Chapter 4

ARIMA–Models

4.1 Introductionary Remarks

Forecasting based on ARIMA (autoregressive integrated moving averages) models, commonly know as the Box–Jenkins approach, comprises following stages:

i.) *Model identification*

ii.) *Parameter estimation*

iii.) *Diagnostic checking*

These stages are repeated until a "suitable" model for the given data has been identified (e.g. for prediction). The following three sections show some facilities that **R** offers for assisting the three stages in the Box–Jenkins approach.

4.2 Analysis of Autocorrelations and Partial Autocorrelations

A first step in analyzing time series is to examine the autocorrelations (ACF) and partial autocorrelations (PACF). **R** provides the functions `acf( )` and `pacf( )` for computing and plotting of ACF and PACF. The order of "pure" AR and MA processes can be identified from the ACF and PACF as shown below:

```
sim.ar<-arima.sim(list(ar=c(0.4,0.4)),n=1000)
sim.ma<-arima.sim(list(ma=c(0.6,-0.4)),n=1000)
par(mfrow=c(2,2))
acf(sim.ar,main="ACF of AR(2) process")
acf(sim.ma,main="ACF of MA(2) process")
pacf(sim.ar,main="PACF of AR(2) process")
pacf(sim.ma,main="PACF of MA(2) process")
```

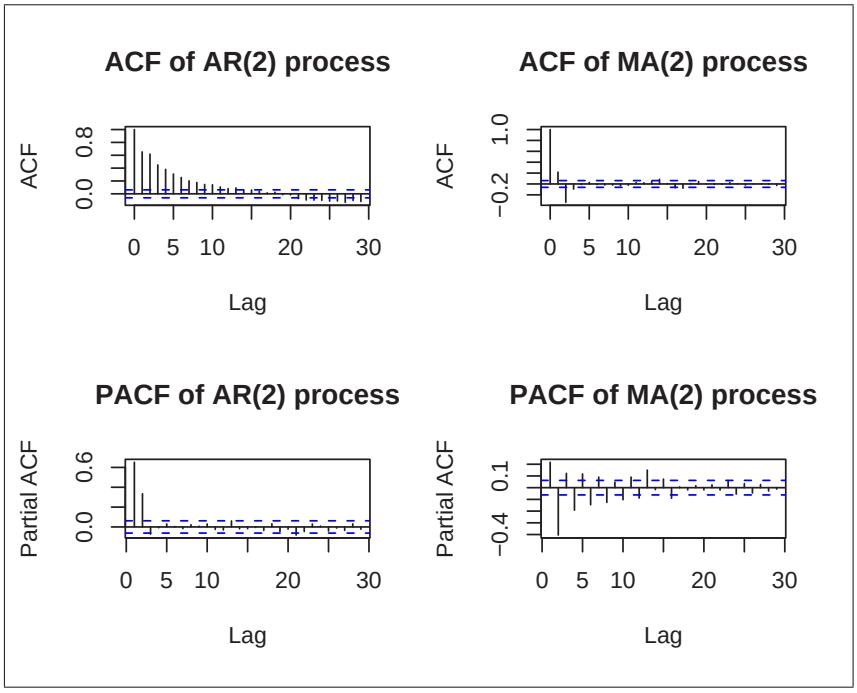

Figure 4.1: ACF and PACF of AR– and MA–models

The function `arima.sim( )` was used to simulate ARIMA(p,d,q)–models ; in the first line 1000 observations of an ARIMA(2,0,0)–model (i.e. AR(2)–model) were simulated and saved as `sim.ar`. Equivalently, the second line simulated 1000 observations from a MA(2)–model and saved them to `sim.ma`.
An useful command for graphical displays is `par(mfrow=c(h,v))` which splits the graphics window into (`h`×`v`) regions — in this case we have set up 4 seperate regions within the graphics window.
The last four lines created the ACF and PACF plots of the two simulated processes. Note that by default the plots include confidence intervals (based on uncorrelated series).

4.3 Parameter–Estimation of ARIMA–Models

Once the order of the ARIMA(p,d,q)–model has been specified, the function `arima( )` from the `ts`–library can be used to estimate the parameters:

```
arima(data,order=c(p,d,q))
```

Fitting e.g. an ARIMA(1,0,1)–model on the `LakeHuron`–dataset (annual levels of the Lake Huron from 1875 to 1972) is done using

The Box–Pierce (and Ljung–Box) test examines the Null of independently distributed residuals. It's derived from the idea that the residuals of a "correctly specified" model are independently distributed. If the residuals are not, then they come from a miss–specified model. The function `Box.test( )` computes the test statistic for a given lag:

```
Box.test(fit$residuals,lag=1)
```

4.5 Prediction of ARIMA–Models

Once a model has been identified and its parameters have been estimated, one purpose is to predict future values of a time series. Lets assume, that we are satisfied with the fit of an ARIMA(1,0,1)–model to the `LakeHuron`–data:

```
fit<-arima(LakeHuron,order=c(1,0,1))
```

As with Exponential Smoothing, the function `predict( )` can be used for predicting future values of the levels under the model:

```
LH.pred<-predict(fit,n.ahead=8)
```

Here we have predicted the levels of Lake Huron for the next 8 years (i.e. until 1980). In this case, `LH.pred` is a list containing two entries, the predicted values `LH.pred$pred` and the standard errors of the prediction `LH.pred$se`. Using a rule of thumb for an approximate confidence interval (95%) of the prediction, *"prediction ± 2·SE"*, one can e.g. plot the Lake Huron data, predicted values and an approximate confidence interval:

```
plot(LakeHuron,xlim=c(1875,1980),ylim=c(575,584))
LH.pred<-predict(fit,n.ahead=8)
lines(LH.pred$pred,col="red")
lines(LH.pred$pred+2*LH.pred$se,col="red",lty=3)
lines(LH.pred$pred-2*LH.pred$se,col="red",lty=3)
```

First, the levels of Lake Huron are plotted. To leave some space for adding the predicted values, the x-axis has been "limited" from 1875 to 1980 with `xlim=c(1875,1980)` ; the use of `ylim` is purely for cosmetic purposes here. The prediction takes place in the second line using `predict( )` on our fitted model. Adding the prediction and the approximate confidence interval is done in the last three lines. The confidence bands are drawn as a red, dotted line (using the options `col="red"` and `lty=3`):

```
data(LakeHuron)
fit<-arima(LakeHuron,order=c(1,0,1))
```

Here, `fit` is a list containing e.g. the coefficients (`fit$coef`), residuals (`fit$residuals`) and the Akaike Information Criterion AIC (`fit$aic`).

4.4 Diagnostic Checking

A first step in diagnostic checking of fitted models is to analyze the residuals from the fit for any signs of non–randomness. **R** has the function `tsdiag( )`, which produces a diagnostic plot of a fitted time series model:

```
fit<-arima(LakeHuron,order=c(1,0,1))
tsdiag(fit)
```

It produces following output containing a plot of the residuals, the autocorrelation of the residuals and the p-values of the Ljung–Box statistic for the first 10 lags:

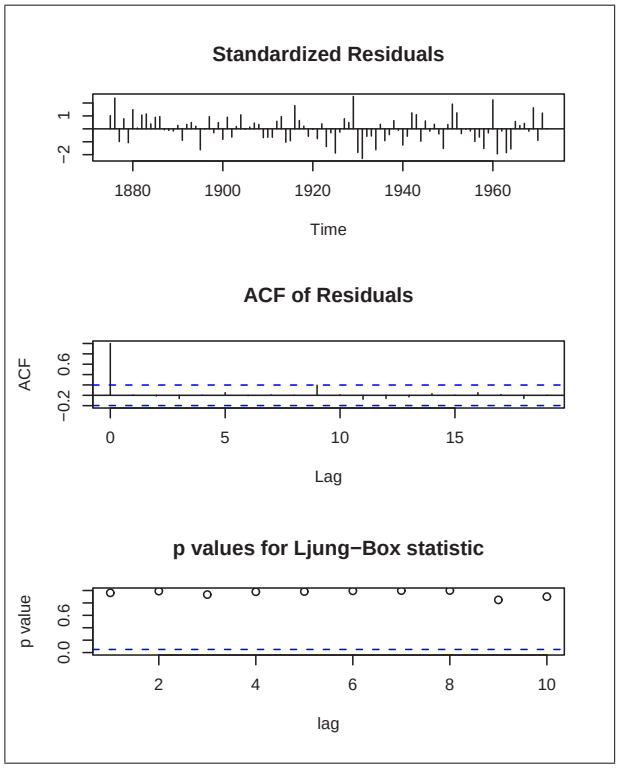

Figure 4.2: Output from `tsdiag`

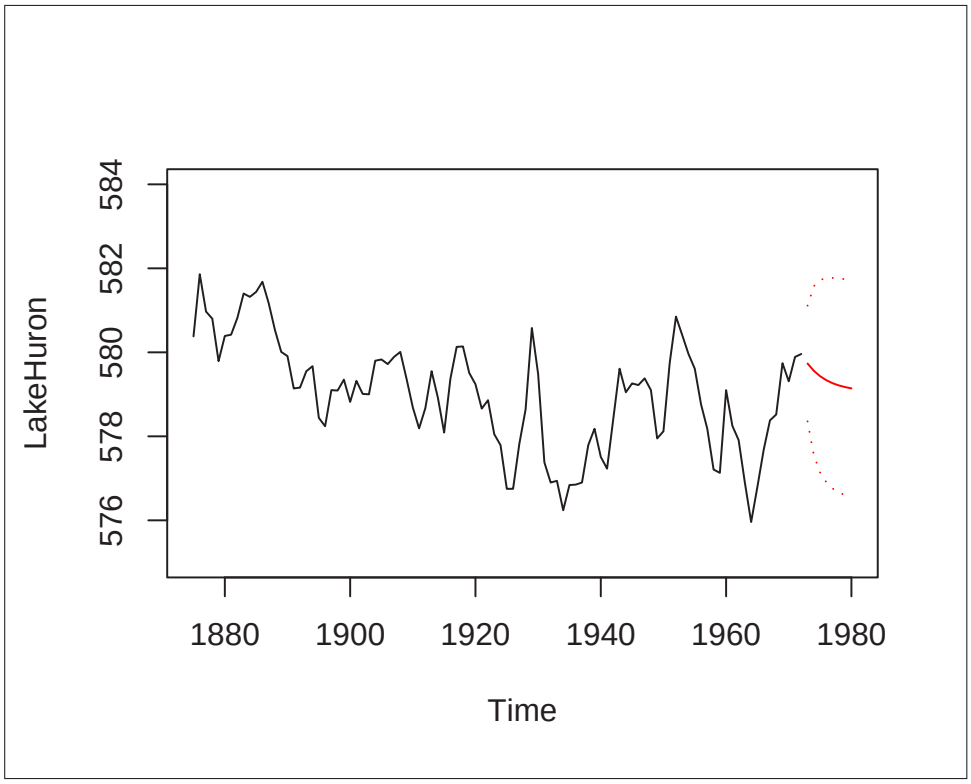

Figure 4.3: Lake Huron levels and predicted values