STAT 443: Time Series and Forecasting

Chapter 3

Estimation and Model Fitting for Time Series

Statistics is the grammar of science.

/Karl Pearson/

Overview

Given the class of stochastic models introduced in Chapter 2, how to decide which of the models would be a good "fit" for the data?

1. Estimate the process mean and autocorrelation function
   - Study properties of the proposed estimators (e.g., bias, efficiency)
2. Find a suitable model from the class of ARMA processes
   - The sample acf would be the basis for model selection, in which case we deal with estimation or inference in the time domain
3. Fit parameters of the chosen model
4. Assess goodness-of-fit (model diagnostics)

Estimation in the time domain - setup

- Time series data: $x_1, \ldots, x_n$
- Assumption: the series $\{x_t\}$ is either stationary or has been pre-processed (e.g., by removing trend and/or seasonal variation) to look stationary
- We will be looking for a simple (parsimonious) model from the class of ARMA(p,q) models that provides a good fit to the data
- Notation: an estimate of a parameter is denoted by the same symbol with a "hat" on it; e.g. $\hat{\alpha}$ is an estimate for parameter α
- The stochastic process $\{X_t\}$ can be thought of as the data generating process
  - μ, σ_X^2 and $\gamma(h)$ denote the mean, variance and autocovariance at lag h of X_t, respectively

Estimation of the autocovariance function (acvf)

- The acvf $\gamma(h)$ is usually estimated by the sample acvf:

$$c_h := \frac{1}{n}\sum_{t=1}^{n-h}(X_t - \overline{X})(X_{t+h} - \overline{X})$$

  - c_h is a biased estimator of $\gamma(h)$; it can be shown

$$\mathbb{E}(c_h) = \gamma(h) + B_n,$$

    where the bias B_n is of order $1/n$
  - However, $B_n \to 0$ as $n \to \infty$, thus c_h is asymptotically unbiased for $\gamma(h)$
  - So, c_h is a reasonable choice for $\hat{\gamma}(h)$ when n is large

- There are techniques to reduce the bias (e.g., a "jack-knife" estimator has bias of order $1/n^2$), but few if any software packages implement it

Estimation of the autocorrelation function (acf)

- The acf $\rho(h)$ is estimated by the sample acf: $r_h := \frac{c_h}{c_0}$
- As mentioned earlier, for a completely random process (e.g., white noise), under some weak conditions,

$$r_h \sim \mathcal{N}(-1/n, 1/n) \qquad \text{asymptotically}$$

- Examining the correlogram, the plot of r_h against lag h, is often helpful in determining which ARMA model might be appropriate

  Recall:

  - For an MA(q) process, the acf will cut off sharply at lag q
  - For an AR(p) process, the acf will decay

Estimation of the mean of the process

- Unlike in classic statistics for i.i.d. data, the problem of the estimation of the (process) mean for time series is less straightforward

- Consider the following hypothetical situation:

  - Suppose we are given m realizations of a stochastic process, i.e., m time series each of length n, and let $\overline{X}_j$ be the sample mean for each of them ($j = 1, \ldots, m$)

  - Then the mean of these sample means $\hat{\mu} = \frac{1}{m} \sum_{j=1}^{m} \overline{X}_j$ converges to the true mean μ in mean square:

$$\lim_{m \to \infty} \mathbb{E}[(\hat{\mu} - \mu)^2] = 0$$

- But in reality m=1 as only one sequence of observations is available!

Estimation of the mean of the process (cont'd)

- This gives rise to the question:

  To what extent does the mean of a sample $\overline{X} = \frac{1}{n}\sum_{t=1}^{n} X_t$ constitute a good estimator of μ?

- It can be shown, provided the sample comes from a stationary process for which $\gamma(h) \to 0$ as $h \to \infty$, the following holds:

  - $\overline{X}$ is unbiased for μ: $\mathbb{E}(\overline{X}) = \mu$
  - $\overline{X}$ is consistent for μ: $Var(\overline{X}) \to 0$ as $n \to \infty$

How does temporal dependence affect precision of $\overline{X}$?

Activity: Properties of the Sample Mean

- For i.i.d. data: $Var(\overline{X}) = \frac{1}{n^2} Var(X_1 + \cdots X_n) = \frac{\sigma_X^2}{n}$
- For correlated data:

$$Var(\overline{X}) = \frac{\sigma_X^2}{n}\left(1 + 2\sum_{h=1}^{n-1}\left(1 - \frac{h}{n}\right)\rho(h)\right)$$

  - ✤ The term in the bracket can be quite large if autocorrelations $\rho(h)$ are large

Example: For an AR(1) process $X_t - \mu = \alpha(X_{t-1} - \mu) + Z_t$

$$Var(\overline{X}) \approx \frac{\sigma_X^2}{n}\left(\frac{1+\alpha}{1-\alpha}\right)$$

- The factor $\frac{1+\alpha}{1-\alpha}$ is what differentiates variability of $\overline{X}$ for the AR(1) process from the i.i.d. sequence

- If $\alpha > 0$, then $\frac{1+\alpha}{1-\alpha} > 1$ and hence variance of $\overline{X}$ is higher than in the i.i.d. case due to positive autocorrelations in the data
  - if one observation is above μ, then subsequent observations are also likely to be above μ which adversely impacts $\overline{X}$ as an estimator of μ

- If $\alpha < 0$, variance of $\overline{X}$ is lower than in the i.i.d. case
  - negatively autocorrelated series will tend to avoid sequences of observations on the same side of μ

Fitting an AR model

Suppose, after examining the correlogram, we decide that an AR model is suitable for the data in hand

The next steps involve:

1. determine the order p
2. estimate the parameters μ, $\alpha_1, \ldots, \alpha_p$ and σ^2

Remarks:

- Step 2 is fairly straightforward using standard statistical estimation methods
- Step 1 can in general be quite hard
- So we first discuss the easier problem in Step 2 and then return to Step 1

Parameter estimation for an AR(p) model

The general AR(p) process with mean μ:

$$\boxed{X_t - \mu = \alpha_1(X_{t-1} - \mu) + \cdots + \alpha_p(X_{t-p} - \mu) + Z_t}$$

- Natural procedure: least squares estimation
  (note: the process is essentially a linear regression model)
  i.e.,

  $$(\hat{\mu}, \hat{\alpha}_1, \ldots, \hat{\alpha}_p) = \underset{\mu, \alpha_1, \ldots, \alpha_p}{\arg\min}\ S,$$

  where

  $$S := \sum_{t=p+1}^{n} (x_t - \mu - \alpha_1(x_{t-1} - \mu) - \cdots - \alpha_p(x_{t-p} - \mu))^2$$

  is the sum of squared errors, the differences between the observed and model-predicted values

Parameter estimation - AR(1) model

A look at the AR(1) case:

$$\boxed{X_t - \mu = \alpha(X_{t-1} - \mu) + Z_t, \qquad |\alpha| < 1, \qquad \{Z_t\} \sim WN(0, \sigma^2)}$$

- The minimization problem can be solved by hand, and with mild approximations we get the following estimates:

$$\hat{\mu} = \overline{x}, \qquad \hat{\alpha} = r_1 \quad \text{(sample acf at lag 1)}$$

- This is appealing since recall $\rho(h) = \alpha^{|h|}$ and so $\rho(1) = \alpha$
- To estimate σ^2, we can use the residual mean square:

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{t=2}^{n} \hat{z}_t^2$$

where $\hat{z}_t = (x_t - \hat{\mu}) - \hat{\alpha}(x_{t-1} - \hat{\mu})$ is the fitted residual at time t

Parameter estimation - AR(2) model

$$\boxed{X_t - \mu = \alpha_1(X_{t-1} - \mu) + \alpha_2(X_{t-2} - \mu) + Z_t, \qquad \{Z_t\} \sim WN(0, \sigma^2)}$$

- The least squares procedure gives:

$$\hat{\mu} = \overline{x}, \qquad \hat{\alpha}_1 = \frac{r_1(1 - r_2)}{1 - r_1^2}, \qquad \hat{\alpha}_2 = \frac{r_2 - r_1^2}{1 - r_1^2}$$

- Note: if we were to fit an AR(2) process to data which really comes from an AR(1) process (and so $\rho(2) = \alpha_1^2$), then the estimates above reduce to $\hat{\alpha}_1 \approx r_1$ and $\hat{\alpha}_2 \approx 0$, which is intuitively appealing

- The value $\hat{\alpha}_2$ is called the (sample) partial autocorrelation coefficient of order 2 as it measures the "extra" correlation between X_t and X_{t-2} not accounted for by $\hat{\alpha}_1$

Determining the order of an AR model

- As already mentioned, it can be hard to determine a suitable value p for an AR(p) model
  - For an AR process, the acf always decays to zero with the lag (either exponentially or as a "damped" sine curve), but this does not suggest the order
- One idea is to make use of the (sample) partial autocorrelation function (pacf)
- In analogy with the AR(2) case we have just seen, we can extend the definition of the partial autocorrelation coefficient to general order
  - When fitting an AR(k) process, the last coefficient α_k measures the "extra" correlation between X_t and X_{t-k} not accounted for by the AR(k-1) model

Determining the order of an AR model (cont'd)

- The sample partial autocorrelation coefficient of order k, denoted $\hat{\alpha}_{kk}$, is an estimate of the coefficient on the term in the model of the highest lag

- For a true AR(p) process, the pacf should "cut-off" at order p

- For n large, we have approximately

$$\hat{\alpha}_{kk} \sim \mathcal{N}(0, 1/n), \qquad k > p$$

  which can be used to determine those partial autocorrelations which are significantly different from zero

  - Plot the sample pacf $\{\hat{\alpha}_{kk}\}_{k=1,2,\ldots}$
  - Choose order p so that $\hat{\alpha}_{kk}$'s remain mostly within the range $\pm 2/\sqrt{n}$ for $k > p$

Activity: AR processes and the Partial Autocorrelation Function

Activity: Model Fitting

Fitting an MA model

- The order q can be determined using the correlogram as the acf for an MA(q) process "cuts-off" to zero at lag q

- Parameter estimation, however, is more involved than in the AR case; while several approaches exist, we briefly outline one of them, which is most commonly used

Parameter estimation - MA(1) model

$$\boxed{X_t = \mu + Z_t + \beta Z_{t-1}, \qquad \{Z_t\} \sim WN(0, \sigma^2)}$$

- Choose starting values for estimates of μ and β
  - E.g. $\hat{\mu} = \overline{x}$ and the model-based estimate of β: solution to $r_1 = \dfrac{\hat{\beta}}{1 + \hat{\beta}^2}$
- From the model equation above we have $Z_t = X_t - \mu - \beta Z_{t-1}$
- Recursively writing down residuals starting with $z_0 = 0$

$$\begin{aligned} z_1 &= x_1 - \mu \\ z_2 &= x_2 - \mu - \beta z_1 \\ &\vdots \end{aligned}$$

  we can calculate the residual sum of squares $RSS := \sum_{t=1}^{n} z_t^2$
- A numeric minimization of RSS can be done using e.g. grid search over a range of values for the pair $(\hat{\mu}, \hat{\beta})$

Parameter estimation - MA processes

- The same idea extends naturally to MA processes of higher order
- As before, σ^2 can be estimated by the residual mean square

Fitting an ARIMA model

- Fitting a general ARIMA(p,d,q) model is not an exact science
- The time plot and a very slowly decaying correlogram at large lags will indicate departures from stationarity
- For a (non-seasonal) non-stationary series, it is common to difference it to make it appear stationary
  - Differencing once suffices in most cases, so usually $d = 1$ or occasionally $d = 2$
- An ARMA model can then be fitted to the differenced series
  - Parameter estimation must be done iteratively as for the MA processes

Fitting an ARIMA model - final remarks

- "Computer revolution" opened the door for a wide use of the maximum likelihood (ML) estimation, which is another commonly used method to fit time series models
  - For very long time series, the full ML estimation might still be computationally prohibitive though
  - Conditional ML estimation may be adopted in this case, often leading to fairly similar results
- Statistical software packages will often allow ARIMA(p, d, q) models to be fitted up to specified values of p, d and q, and may suggest a choice from those models considered, though experience and parsimony should determine the final model adopted

ARMA models and sample acf and pacf

- To re-iterate: the acf and pacf are often helpful in choosing a suitable model from the ARMA class
- Below is the summary of the expected behaviour of these functions for each model, which can be used as a rule of thumb when examining the sample acf and pacf

Model	Acf	Pacf
MA(q)	Cuts-off at lag q	Tails off, no pattern
AR(p)	Tails off (exponentially or like a "damped" sine wave)	Cuts-off at lag p
ARMA(p,q)	No pattern up to lag q, then tails off as in AR case	Tails off, no pattern

Activity: Model specification using sample acf and pacf

Model specification for a SARIMA$(p, d, q) \times (P, D, Q)_s$ process

1. Using the time plot, explore features of the time series such as trend and/or seasonality

2. If necessary, apply suitable differencing to remove non-stationarity
   - Series with seasonal variation of period s but no trend: difference at lag s
     - $Y_t = \nabla_s X_t = X_t - X_{t-s}$ (R code: `y = diff(x, lag=s)`)
   - Series with linear trend and no seasonal variance: difference at lag 1
     - $Y_t = \nabla X_t = X_t - X_{t-1}$ (R code: `y = diff(x, lag=1)`)
   - Series with trend and seasonal variation: apply seasonal differencing, check time plot for potential trend and, if necessary, apply differencing at lag 1 (cf., Lab 4)
   - If the series has no apparent deviations from stationarity, do not difference

Model specification for a SARIMA$(p, d, q) \times (P, D, Q)_s$ process (cont'd)

3. For the differenced series, if differencing was done, or otherwise for the original series, examine sample acf and pacf plots to determine p, P, q, Q

   - There will not always be a definitive answer from this analysis, but you may be able to narrow down options for a potentially adequate model
   - Values for low lags (1,2,3,...) are used to decide on values of p and q for pure AR and MA processes (recall the previous two activities)
   - Values corresponding to multiples of seasonal period s are used to decide on values of P and Q for seasonal AR or MA components

4. Fit parameters for the selected SARIMA$(p, d, q) \times (P, D, Q)_s$ model (R code:

```
fm = arima(x, order=c(p,d,q),
           seasonal=list(order=c(P,D,Q), period=s))
```
)

5. Perform model diagnostics to see if the model fits well (to be discussed shortly)

Model-selection criteria

- The two standard tools we have already discussed for identifying ARMA models are the sample acf and sample pacf
  - They require a subjective choice being made by matching the model characteristics with those observed in the sample
- The most commonly used model-selection statistic is the Akaike's Information Criterion (AIC) defined (approximately) as

$$AIC := -2 \log(\text{maximum likelihood}) + 2r,$$

where r is the number of free parameters in the fitted model

  - i.e., the model choice is made based on the "best" fit, determined by the likelihood function, but penalized by the number of parameters in the model
  - For ARMA(p,q), $r = p + q + 1$

Model-selection criteria (cont'd)

- For small samples, the AIC is biased
- The biased-corrected version, denoted AIC_C and given (approx.) by

$$AIC_C := -2\log(\text{maximum likelihood}) + 2r\left(\frac{n}{n-r-1}\right)$$

  is often recommended

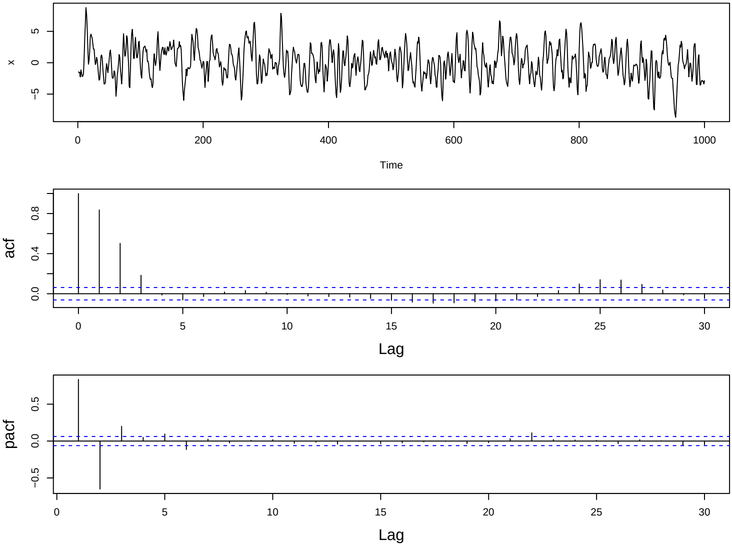
- The Bayesian Information Criterion (BIC) is also quite popular

$$BIC := -2\log(\text{maximum likelihood}) + (r + r\log n)$$

  ✤ it has a larger penalty for inclusion of extra parameters compared to AIC

Example: Model selection using AIC

Consider the following time series:



Example: Model selection using AIC (cont'd)

- Below are the values of AIC for each of the considered ARMA(p,q) models:

	$q=0$	$q=1$	$q=2$	$q=3$	$q=4$	$q=5$
$p=0$	4714.3	3683.1	3229.4	2986.6	2915.2	2911.6
$p=1$	3514.3	3050.8	2987.3	2921.2	2912.4	2913.6
$p=2$	2961.8	2926.8	2928.1	**2904.3**	2906.2	2907.1
$p=3$	2921.8	2919.4	2927.3	2906.1	2907.2	2908.0
$p=4$	2921.2	2922.0	2915.5	2906.8	2908.7	2911.2
$p=5$	2914.6	2906.2	2905.5	2907.5	2908.1	2909.8

- The lowest value of 2904.3 corresponds to ARMA(2,3), which is indeed the true model

Example: Model selection using AIC (cont'd)

R code:

```
set.seed(166) # data simulation
x <- arima.sim(n=1000, list(ar=c(.9,-.4),ma=c(.6,.4,.3)))

par(mfrow=c(3,1), mar=c(4.5,4.5,1,1))
plot.ts(x, ylab="x", cex.lab=1.5)
acf(x, lag.max = 30, main="",ylab="acf", cex.lab=1.5)
pacf(x, lag.max = 30, main="",ylab="pacf",cex.lab=1.5)

out.aic <- matrix(nrow=6,ncol=6) # output matrix to store AIC values

for (p in 0:5) for (q in 0:5)
{
    fm <- arima(x,order=c(p,0,q))
    out.aic[p+1,q+1] <- AIC(fm)
}

round(out.aic,1)

# return indices corresponding to the smallest AIC value
which(out.aic == min(out.aic), arr.ind = TRUE) # return indices corresponding to the smallest
AIC value
```

Model diagnostics

- Suppose we have now fitted a suitable model from the ARIMA family
- Final step: Check that the model fits the data reasonably well with no patterns in the data that the model is not detecting
- General approach to model diagnostics is based on examining the residuals of the model

$$\hat{z}_t := x_t - \hat{x}_t, \qquad t = 1, \ldots, n$$

where $\hat{x}_t$ is the value fitted by the model at time t

- Example: For a fitted AR(1) model, if $\hat{\alpha}$ denotes the estimate of parameter α then the residual at time t is

$$\hat{z}_t = x_t - \hat{\alpha} x_{t-1}$$

which can be thought of as an estimate of the white noise term z_t in the definition of the model

Model diagnostics (cont'd)

- For a model that fits the data well and leaves no "residual" pattern in the data, the residuals will be small and look "random", like a realization of a white noise process
- You already know a simple way to check "randomness" of residuals

Model diagnostics (cont'd)

- For a model that fits the data well and leaves no "residual" pattern in the data, the residuals will be small and look "random", like a realization of a white noise process
- You already know a simple way to check "randomness" of residuals: the sample acf for residuals should not have significantly large values even at small lags

Model diagnostics (cont'd)

- For a model that fits the data well and leaves no "residual" pattern in the data, the residuals will be small and look "random", like a realization of a white noise process
- You already know a simple way to check "randomness" of residuals: the sample acf for residuals should not have significantly large values even at small lags
- There exist several model diagnostic tests based on the residuals
- Notation: let $r_h(\hat{z})$ denote the sample autocorrelation coefficient of the residuals at lag h
- Assume an ARMA(p,q) model has been fitted to n data points
  - If the data had to be first differenced, then n corresponds to the length of the differenced series which has less terms than the original series

Model diagnostic tests

Portmanteau lack-of-fit test

- The test statistic is given by

$$Q_1 := n \sum_{h=1}^{m} r_h(\hat{z})^2$$

where

  - n: number of terms in the series (possibly after differencing)
  - m: an integer (less than n), usually between 15 and 30

- If the fitted ARMA(p,q) model is reasonable, then

$$Q_1 \sim \chi^2_{m-p-q}$$

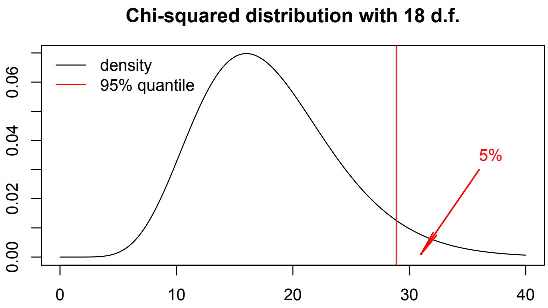
where χ^2_ν denotes the Chi-squared distribution with ν degrees of freedom

Model diagnostic tests (cont'd)

- If model fits well, Q_1 would be consistent with χ^2_{m-p-q} distribution
- If model fits poorly, Q_1 would be inflated and hence lie in the far upper tail of χ^2_{m-p-q} distribution
- The rule of thumb: re-consider the model if Q_1 exceeds 95% quantile of χ^2_{m-p-q} distribution

E.g.:

$$m = 20, \; p = q = 1$$



Model diagnostic tests (cont'd)

Ljung-Box-Pierce test

- A variant of the portmanteau lack-of-fit test with test statistic

$$Q_2 = n(n+2)\sum_{h=1}^{m} \frac{r_h(\hat{z})^2}{n-h}$$

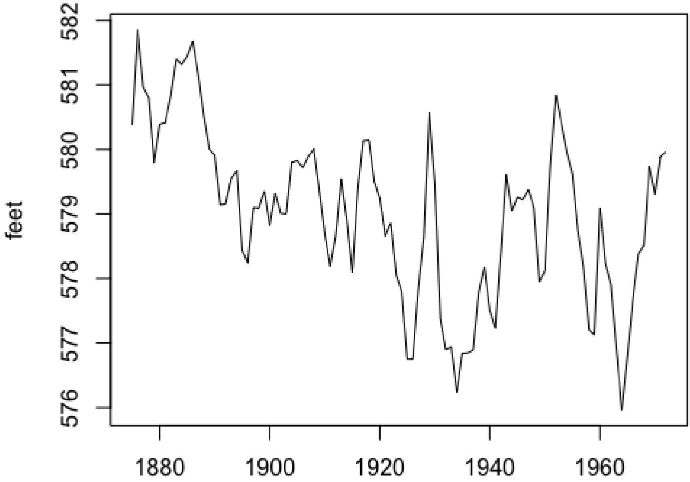
- Again, under the hypothesis of ARMA(p,q) model, $Q_2 \sim \chi^2_{m-p-q}$

Model diagnostic tests (cont'd)

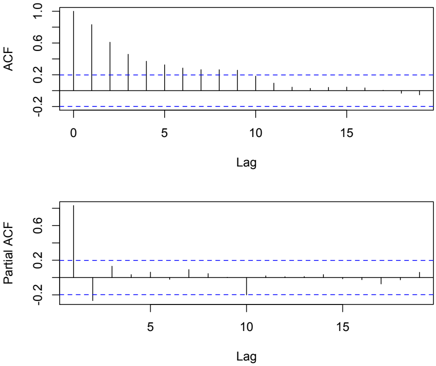
A few warning remarks:

- Unfortunately, these tests are by nature rather inconclusive
- They tend to accept an unacceptable model as being adequate rather too often
- However, no loss in considering these indicators especially since software packages provide them as part of their output

Example: Lake Huron water level



Example: Model specification



- The acf is decaying slowly, indicating possibly a non-stationary model, or else an AR (or maybe ARMA) model with long–term dependence
- The pacf cuts off noticeably at lag 2, suggesting an AR(2) could be suitable

Example: Lake Huron water level (cont'd)

- R commands and partial output are:

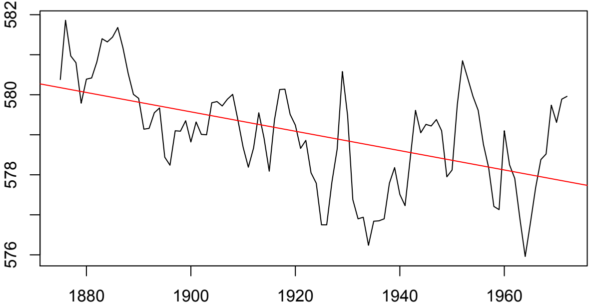
```
> (ar2Huron <- ar(LakeHuron, order.max=2, method="ols"))

Coefficients:
      1        2
 1.0217  -0.2376
Intercept: -0.02382 (0.06878)
Order selected 2  sigma^2 estimated as  0.454
```

- Hence, the fitted model is

$$X_t - \hat{\mu} = 1.0217(X_{t-1} - \hat{\mu}) - 0.2376(X_{t-2} - \hat{\mu}) + Z_t$$

with $\hat{\mu} = -0.0238$ and $Z_t \sim WN(0, 0.454)$, i.e. $\hat{\sigma} = \sqrt{0.454} = 0.6738$

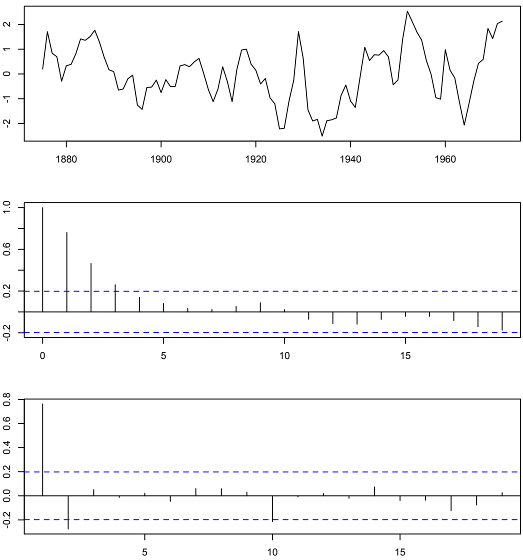
Looking at the time plot again, there appears to be a downward trend:



To remove the trend, we can use:

```
t = time(LakeHuron)-1874
LakeHuron.dt = ts(lm(LakeHuron ~ t)$residuals, start=1875, freq=1)
```

De-trended time series with ACF and PACF plots look as follows:

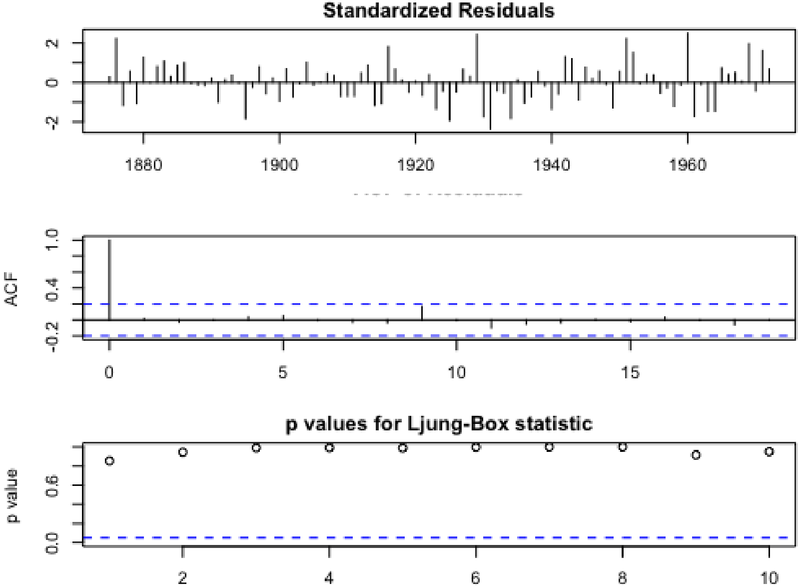


Re-fitting the AR(2) model to de-trended series gives:

```
> (ar(LakeHuron.dt, order.max=2, method="ols", demean=F))
Coefficients:
      1        2
 1.0020  -0.2834
Order selected 2  sigma^2 estimated as  0.4436

> round(coef(lm(LakeHuron ~ t)),3)
(Intercept)           t
    580.202      -0.024
```

Exercise: Write down the final fitted model

The model diagnostic plots based on the residuals from the fitted model look as follows:



Summary: Steps in model building

1. Model formulation/specification (the most tricky part!)
2. Model estimation/fitting (routine with computer software for well-established models)
3. Model checking/verification (an important step not to be missed!)