X_t = \phi_0 + \phi_1 X_{t-1} + \varepsilon_t$$

darstellen lässt. Dabei ist $(\varepsilon_t) \sim \text{WN}(0, \sigma_\varepsilon^2)$ ein White Noise (**“Zufallschock”, “Innovation”**). ϕ_0 und ϕ_1 sind Konstanten (**Parameter des Prozesses**).

Ist ein AR(1)-Prozess **stationär**, so muss gelten $\mu = \mathbb{E}[X_t] = \frac{\phi_0}{1-\phi_1}$ und

$$X_t - \mu = \phi_1(X_{t-1} - \mu) + \varepsilon_t \quad (4)$$

Daher wird oft O.B.d.A. $\mu = 0$ vorausgesetzt und die Gleichung

$$X_t = \phi_1 X_{t-1} + \varepsilon_t \quad (5)$$

zur Definition eines AR(1)-Prozesses genutzt.

AR(1)-Prozess

Ist ein AR(1)-Prozess **stationär**, so muss gelten:

$$\mu = \mathbb{E}[X_t] = \frac{\phi_0}{1 - \phi_1} \quad \text{und} \quad \sigma^2 = \text{var}(X_t) = \frac{\sigma_\varepsilon^2}{1 - \phi_1^2}$$

– geht nur wenn $|\phi_1| < 1$. In dem Fall erhält man durch Iteration von (4)

$$X_t - \mu = \varepsilon_t + \phi_1 \varepsilon_{t-1} + \phi_1^2 \varepsilon_{t-2} + \cdots = \sum_{n=0}^{\infty} \phi_1^n \varepsilon_{t-n},$$

d.h. (X_t) ist ein **kausaler linearer Prozess**.

Umgekehrt, ist $|\phi_1| < 1$, so ist der **kausale lineare Prozess**

$$X_t := \mu + \sum_{n=0}^{\infty} \phi_1^n \varepsilon_{t-n}$$

die einzige stationäre Lösung von (4).

AR(1)-Prozess

- Für $\phi_1 = 1$ erhalten wir Random Walk \rightarrow **nicht stationär**.
- Auch für $\phi_1 = -1$ hat (4) keine stationäre Lösung.
- AR(1)-Prozesse mit $|\phi_1| > 1$ sind **stationär** aber **nicht kausal**. Man kann sie allerdings durch AR(1)-Modelle mit $\tilde{\phi}_1 = \frac{1}{\phi_1}$ und anderem White Noise ($\tilde{\varepsilon}_t$) darstellen, so dass sie wie AR(1)-Prozesse mit $|\phi_1| < 1$ angesehen werden können.

In Folgendem betrachten wir also nur den Fall $|\phi_1| < 1$.

AR(1)-Prozess

Die Gleichung (5) kann auch mit dem **Backshift-Operator** geschrieben werden:

$$(1 - \phi_1 B)X_t = \varepsilon_t.$$

Dabei benutzen wir das **Lag-Polynom**

$$\phi(B) := 1 - \phi_1 B.$$

Ersetzt man B durch eine Variable z , so erhält man die **charakteristische Gleichung** eines AR(1)- Prozesses:

$$1 - \phi_1 z = 0 \quad \text{hier mit der einzigen Lösung} \quad z = \frac{1}{\phi_1}.$$

Die Stationaritätsbedingung $|\phi_1| < 1$ ist also äquivalent zu der Forderung, dass die Lösung der charakteristischen Gleichung z betragsmäßig größer als 1 sein soll.

AR(1)-Prozess

Die **Autokovarianzfunktion** eines AR(1)-Prozesses erhält man durch die rekursive Gleichung

$$\gamma(\tau) = \phi_1 \gamma(\tau - 1) \quad \text{mit} \quad \gamma(0) = \sigma^2 = \frac{\sigma_\varepsilon^2}{1 - \phi_1^2},$$

es gilt also

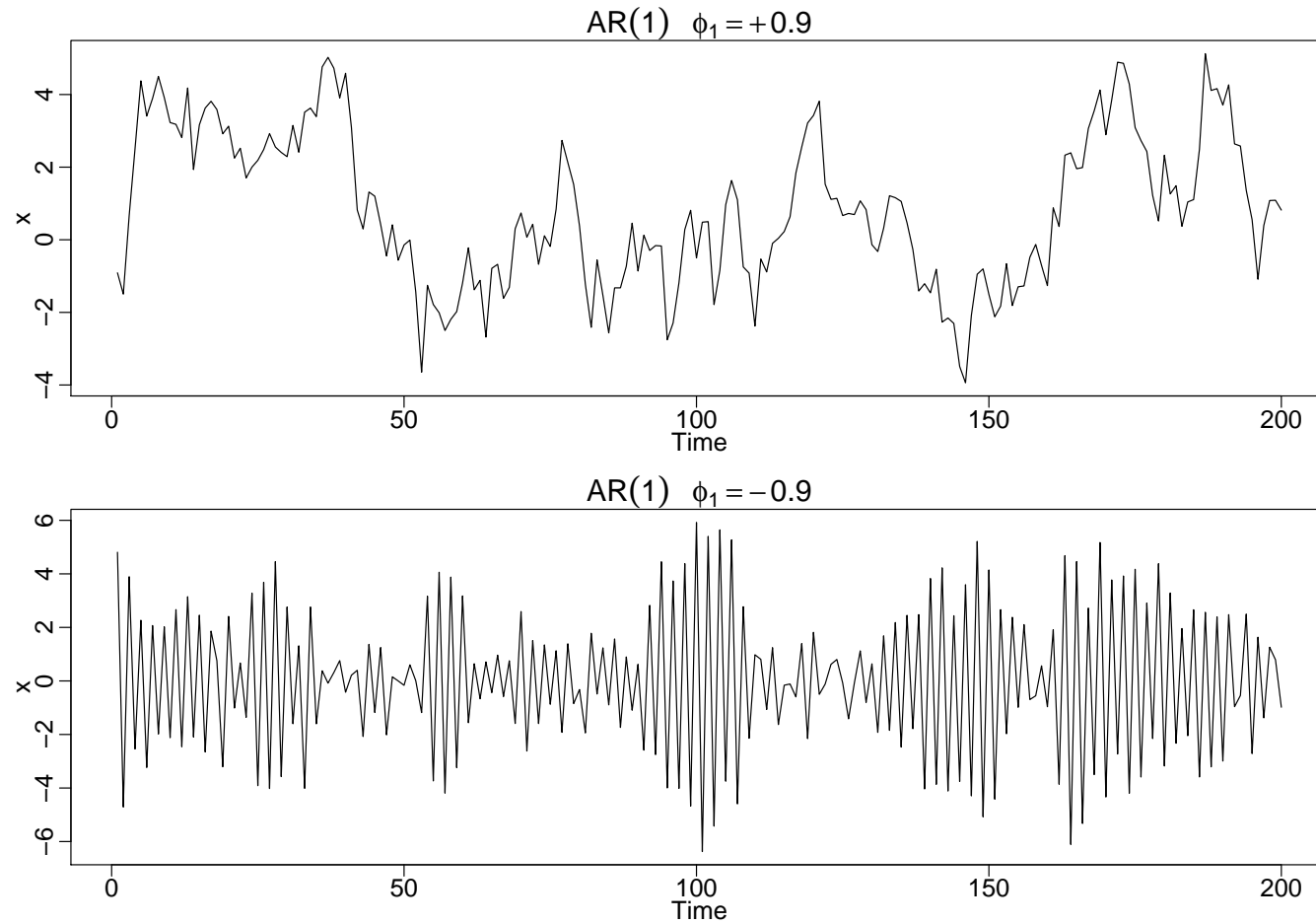
$$\gamma(\tau) = \phi_1^\tau \frac{\sigma_\varepsilon^2}{1 - \phi_1^2} \quad \text{und} \quad \rho(\tau) = \phi_1^\tau, \quad \tau \in \mathbb{N}.$$

D.h. die Autokorrelationen klingen exponentiell ab, monoton falls $\phi_1 > 0$ und alternierend falls $\phi_1 < 0$.

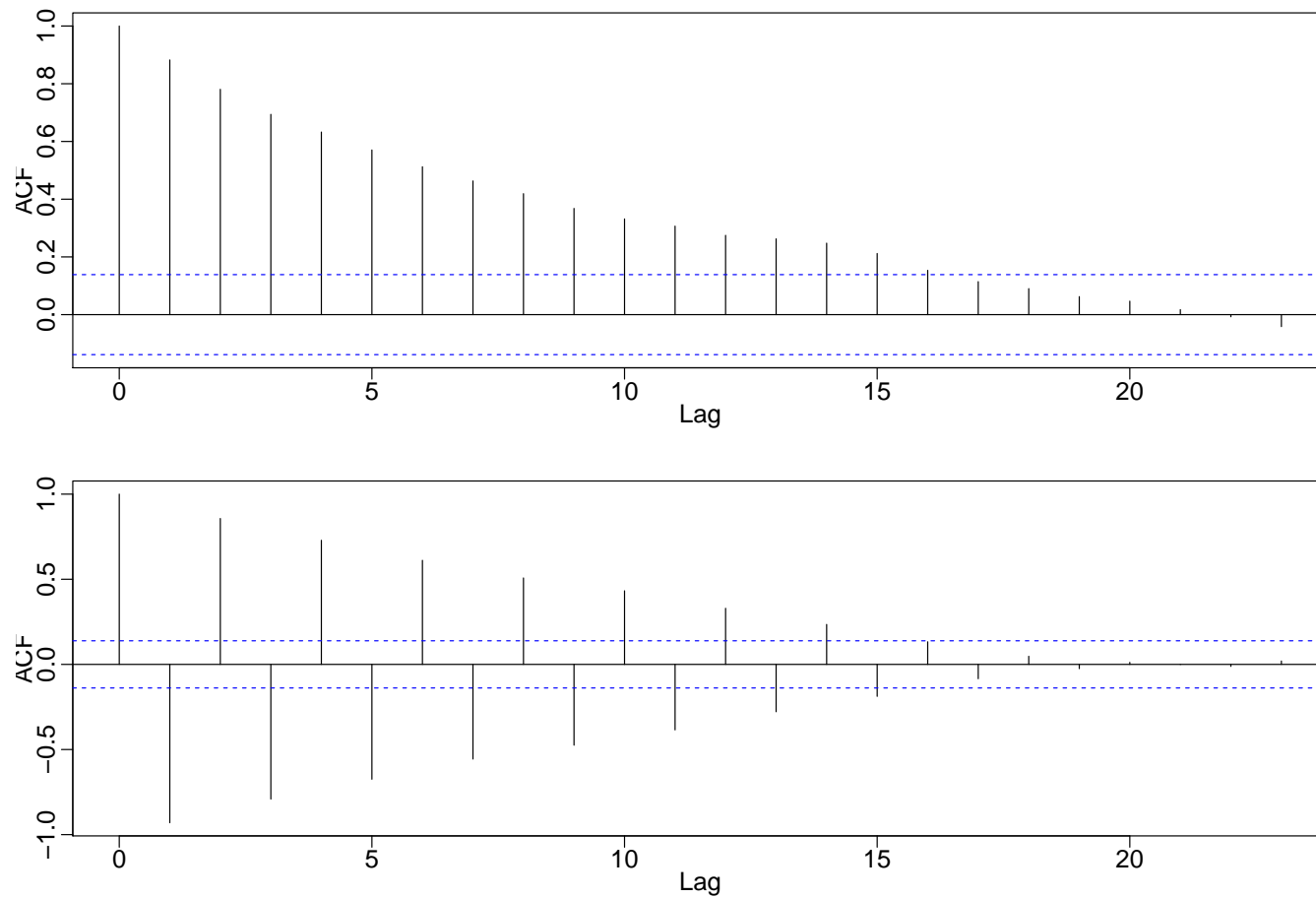
Man kann die Autokorrelationsfunktion auch mit dem Lag-Polynom ausdrücken:

$$(1 - \phi_1 B)\rho(\tau) = 0.$$

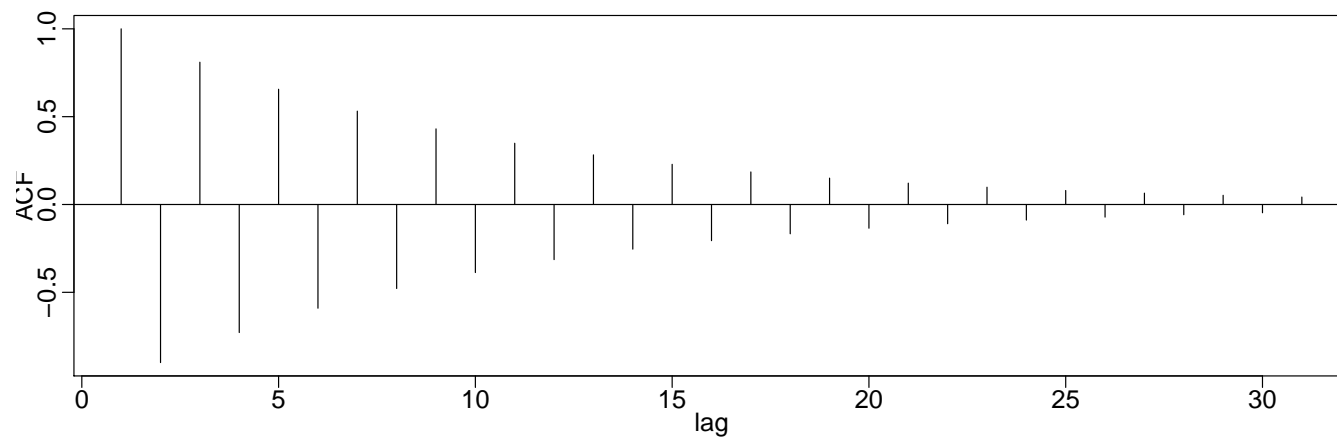
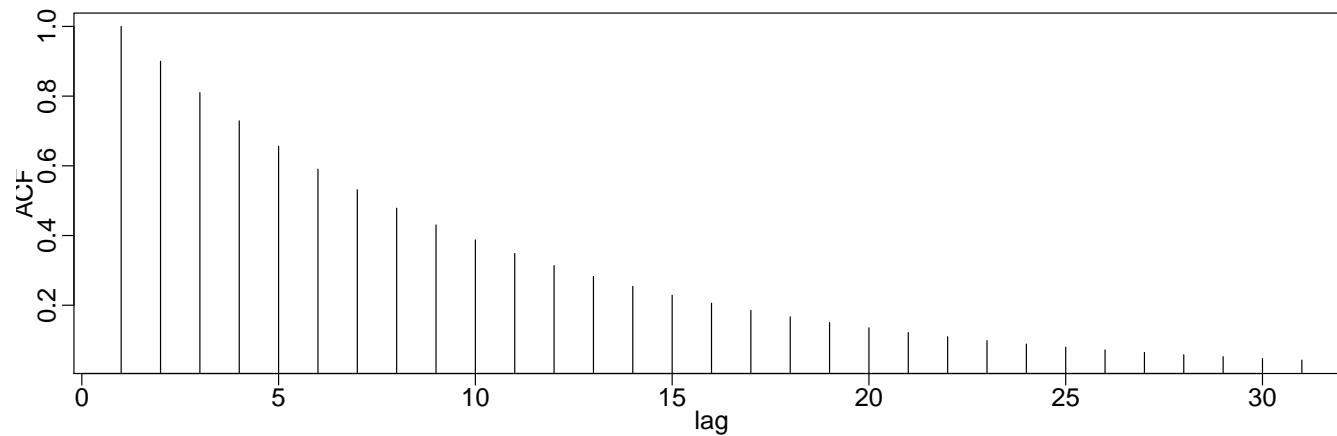
AR(1)-Prozess



AR(1)-Prozess



AR(1)-Prozess



AR(1)-Prozess

```
#Vergleich
par(mfrow=c(2,1), mar=c(3,2,1,0)+.5, mgp=c(1.6,.6,0))
x1=arima.sim(list(order=c(1,0,0), ar=.9), n=200)
x2=arima.sim(list(order=c(1,0,0), ar=-.9), n=200)
plot(x1, ylab="x", main=(expression(AR(1)~~~phi[1]==+.9)))
plot(x2, ylab="x", main=(expression(AR(1)~~~phi[1]==-.9)))
acf(x1)
acf(x2)

ACF = ARMAacf(ar=.9, ma=0, 30)
plot(ACF, type="h", xlab="lag")
abline(h=0)

ACF = ARMAacf(ar=-.9, ma=0, 30)
plot(ACF, type="h", xlab="lag")
abline(h=0)
```

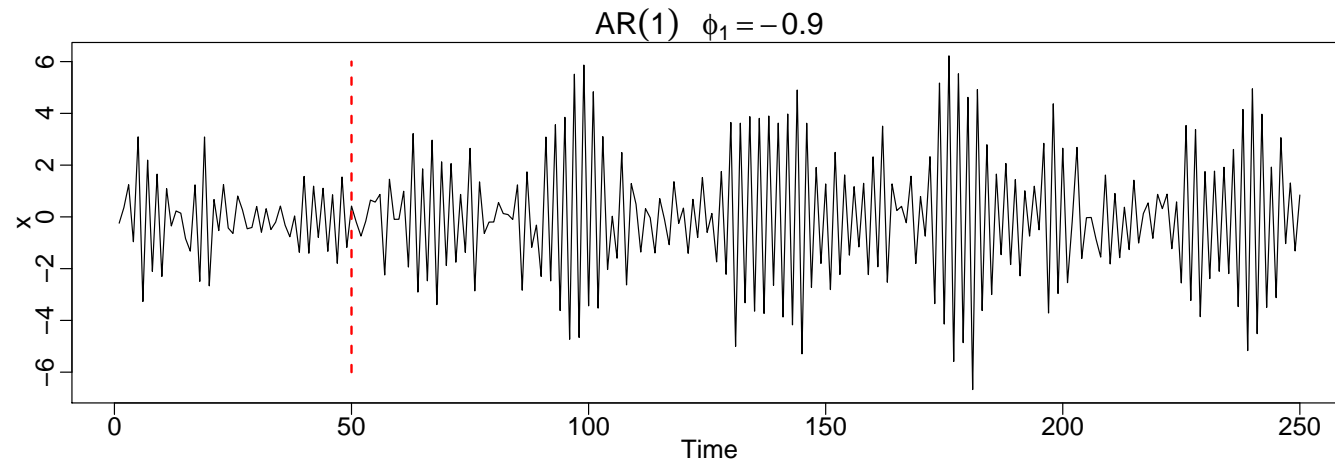
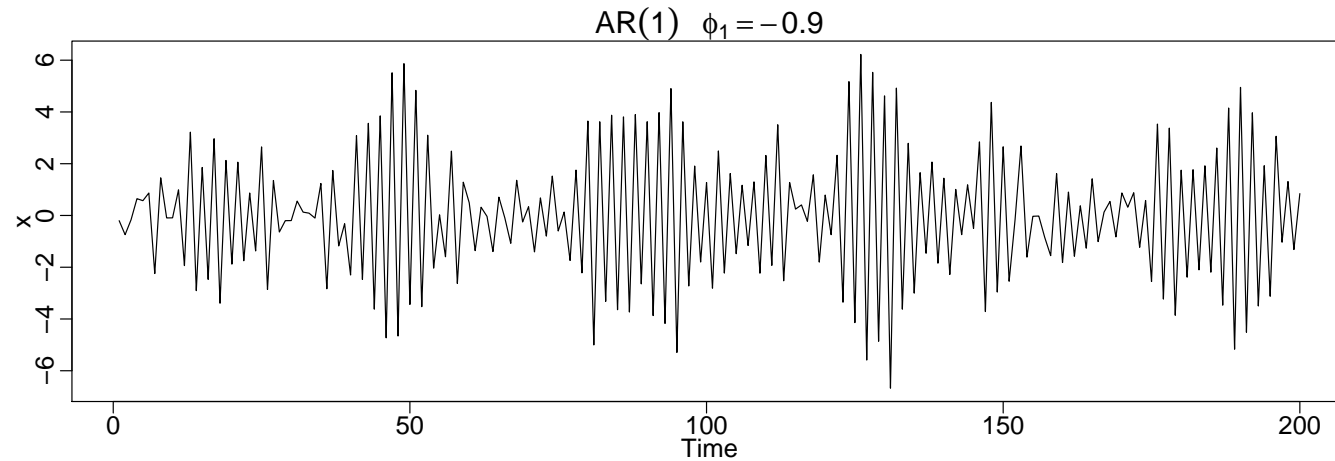
AR(1)-Prozess

```
#AR(1)-Simulation
w = rnorm(250,0,1) # 50 extra zum Einschwingen
x = filter(w, filter=c(-.9), method="recursive")[-(1:50)]
plot.ts(x, main=(expression(AR(1)~~~phi[1]==-.9)))

#Einschwingen
x = filter(w, filter=c(-.9), method="recursive")[-(1:50)]
plot.ts(x, main=(expression(AR(1)~~~phi[1]==-.9)))
lines(c(50,50), c(-6,6), lty=2, lwd=2, col=2)

#oder
n=200
x1=w; t=3
while(t<(n+50)){t=t+1; x1[t]=sum(-.9*x1[t-1])+w[t]}
plot.ts(x1[-(1:50)], main=(expression(AR(1)~~~phi[1]==-.9)))
#oder
x2 = arima.sim(list(order=c(1,0,0),ar=c(-.9)),n=200)
plot.ts(x2, main=(expression(AR(1)~~~phi[1]==-.9)))
```

AR(1)-Prozess



AR(2)-Prozess

Ein autoregressiver Prozess der Ordnung 2 oder ein AR(2)-Prozess ist von der Form

$$X_t = \phi_0 + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \varepsilon_t,$$

mit Parametern ϕ_0, ϕ_1 und ϕ_2 und (ε_t) wie gehabt.

Ist ein AR(2)-Prozess stationär, so muss gelten $\mu = \mathbb{E}[X_t] = \frac{\phi_0}{1-\phi_1-\phi_2}$ und somit

$$(X_t - \mu) - \phi_1(X_{t-1} - \mu) - \phi_2(X_{t-2} - \mu) = \varepsilon_t$$

mit dem Lag-Polynom

$$\phi(B) := 1 - \phi_1 B - \phi_2 B^2$$

und der charakteristischen Gleichung

$$1 - \phi_1 z - \phi_2 z^2 = 0.$$

AR(2)-Prozess

Auch für einen AR(2)-Prozess gilt: (X_t) ist ein **stationärer kausaler linearer** Prozess genau dann, wenn die beiden Lösungen z_1, z_2 der charakteristischen Gleichung

$$1 - \phi_1 z - \phi_2 z^2 = 0$$

außerhalb des Einheitskreises liegen (in der **komplexen Ebene**). Für stationär reicht schon, wenn $|z_{1,2}| \neq 1$.

Die Autokovarianzfunktion eines AR(2)-Prozesses erhält man rekursiv durch

$$\gamma(\tau) = \phi_1 \gamma(\tau - 1) + \phi_2 \gamma(\tau - 2) \quad \text{für } \tau > 0 \quad \text{und}$$

$$\gamma(0) = \sigma^2 = \phi_1 \gamma(1) + \phi_2 \gamma(2) + \sigma_\varepsilon^2.$$

AR(2)-Prozess

Die Autokorrelationen erfüllen die **homogene Differenzengleichung**

$$\rho(\tau) = \phi_1 \rho(\tau - 1) + \phi_2 \rho(\tau - 2), \quad \tau > 0. \quad (6)$$

Wegen $\rho(0) = 1$ und $\rho(1) = \rho(-1)$ erhält man

$$\begin{pmatrix} \rho(1) \\ \rho(2) \end{pmatrix} = \begin{pmatrix} 1 & \rho(1) \\ \rho(1) & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \end{pmatrix} \quad (7)$$

– diese Gleichungen werden als **Yule-Walker-Gleichungen** bezeichnet. Mit den Anfangsbedingungen

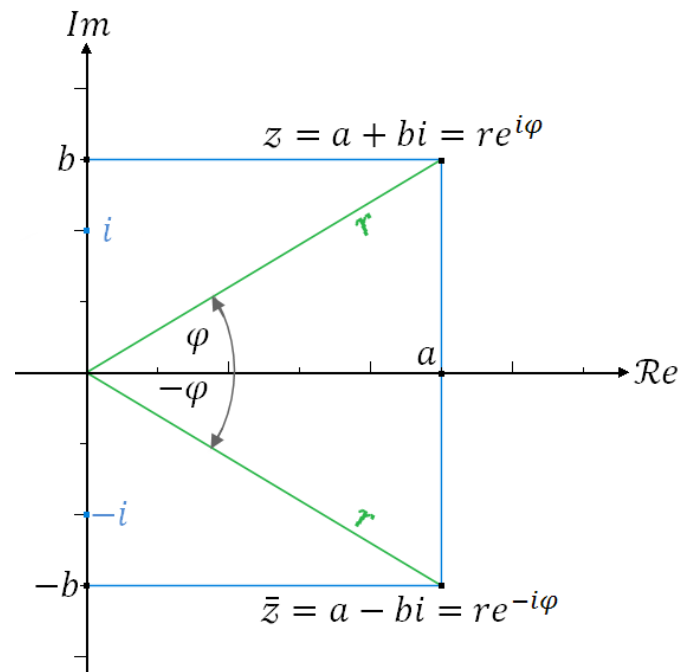
$$\rho(0) = 1, \quad \rho(1) = \frac{\phi_1}{1 - \phi_2} \quad (8)$$

können alle weiteren Korrelationen berechnet werden.

Zur Erinnerung

$$z = a + bi = re^{i\varphi} = r(\cos(\varphi) + i \sin(\varphi)), \quad \bar{z} = a - bi$$

$$r = |z| = \sqrt{a^2 + b^2} = \sqrt{z \cdot \bar{z}}, \quad \varphi = \begin{cases} \arccos\left(\frac{a}{r}\right); & b \geq 0 \\ -\arccos\left(\frac{a}{r}\right); & \text{sonst.} \end{cases}$$



AR(2)-Prozess

Der Verlauf der Autokorrelationsfunktion hängt von den Lösungen z_1 und z_2 der charakteristischen Gleichung ab:

- Sind $z_1 \neq z_2$ beide reell, so gilt $\rho(\tau) = C_1 \left(\frac{1}{z_1}\right)^\tau + C_2 \left(\frac{1}{z_2}\right)^\tau$.
- Ist $z_1 = z_2$ (also reell), so gilt $\rho(\tau) = \left(\frac{1}{z_1}\right)^\tau (1 + C_2 \tau)$
- Sind z_1 und z_2 konjugiert komplex, $z_1 = |z_1|e^{i\theta}$, so folgt $\rho(\tau) = \frac{1}{|z_1|^\tau} (\cos(\tau\theta) + C_2 \sin(\tau\theta))$

Die Konstanten C_1, C_2 ergeben sich aus der Anfangsbedingung (8). Die Stationaritätsbedingung $|z_1| > 1, |z_2| > 1$ gewährleistet, dass die Autokorrelationen exponentiell abklingen. Bei komplexen Wurzeln entwickeln AR(2)-Prozesse ein zyklisches Verhalten mit der Periode

$$p = \frac{2\pi}{\theta} = \frac{2\pi}{\arccos(\phi_1/2\sqrt{-\phi_2})}.$$

AR(2)-Prozess: Beispiele

- $X_t = 1.3X_{t-1} - 0.4X_{t-2} + \varepsilon_t \rightarrow$ Charakteristische Gleichung

$$1 - 1.3z + 0.4z^2 = 0$$

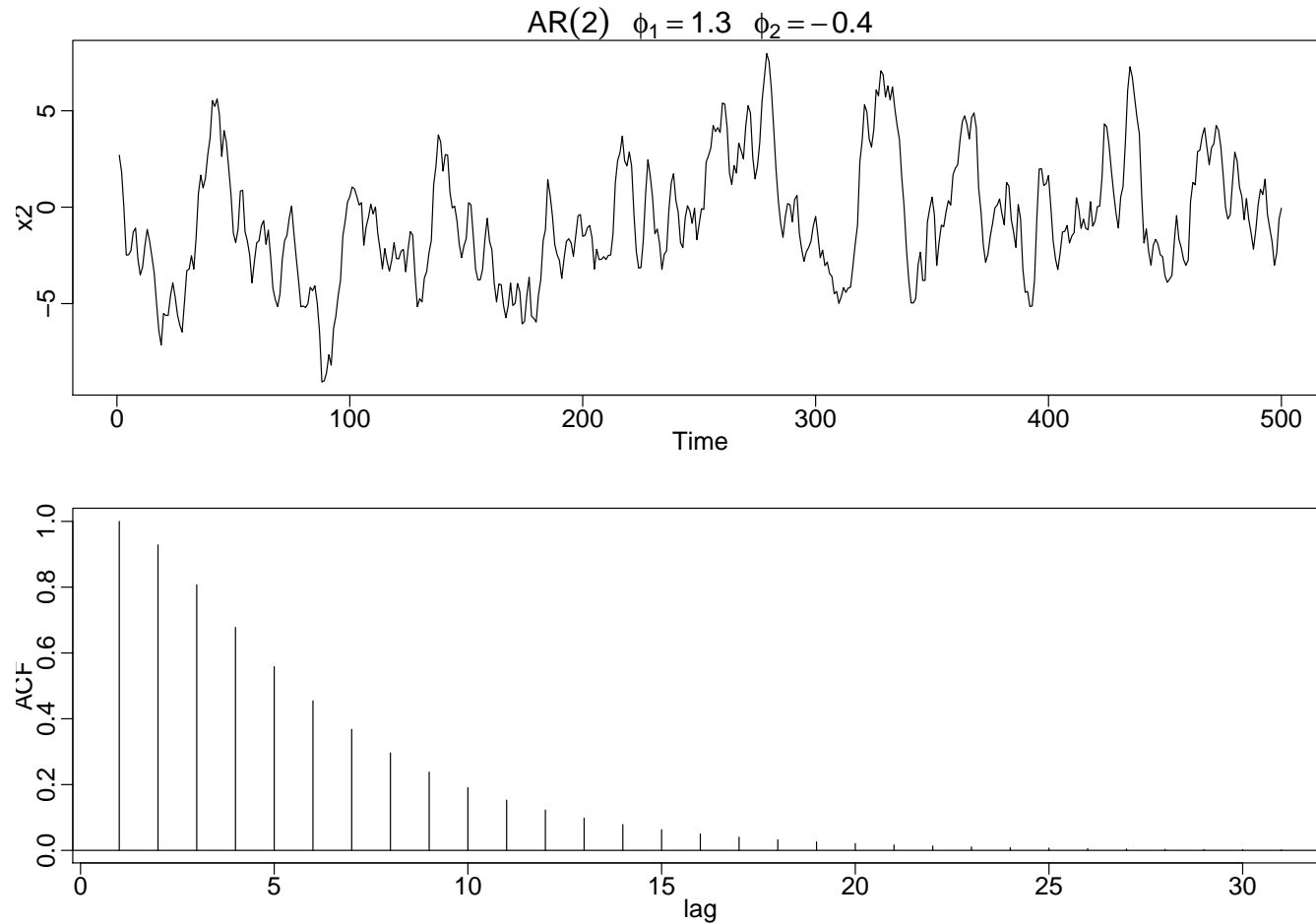
hat zwei reelle Lösungen $z_1 = 1.25$, $z_2 = 2$. Damit folgt: $C_1 = 1.43$,
 $C_2 = -.43$ und $\rho(\tau) = 1.43 \frac{1}{1.25^\tau} - 0.43 \frac{1}{2^\tau}$

- $X_t = X_{t-1} - 0.25X_{t-2} + \varepsilon_t \rightarrow$ Charakteristische Gleichung

$$1 - z + 0.25z^2 = 0$$

hat eine doppelte Lösung $z_1 = 2$. Damit folgt: $C_2 = 0.6$ und
 $\rho(\tau) = \frac{1}{2^\tau} (1 + 0.6\tau)$

AR(2)-Prozess: Beispiele

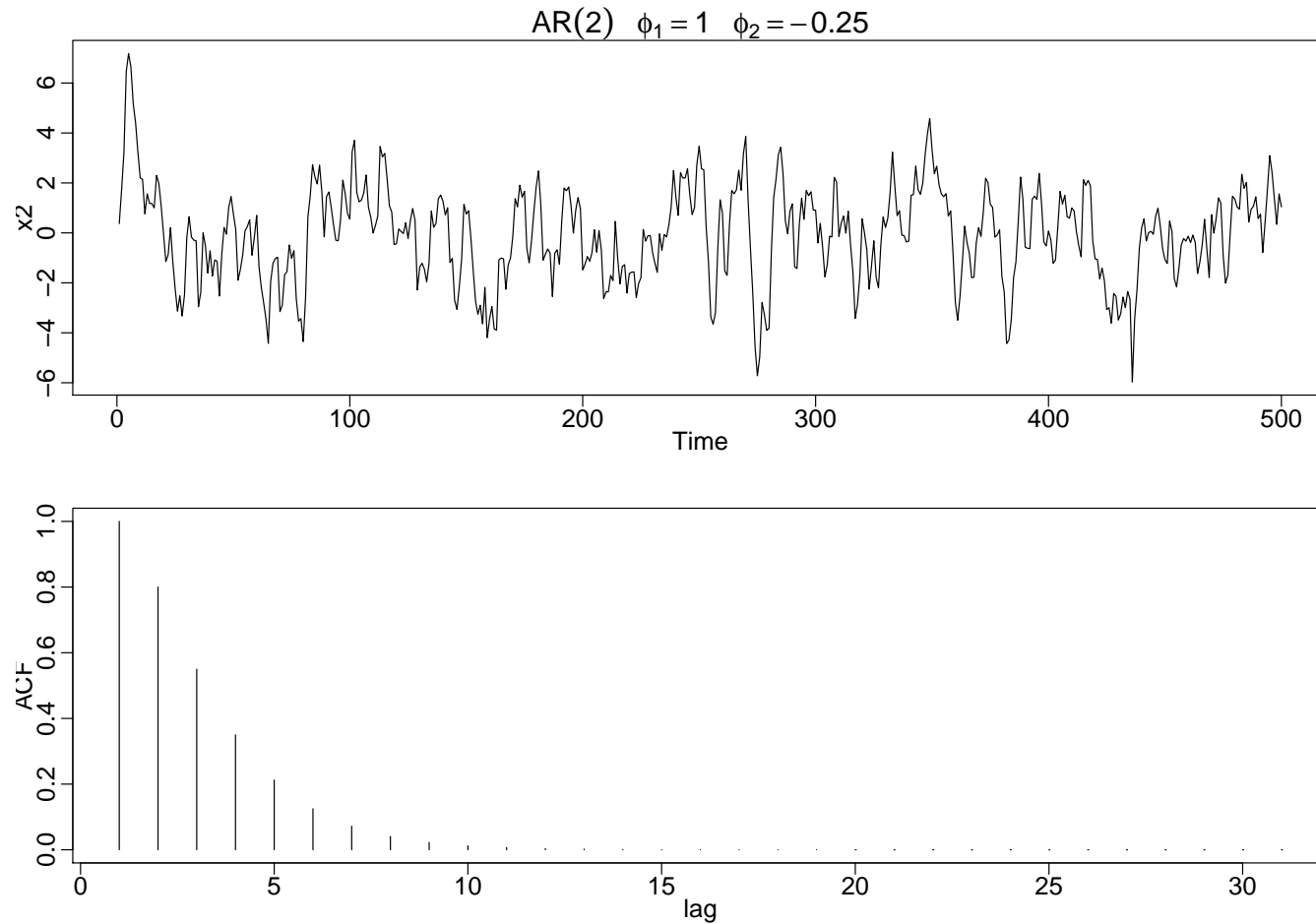


AR(2)-Prozess: Beispiele

```
z = c(1,-1.3,.4)
(a = polyroot(z)) # Nullstellen
K <- rbind(c(1,1), c(1/a[1], 1/a[2]))
b <- c(1, 1.3/(1+.4))
solve(K, b)

w = rnorm(550,0,1) # 50 extra zum Einschwingen
x = filter(w, filter=c(1.3,-.4), method="recursive")[-(1:50)]
plot.ts(x, main=(expression(AR(2)~~~phi[1]==1.3~~~phi[2]==-.4)))
#oder
n=500
x1=w; t=3
while(t<(n+50)){t=t+1; x1[t]=sum(c(1.3,-.4)*x1[t-c(1,2)])+w[t]}
plot.ts(x1, main=(expression(AR(2)~~~phi[1]==1.3~~~phi[2]==-.4)))
#oder
par(mfrow=c(2,1), mar=c(3,2,1,0)+.5, mgp=c(1.6,.6,0))
x2 = arima.sim(list(order=c(2,0,0),ar=c(1.3,-.4)),n=500) # AR(2)
plot(x2, main=(expression(AR(2)~~~phi[1]==1.3~~~phi[2]==-.4)))
ACF = ARMAacf(ar=c(1.3,-.4), ma=0, 30)
plot(ACF, type="h", xlab="lag")
abline(h=0)
```

AR(2)-Prozess: Beispiele



AR(2)-Prozess: Beispiele

- $X_t = 1.5X_{t-1} - 0.75X_{t-2} + \varepsilon_t \rightarrow$ Charakteristische Gleichung

$$1 - 1.5z + 0.75z^2 = 0$$

hat zwei komplexe Lösungen $z_{1,2} = 1 \pm \frac{i}{\sqrt{3}}$. Es ist $|z_{1,2}| = 1.15$ und in Polarkoordinaten

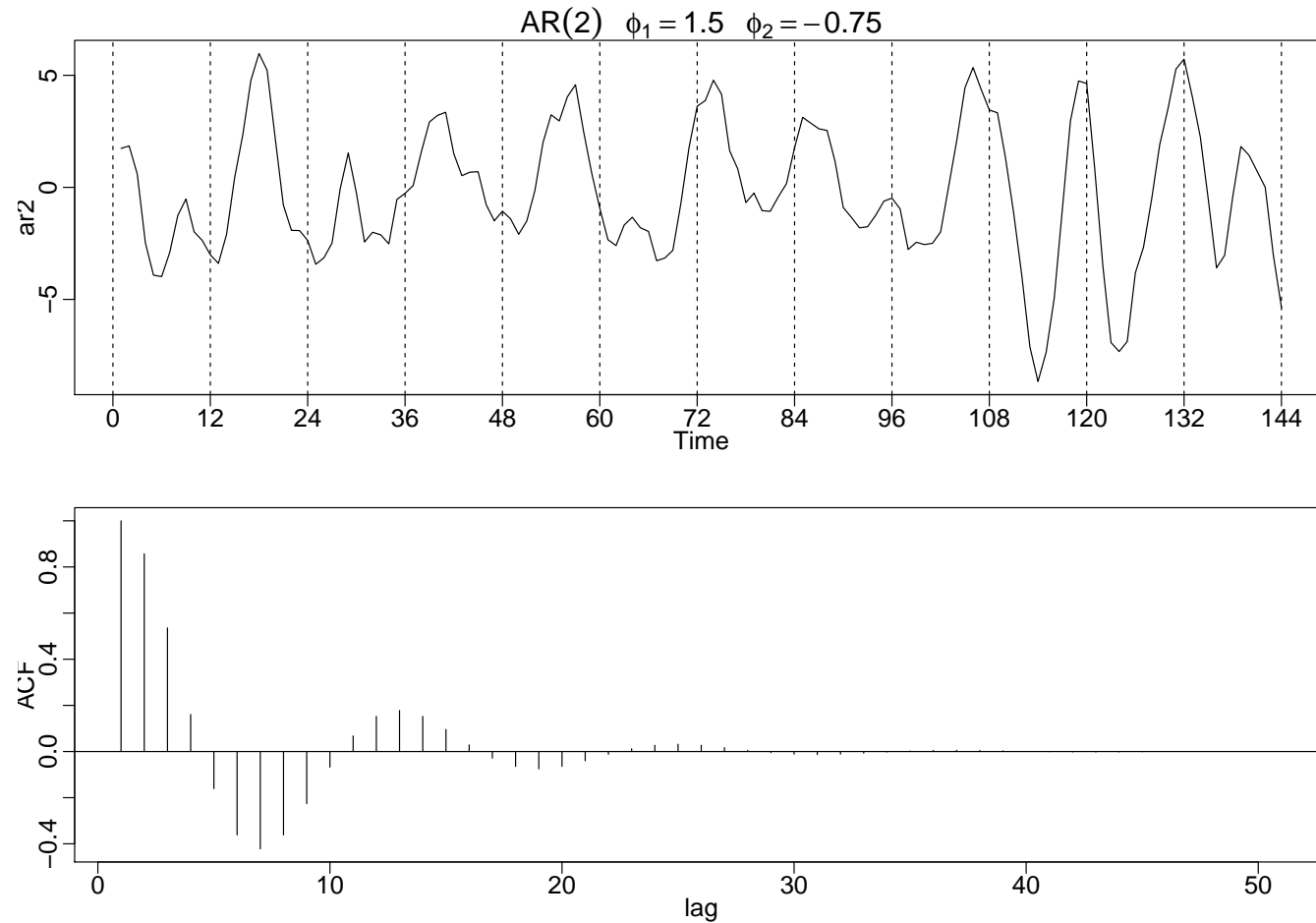
$$z_{1,2} = 1.15e^{\pm i\theta} \quad \text{mit} \quad \theta = \arctan(1/\sqrt{3}) = 2\pi/12 = \pi/6.$$

Damit folgt: $C_2 = 0.25$ und $\rho(\tau) = \frac{1}{1.15^\tau} \left(\cos(\tau \frac{\pi}{6}) + 0.25 \sin(\tau \frac{\pi}{6}) \right)$

\rightarrow Zyklisches Verhalten mit der Periode

$$p = \frac{2\pi}{\theta} = 12.$$

AR(2)-Prozess: Beispiele



AR(2)-Prozess: Beispiele

```
#AR(2) komplexe Wurzeln
z = c(1,-1.5,.75)
polyroot(z)
a = polyroot(z)[1]
(r=Mod(a))
(theta = Arg(a))# theta
(ptheta = Arg(a)/(2*pi))# theta in Radian
1/pttheta #Periode
K <- sin(theta)/r
b <- 1.5/(1+.75)-cos(theta)/r
solve(K, b)
K <- rbind(c(1/a[1], 1/a[2]), c(1/a[1]^2, 2/a[2]^2))
b <- c(1/(1+.25), 1^2/(1+.25)-.25)
(c2=solve(K, b))

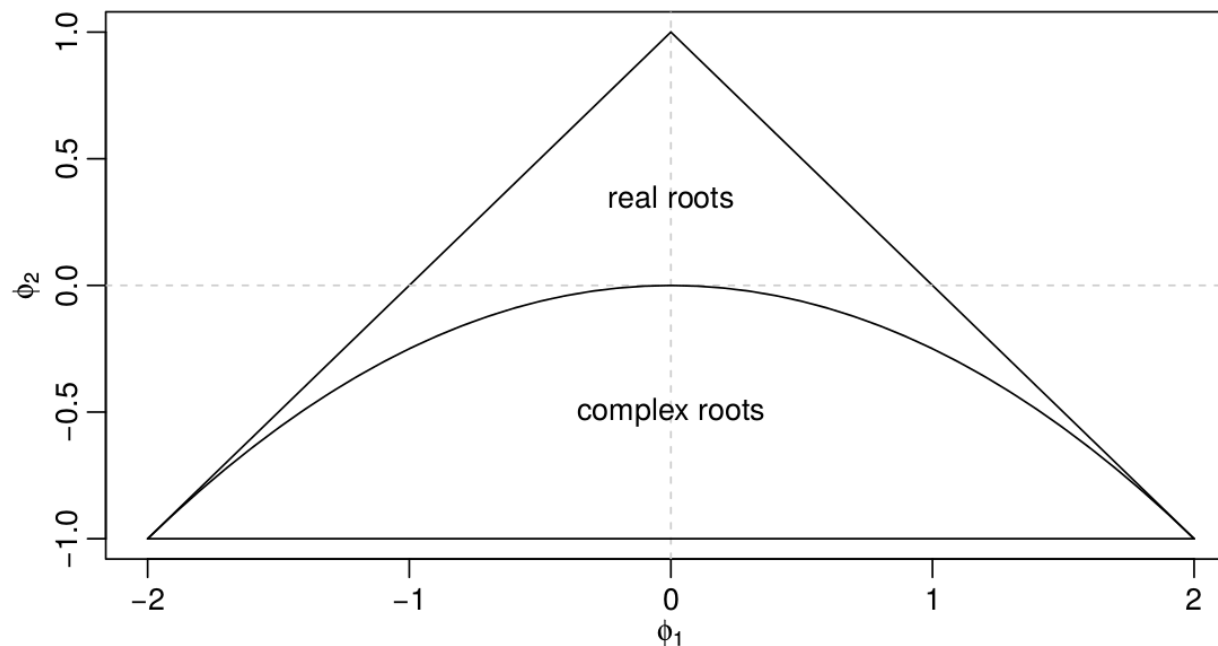
par(mfrow=c(2,1), mar=c(3,2,1,0)+.5, mgp=c(1.6,.6,0))
set.seed(8675309)
ar2 = arima.sim(list(order=c(2,0,0), ar=c(1.5,-.75)), n = 144)
plot(ar2, axes=FALSE, xlab="Time", main=(expression(AR(2)~~~phi[1]==1.5~~~phi[2]==-.75)))
axis(2); axis(1, at=seq(0,144,by=12)); box()
abline(v=seq(0,144,by=12), lty=2)

ACF = ARMAacf(ar=c(1.5,-.75), ma=0, 50)
plot(ACF, type="h", xlab="lag")
abline(h=0)
```

AR(2)-Prozess

Wegen $\phi_1 = \frac{1}{z_1} + \frac{1}{z_2}$ und $\phi_2 = -\frac{1}{z_1 z_2}$ impliziert die Forderung $|z_1| > 1$, $|z_2| > 1$ folgende Bedingungen an die Koeffizienten ϕ_1, ϕ_2 :

$$\phi_1 + \phi_2 < 1, \quad \phi_2 - \phi_1 < 1 \quad \text{und} \quad |\phi_2| < 1.$$



AR(p)-Prozess

Ein stochastischer Prozess (X_t) heißt ein **autoregressiver Prozess der Ordnung p** oder **AR(p)-Prozess**, wenn er von der Form

$$X_t = \phi_0 + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t$$

ist. Dabei ist $(\varepsilon_t) \sim \text{WN}(0, \sigma_\varepsilon^2)$ ein White Noise.

Ein AR(p)-Prozess ist **kausal und stationär**, wenn alle Nullstellen der charakteristischen Gleichung

$$1 - \phi_1 z - \cdots - \phi_p z^p = 0$$

außerhalb des Einheitskreises liegen. In dem Fall gilt

$$X_t - \mu = \phi_1 (X_{t-1} - \mu) + \cdots + \phi_p (X_{t-p} - \mu) + \varepsilon_t$$

mit $\mu = \mathbb{E}[X_t] = \frac{\phi_0}{1 - \phi_1 - \cdots - \phi_p}$, bzw., unter der Annahme $\mu = 0$

$$X_t = \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \varepsilon_t. \quad (9)$$

AR(p)-Prozess

Für die Autokovarianzen bzw. die Autokorrelationen eines AR(p)-Prozesses gelten die **Yule-Walker Gleichungen**

$$\begin{aligned}\gamma(\tau) &= \phi_1 \gamma(\tau - 1) + \phi_2 \gamma(\tau - 2) + \cdots + \phi_p \gamma(\tau - p), & \tau > 0 \\ \gamma(0) &= \phi_1 \gamma(1) + \phi_2 \gamma(2) + \cdots + \phi_p \gamma(p) + \sigma_\varepsilon^2 \\ \rho(\tau) &= \phi_1 \rho(\tau - 1) + \phi_2 \rho(\tau - 2) + \cdots + \phi_p \rho(\tau - p), & \tau > 0.\end{aligned}\tag{10}$$

In der Matrix-Notation $\underline{\gamma}_p = \Gamma_p \underline{\phi}$ bzw. $\underline{\rho}_p = R_p \underline{\phi}$, d.h.

$$\begin{pmatrix} \rho(1) \\ \rho(2) \\ \vdots \\ \rho(p) \end{pmatrix} = \begin{pmatrix} 1 & \rho(1) & \cdots & \rho(p-1) \\ \rho(1) & 1 & \cdots & \rho(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(p-1) & \rho(p-2) & \cdots & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{pmatrix}$$

Schätzen von AR-Parametern

Einige Methoden zur Schätzung der Parameter ϕ_1, \dots, ϕ_p eines AR(p)-Prozesses sind

- Conditional bzw. Unconditional Least Sum of Squares (CLS bzw. ULS)
- Conditional bzw. Unconditional Maximum Likelihood (CML bzw. UML)
- Burg
- Yule Walker (YW)

Wir illustrieren diese am Beispiel eines zentrierten AR(2)-Prozesses

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \varepsilon_t$$

mit der Realisierung x_1, \dots, x_N .

Schätzen von AR-Parametern: C/ULS

Bei der LS-Methode werden die Schätzer $\hat{\phi}_1$, $\hat{\phi}_2$ so bestimmt, dass

$$\sum_{t=1}^N (x_t - \hat{\phi}_1 x_{t-1} - \hat{\phi}_2 x_{t-2})^2 \stackrel{!}{=} \min$$

Das führt auf das Gleichungssystem

$$\begin{aligned} \hat{\phi}_1 \frac{1}{N} \sum_{t=1}^N x_{t-1}^2 + \hat{\phi}_2 \frac{1}{N} \sum_{t=1}^N x_{t-1} x_{t-2} &= \frac{1}{N} \sum_{t=1}^N x_t x_{t-1} \\ \hat{\phi}_1 \frac{1}{N} \sum_{t=1}^N x_{t-1} x_{t-2} + \hat{\phi}_2 \frac{1}{N} \sum_{t=1}^N x_{t-2}^2 &= \frac{1}{N} \sum_{t=1}^N x_t x_{t-2} \end{aligned}$$

Werden dabei $x_{-1} = x_0 = 0 = \mu$ gesetzt (Summation beginnt dann bei 3), so spricht man von **C**LS. Werden für x_{-1} und x_0 Prognosewerte der Reihe in die Vergangenheit genutzt, so spricht man von **U**LS.

Schätzen von AR-Parametern: C/UML

Bei der ML-Methode wird angenommen, dass die Residuen (ε_t) ein **Gaußscher** White Noise sind. Sei $\theta = (\phi_1, \phi_2, \sigma_\varepsilon^2)$. Der Schätzer $\hat{\theta} := (\hat{\phi}_1, \hat{\phi}_2, \hat{\sigma}_\varepsilon^2)$ maximiert die Log-Likelihoodfunktion

$$L(\theta) = \ln(f_{x_1}(x_1; \theta)) + \ln(f_{x_2|x_1}(x_2|x_1; \theta)) \quad (11)$$

$$\begin{aligned} &+ \sum_{t=3}^N \ln(f_{x_t|x_{t-1}, x_{t-2}}(x_t|x_{t-1}, x_{t-2}; \theta)) \\ &= -\frac{N}{2} \ln(2\pi) - \frac{N}{2} \ln(\sigma_\varepsilon^2) + \frac{1}{2} \ln((1 + \phi_1)^2((1 - \phi_2)^2 - \phi_1^2)) \\ &\quad - \frac{1 + \phi_2}{2\sigma_\varepsilon^2} ((1 - \phi_2)x_1^2 - 2\phi_1 x_1 x_2 + (1 - \phi_2)x_2^2) \\ &\quad - \frac{1}{2\sigma_\varepsilon^2} \sum_{t=3}^N (x_t - \phi_1 x_{t-1} - \phi_2 x_{t-2})^2 \stackrel{!}{=} \max \end{aligned} \quad (12)$$

Schätzen von AR-Parametern: C/UML

- Werden dabei x_1 und x_2 als deterministisch vorausgesetzt, so verschwinden die Terme in (11), und eine Maximierung der Log-Likelihoodfunktion läuft auf eine Minimierung von (12) hinaus. In diesem Fall erhält man (bis auf die Konstante $\frac{1}{N-2}$ bzw. $\frac{1}{N-3}$) dieselben Schätzer wie bei Conditional Least Squares. Man spricht dann von CML.
- Wird die exakte Log-Likelihoodfunktion minimiert (das führt auf ein nichtlineares Gleichungssystem), so spricht man von UML.

Schätzen von AR-Parametern: Burg

Basierend auf der Tatsache, dass ein stationärer linearer Prozess **zeitreversibel** ist, d.h. die Kovarianzen von

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} \quad \text{und} \quad Y_t = \phi_1 Y_{t+1} + \phi_2 Y_{t+2}$$

sind gleich, wird die RSS vorwärts und rückwärts minimiert:

$$\sum_{t=3}^N (x_t - \hat{\phi}_1 x_{t-1} - \hat{\phi}_2 x_{t-2})^2 + \sum_{t=1}^{N-2} (x_t - \hat{\phi}_1 x_{t+1} - \hat{\phi}_2 x_{t+2})^2 \stackrel{!}{=} \min$$

Schätzen von AR-Parametern: Yule-Walker

Es werden die empirischen Autokovarianzen ($\hat{\gamma}$) (bzw. Autokorrelationen ($\hat{\rho}$)) in die Yule-Walker Gleichungen (10) eingesetzt und nach $\phi_1, \phi_2, \sigma_\varepsilon^2$ aufgelöst. Für einen AR(2)-Prozess ergibt das die Schätzer

$$\hat{\phi}_1 = \hat{\rho}(1) \frac{1 - \hat{\rho}(2)}{1 - \hat{\rho}(1)^2}, \quad \hat{\phi}_2 = 1 - \frac{1 - \hat{\rho}(2)}{1 - \hat{\rho}(1)^2},$$
$$\hat{\sigma}_\varepsilon^2 = \hat{\gamma}(0)(1 - \hat{\phi}_1 \rho(1) - \hat{\phi}_2 \rho(2)).$$

Allgemein gilt

$$\underline{\hat{\phi}} = (\hat{\phi}_1, \dots, \hat{\phi}_p)' = \hat{R}_p^{-1} \underline{\hat{\rho}}_p, \quad \hat{\sigma}_\varepsilon^2 = \hat{\gamma}(0)(1 - \underline{\hat{\rho}}_p' \hat{R}_p^{-1} \underline{\hat{\rho}}_p). \quad (13)$$

Schätzen von AR-Parametern

- Für alle oben genannten Schätzer gilt

$$\sqrt{N}(\hat{\phi} - \phi) \overset{\text{in Vert.}}{\rightarrow} N(0, \sigma_{\varepsilon}^2 \Gamma_p^{-1}), \quad \text{d.h.}$$
$$\hat{\phi} - \phi \approx N(0, \frac{\sigma_{\varepsilon}^2}{N} R_p^{-1}) \quad (14)$$

- Dies erlaubt **Signifikanztests** der einzelnen Parameterschätzungen mittels t -Tests. Visuell (für 5%-Konfidenzniveau): Ist $0 \in [\hat{\phi} \pm 1.96 \cdot \text{s.e.}]$? Dann ist der Parameter nicht signifikant. In R mit **“coefest”** bei **“arima”**-Schätzungen, benötigt das Paket **“lmtest”**. Oder mit **“sarima”** schätzen und direkt in **“ttable”** schauen.

Schätzen von AR-Parametern

- Yule-Walker Schätzer sind konsistent und effizient, d.h. es existieren keine Schätzer mit kleinerer asymptotischer Varianz.
- Für kürzere Reihen erhält man i.d. Regel bessere Ergebnisse mit ML.
- Nur die Yule-Walker Methode liefert Schätzwerte, die automatisch zu einem stationären Prozess führen. In R ist es auch bei anderen Methoden default, allerdings **nicht für Subsetmodelle**.

Schätzen von AR-Parametern

```
> ar2 = arima.sim(list(order=c(2,0,0), ar=c(1.5,-.75)), n = 144)
> (ar.yw=ar.yw(ar2, order=2))
```

Call:

```
ar.yw.default(x = ar2, order.max = 2)
```

Coefficients:

```
      1      2
1.4379 -0.7051
```

Order selected 2 sigma^2 estimated as 1.327

```
> (ar.ols=ar.ols(ar2, order=2))
```

Call:

```
ar.ols(x = ar2, order.max = 2)
```

Coefficients:

```
      1      2
1.5323 -0.7884
```

Intercept: -0.0186 (0.08249)

Order selected 2 sigma^2 estimated as 0.9655

```
> (ar.mle=ar.mle(ar2, order=2))
```

Call:

```
ar.mle(x = ar2, order.max = 2)
```

Coefficients:

```
      1      2
1.5246 -0.7802
```

Order selected 2 sigma^2 estimated as 0.9557

```
> (ar.burg=ar.burg(ar2, order=2))
```

Call:

```
ar.burg.default(x = ar2, order.max = 2)
```

Coefficients:

```
      1      2
1.5181 -0.7790
```

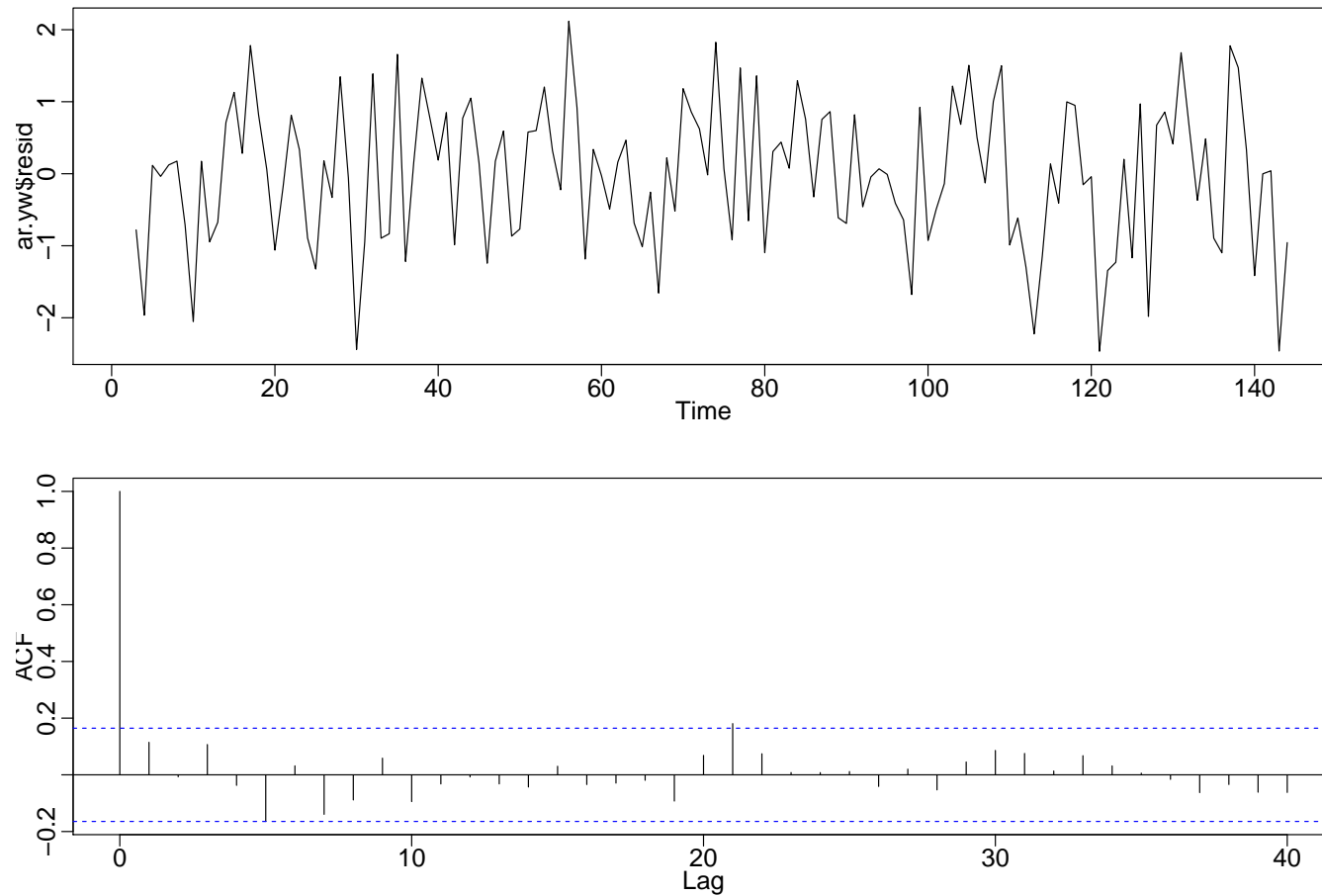
Order selected 2 sigma^2 estimated as 0.9563

Schätzen von AR-Parametern

```
#Werte der Yule-Walker-Schaetzung
names(ar.yw)
ar.yw$x.mean; mean(ar2) # (Mittelwertschaetzung)
ar.yw$ar      # (Parameterschaetzung)
ar.yw$asy.var.coef # Kovarianzmatrix der Parameterschaetzungen
(v=sqrt(diag(ar.yw$asy.var.coef))) # (Standardabweichungen der Parameter)
ar.yw$var.pred # (Varianz von White Noise)
ar.yw$ar[1]-1.96*v[1]; ar.yw$ar[1]+1.96*v[1] # Konfidenzintervale
ar.yw$ar[2]-1.96*v[2]; ar.yw$ar[2]+1.96*v[2]

#Yule-Walker direkt:
acf(ar2)[1:2]; names(acf(ar2))
(a=(acf(ar2)[1:2]$acf)) #rho(1), rho(2) genau
(s=sum((ar2-mean(ar2))^2)/144); sqrt(s) #gamma(0)=sigma^2, sigma
(rhom=matrix(c(1,a[1],a[1],1),2,2)) #R_2
(rhoi=ginv(rhom)) # inverse
(phi=rhoi%%a); ar.yw$ar # Schaetzung der Parameter, zum Vergleich Werte aus ar.yw
(sigma=s*(1-a%%phi)); (sigma.r=sigma*144/141); ar.yw$var.pred # Varianz von White Noise
#zum Vergleich Werte aus ar.yw: R nimmt 144-3=141 als Normierungskonstante für gamma(0)
(as=matrix(rep(1/length(ar2)*sigma/s,4),2,2)*rhoi); ar.yw$asy.var.coef # Kovarianzmatrix
# etwas anders als in ar.yw
(v=sqrt(as[1,1])) #=Standardabweichung der Parameter (hier gleich für beide)
(o1=phi[1]+qnorm(0.975)*v) #Konfidenzintervale
(u1=phi[1]-qnorm(0.975)*v)
(o2=phi[2]+qnorm(0.975)*v)
(u2=phi[2]-qnorm(0.975)*v)
```

Schätzen von AR-Parametern



Schätzen von AR-Parametern

Überprüfung der Unkorreliertheit der Residuen: Ljung-Box-Pierce-Test.

Die Test-Statistik

$$Q = N(N + 2) \sum_{j=1}^k \frac{\hat{\rho}_j^2}{N - j}$$

ist unter der Nullhypothese $\rho(1) = \dots = \rho(k) = 0$ asymptotisch χ^2 -verteilt mit $k - p$ Freiheitsgraden, wobei p die Anzahl der geschätzten AR-Parameter ist. Man kann dabei verschiedene k -Werte wählen, typisch 6,12,18,20... Auf jeden Fall sollte $k > p$ sein.

Schätzen von AR-Parametern

```
#Residuen Yule-Walker
plot(ar.yw$resid)
res=na.omit(ar.yw$resid)
acf(res,40)
Box.test(res, lag=20, type="Ljung-Box", fitdf=2) #p=2 Parameter geschätzt
Box.test(res, lag=18, type="Ljung-Box", fitdf=2) #p=2 Parameter geschätzt
qqnorm(res, main="Normal QQ-Plot") # qq-plots
qqline(res, col=2)

hist(res, prob=TRUE) # histogram
lines(density(res))
dn=dnorm(x=seq(min(res),max(res),length.out=500), mean(res), sd(res))
lines(x=seq(min(res),max(res),length.out=500), dn, col=2)

shapiro.test(res) # test for normality
lillie.test(res)
```

Levinson-Durbin-Rekursion

Statt die inverse Matrix in (13) zu berechnen, kann man auch rekursiv vorgehen und aus den Schätzungen der Parameter eines AR(p-1)-Prozesses die Parameter für einen AR(p)-Prozess zu schätzen. Das wird als **Levinson-Durbin-Rekursion bezeichnet**. Wir fangen an mit

$$\begin{aligned}\hat{\phi}_{00} &:= 0 \quad \text{und weiter rekursiv} \\ \hat{\phi}_{pp} &= \frac{\hat{\rho}(p) - \sum_{k=1}^{p-1} \hat{\phi}_{p-1,k} \hat{\rho}(p-k)}{1 - \sum_{k=1}^{p-1} \hat{\phi}_{p-1,k} \hat{\rho}(k)}\end{aligned}\tag{15}$$

wobei für $p \geq 2$

$$\hat{\phi}_{p,k} = \hat{\phi}_{p-1,k} - \hat{\phi}_{pp} \hat{\phi}_{p-1,p-k}, \quad k = 1, \dots, p-1.$$

Partielle Autokorrelationen

Die **partielle Korrelation** zweier Zufallsvariablen X und Y **unter Konstanthalten der Zufallsvariablen** Z_1, \dots, Z_k ist definiert als

$$\rho_{XY|Z_1, \dots, Z_k} := \text{corr}(X - \hat{X}, Y - \hat{Y}),$$

wobei \hat{X} die (LS-) lineare Regression von X durch Z_1, \dots, Z_k bezeichnet, d.h. $\hat{X} = a_1 Z_1 + \dots + a_k Z_k$ mit

$$\mathbb{E} [(X - a_1 Z_1 + \dots + a_k Z_k)^2] \stackrel{!}{=} \min.$$

\hat{Y} ist analog definiert.

→ Partielle Korrelation ist die Korrelation der vom linearen Einfluss von Z_1, \dots, Z_k bereinigten Zufallsvariablen $X - \hat{X}$ und $Y - \hat{Y}$.

Partielle Autokorrelationen

Die **partielle Autokorrelationsfunktion (PACF)** (π_τ) eines **stationären Prozesses** (X_t) ist die Folge der partiellen Korrelationen von X_t und $X_{t+\tau}$ unter Festhalten der Werte $X_{t+1}, \dots, X_{t+\tau-1}$. Wir setzen

$$\pi_0 := 1, \quad \pi_1 := \rho(1), \quad \pi_{-\tau} := \pi_\tau.$$

Wegen (1) gilt immer $\pi_2 = \frac{\rho(2) - \rho(1)^2}{1 - \rho(1)^2}$.

Die (**theoretische**) PACF eines stationären Prozesses ist gegeben durch die Werte $\pi_\tau = \phi_{\tau\tau}$ in der Durbin-Levinson Rekursion (15), wenn man statt den empirischen Autokorrelationen $\hat{\rho}$ die theoretischen Autokorrelationen ρ einsetzt.

Die **empirische** PACF ergibt sich entsprechend aus der Durbin-Levinson Rekursion (15) mit den empirischen Autokorrelationen $\hat{\rho}$.

Partielle Autokorrelationen

Für einen AR(1)-Prozess $X_t = \phi_1 X_{t-1} + \varepsilon_t$ gilt:

$$\pi_1 := \rho(1) = \phi_1, \quad \pi_\tau = 0 \quad \text{für } \tau \geq 2.$$

Für einen AR(2)-Prozess $X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \varepsilon_t$ gilt:

$$\pi_1 := \rho(1) = \frac{\phi_1}{1 - \phi_2}, \quad \pi_2 = \phi_2, \quad \pi_\tau = 0 \quad \text{für } \tau \geq 3.$$

Entsprechend erhält man für einen AR(p)-Prozess $\pi_p = \phi_p$ und $\pi_\tau = 0$ für $\tau > p$. Es gilt sogar:

Satz 7 (X_t) ist genau dann ein AR(p)-Prozess, wenn $\pi_p \neq 0$ und $\pi_\tau = 0$ für alle $\tau > p$. Dann gilt $\phi_p = \pi_p$.

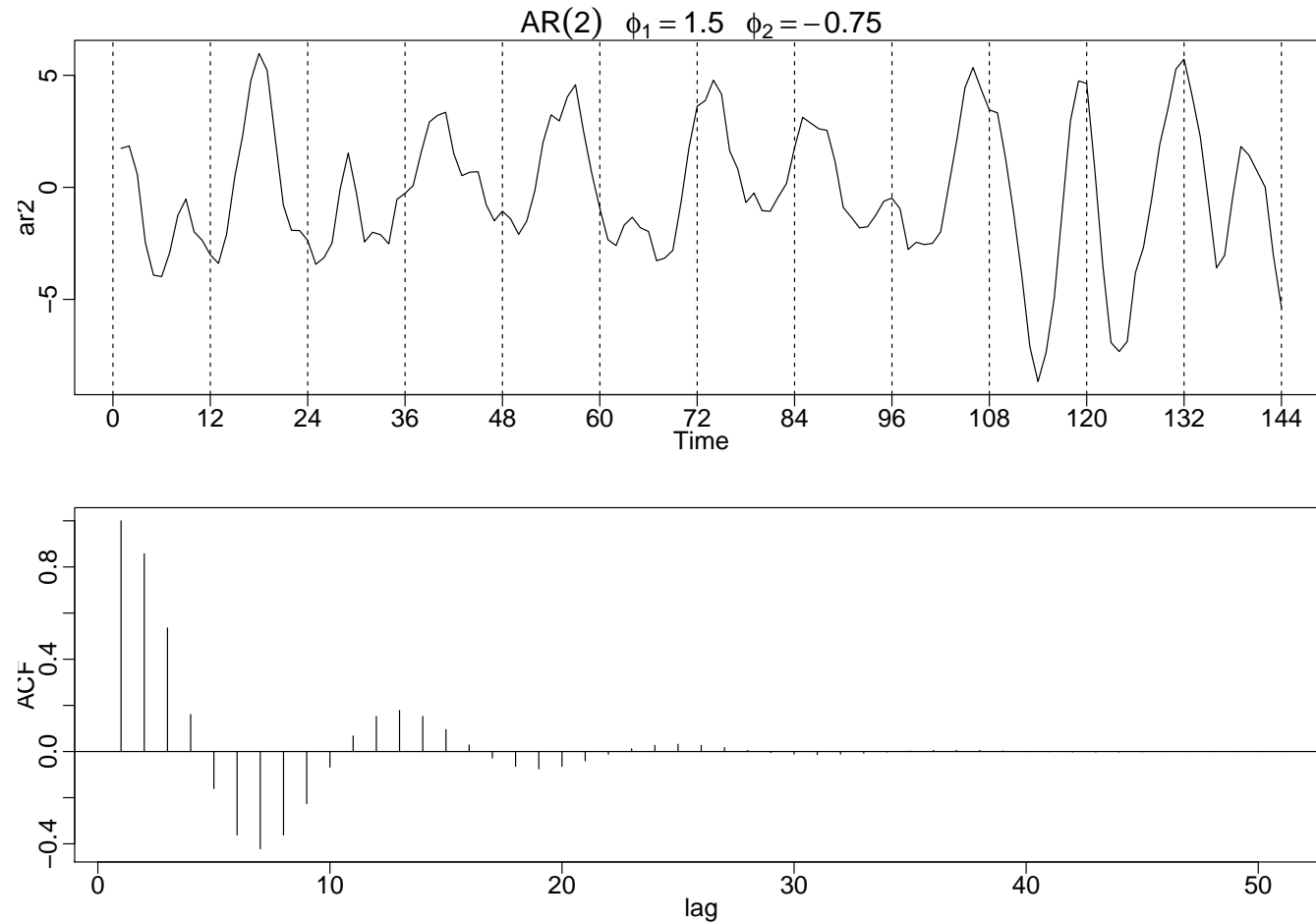
Partielle Autokorrelationen

Für die empirische PACF ($\hat{\pi}_\tau$) eines AR(p)-Prozesses gilt wegen $\hat{\pi}_\tau = \hat{\phi}_\tau$ in der Yule-Walker-Schätzung und wegen (14):

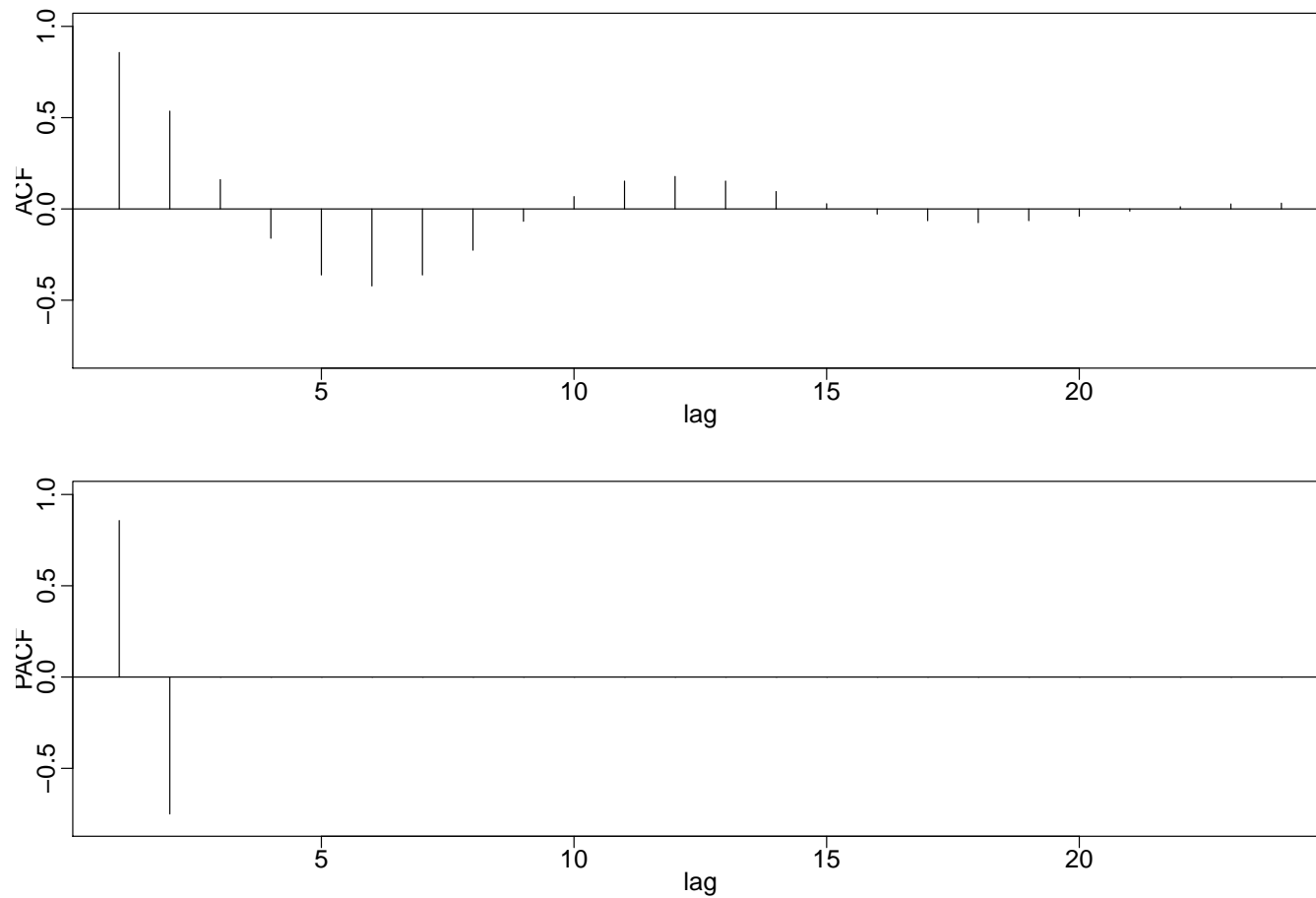
$$\sqrt{N}\hat{\pi}_\tau \xrightarrow{\text{in Vert.}} N(0, 1) \quad \text{für} \quad \tau > p.$$

Damit erhalten wir ein Kriterium und Konfidenzintervalle für Identifikation eines AR(p)-Prozesses: z.B. müssen ung. 95% der ersten $n > p$ empirischen PACF-Werte eines AR(p)-Prozesses innerhalb des 95%-Konfidenzintervalls $\pm \frac{1.96}{\sqrt{N}}$ liegen.

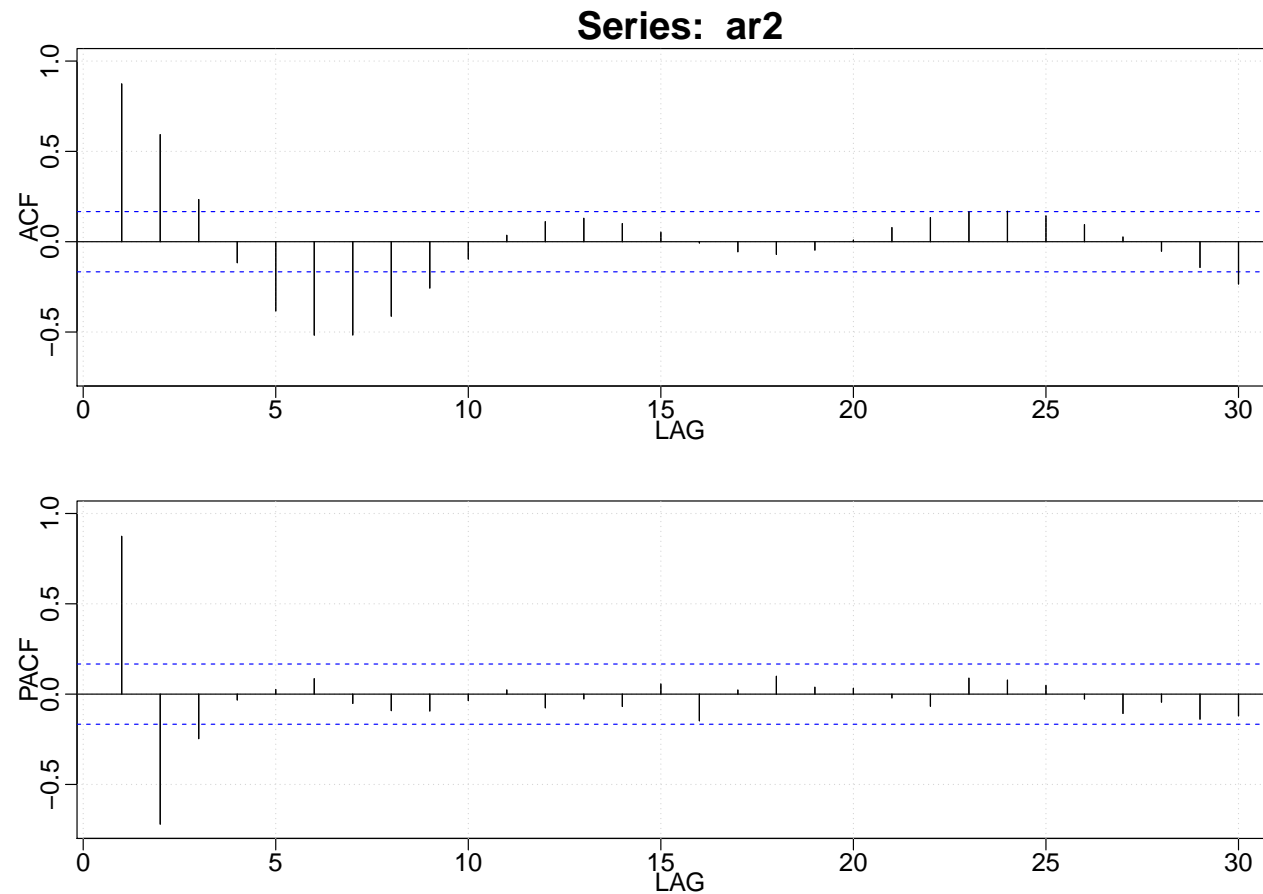
AR(2)-Prozess: Beispiel



AR(2)-Prozess: Theoretische ACF und PACF



AR(2)-Prozess: Empirische ACF und PACF



AR(2)-Prozess: Beispiele

```
ACF = ARMAacf(ar=c(1.5,-.75), ma=0, 24)[-1]
PACF = ARMAacf(ar=c(1.5,-.75), ma=0, 24, pacf=TRUE)
par(mfrow=c(2,1), mar=c(3,2,1,0)+.5, mgp=c(1.6,.6,0))
plot(ACF, type="h", xlab="lag", ylim=c(-.8,1)); abline(h=0)
plot(PACF, type="h", xlab="lag", ylim=c(-.8,1)); abline(h=0)

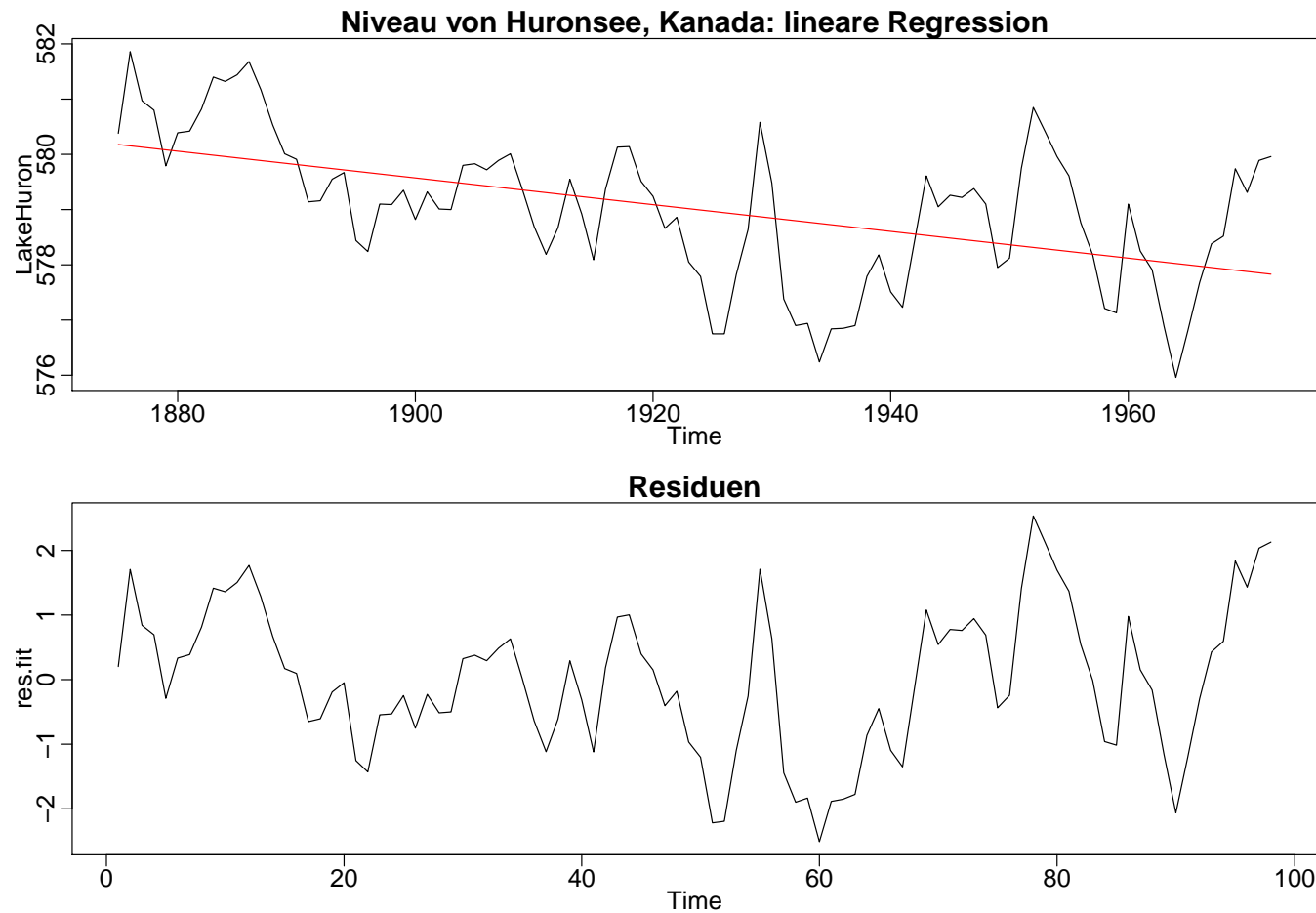
set.seed(8675309)
ar2 = arima.sim(list(order=c(2,0,0), ar=c(1.5,-.75)), n = 144)
acf2(ar2,30)
```

AR-Prozesse: Identifikation der Ordnung

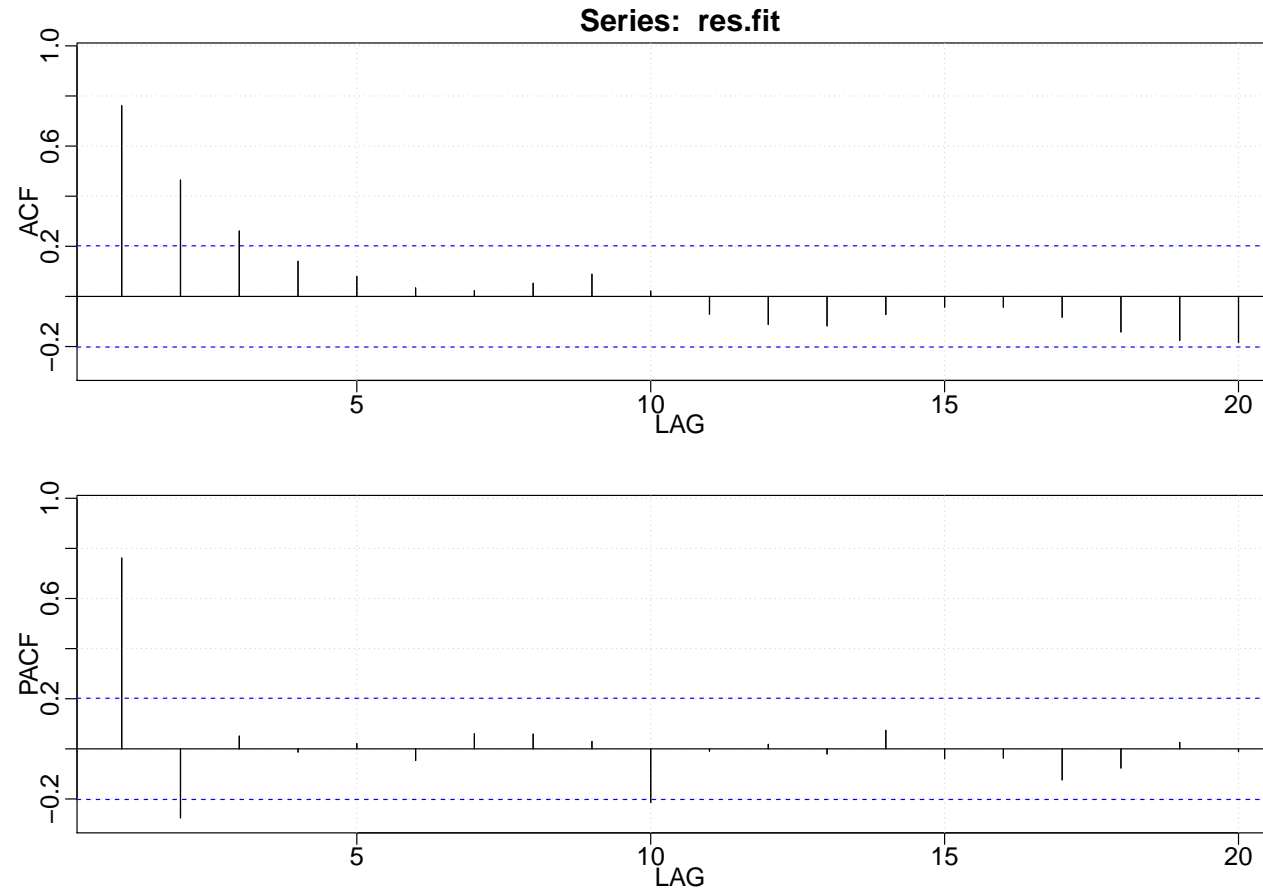
Außer der empirischen ACF und PACF kann man noch AIC und BIC zu Identifikation von AR-Modellen heranziehen: Es werden sukzessiv AR(1), AR(2), ..., AR(p) Modelle bis zu einer vorgegebenen Maximalordnung p angepasst. Das beste Modell wird mittels AIC bzw. BIC ausgesucht.

- Wenn der Unterschied in AIC bzw. BIC klein ist, sollten Modelle mit weniger Parametern gewählt werden.
- Wenn einige geschätzte Parameter sich nicht signifikant von Null unterscheiden, sollten sie weggelassen und das Modell neu angepasst werden (Subsetmodelle)
- Bei N groß ($N \geq 100$) überschätzt AIC relativ häufig die Modellordnung, BIC liegt da besser. BIC ist stark konsistent, AIC nicht.

AR-Prozesse: Identifikation der Ordnung



AR-Prozesse: Identifikation der Ordnung



AR-Prozesse: Identifikation der Ordnung

```
> (mm1=ar.mle(res.fit))
```

Call:

```
ar.mle(x = res.fit)
```

Coefficients:

1	2
1.0047	-0.2920

Order selected 2 sigma^2 estimated as 0.4571

```
> (mm2=ar.yw(res.fit))
```

Call:

```
ar.yw.default(x = res.fit)
```

Coefficients:

1	2
0.9714	-0.2754

Order selected 2 sigma^2 estimated as 0.501

```
> (mm3=ar.ols(res.fit))
```

Call:

```
ar.ols(x = res.fit)
```

Coefficients:

1	2
1.0021	-0.2838

Intercept: -0.007852 (0.06803)

Order selected 2 sigma^2 estimated as 0.4435

```
> (mm4=ar.burg(res.fit))
```

Call:

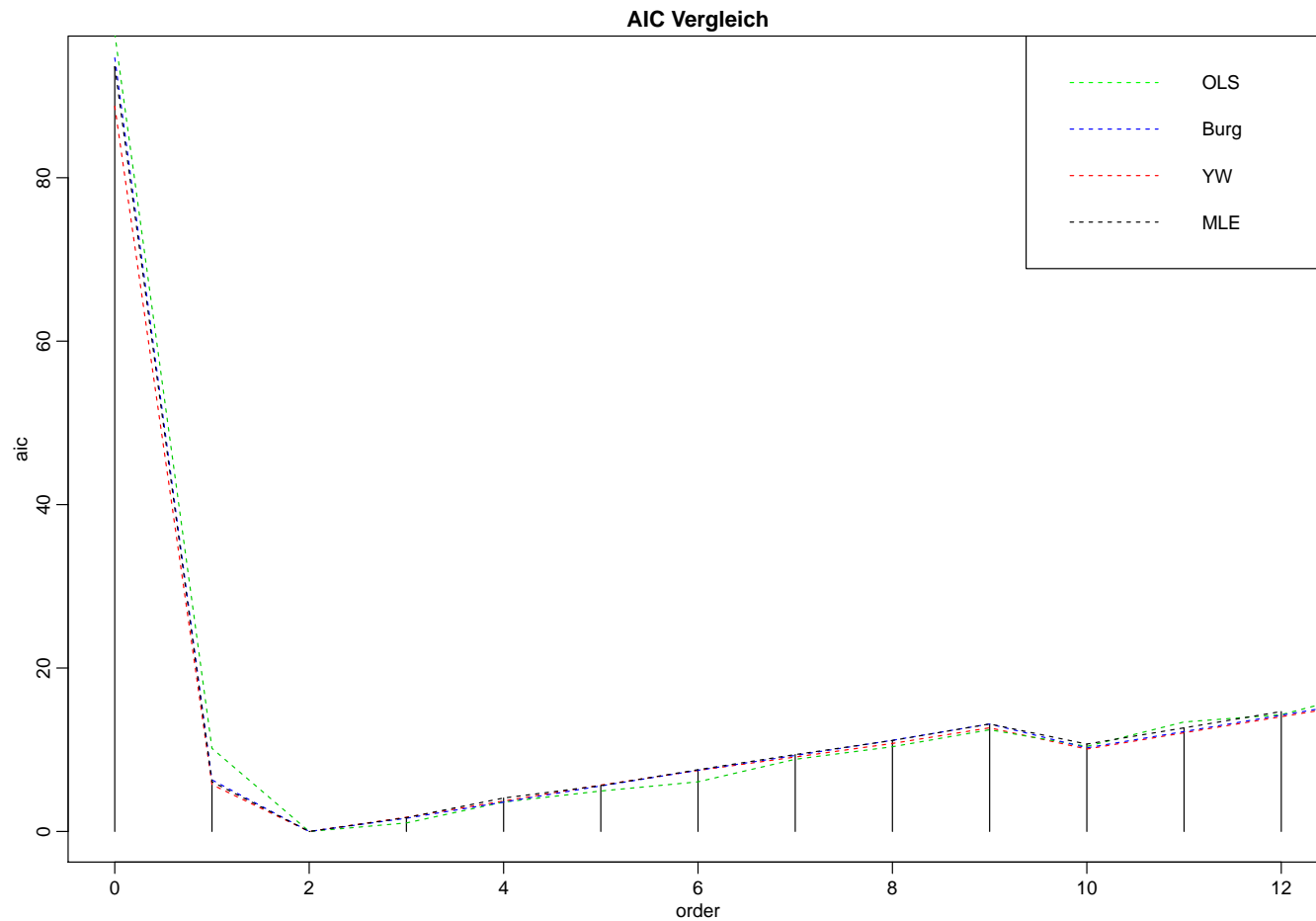
```
ar.burg.default(x = res.fit)
```

Coefficients:

1	2
0.9974	-0.2851

Order selected 2 sigma^2 estimated as 0.4573

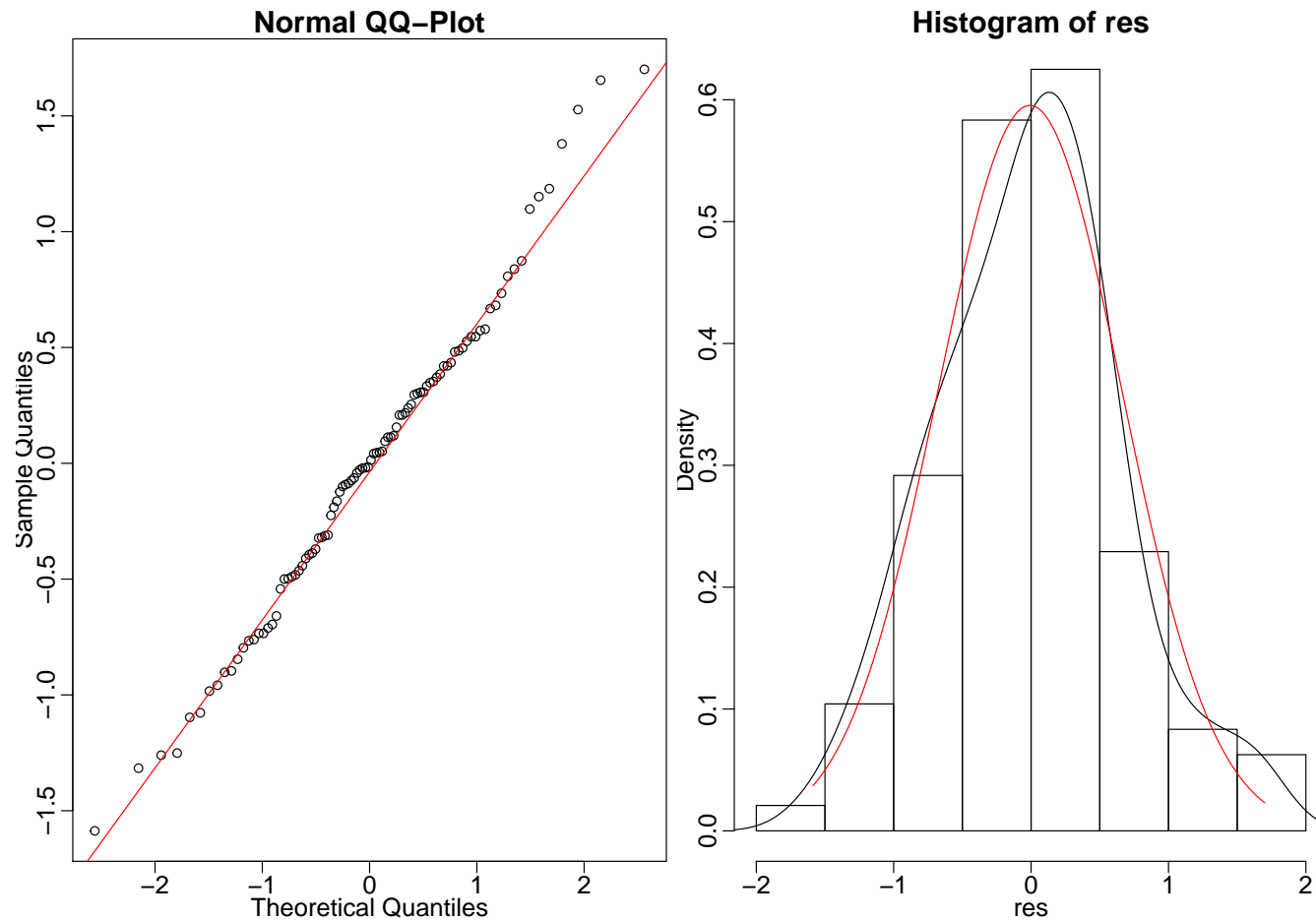
AR-Prozesse: Identifikation der Ordnung



AR-Prozesse: Identifikation der Ordnung

```
names(mm1)
print(mm1$aic,digits=3)
aic=mm1$aic # For plotting below.
plot(c(0:(length(aic)-1)),aic,type='h',xlab='order',ylab='aic')
lines(0:(length(mm2$aic)-1), mm2$aic, lty=2, col=2)
lines(0:(length(mm3$aic)-1), mm3$aic, lty=2, col=3)
lines(0:(length(mm4$aic)-1), mm4$aic, lty=2, col=4)
lines(0:(length(aic)-1),aic,lty=2)
colors <- c("green","blue","red", "black")
legend("topright", c(paste("OLS"),
                    paste("Burg"),
                    paste("YW"),
                    paste("MLE")),
      lwd = 1, cex=1, col=colors, pt.lwd = 1, lty=2)
```


AR-Prozesse: Identifikation der Ordnung

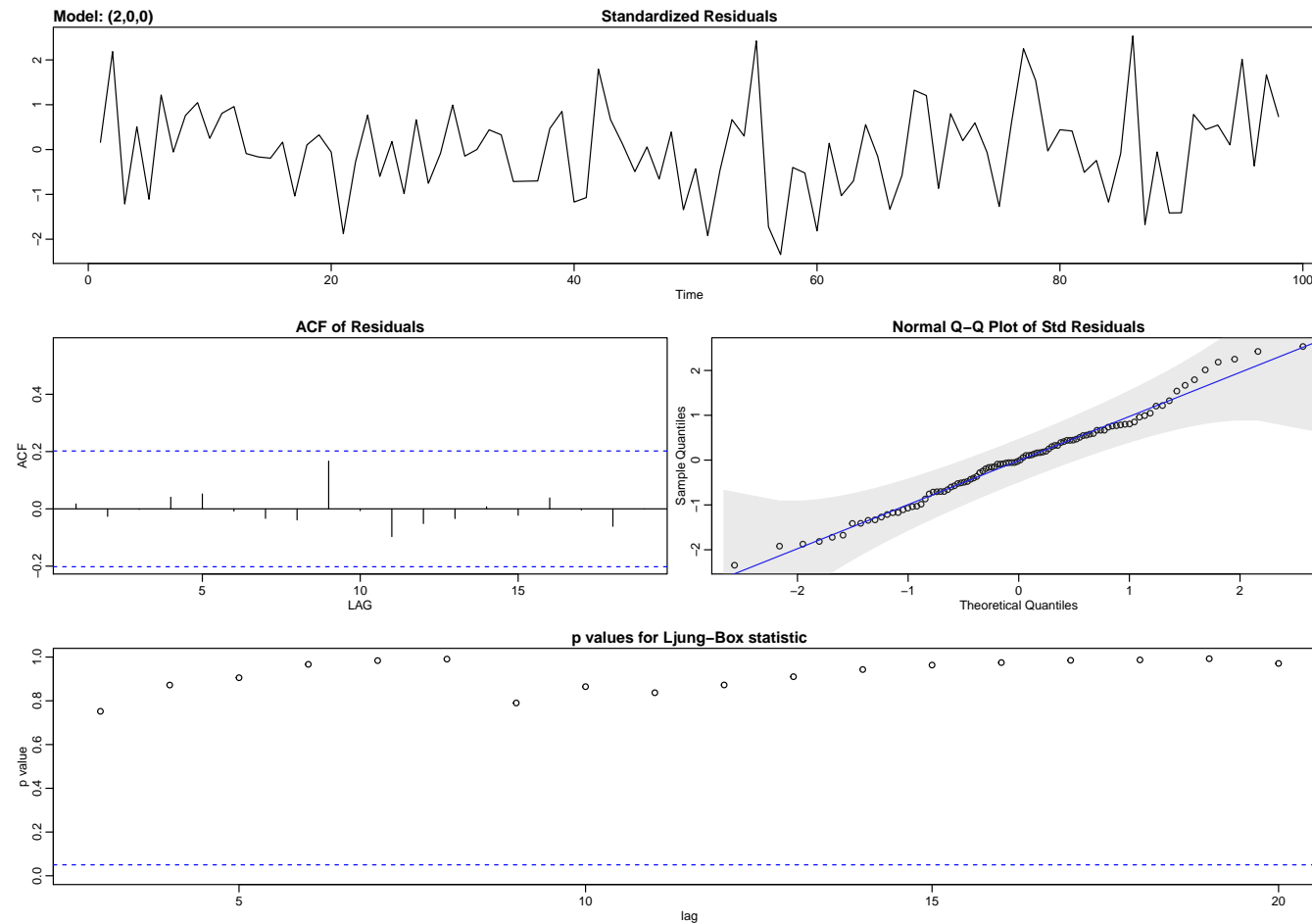


AR-Prozesse: Identifikation der Ordnung

```
#Fehleranalyse
plot.ts(mm2$resid)
res=na.omit(mm2$resid)
acf(res,30)
Box.test(res, lag=20, type="Ljung-Box", fitdf=2)
par(mfrow=c(1,2))
qqnorm(res, main="Normal QQ-Plot") # qq-plots
qqline(res, col=2)
hist(res, prob=TRUE) # histogram
lines(density(res))
dn=dnorm(x=seq(min(res),max(res),length.out=500), mean(res), sd(res))
lines(x=seq(min(res),max(res),length.out=500), dn, col=2)

#auch möglich
(m=arima(res.fit, c(2,0,0)))
coeftest(m) #test (p-values fuer koeffizienten), braucht packet lmtest
tsdiag(m)
(mm=sarima(res.fit, 2,0,0))
mm$tttable
```

AR-Prozesse: Identifikation der Ordnung



Regression mit autokorrelierten Störungen

```
(t=1:length(LakeHuron))
(fit=lm(LakeHuron~t))
plot(LakeHuron, main="Niveau von Huronsee, Kanada: lineare Regression")
lines(ts(fit$fitted.values, start=1875, frequency = 1), col=2)
summary(fit)
vcov(fit)
# Varianz der Schaetzer \beta_0, \beta_1, falls Residuen unabhaengig: 0.052; 0.000016
#Residuen AR(2), jetzt Regression und Residuelmodellierung zusammen:
(m=sarima(LakeHuron,2,0,0,xreg=t, details = F))
m$tttable
m$fit$var.coef
# Varianz der Schaetzer \beta_0, \beta_1, falls Residuen AR(2): 0.2149; 0.000066 -
# dreimal so gross wie vorher!
> vcov(fit)
```

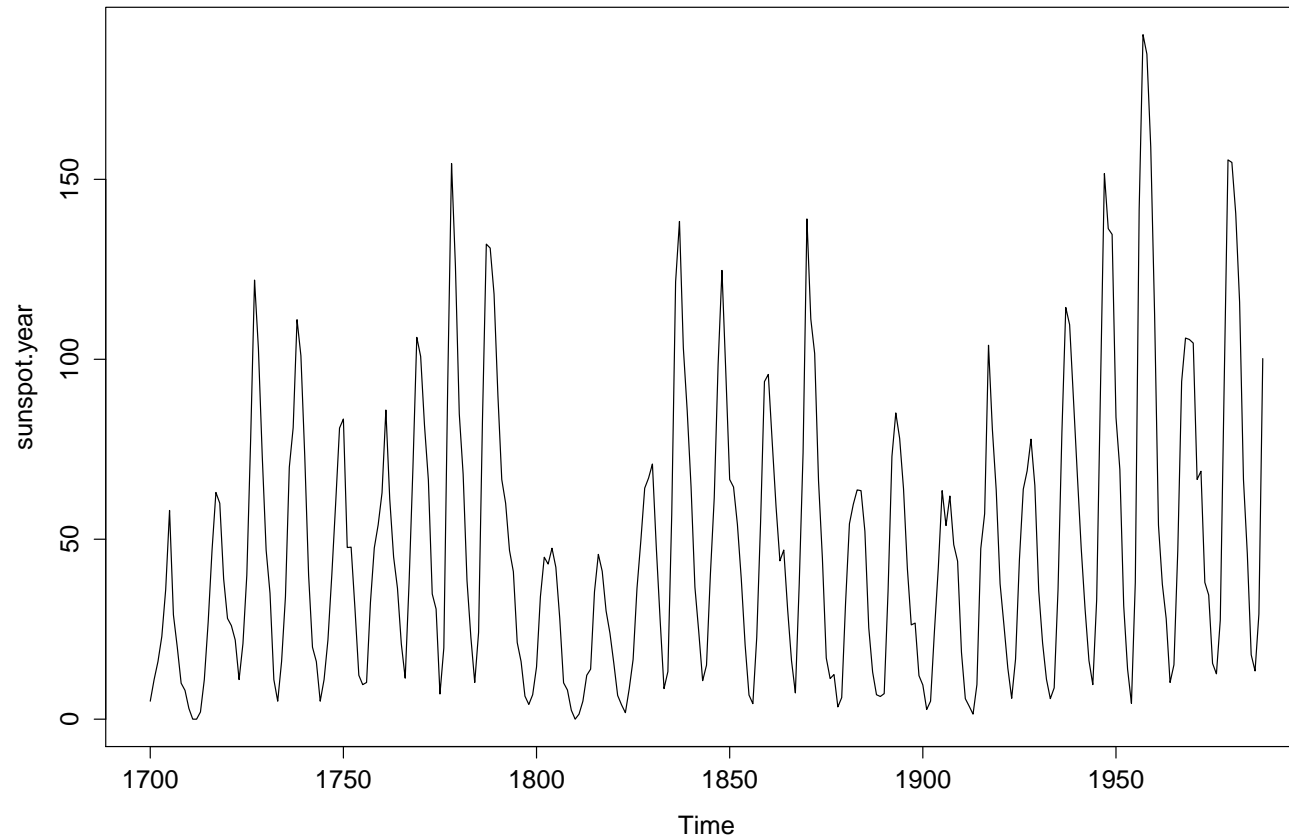
	(Intercept)	t
(Intercept)	0.0529511879	-8.063633e-04
t	-0.0008063633	1.629017e-05

```
> m$fit$var.coef
```

	ar1	ar2	intercept	xreg
ar1	9.527944e-03	-7.612603e-03	-1.710798e-03	2.054033e-05
ar2	-7.612603e-03	1.007317e-02	-4.652789e-05	3.424910e-05
intercept	-1.710798e-03	-4.652789e-05	2.149140e-01	-3.233833e-03
xreg	2.054033e-05	3.424910e-05	-3.233833e-03	6.559041e-05

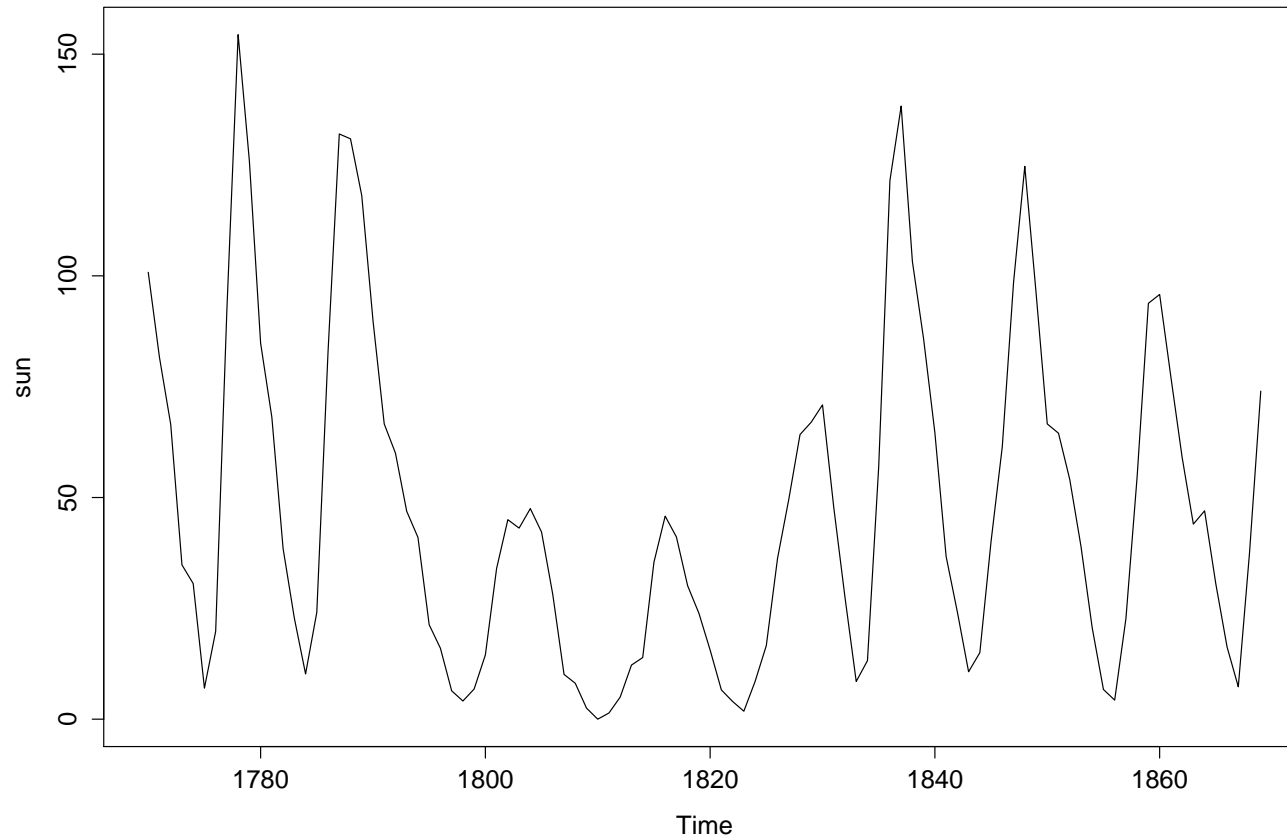
AR-Prozesse: Identifikation der Ordnung

Jährliche Anzahl Sonnenflecken 1700–1988

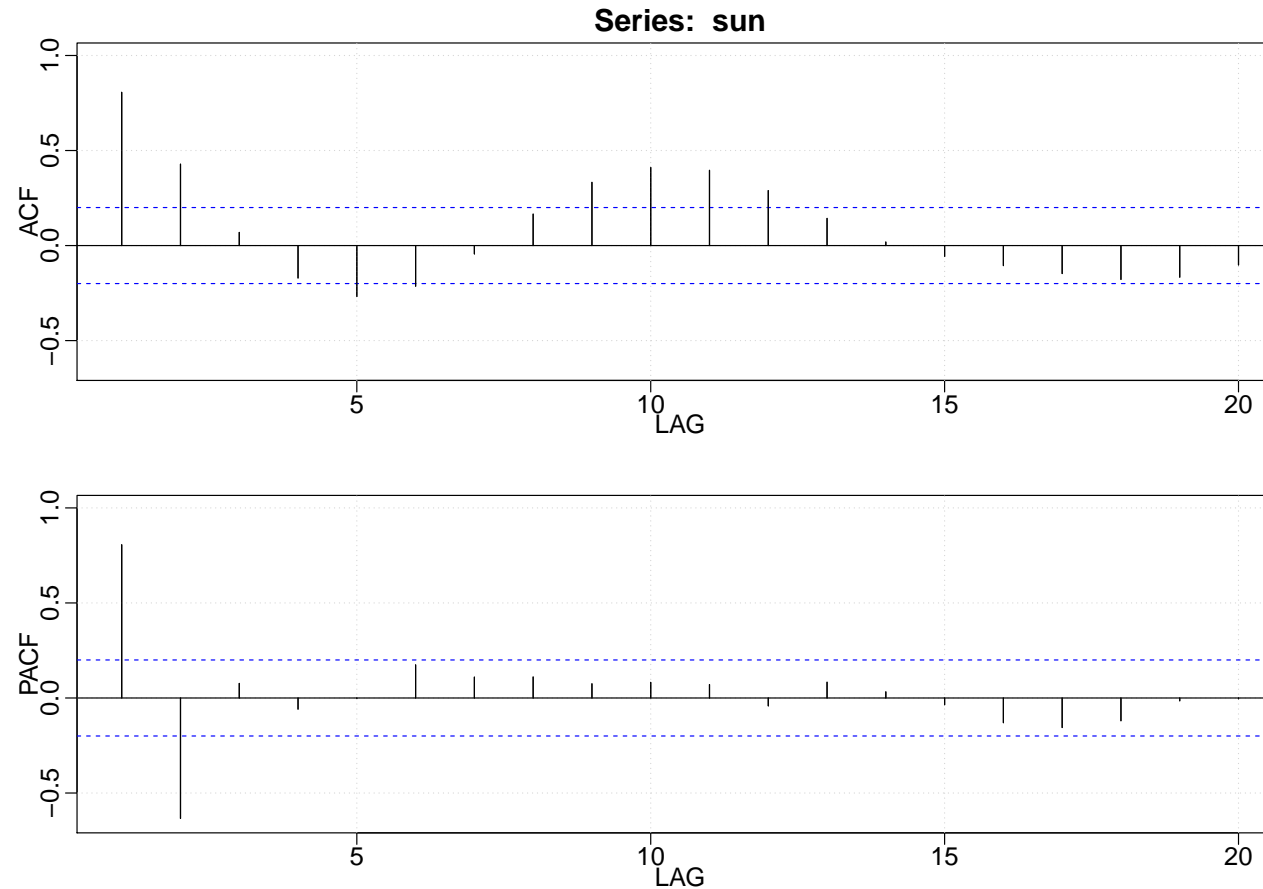


AR-Prozesse: Identifikation der Ordnung

Ausschnitt 1700–1869



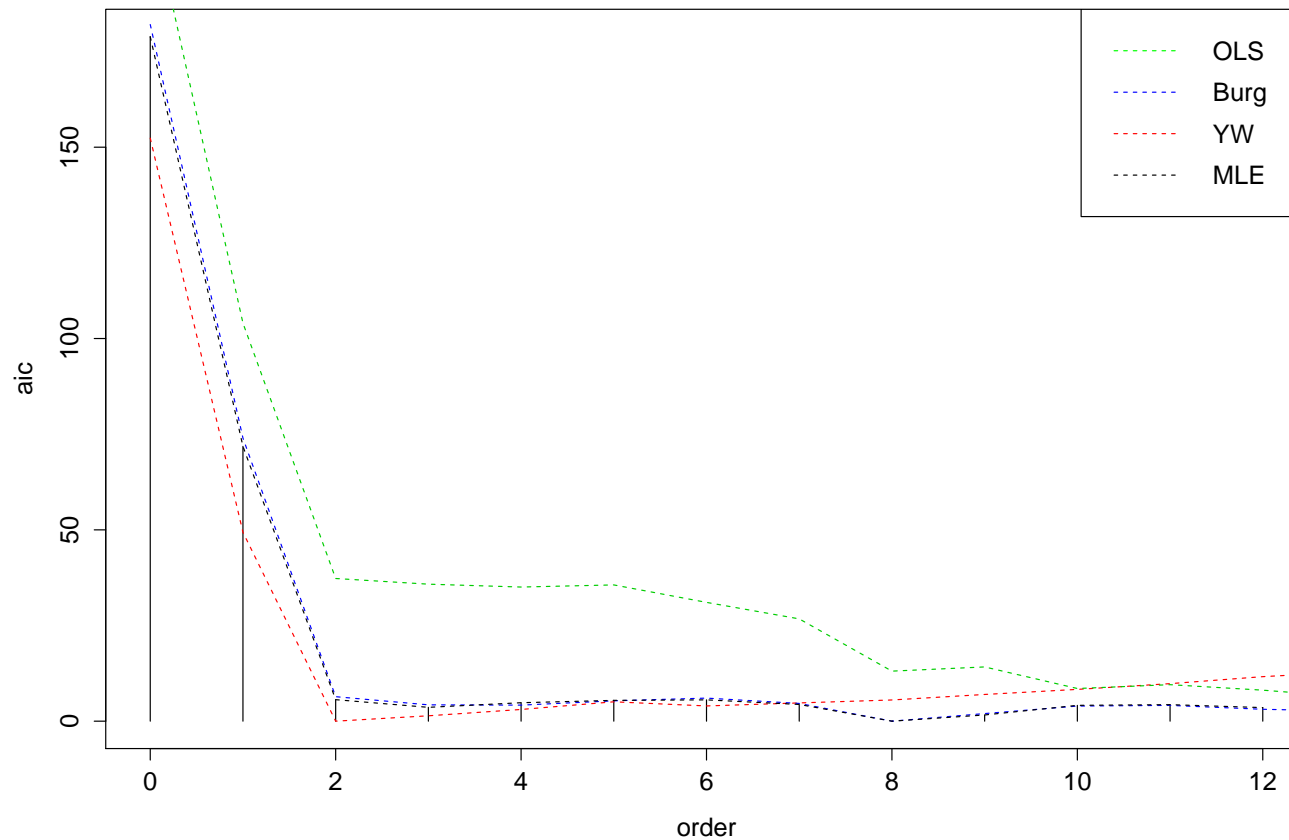
AR-Prozesse: Identifikation der Ordnung



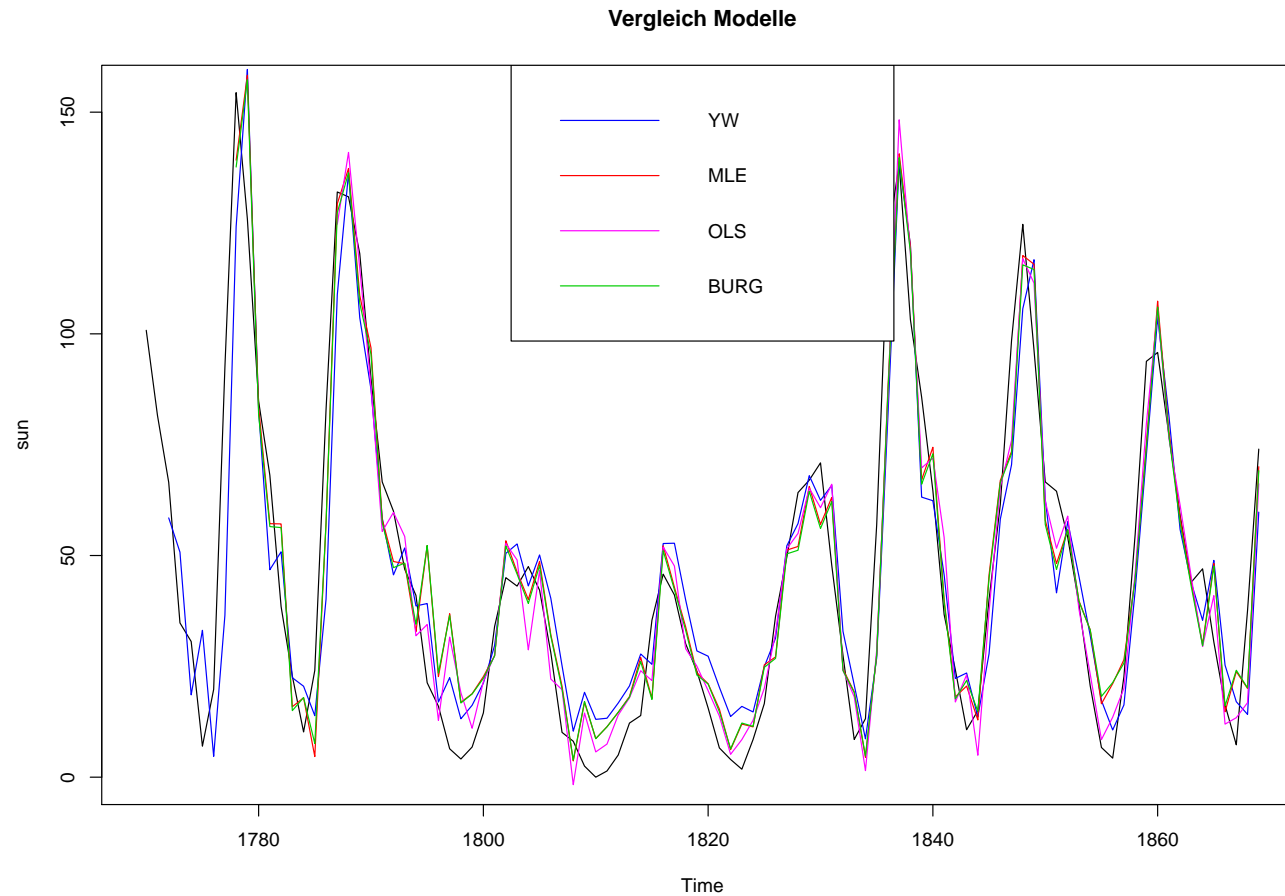
AR-Prozesse: Identifikation der Ordnung

```
> mm1=ar.mle(sun)
> mm1$order # Find the identified order
[1] 8
> mm2=ar.yw(sun)
> mm2$order # Find the identified order
[1] 2
> mm3=ar.ols(sun)
> mm3$order # Find the identified order
[1] 17
> mm4=ar.burg(sun)
> mm4$order # Find the identified order
[1] 8
```


AR-Prozesse: Identifikation der Ordnung

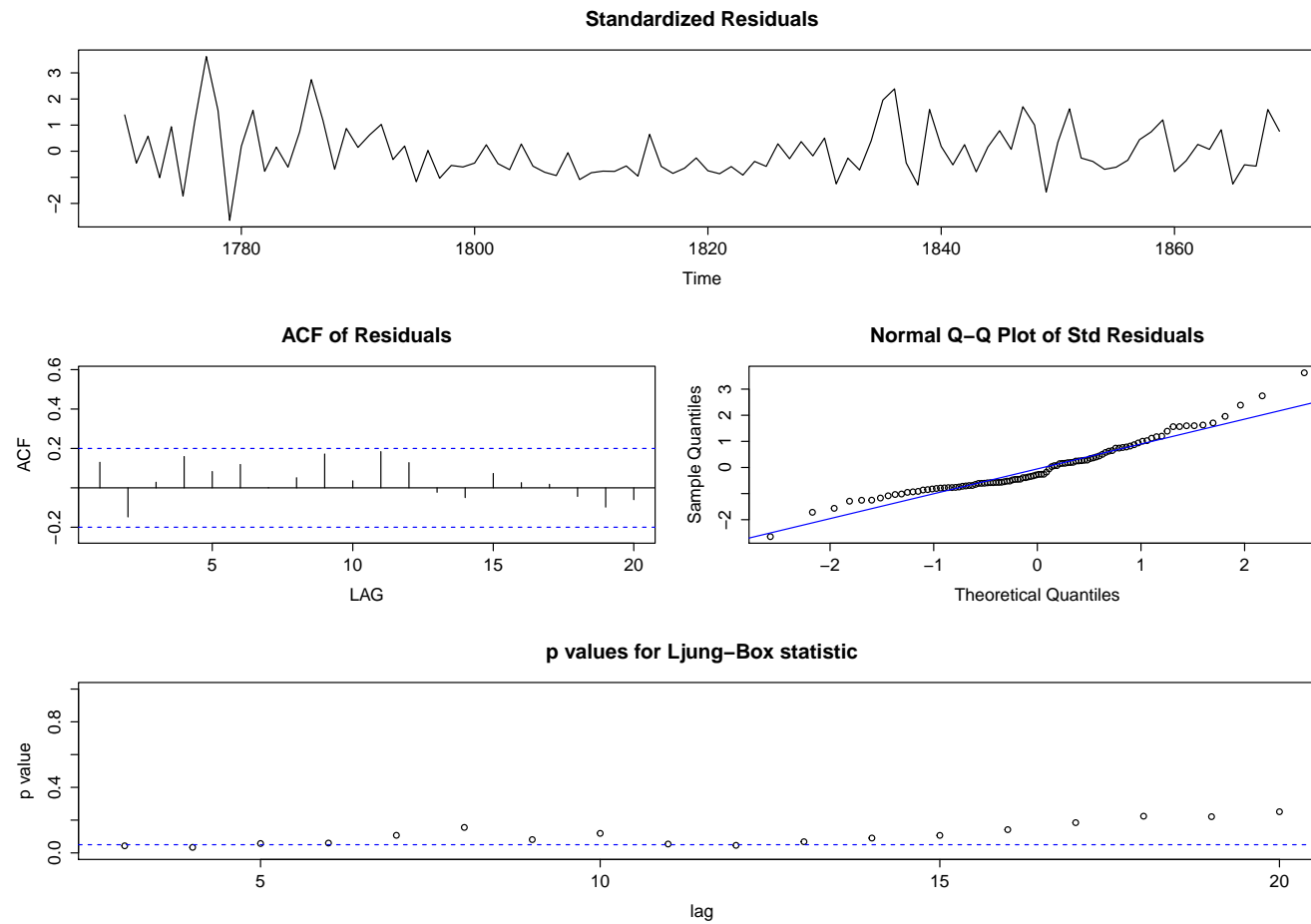


AR-Prozesse: Identifikation der Ordnung



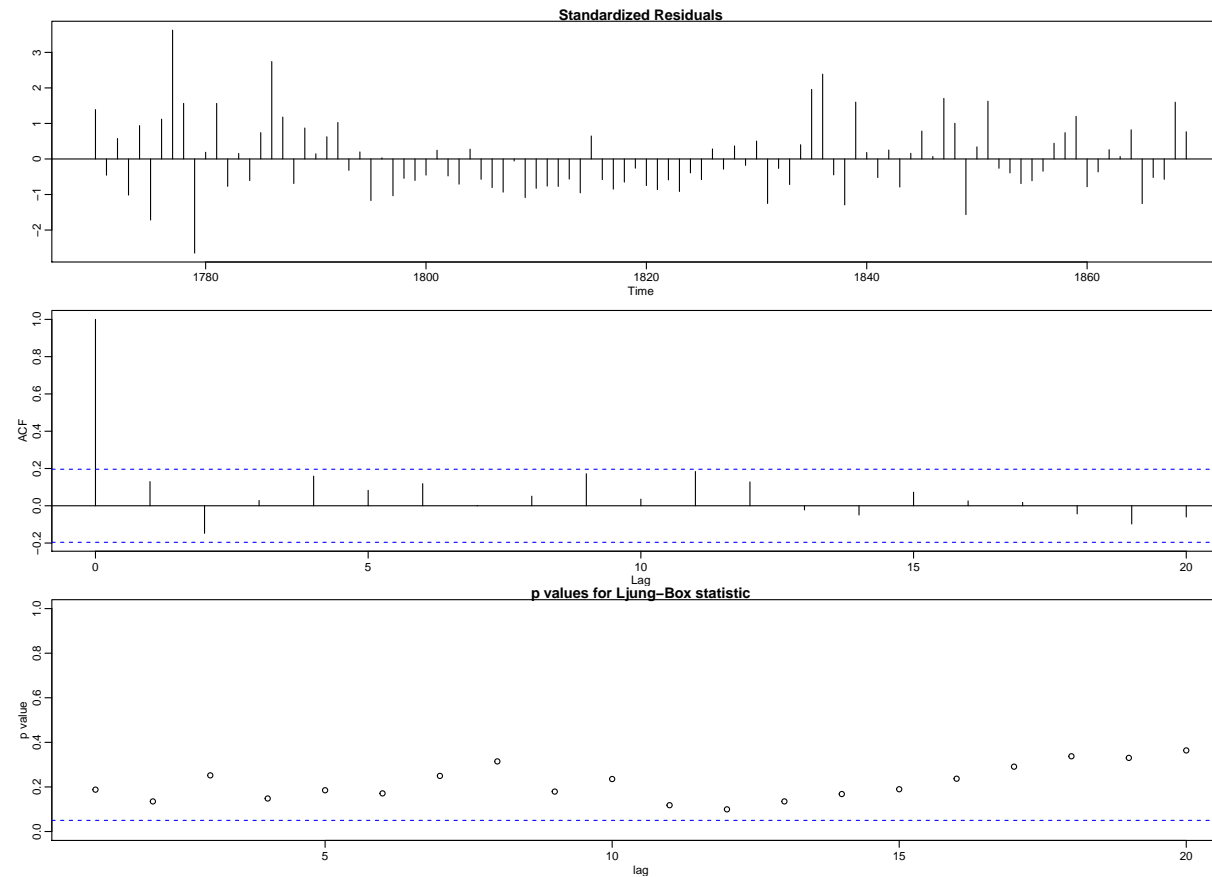
Residuen des AR(2)-Prozesses

Mit “sarima”:



Residuen des AR(2)-Prozesses

Mit “arima” und “tsdiag”



Freiheitsgrade beim Ljung-Box-Test nicht korrekt!

AR-Prozesse: Subsetmodelle

```
> sarima(sun, 2,0,0, details = F)$ttable # alle signifikant, aber Ljung-Box nicht optimal
```

	Estimate	SE	t.value	p.value
ar1	1.4059	0.0706	19.9212	0
ar2	-0.7111	0.0702	-10.1235	0
xmean	48.2642	4.9747	9.7019	0

```
> sarima(sun, 8,0,0, details = F)$ttable # 4-7 nicht signifikant
```

	Estimate	SE	t.value	p.value
ar1	1.5271	0.0966	15.8098	0.0000
ar2	-1.1107	0.1819	-6.1066	0.0000
ar3	0.4994	0.2216	2.2540	0.0266
ar4	-0.2051	0.2370	-0.8654	0.3891
ar5	-0.0035	0.2457	-0.0142	0.9887
ar6	0.1427	0.2415	0.5910	0.5560
ar7	-0.2681	0.2054	-1.3052	0.1951
ar8	0.2807	0.1084	2.5906	0.0112
xmean	50.1590	9.3697	5.3533	0.0000

```
> (sm=arima(sun, c(8,0,0), fixed=c(NA,NA,NA,0,0,0,0,NA, NA)) )
```

Call:

```
arima(x = sun, order = c(8, 0, 0), fixed = c(NA, NA, NA, 0, 0, 0, 0, NA, NA))
```

Coefficients:

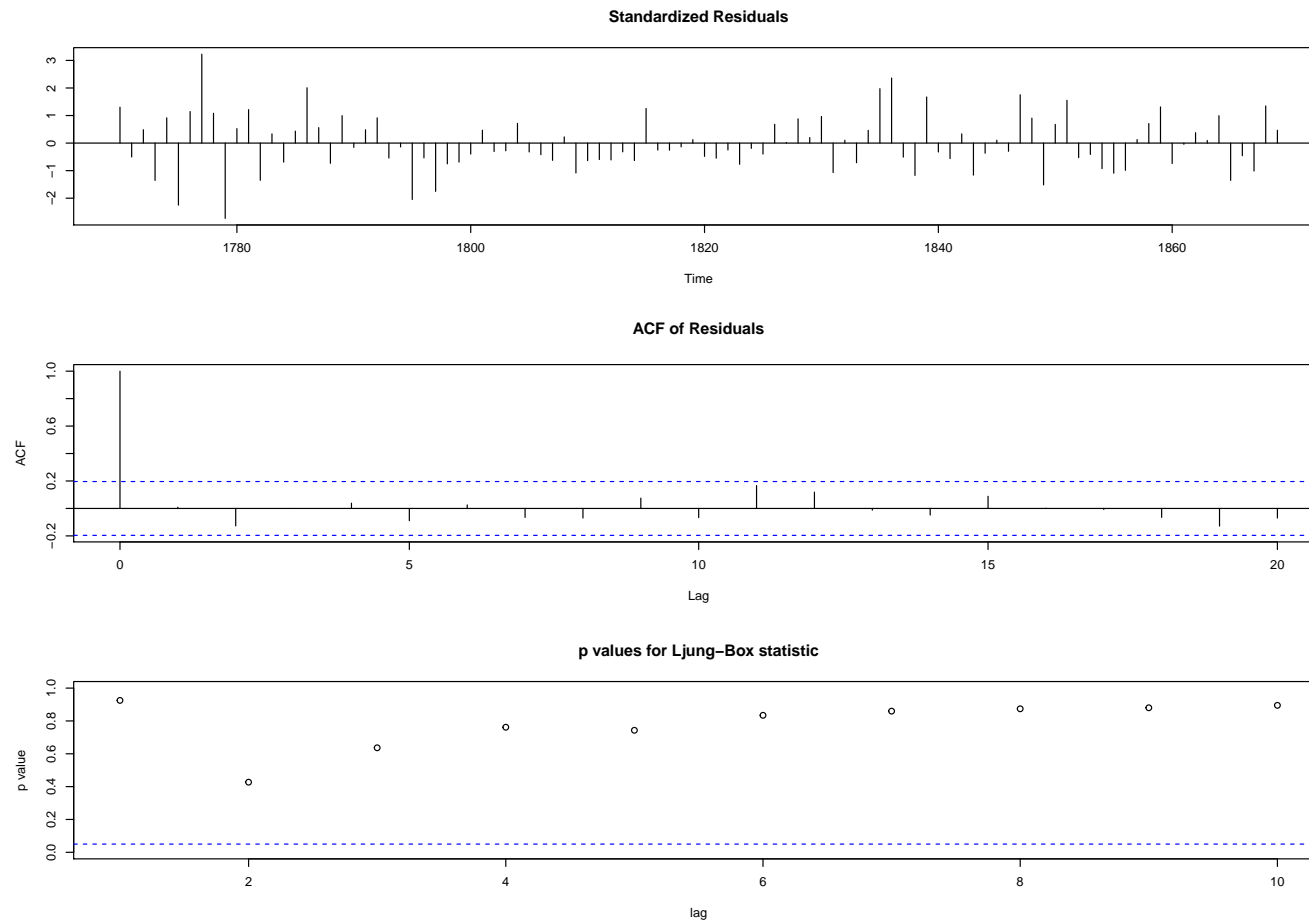
	ar1	ar2	ar3	ar4	ar5	ar6	ar7	ar8	intercept
	1.5016	-0.9280	0.1980	0	0	0	0	0.1114	49.8672
s.e.	0.0953	0.1503	0.0942	0	0	0	0	0.0392	11.2427

sigma^2 estimated as 203.1: log likelihood = -409.19, aic = 830.38

Warning message:

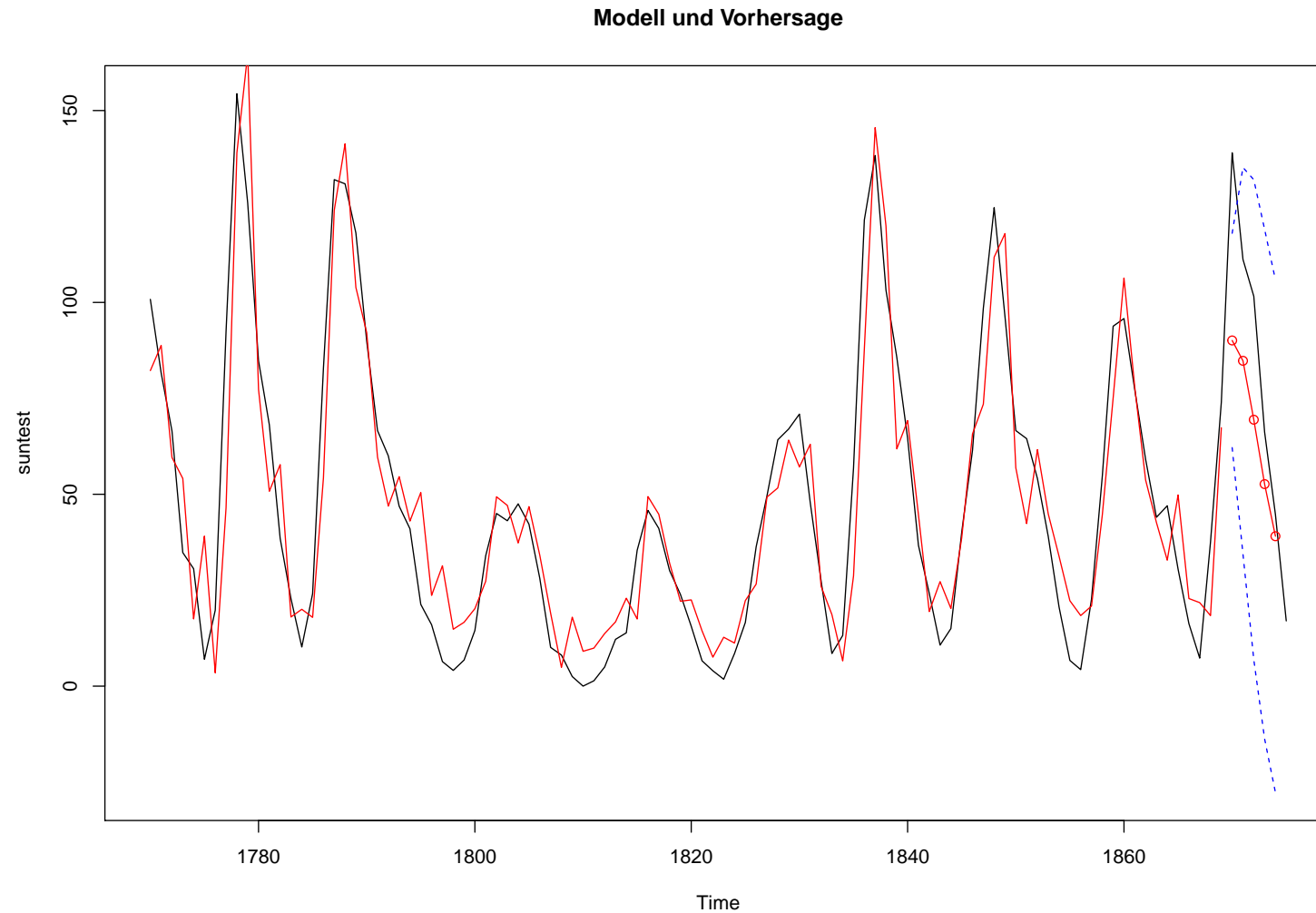
```
In arima(sun, c(8, 0, 0), fixed = c(NA, NA, NA, 0, 0, 0, 0, NA, :
some AR parameters were fixed: setting transform.pars = FALSE
```

AR-Prozesse: Identifikation der Ordnung

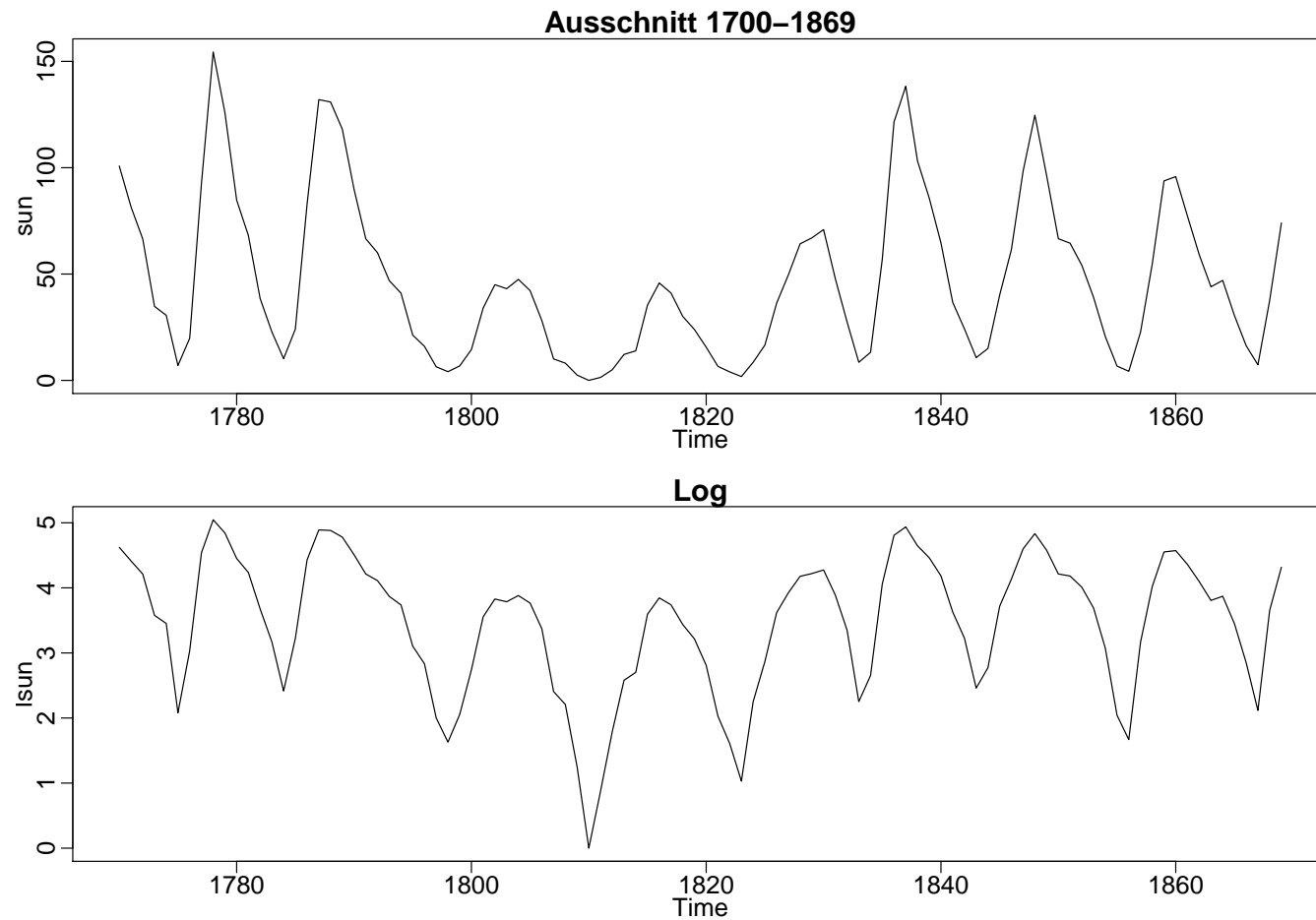


$$\text{Modell: } X_t = 5.83 + 1.5X_{t-1} - 0.93X_{t-2} + 0.2X_{t-3} + 0.11X_{t-8} + \varepsilon_t, \quad \sigma_\varepsilon^2 = 203.06$$

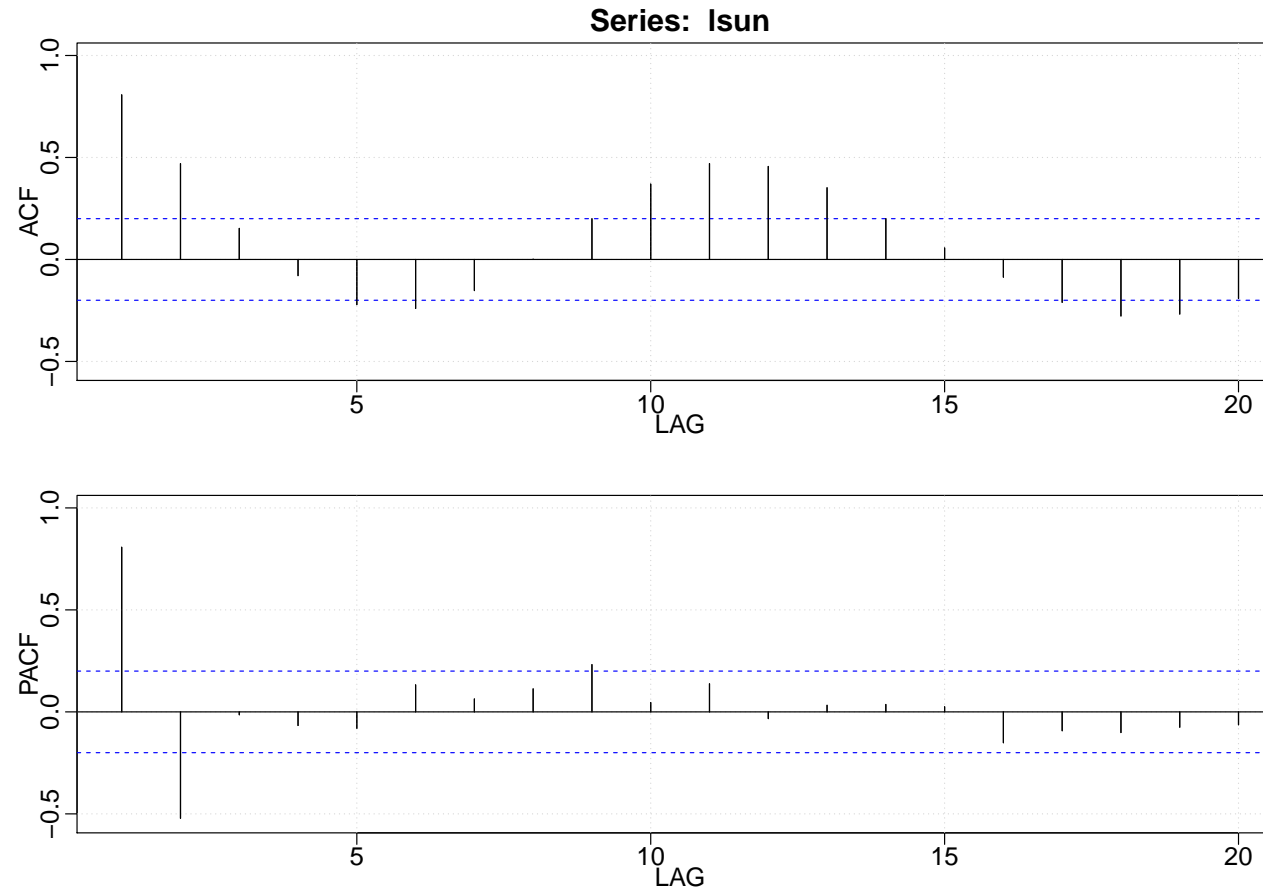
AR-Prozesse: Identifikation der Ordnung



AR-Prozesse: Identifikation der Ordnung



AR-Prozesse: Identifikation der Ordnung



AR-Prozesse: Identifikation der Ordnung

lsun=log(sun+1)

```
> (mm1=ar.mle(lsun))
```

```
Call:  
ar.mle(x = lsun)
```

```
Coefficients:  
      1      2  
1.2690 -0.5542
```

```
Order selected 2  sigma^2 estimated as  0.2494
```

```
> (mm2=ar.yw(lsun))
```

```
Call:  
ar.yw.default(x = lsun)
```

```
Coefficients:  
      1      2  
1.2285 -0.5218
```

```
Order selected 2  sigma^2 estimated as  0.2822
```

```
> (mm3=ar.ols(lsun))
```

```
Call:  
ar.ols(x = lsun)
```

```
Coefficients:  
      1      2      3      4      5      6      7      8      9  
1.2473 -0.5360 -0.0438  0.1908 -0.1847  0.0019  0.1035 -0.1790  0.2403
```

```
Intercept: -0.0104 (0.04639)
```

```
Order selected 9  sigma^2 estimated as  0.1951
```

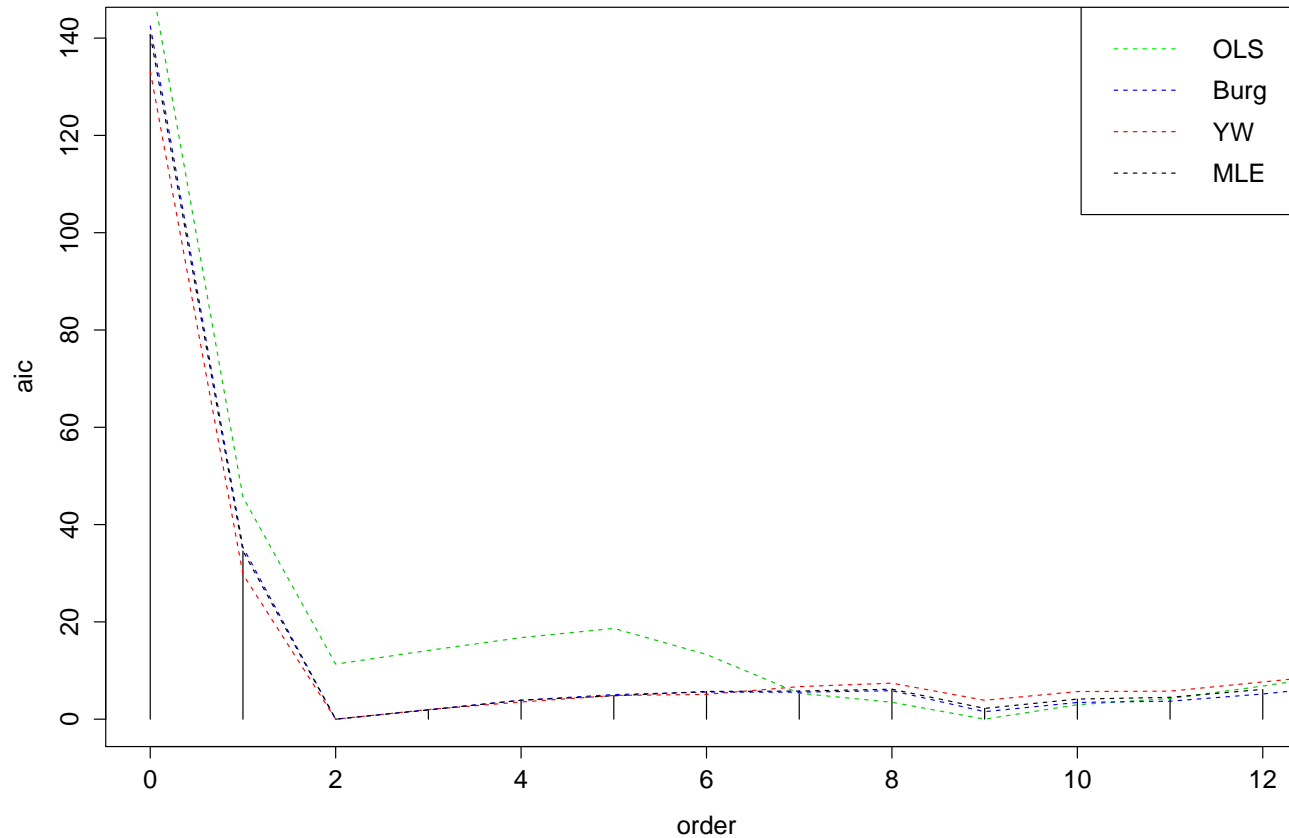
```
> (mm4=ar.burg(lsun))
```

```
Call:  
ar.burg.default(x = lsun)
```

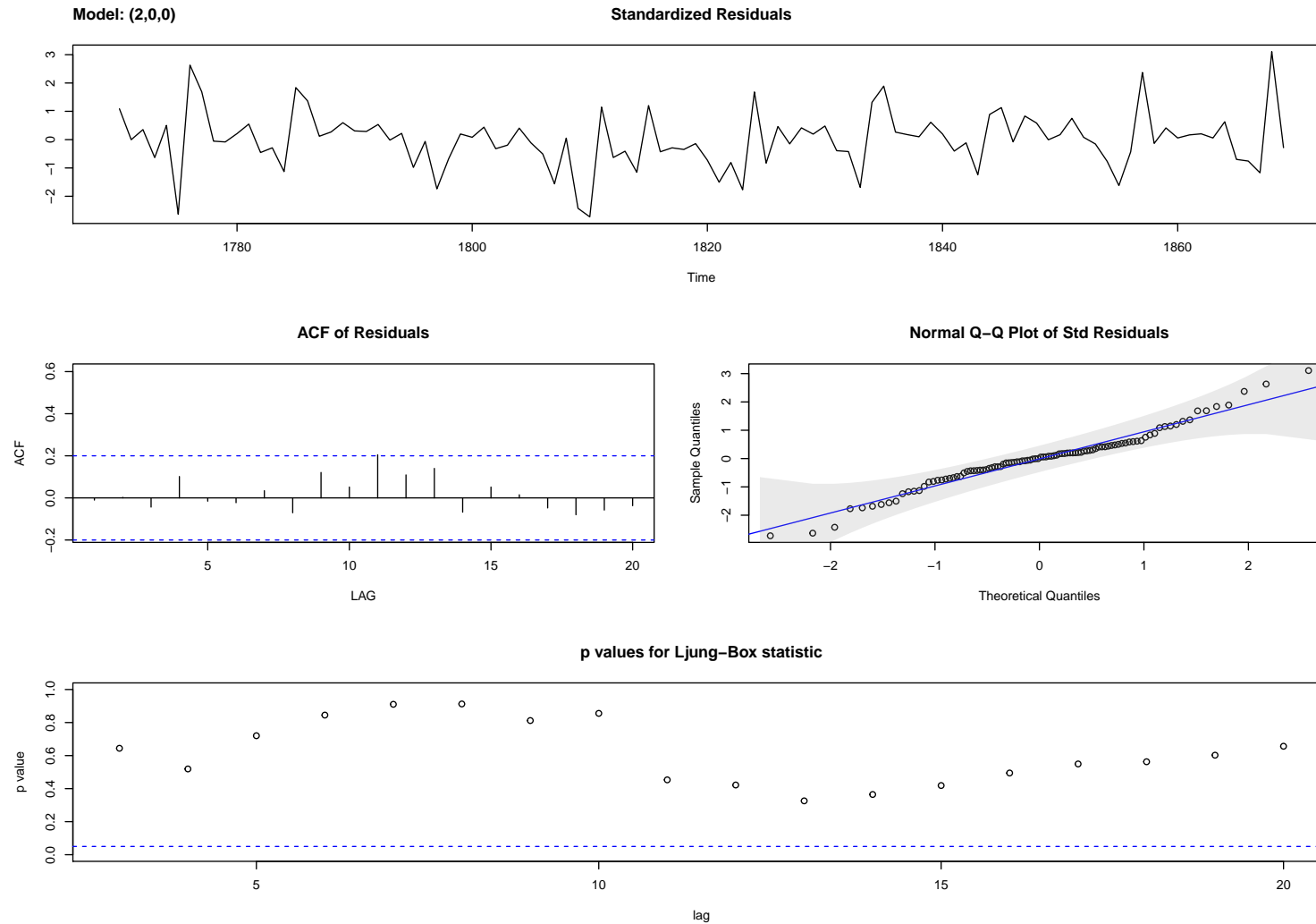
```
Coefficients:  
      1      2  
1.2701 -0.5583
```

```
Order selected 2  sigma^2 estimated as  0.2494
```

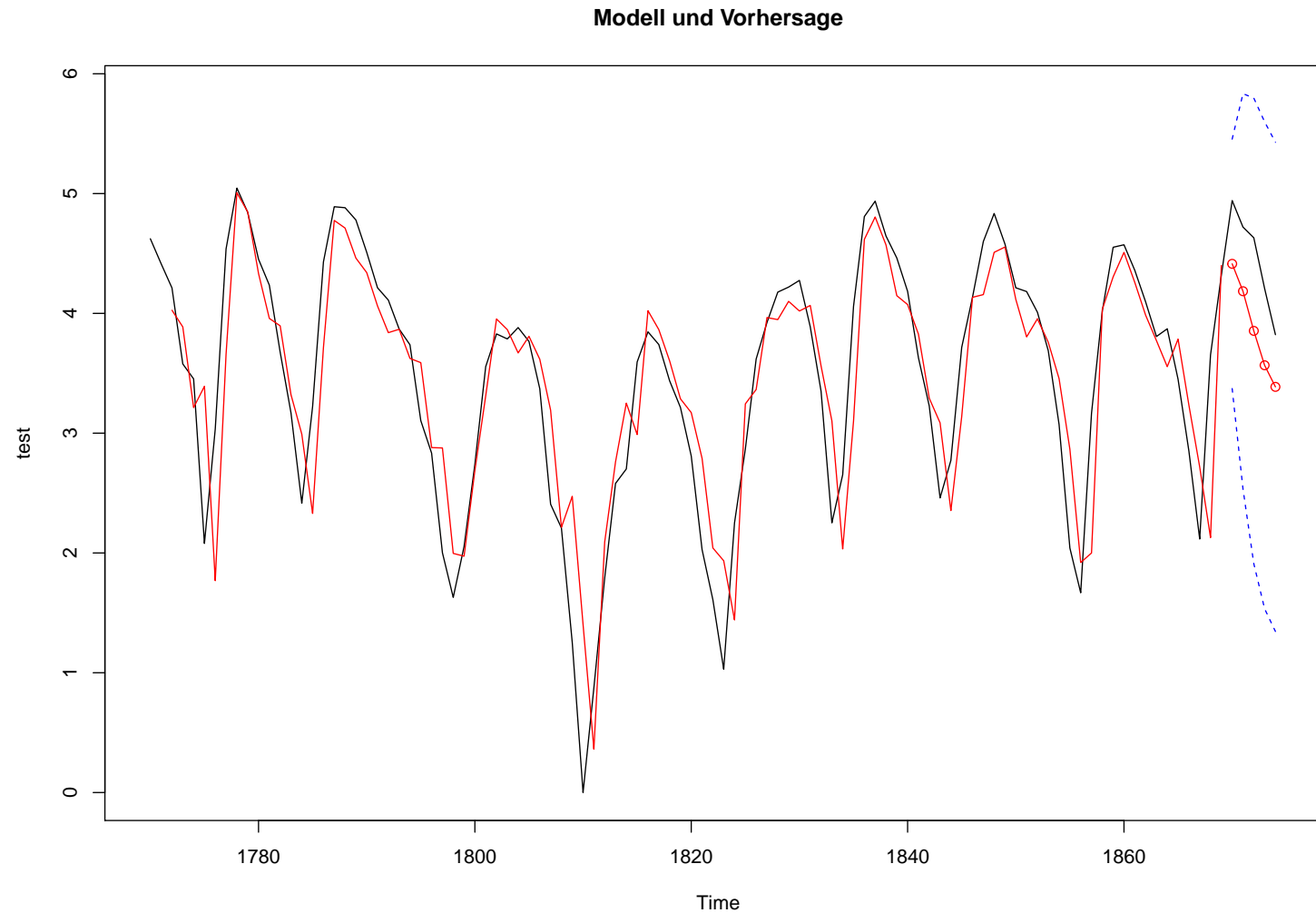
AR-Prozesse: Identifikation der Ordnung



AR-Prozesse: Identifikation der Ordnung



AR-Prozesse: Identifikation der Ordnung



Gliederung der Vorlesung

- 1) Einführung und Grundlagen
- 2) Klassische Zeitreihenanalyse
 - a) Trendbestimmung
 - b) Glättung von Zeitreihen durch Filter (lokale Trendbestimmung)
 - c) Saisonbereinigung
 - d) Exponentielles Glätten, Prognosen
- 3) Modellierung von Reihen durch stochastische Prozesse
 - a) Autoregressive (AR) Modelle
 - b) Moving Average (MA) Modelle
 - c) ARMA Modelle
 - d) ARIMA Modelle

MA(q)-Prozesse

Ein stochastischer Prozess (X_t) heißt ein **Moving Average Prozess der Ordnung q** oder **MA(q)-Prozess**, wenn er von der Form

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q}$$

ist. Dabei ist $(\varepsilon_t) \sim \text{WN}(0, \sigma_\varepsilon^2)$; Konstanten θ_i heißen **Parameter des Prozesses**. Ein MA(q)-Prozess ist **linear, kausal und stationär** für jede Wahl der Parameter θ_i .

Oft wird (o.B.d.A.) $\mathbb{E}[X_t] = \mu = 0$ angenommen und

$$\begin{aligned} X_t &= \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} \\ &= (1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q) \varepsilon_t =: \theta(B) \varepsilon_t \end{aligned} \quad (16)$$

für die Definition eines MA(q)-Prozesses genutzt.

MA(1)-Prozess

Für einen MA(1)-Prozess

$$X_t = \varepsilon_t + \theta \varepsilon_{t-1}$$

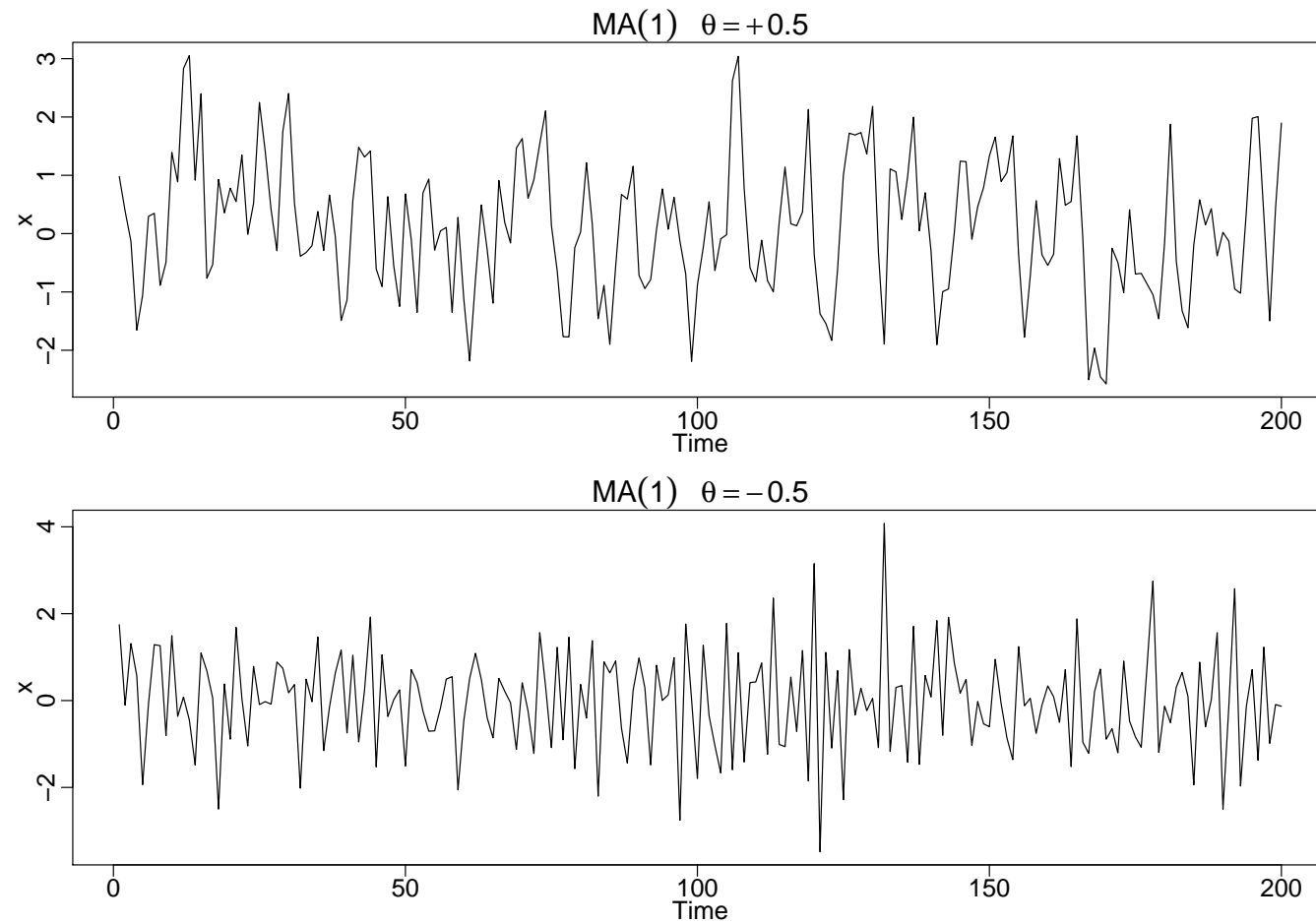
gilt $\mathbb{E}[X_t] = 0$ und

$$\gamma(\tau) = \begin{cases} (1 + \theta^2)\sigma_\varepsilon^2 & \tau = 0 \\ \theta\sigma_\varepsilon^2 & \tau = \pm 1 \\ 0 & \text{sonst} \end{cases} \quad \text{und} \quad \rho(\tau) = \begin{cases} 1 & \tau = 0 \\ \frac{\theta}{1+\theta^2} & \tau = \pm 1 \\ 0 & \text{sonst} \end{cases}$$

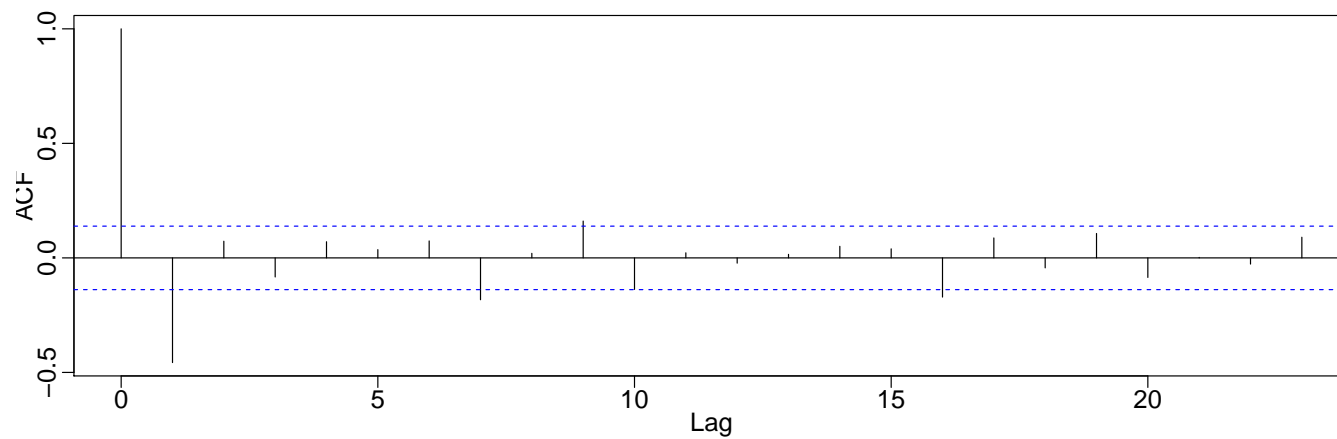
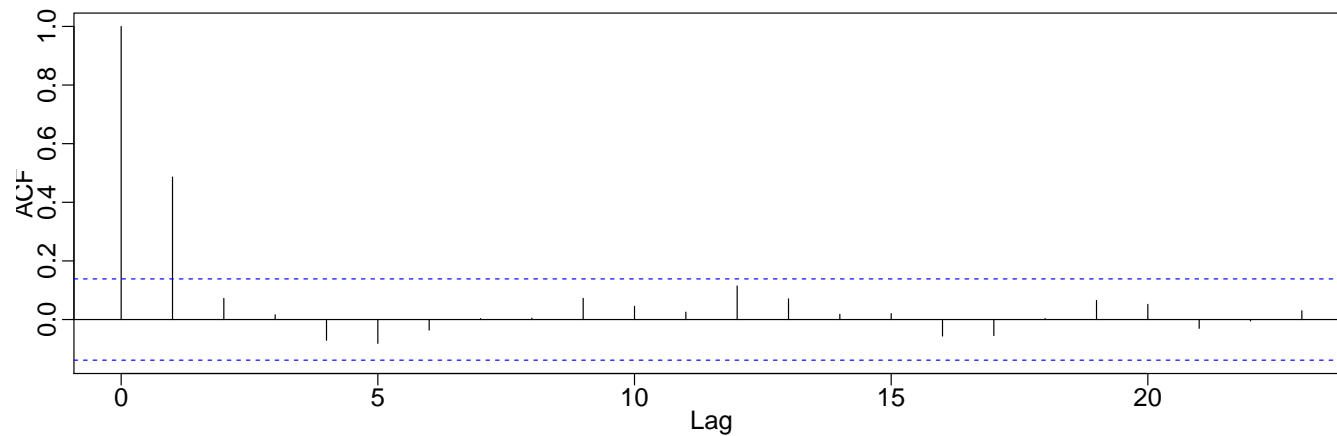
Es gilt immer $|\rho(1)| \leq 0.5$.

Z.B. für $\theta = 0.5$ erhält man $\rho(1) = 0.4$, für $\theta = -0.5$ ist $\rho(1) = -0.4$.

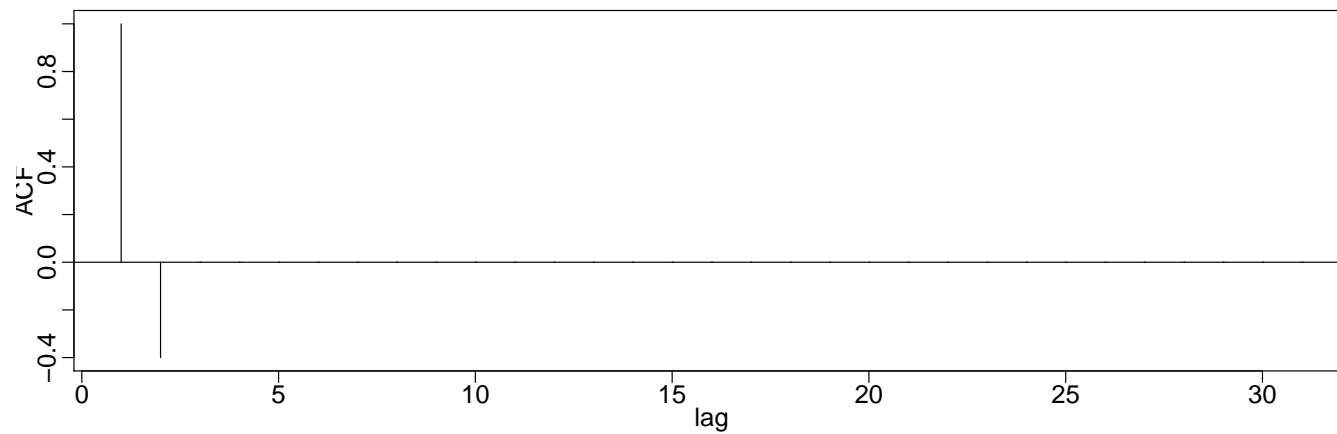
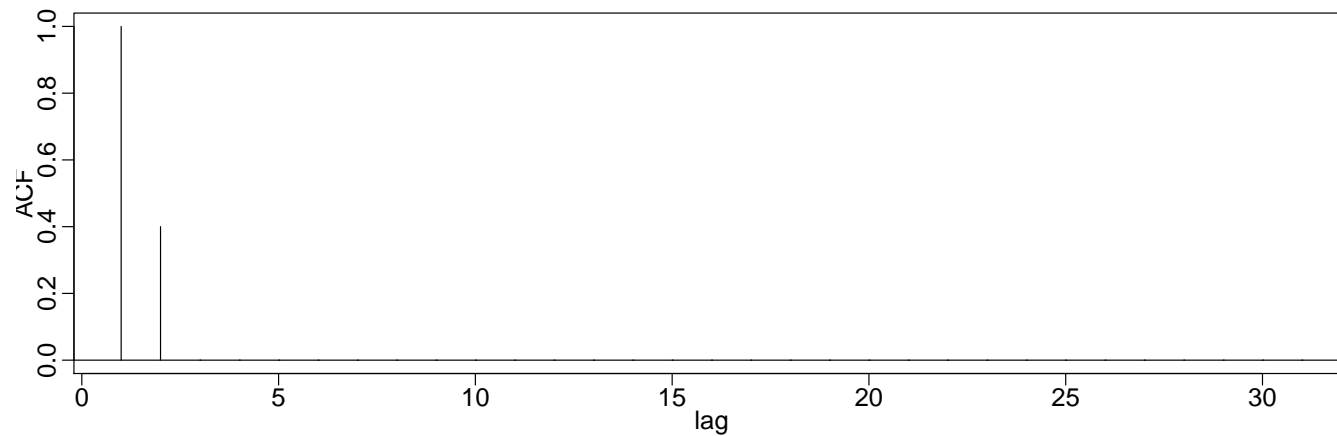
MA(1)-Prozess



MA(1)-Prozess



MA(1)-Prozess



MA(1)-Prozess

```
#Vergleich
par(mfrow=c(2,1), mar=c(3,2,1,0)+.5, mgp=c(1.6,.6,0))
x1=arima.sim(list(order=c(0,0,1), ma=.5), n=200)
x2=arima.sim(list(order=c(0,0,1), ma=-.5), n=200)
plot(x1, ylab="x", main=(expression(MA(1)~~~theta==+.5)))
plot(x2, ylab="x", main=(expression(MA(1)~~~theta==-.5)))
acf(x1)
acf(x2)

ACF = ARMAacf(ar=0, ma=0.5, 30)
plot(ACF, type="h", xlab="lag")
abline(h=0)

ACF = ARMAacf(ar=0, ma=-0.5, 30)
plot(ACF, type="h", xlab="lag")
abline(h=0)
```

MA(1)-Prozess

Für einen MA(1)-Prozess gilt

$$\gamma(\tau) = \begin{cases} (1 + \theta^2)\sigma_\varepsilon^2 & \tau = 0 \\ \theta\sigma_\varepsilon^2 & \tau = \pm 1 \\ 0 & \text{sonst} \end{cases} \quad \text{und} \quad \rho(\tau) = \begin{cases} 1 & \tau = 0 \\ \frac{\theta}{1+\theta^2} & \tau = \pm 1 \\ 0 & \text{sonst} \end{cases}$$

Wegen $\frac{\theta}{1+\theta^2} = \frac{1/\theta}{1+1/\theta^2}$ ist ein MA(1)-Prozess **nicht eindeutig bestimmt** durch seine Parameter. Z.B. $\theta = 5$, $\sigma_\varepsilon = 1$ und $\theta = 1/5$, $\sigma_\varepsilon = 25$ liefern dieselbe Autokovarianzfunktion

$$\gamma(\tau) = \begin{cases} 26 & \tau = 0 \\ 5 & \tau = \pm 1 \\ 0 & \text{sonst.} \end{cases}$$

MA(1)-Prozess

Wegen $\frac{\theta}{1+\theta^2} = \frac{1/\theta}{1+1/\theta^2}$ ist ein MA(1)-Prozess **nicht eindeutig bestimmt** durch seine Parameter. Z.B. $\theta = 5$, $\sigma_\varepsilon = 1$ und $\theta = 1/5$, $\sigma_\varepsilon = 25$ liefern dieselbe Autokovarianzfunktion

$$\gamma(\tau) = \begin{cases} 26 & \tau = 0 \\ 5 & \tau = \pm 1 \\ 0 & \text{sonst,} \end{cases}$$

d.h. die Prozesse

$$X_t = \varepsilon_t + \frac{1}{5}\varepsilon_{t-1}, \quad \varepsilon_t \sim N(0, 25)$$

und

$$\tilde{X}_t = \tilde{\varepsilon}_t + 5\tilde{\varepsilon}_{t-1}, \quad \tilde{\varepsilon}_t \sim N(0, 5)$$

sind ununterscheidbar. Daher führt man das Kriterium der **Invertierbarkeit** ein.

MA(1)-Prozess

Wir haben gesehen, dass ein AR(1)-Prozess mit $|\phi_1| < 1$ sich als ein kausaler linearer Prozess – ein **MA(∞)-Prozess** – schreiben lässt:

$$X_t = \phi_1 X_{t-1} + \varepsilon_t = \sum_{n=0}^{\infty} \phi_1^n \varepsilon_{t-n}$$

Umgekehrt, kann ein MA(1)-Prozess $X_t = \varepsilon_t + \theta \varepsilon_{t-1}$ als ein **AR(∞)-Prozess** dargestellt werden:

$$\begin{aligned} X_t &= \varepsilon_t + \theta \varepsilon_{t-1} = \varepsilon_t + \theta X_{t-1} - \theta^2 \varepsilon_{t-2} \\ &= \varepsilon_t + \theta X_{t-1} - \theta^2 X_{t-2} + \theta^3 \varepsilon_{t-3} = \dots = \varepsilon_t - \sum_{n=1}^{\infty} (-\theta)^n X_{t-n} \end{aligned}$$

Diese Darstellung macht nur Sinn, wenn $|\theta| < 1$ ist. Dann heißt (X_t) **invertierbar** (weil dann $\varepsilon_t = \sum_{n=0}^{\infty} (-\theta)^n X_{t-n} =: \theta^{-1}(B)X_t$, vgl. (16)).

MA(q)-Prozess

Ein MA(q)-Prozess

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} = (1 + \theta_1 B + \cdots + \theta_q B^q) \varepsilon_t$$

ist **invertierbar**, wenn alle Nullstellen seiner **charakteristischen Gleichung**

$$1 + \theta_1 z + \cdots + \theta_q z^q = 0$$

außerhalb des Einheitskreises liegen. Wir werden im Folgendem nur invertierbare MA-Prozesse betrachten.

MA(q)-Prozess

Für die Autokovarianzfunktion eines MA(q)-Prozesses gilt (mit $\theta_0 := 1$)

$$\gamma(\tau) = \begin{cases} \sigma_\varepsilon^2 \sum_{j=0}^q \theta_j^2 & \tau = 0 \\ \sigma_\varepsilon^2 \sum_{j=0}^{q-\tau} \theta_j \theta_{j+\tau} & \tau = 1, \dots, q \\ 0 & \tau > q \end{cases}$$

und für die ACF

$$\rho(\tau) = \begin{cases} \frac{\sum_{j=0}^{q-\tau} \theta_j \theta_{j+\tau}}{\sum_{j=0}^q \theta_j^2} & \tau = 1, \dots, q \\ 0 & \tau > q \end{cases}$$

– eine **nichtlineare** Funktion von den Parametern, im Gegensatz zu AR-Modellen. Das macht Schätzung von Parametern schwieriger.

MA(q)-Prozess

Nach Satz 4 sind die empirischen Autokorrelationen $\hat{\rho}(\tau)$ für $\tau > q$ asymptotisch normalverteilt mit

$$\mathbb{E}[\hat{\rho}(\tau)] = 0, \quad \text{var}(\hat{\rho}(\tau)) = \frac{1}{N}(1 + 2\rho(1)^2 + \dots + 2\rho(q)^2)$$

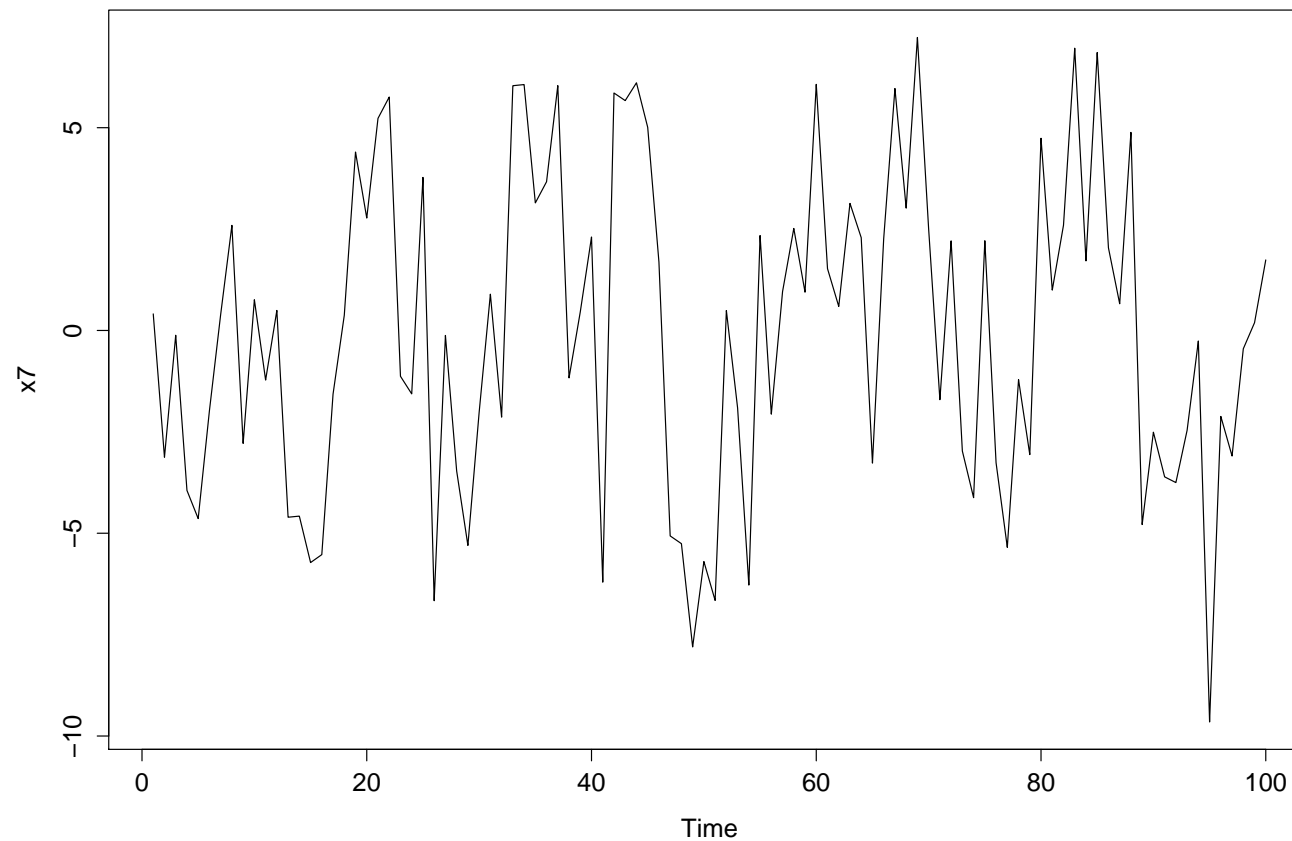
(Formel von Barlett). Dies erlaubt Identifikation von MA-Prozessen: Für jeden Lag τ berechnet man das zugehörige 95%-Konfidenzintervall aus der empirischen ACF:

$$\pm \frac{1.96}{\sqrt{N}} \sqrt{1 + 2\hat{\rho}(1)^2 + \dots + 2\hat{\rho}(\tau)^2}.$$

Der kleinste Lag τ , von dem an dieses Konfidenzintervall 95% aller nachfolgenden Autokorrelationen einschließt ist eine (erste) Schätzung für die Ordnung des Modells.

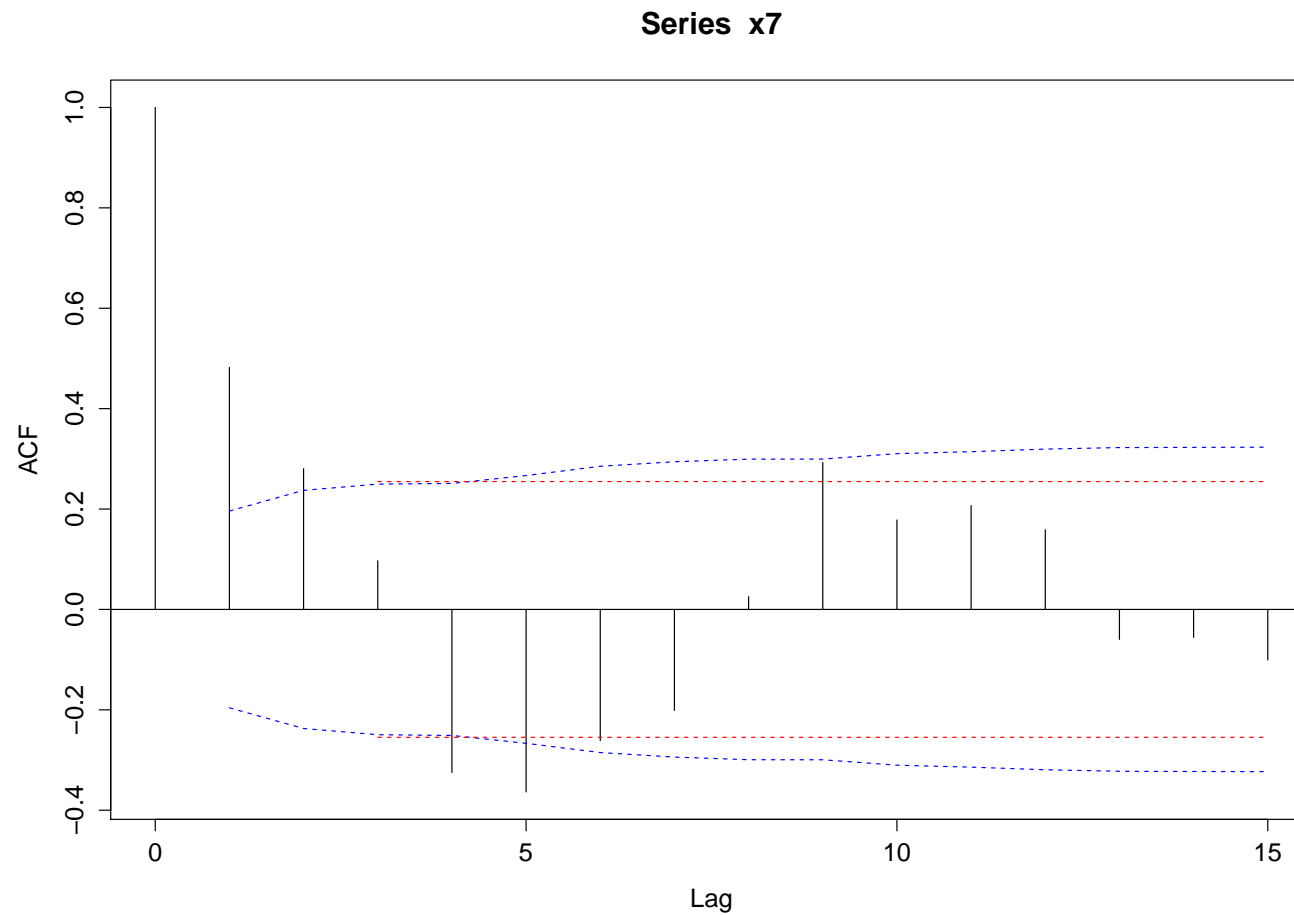
Identifikation von MA-Prozessen

$$X_t = \varepsilon_t + 0.8\varepsilon_{t-1} - 1.8\varepsilon_{t-4} - 1.64\varepsilon_{t-5} - 2.5\varepsilon_{t-7}$$



Identifikation von MA-Prozessen

$$X_t = \varepsilon_t + 0.8\varepsilon_{t-1} - 1.8\varepsilon_{t-4} - 1.64\varepsilon_{t-5} - 2.5\varepsilon_{t-7}$$



Identifikation von MA-Prozessen

```
x7=arima.sim(list(order=c(0,0,7), ma=c(.8, 0,0,-1.8,-1.64,0,-2.5)), n=100)
plot(x7)
acf(x7,ci.type="ma") # ci.type ergibt Barlett-Grenzen
a=acf(x7,ci.type="ma",15)
a$acf[1:3]
lines(c(3,15), rep(2*sqrt((1+2*a$acf[2]^2+2*a$acf[3]^2)/100),2), col=2, lty=2)
lines(c(3,15), rep(-2*sqrt((1+2*a$acf[2]^2+2*a$acf[3]^2)/100),2), col=2, lty=2)
```

MA(q)-Prozess

Was ist mit PACF?

Wegen seiner $AR(\infty)$ Darstellung verschwindet die PACF eines MA-Prozesses nie. Die Werte können mit Durbin-Levinson-Rekursion berechnet werden, z.B. für einen MA(1)-Prozess $X_t = \varepsilon_t + \theta\varepsilon_{t-1}$:

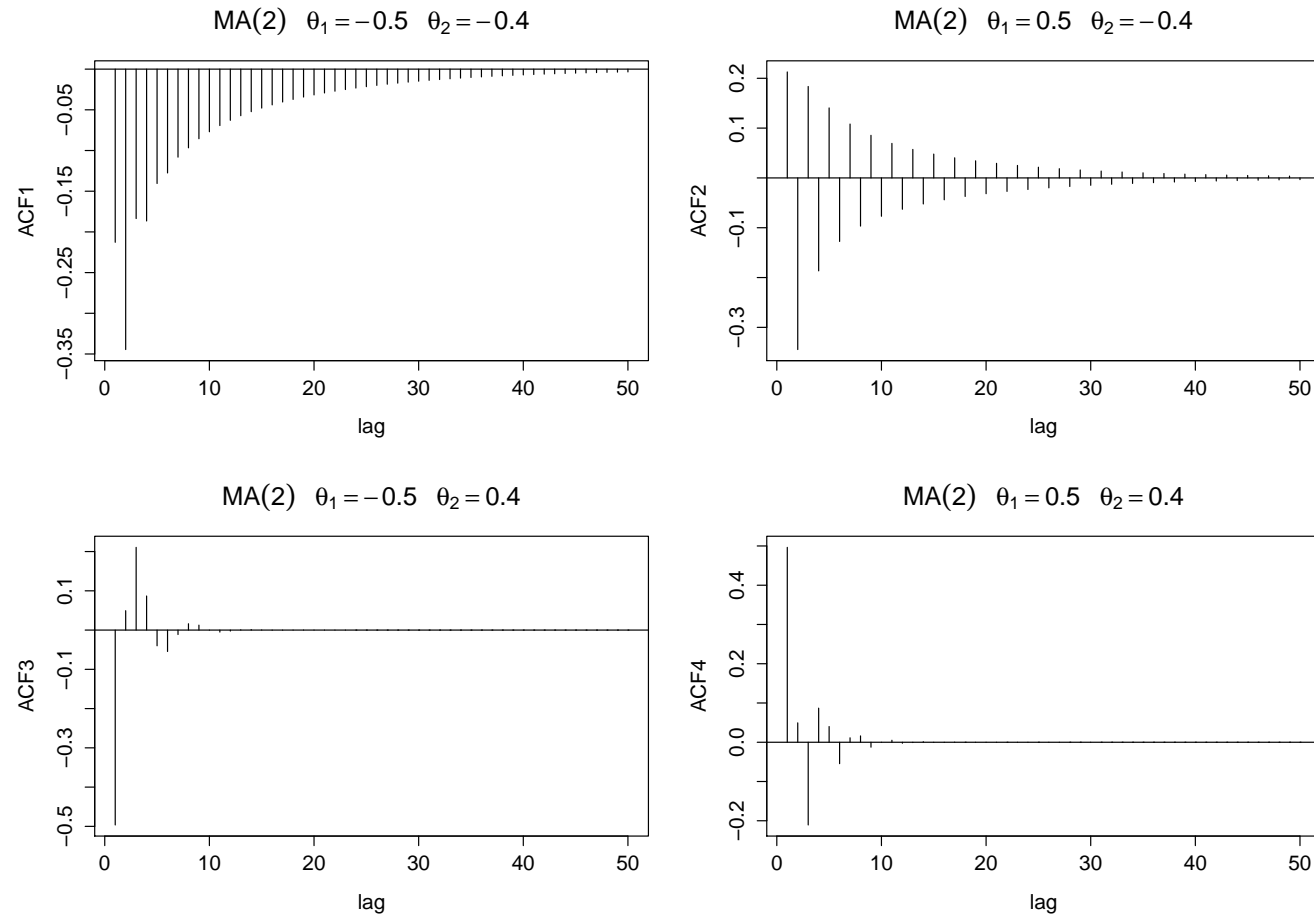
$$\pi_1 = \phi_{11} = \rho(1) = \frac{\theta}{1 + \theta^2},$$

$$\pi_2 = \phi_{22} = \frac{\rho(2) - \rho(1)^2}{1 - \rho(1)^2} = \frac{-\theta^2}{1 + \theta^2 + \theta^4},$$

$$\pi_\tau = \phi_{\tau\tau} = \frac{-(-\theta)^\tau}{1 + \theta^2 + \dots + \theta^{2\tau}}$$

→ exponentielles Abklingen.

MA(q)-Prozess



Schätzen von MA-Parametern

Einige Methoden zur Schätzung der Parameter $\theta_1, \dots, \theta_q$ eines MA(q)-Prozesses sind

- Conditional bzw. Unconditional Least Sum of Squares (CLS bzw. ULS)
- Maximum Likelihood (ML)
- Innovationsalgorithmus (nicht hier)

Schätzung **schwieriger** als bei AR-Modellen wegen der $AR(\infty)$ -Darstellung (“unendliche” Abhängigkeit von den beobachteten Werten). Beide LS und ML führen zu **nichtlinearen** Gleichungen und müssen numerisch gelöst werden.

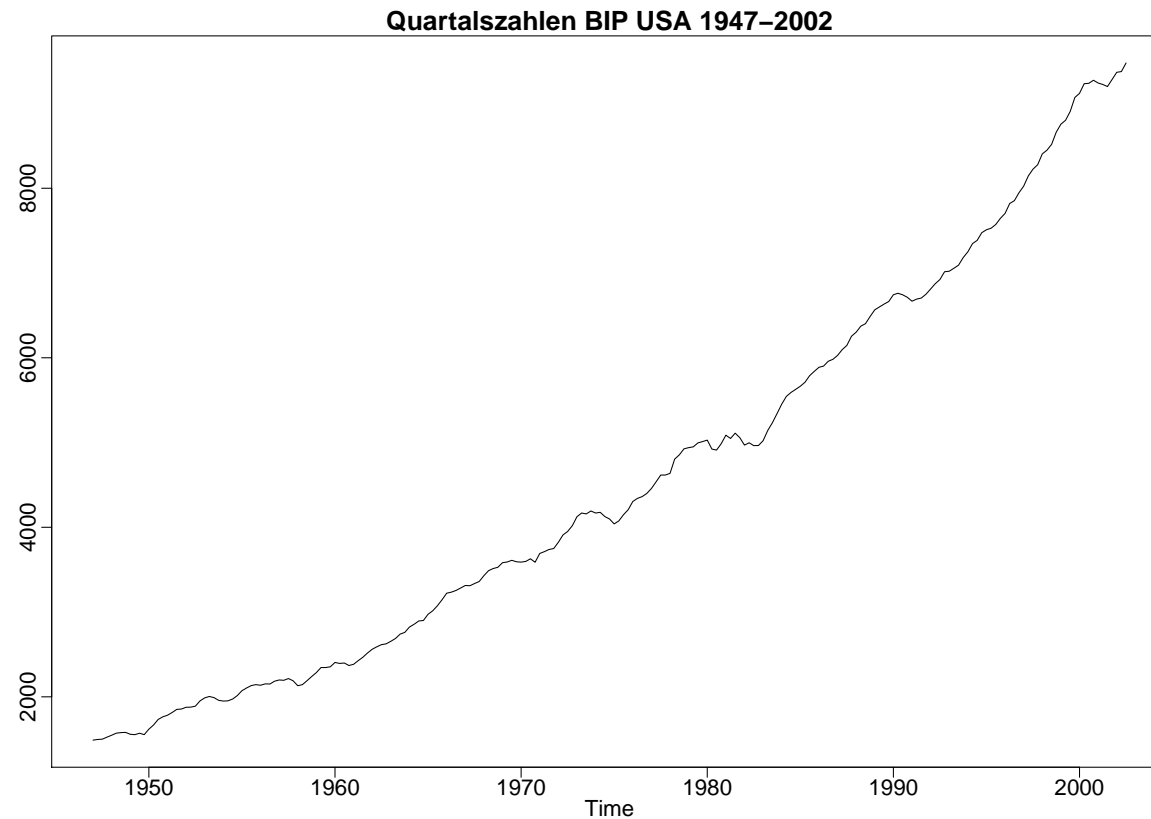
Schätzen von MA-Parametern

Werden die Anfangswerte festgesetzt, so führt es auf **CLS** – die Optimierung ist dann einfacher als bei den “exakten” **ULS** bzw. **UML**. Allerdings ist das Festsetzen von Anfangswerten **problematisch** bei

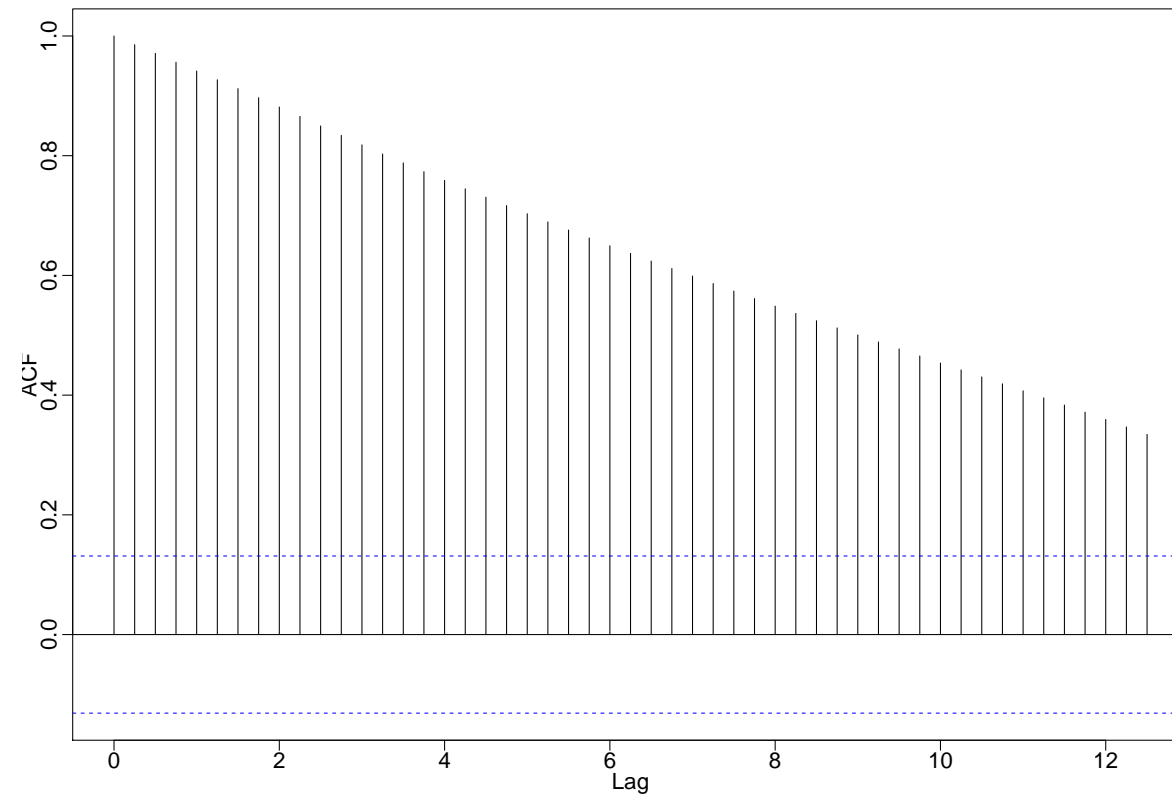
- kurzen Reihen
- Parametern in der Nähe des Invertierbarkeitsbereichs
- Modellen hoher Ordnung

Typische Vorgehensweise (Default bei “arima” in R): Nutze CLS rückwärts, um Anfangswerte zu schätzen (“Backcasting”), und mit diesen Werten dann ML.

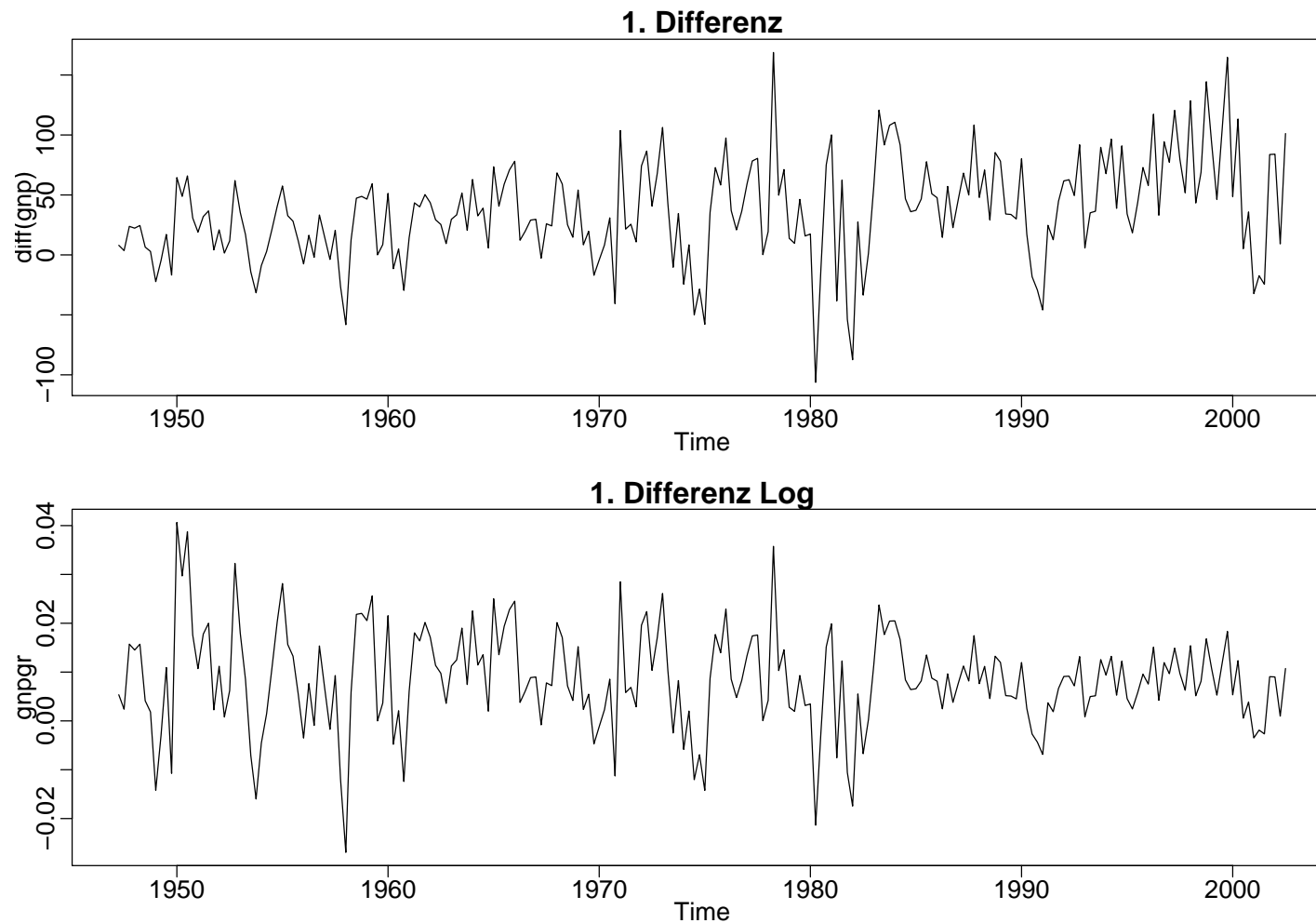
Identifikation von MA-Prozessen



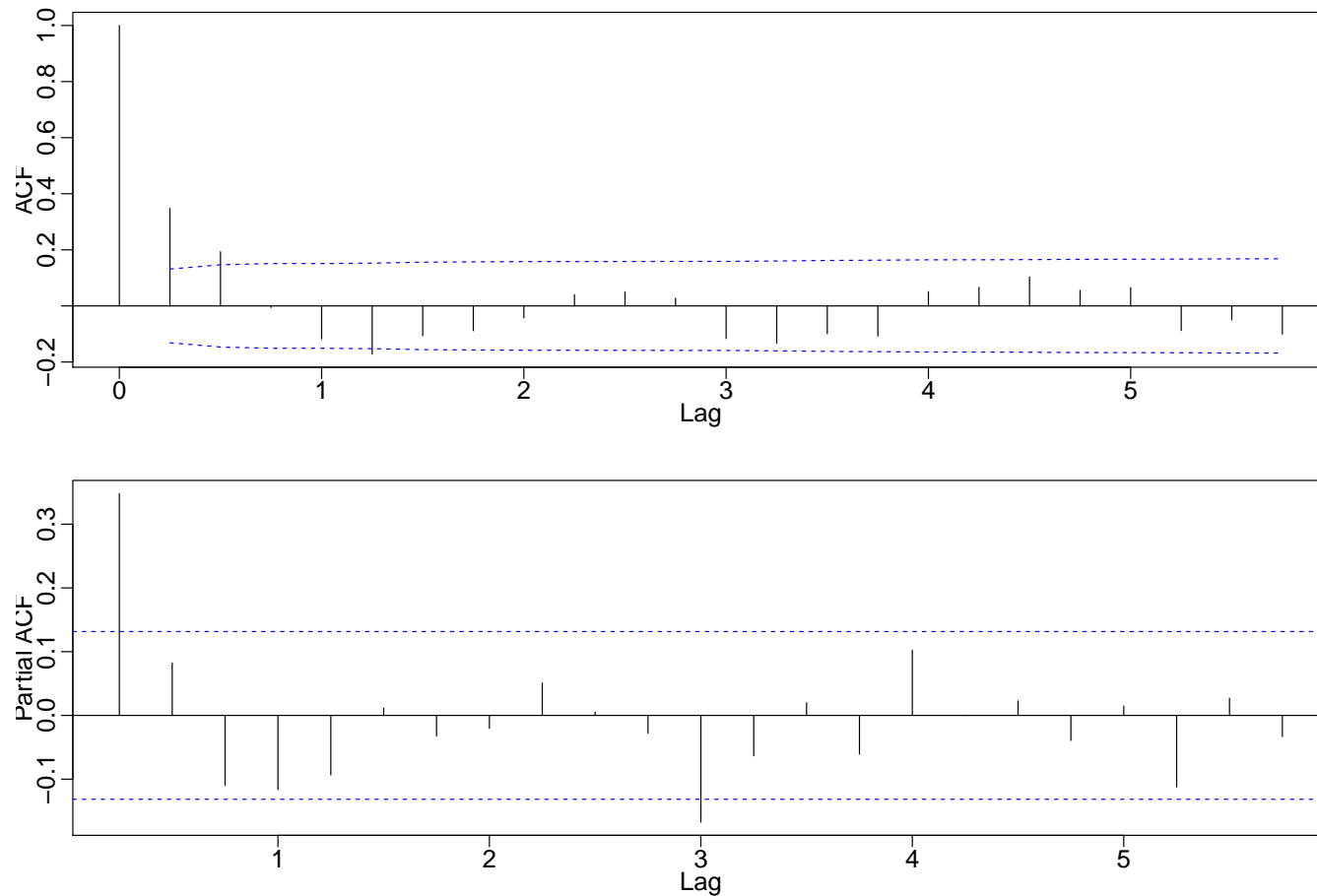
Identifikation von MA-Prozessen



Identifikation von MA-Prozessen



Identifikation von MA-Prozessen



AR(1) oder MA(2)?

Identifikation von MA-Prozessen

```
> bipwr = diff(log(gnp)) # growth rate
> (ar1=arima(bipwr, order=c(1,0,0)))
```

Call:

```
arima(x = bipwr, order = c(1, 0, 0))
```

Coefficients:

	ar1	intercept
	0.3467	0.0083
s.e.	0.0627	0.0010

sigma^2 estimated as 9.03e-05: log likelihood = 718.61, aic = -1431.22

```
> (ma2=arima(bipwr, order=c(0,0,2)))
```

Call:

```
arima(x = bipwr, order = c(0, 0, 2))
```

Coefficients:

	ma1	ma2	intercept
	0.3028	0.2035	0.0083
s.e.	0.0654	0.0644	0.0010

sigma^2 estimated as 8.919e-05: log likelihood = 719.96, aic = -1431.93

```
> BIC(ar1); BIC(ma2)
```

```
[1] -1421.013
```

```
[1] -1418.319
```

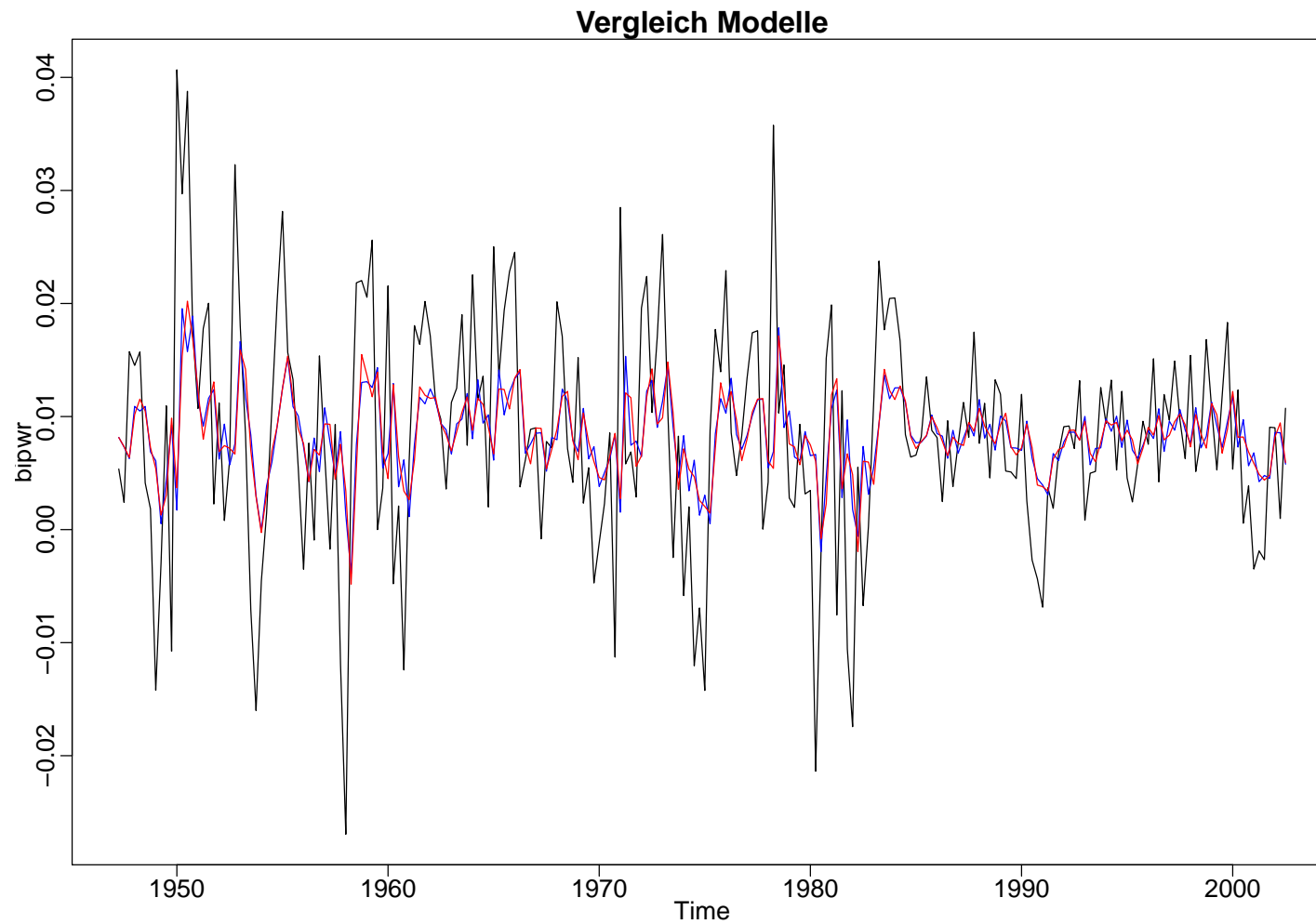
```
> ARMAtoMA(ar=ar1$coef[1], ma=0, 3) # koeffizienten der ma-darstellung
```

```
[1] 0.34665450 0.12016934 0.04165724
```

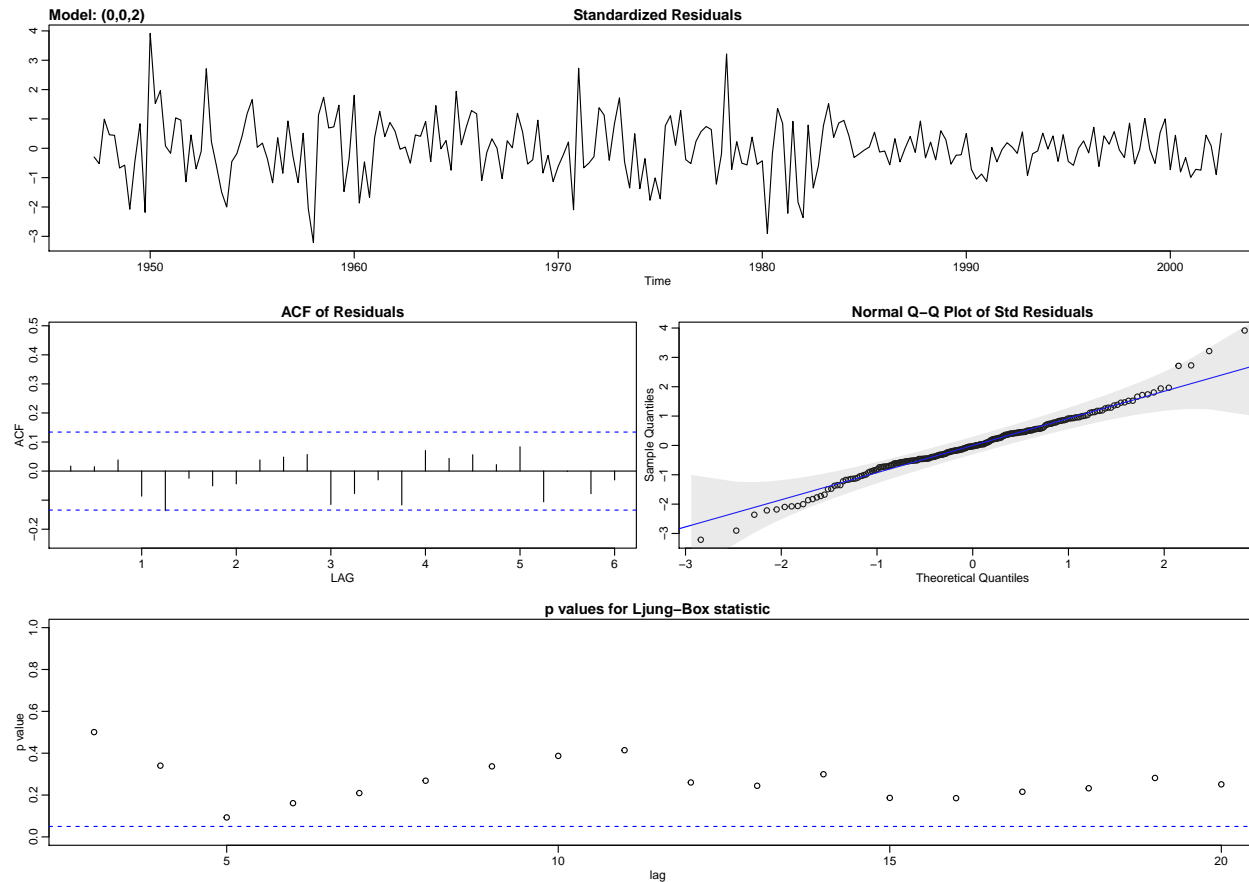
```
> ma2$coef[1:2] #ma1,2
```

	ma1	ma2
	0.3028085	0.2035189

Identifikation von MA-Prozessen



Identifikation von MA-Prozessen



mit

```
sarima(bipwr, 1, 0, 0) # AR(1)  
sarima(bipwr, 0, 0, 2) # MA(2)
```