Lab 6

Pstat 174/274

May 11, 2017

Model Estimation

Summary of `R` commands for estimation:

1. To estimate parameters of an AR model:

```
ar(data, aic = TRUE, order.max = NULL, method = c("..."))
```

If `aic = TRUE`, Akaike's Information Criterion is used to select the order of the AR model to fit. If `aic = FALSE`, the model of order equal to `order.max` is fitted. `method` could be set equal to `"yule-walker"` or `"mle"` for different estimation methods of AR parameters.

2. To estimate parameters of an ARMA model:

```
arima(data, order = c(p, 0, q), method = c("..."))
```

The middle number in `order` is the number of times d the data should be differenced before fitting an ARMA model. For example, `order = c(2, 1, 2)` would fit an ARMA(2,2) once the original time series has been differenced (e.g., to remove the trend component). For regular ARMA models, put 0 for this number. Refer to help file (type `?arima`) for different methods of estimation.

3. To compare models using AICC:

```
AICc(fittedModel)
```

The `qpcR` package needs to be downloaded and installed in `R` first before using this function. This function will give the Akaike's Information Criterion corrected for bias given a fitted model object from `arima()`. We want to select the model with the *lowest* AICC.

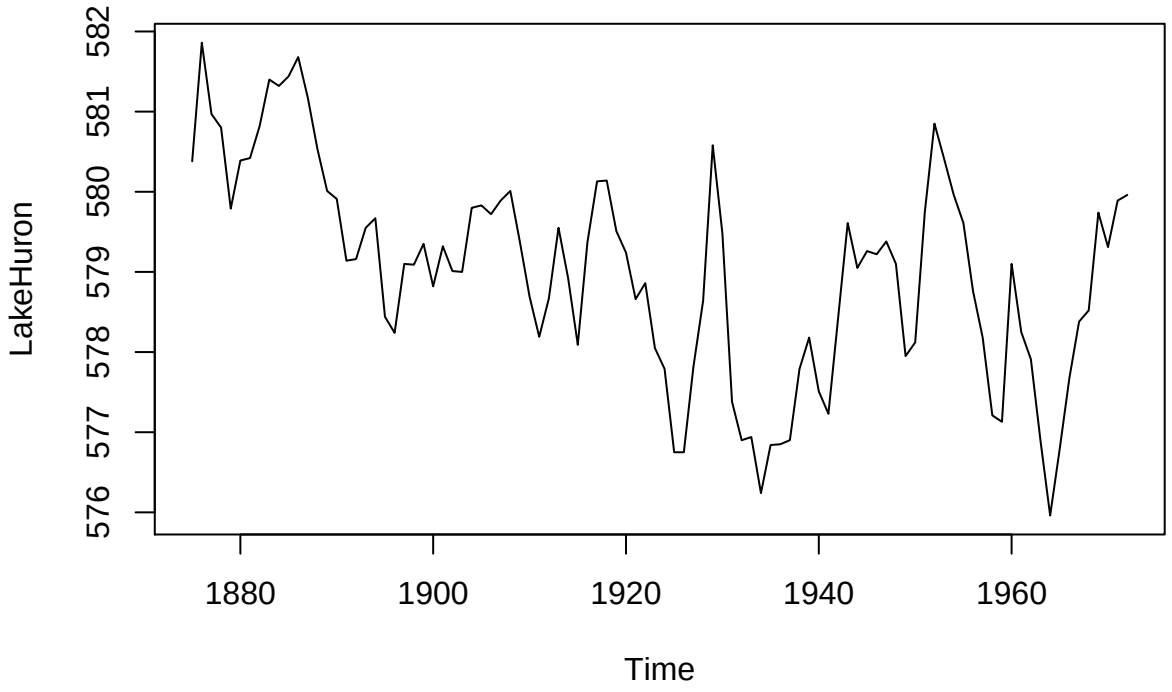
Lake Huron data

1. Analyze the Lake Huron data (preloaded data set in `R`). Load the data set using the command

```
data(LakeHuron)
```

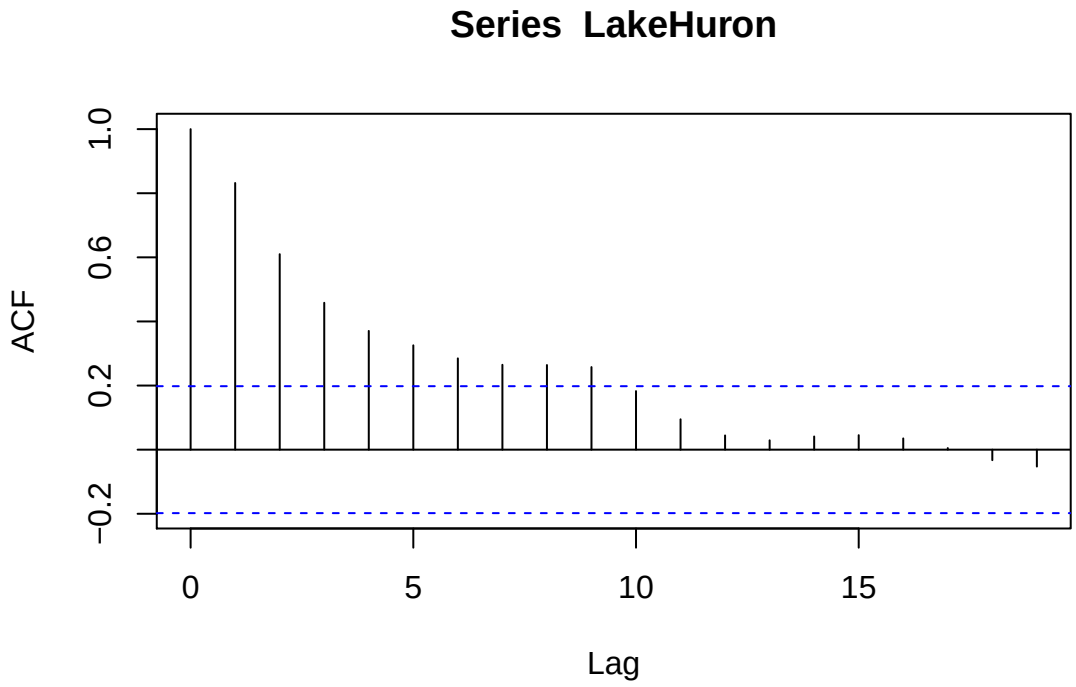
and plot the data set using

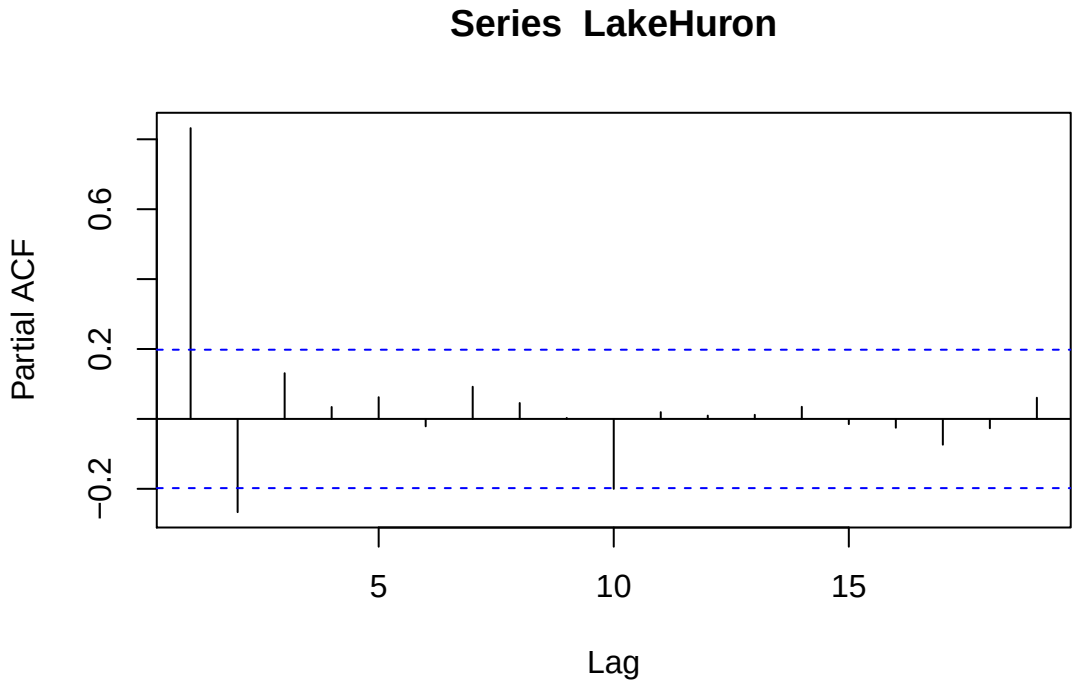
```
ts.plot(LakeHuron)
```



2. Plot the ACF and PACF. What kind of model do they suggest?

```
op <- par(mfrow=c(2,1))
acf(LakeHuron)
pacf(LakeHuron)
par(op)
```





3. Consider some possible AR models that might fit the data and perform preliminary Yule-Walker estimation of the model parameters (Hint: use the function `ar()`).

```
# Preliminary estimation using Yule-Walker
ar(LakeHuron, method="yule-walker")
```

```
##
## Call:
## ar(x = LakeHuron, method = "yule-walker")
```

```
##
## Coefficients:
##       1        2
##  1.0538  -0.2668
##
## Order selected 2  sigma^2 estimated as  0.5075
```

4. Consider some possible MA models that might fit the data and perform preliminary estimation of model parameters using the innovations algorithm (see `lab6RCode.r` for using the innovations algorithm using the function `innovations.algorithm()` in `innovations.r` file).

```
# Preliminary estimation using innovations algorithm
source("innovations.r")
?acf
acvf = acf(LakeHuron, plot=FALSE, lag.max = length(LakeHuron))$acf[,1,1] * var(LakeHuron)
m = length(acvf)
lh.ia = innovations.algorithm(m+1, acvf)
lh.ia$thetas[9,1:9] # Preliminary estimates of coefficients for MA(9)
```

```
## [1] 1.0810868 0.7691600 0.5218429 0.3315308 0.3085947 0.2479798 0.1932361
## [8] 0.1612645 0.2576989
```

```
lh.ia$thetas[10,1:10] # Preliminary estimates of coefficients for MA(10)
```

```
##  [1] 1.0816255 0.7781248 0.5367164 0.3291559 0.3160040 0.2513755 0.2051537
##  [8] 0.1441070 0.3431868 0.1827401
```

```
lh.ia$thetas[11,1:11] # Preliminary estimates of coefficients for MA(11)
```

```
##  [1] 1.08549783 0.77865745 0.54561580 0.34486948 0.31306321 0.26042628
##  [7] 0.20956724 0.16011877 0.31610604 0.33734448 0.09479822
```

5. Fit the different models under consideration using maximum likelihood estimation and compare the model fits using AICC (Hint: use `arima()` for estimation and `AICc()` in `R` library `qpcR` for model comparison - you will need to install this package into `R` first). Which model is preferred?

```
?arima
# Fit models using maximum likelihood estimation
fit_ar2 = arima(LakeHuron, order = c(2,0,0), method = "ML")
fit_ma9 = arima(LakeHuron, order = c(0,0,9), method = "ML")
fit_ma10 = arima(LakeHuron, order = c(0,0,10), method = "ML")
fit_ma11 = arima(LakeHuron, order = c(0,0,11), method = "ML")
```

Compare models using AICC

```
# install.packages("qpcR")
library(qpcR)
AICc(fit_ar2)
```

```
## [1] 215.5218
```

```
AICc(fit_ma9)
```

```
## [1] 228.0829
```

```
AICc(fit_ma10)
```

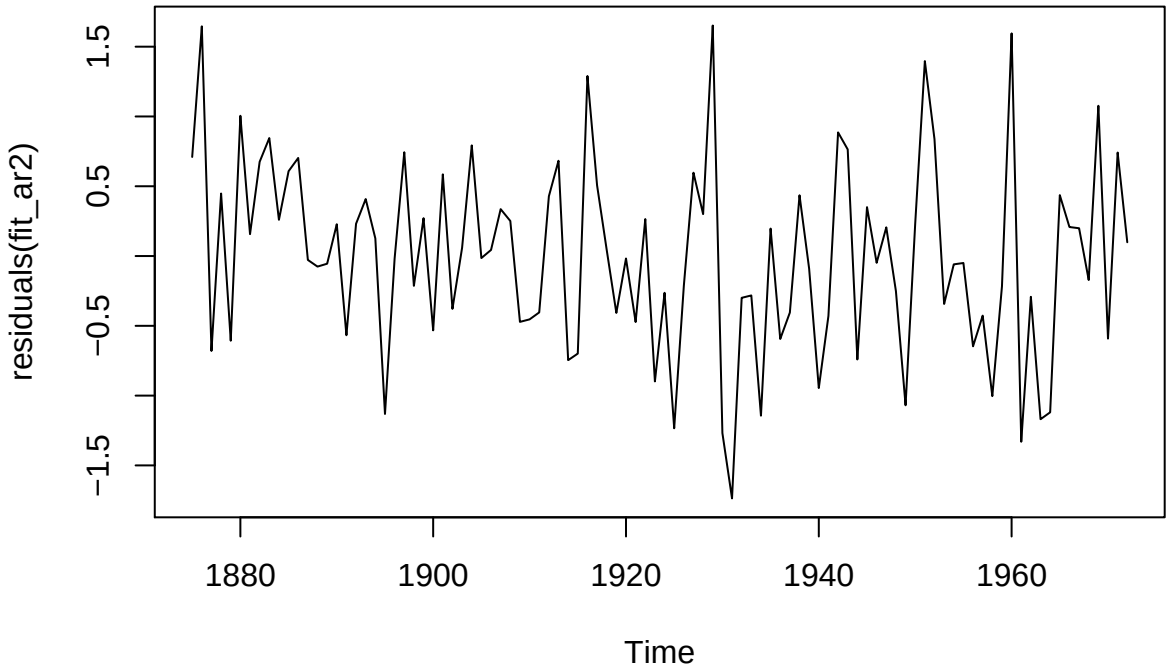
```
## [1] 228.9409
```

```
AICc(fit_ma11)
```

```
## [1] 230.2403
```

6. Plot the residuals of the chosen model. Do they look like white noise? Test using the Ljung Box test (Hint: function `Box.text(residuals, type="Ljung")`).

```
# Choose AR(2) - check residuals
plot(residuals(fit_ar2))
```

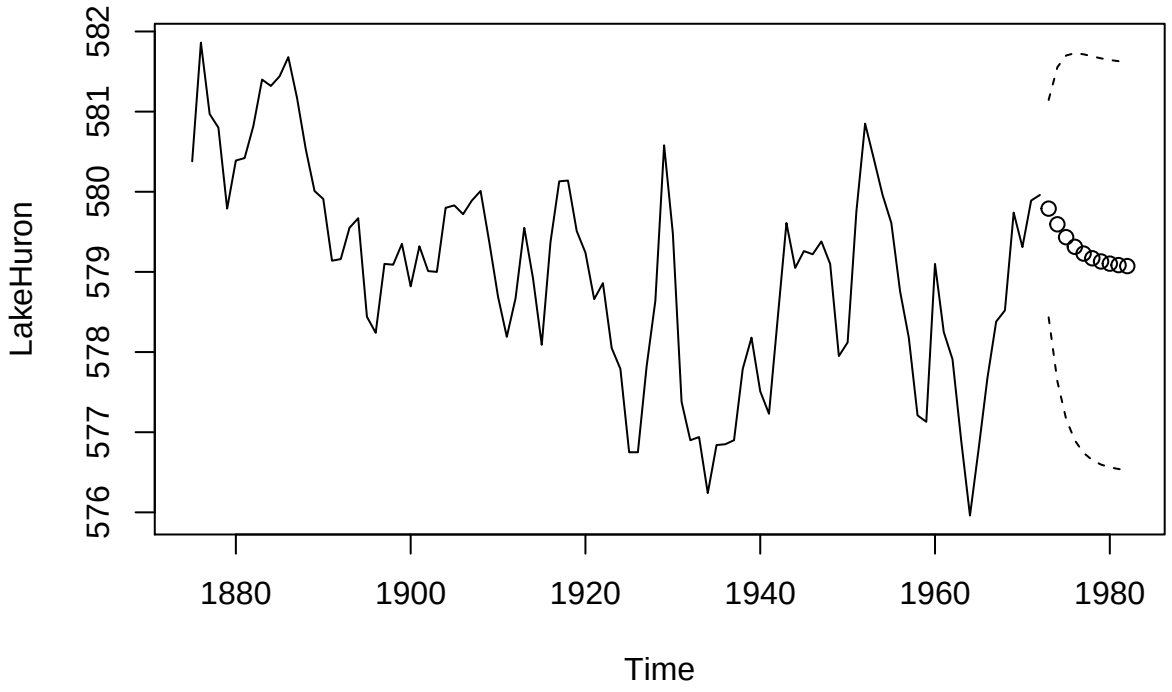



```
Box.test(residuals(fit_ar2), type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  residuals(fit_ar2)
## X-squared = 0.092714, df = 1, p-value = 0.7608
```

7. Forecast the next 10 observations using your model (Hint: function `predict(fittedModel, n.ahead=10)`).

```
# Predict next 10 time points and plot
pred = predict(fit_ar2, n.ahead=10)
ts.plot(LakeHuron,xlim=c(1875,1982))
points(1973:1982,pred$pred)
lines(1973:1982,pred$pred-1.96*pred$se,lty=2)
lines(1973:1982,pred$pred+1.96*pred$se,lty=2)
```


LakeHuron
582
581
580
579
578
577
576
1880
1900
1920
1940
1960
1980
Time