Module 6 - Parameter Estimation

MATH1318 Time Series Analysis

Dr. Haydar Demirhan - RMIT University, School of Science

All the contents in this presentation are mainly based on the textbook of MATH1318 Time Series Analysis course: 'Cryer and Chan, Time Series Analysis with R, Springer, 2008.' Other resources than the textbook are cited accordingly.

Introduction

After ensuring the series is stationary and specification of orders of autoregressive and moving average elements of ARIMA models, the next step is to estimate parameters of the specified tentative models.

For this aim, we use the following estimation methods:

- the method of moments,
- least squares estimation,
- maximum likelihood and unconditional least squares, and
- bootstrap approach to ARIMA models.

In this module, we will use `arima()` and `ar()` functions.

However, `arima()` and `ar()` functions only give parameter estimates with a specified method.

To obtain significance tests for each parameter, we will use `coeftest()` function from `lmtest` package.

The Method of Moments

The method of moments is the easiest one to apply.

The method consists of equating **sample moments** to corresponding **theoretical moments** and solving the resulting equations to obtain estimates of any unknown parameters.

Numerical Examples

The following table summarizes the method of moment parameter estimates for various simulated time series.

	True Parameters			Method-of-Moments Estimates			
Model	θ	ϕ_1	ϕ_2	θ	ϕ_1	ϕ_2	n
MA(1)	−0.9			−0.554			120
MA(1)	0.9			0.719			120
MA(1)	−0.9			NA†			60
MA(1)	0.5			−0.314			60
AR(1)		0.9			0.831		60
AR(1)		0.4			0.470		60
AR(2)		1.5	−0.75		1.472	−0.767	120

† No method-of-moments estimate exists since $r_1 = 0.544$ for this simulation.

```
estimate.ma1.mom=function(x){
  r=acf(x,plot=F)$acf[1] # gives us r_1
  if (abs(r)<0.5)
    return((-1+sqrt(1-4*r^2))/(2*r))
  else
    return(NA)
}

data(ma1.2.s)
data(ma1.1.s)
```

```
ma1.3.s=arima.sim(list(ma=c(0.9)),n=60)
ma1.4.s=arima.sim(list(ma=c(-0.5)),n=60)
estimate.ma1.mom(ma1.2.s)
```

```
## [1] -0.5554273
```

```
estimate.ma1.mom(ma1.1.s)
```

```
## [1] 0.7196756
```

```
estimate.ma1.mom(ma1.3.s)
```

```
## [1] -0.6157487
```

```r
estimate.ma1.mom(ma1.4.s)
```

```
## [1] 0.1645242
```

```r
data(ar1.s)
data(ar1.2.s)
ar(ar1.s,order.max=1,AIC=F,method='yw')
```

```
##
## Call:
## ar(x = ar1.s, order.max = 1, method = "yw", AIC = F)
##
## Coefficients:
##      1
## 0.8314
##
## Order selected 1  sigma^2 estimated as  1.382
```

```r
ar(ar1.2.s,order.max=1,AIC=F,method='yw')
```

```
##
## Call:
## ar(x = ar1.2.s, order.max = 1, method = "yw", AIC = F)
##
## Coefficients:
##      1
## 0.4699
##
## Order selected 1  sigma^2 estimated as  0.9198
```

```r
data(ar2.s)
ar(ar2.s,order.max=2,AIC=F,method='yw')
```

```
##
## Call:
## ar(x = ar2.s, order.max = 2, method = "yw", AIC = F)
##
## Coefficients:
##       1        2
##  1.4694  -0.7646
##
## Order selected 2  sigma^2 estimated as  1.051
```

While the estimates for all the autoregressive models are fairly good, the estimates for the moving average models are not acceptable.

This is due to the inefficiency of the method of moments estimators for models containing moving average terms.

Consider the Canadian hare abundance series. Note that we have found that the square root transformation is appropriate for this series.

So, we will proceed with the square root transformation in parameter estimation. First, we will fit an AR(2) model for this series with the method of moments.

The first two sample autocorrelations are $r_1 = 0.736$ and $r_2 = 0.304$.

Then, we find $\hat{\phi}_1 = 1.1178$ and $\hat{\phi}_{\cancel{1}\,2} = -0.519$.

The sample mean and variance for this series are 5.82 and 5.88, respectively.

The noise variance is estimated as $\hat{\sigma}_e^2 = 1.97$.

The prediction model is then

$$\sqrt{Y_t} = 2.335 + 1.1178\sqrt{Y_{t-1}} - 0.519\sqrt{Y_{t-2}} + e_t.$$

sqrt

$$Y_t = \hat{\alpha} + \hat{\phi}_1 Y_{t-1} + \hat{\phi}_2 Y_{t-2} + e_t$$

Let's consider the oil prices series.

We have specified an MA(1) model for the first difference of natural logarithms of this series.

For this series $r_1 = 0.21$, so the method of moments estimate of θ is $\hat{\theta} = -0.222$.

Hence the prediction model is

$$\log Y_t = \log(Y_{t-1}) + 0.004e_t + 0.222e_{t-1}.$$

with estimated noise variance of $\hat{\sigma}_e^2 = 0.00686$.

Least Squares and Maximum Likelihood Estimation

Although the method of moments approach is easy to apply for simple models, it is mostly inefficient.

So, we have other more efficient (with smaller estimation variance) approaches for the parameter estimation of time series.

The least squares approach is the first one we will consider.

In this approach, we minimise the conditional or unconditional sum of squares function according to the parameters to find parameter estimates.

Maximum Likelihood and Unconditional Least Squares

In the maximum likelihood (ML) estimation, we use all of the information contained in the sample than just the first and second moments.

Another advantage is that many large-sample results are known under very general conditions.

One disadvantage is that, for the first time, we must work specifically with the joint probability density function of the process.

Maximum Likelihood Estimation

The likelihood function is obtained over the joint pdf of the sample in all kinds of data.

The maximum likelihood estimators are then defined as those values of the parameters for which the data actually observed are most likely, that is, the values that maximize the likelihood function.

The most common assumption of ML estimation is that the white noise terms are independent, normally distributed random variables with zero means and common standard deviation of σ_e .

Illustrations of Parameter Estimation

Consider the simulated MA(1) series with $\theta = -0.9$.

The method of moments estimate of θ was found -0.554.

In contrast, the maximum likelihood estimate is -0.915, the unconditional sum-of-squares estimate is -0.923, and the conditional least squares estimate is -0.879.

The maximum likelihood estimate is closest to the true value used in the simulation.

The estimate of the standard error is about 0.04. So none of the maximum likelihood, conditional sum-of-squares, or unconditional sum-of-squares estimates are significantly far from the true value of -0.9.

For the MA(1) simulation with $\theta = 0.9$,

the method of moments estimate was 0.719.

The conditional sum of squares estimate is 0.958,

the unconditional sum-of-squares estimate is 0.983, and

the ML estimate is 1.000 which is a little disconcerting since it corresponds to a noninvertible model.

For the MA(1) simulation with $\theta = -0.9$, the method of moments estimate was -0.719.

The conditional and unconditional sum of squares estimates are -0.979 and -0.961, respectively.

For the AR(1) models, we have the following results:

Parameter ϕ	Method-of-Moments Estimate	Conditional SS Estimate	Unconditional SS Estimate	Maximum Likelihood Estimate	n
0.9	0.831	0.857	0.911	0.892	60
0.4	0.470	0.473	0.473	0.465	60

```r
data(ar1.s)
data(ar1.2.s)
# Method of moments estimate of the AR coefficient
# for simulated AR(1) model
ar(ar1.s,order.max=1,AIC=F,method='yw')
```

```
##
## Call:
## ar(x = ar1.s, order.max = 1, method = "yw", AIC = F)
##
## Coefficients:
##      1
## 0.8314
##
## Order selected 1  sigma^2 estimated as  1.382
```

```r
# Least squares estimate of the AR coefficient
# for simulated AR(1) model
ar(ar1.s,order.max=1,AIC=F,method='ols')
```

```
##
## Call:
## ar(x = ar1.s, order.max = 1, method = "ols", AIC = F)
##
## Coefficients:
##      1
## 0.857
##
## Intercept: 0.02499 (0.1308)
##
## Order selected 1  sigma^2 estimated as  1.008
```

```r
# Least squares estimate of the AR
# coefficient with significance tests
# for simulated AR(1) model
coeftest(arima(ar1.s,order=c(1,0,0),method='CSS'))
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1       0.857041   0.062783 13.6508  < 2e-16 ***
## intercept 2.161228   0.908597  2.3786  0.01738 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Maximum likelihood estimate of the AR
# coefficient for simulated AR(1) model
ar(ar1.s,order.max=1,AIC=F,method='mle')
```

```
##
## Call:
## ar(x = ar1.s, order.max = 1, method = "mle", AIC = F)
##
## Coefficients:
##      1
## 0.8924
##
## Order selected 1  sigma^2 estimated as  1.041
```

```r
# Maximum likelihood estimate of the AR
# coefficient with significance tests
# for simulated AR(1) model
coeftest(arima(ar1.s,order=c(1,0,0),method='ML'))
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1       0.892436   0.059822 14.9181   <2e-16 ***
## intercept 1.263068   1.139865  1.1081   0.2678
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Method of moments estimate of the AR coefficient
# for simulated AR(1) model
ar(ar1.2.s,order.max=1,AIC=F,method='yw')
```

```
##
## Call:
## ar(x = ar1.2.s, order.max = 1, method = "yw", AIC = F)
##
## Coefficients:
##      1
## 0.4699
##
## Order selected 1  sigma^2 estimated as  0.9198
```

```r
# Least squares estimate of the AR coefficient
# for simulated AR(1) model
ar(ar1.2.s,order.max=1,AIC=F,method='ols')
```

```
##
## Call:
## ar(x = ar1.2.s, order.max = 1, method = "ols", AIC = F)
##
## Coefficients:
##      1
## 0.4731
##
## Intercept: -0.006084 (0.1237)
##
## Order selected 1  sigma^2 estimated as  0.9024
```

```r
# Least squares estimate of the AR
# coefficient with significance tests
# for simulated AR(1) model
coeftest(arima(ar1.2.s,order=c(1,0,0),method='CSS'))
```

p d q

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1         0.47310    0.11423  4.1416 3.449e-05 ***
## intercept -0.12051    0.23281 -0.5176    0.6047
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Maximum likelihood estimate of the AR
# coefficient for simulated AR(1) model
ar(ar1.2.s,order.max=1,AIC=F,method='mle')
```

```
##
## Call:
## ar(x = ar1.2.s, order.max = 1, method = "mle", AIC = F)
##
## Coefficients:
##       1
## 0.4654
##
## Order selected 1  sigma^2 estimated as  0.8875
```

```r
# Maximum likelihood estimate of the AR
# coefficient with significance tests
# for simulated AR(1) model    p d q
coeftest(arima(ar1.2.s,order=c(1,0,0),method='ML'))
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1         0.46534    0.11274  4.1275 3.668e-05 ***
## intercept -0.11785    0.22428 -0.5255    0.5993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All four methods estimate reasonably well for AR(1) models and ML estimates are the closest ones to the true parameter values with an estimated standard error of 0.07 or 0.11.

For the simulated AR(2) models, we have the following results:

Parameters	Method-of-Moments Estimates	Conditional SS Estimates	Unconditional SS Estimates	Maximum Likelihood Estimate	*n*
$\phi_1 = 1.5$	1.472	1.5137	1.5183	1.5061	120
$\phi_2 = -0.75$	–0.767	–0.8050	–0.8093	–0.7965	120

```r
data(ar2.s)
# Method of moments estimates of the AR
# coefficients for simulated AR(2) model
ar(ar2.s,order.max=2,AIC=F,method='yw')
```

```
##
## Call:
## ar(x = ar2.s, order.max = 2, method = "yw", AIC = F)
##
## Coefficients:
##       1        2
##  1.4694  -0.7646
##
## Order selected 2  sigma^2 estimated as  1.051
```

```r
# Least squares estimates of the AR
# coefficients for simulated AR(2) model
ar(ar2.s,order.max=2,AIC=F,method='ols')
```

```
##
## Call:
## ar(x = ar2.s, order.max = 2, method = "ols", AIC = F)
##
## Coefficients:
##       1        2
##  1.5137  -0.8050
##
## Intercept: 0.02043 (0.08594)
##
## Order selected 2  sigma^2 estimated as  0.8713
```

```r
# Maximum likelihood estimates of the AR
# coefficients with significance tests
# for simulated AR(2) model
coeftest(arima(ar1.2.s,order=c(1,0,0),method='ML'))
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1         0.46534    0.11274  4.1275 3.668e-05 ***
## intercept -0.11785    0.22428 -0.5255    0.5993
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Maximum likelihood estimates of the AR
# coefficients for simulated AR(2) model
ar(ar2.s,order.max=2,AIC=F,method='mle')
```

```
##
## Call:
## ar(x = ar2.s, order.max = 2, method = "mle", AIC = F)
##
## Coefficients:
##       1        2
##  1.5061  -0.7964
##
## Order selected 2  sigma^2 estimated as  0.862
```

```r
# Maximum likelihood estimates of the AR
# coefficients with significance tests
# for simulated AR(2) model
coeftest(arima(ar2.s,order=c(2,0,0),method='ML'))
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error  z value Pr(>|z|)
## ar1        1.506146   0.053739  28.0269   <2e-16 ***
## ar2       -0.796453   0.053322 -14.9367   <2e-16 ***
## intercept  0.237908   0.292719   0.8128   0.4164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Again, considering the size of the standard errors, all four methods estimate reasonably well for AR(2) models.

For the simulated ARMA(1,1) model, we have the following results:

True Parameters	Method-of-Moments Estimates	Conditional SS Estimates	Unconditional SS Estimates	Maximum Likelihood Estimate	n
$\phi = 0.6$	0.637	0.5586	0.5691	0.5647	100
$\theta = -0.3$	−0.2066	−0.3669	−0.3618	−0.3557	100

```r
data(arma11.s)
# Least squares estimates of the coefficients for
# simulated ARMA(1,1) model
arima(arma11.s, order=c(1,0,1),method='CSS')
```

```
##
## Call:
## arima(x = arma11.s, order = c(1, 0, 1), method = "CSS")
##
## Coefficients:
##          ar1     ma1  intercept
##       0.5586  0.3669     0.3928
## s.e.  0.1219  0.1564     0.3380
##
## sigma^2 estimated as 1.199:  part log likelihood = -150.98
```

```r
# Least squares estimates of the coefficients
# with significance tests for simulated ARMA(!,1) model
coeftest(arima(arma11.s,order=c(1,0,1),method='CSS'))
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## ar1          0.55858    0.12192  4.5814 4.618e-06 ***
## ma1          0.36688    0.15645  2.3451   0.01902 *
## intercept    0.39277    0.33804  1.1619   0.24527
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Maximum likelihood estimates of the coefficients for
# simulated ARMA(1,1) model
arima(arma11.s, order=c(1,0,1),method='ML')
```

```
##
## Call:
## arima(x = arma11.s, order = c(1, 0, 1), method = "ML")
##
## Coefficients:
##          ar1     ma1  intercept
##       0.5647  0.3557     0.3216
## s.e.  0.1205  0.1585     0.3358
##
## sigma^2 estimated as 1.197:  log likelihood = -151.33,  aic = 308.65
```

```r
# Maximum likelihood estimates of the coefficients
# with significance tests for simulated ARMA(1,1) model
coeftest(arima(arma11.s,order=c(1,0,1),method='ML'))
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value  Pr(>|z|)
## ar1        0.56475    0.12046  4.6882 2.756e-06 ***
## ma1        0.35570    0.15847  2.2446   0.02479 *
## intercept  0.32162    0.33577  0.9579   0.33813
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We have specified an AR(1) model for the industrial chemical property time series. Parameter estimates for this model are shown below:

Parameter	Method-of-Moments Estimate	Conditional SS Estimate	Unconditional SS Estimate	Maximum Likelihood Estimate	n
ϕ	0.5282	0.5549	0.5890	0.5703	35

```r
data(color)
# Method of moments estimates
ar(color,order.max=1,AIC=F,method='yw')
```

```
##
## Call:
## ar(x = color, order.max = 1, method = "yw", AIC = F)
##
## Coefficients:
##      1
## 0.5282
##
## Order selected 1  sigma^2 estimated as  27.56
```

```r
# Least squares estimates
ar(color,order.max=1,AIC=F,method='ols')
```

```
##
## Call:
## ar(x = color, order.max = 1, method = "ols", AIC = F)
##
## Coefficients:
##      1
## 0.5549
##
## Intercept: 0.1032 (0.8474)
##
## Order selected 1  sigma^2 estimated as  24.38
```

```r
# Least squares estimates wit significance tests
coeftest(arima(color,order=c(1,0,1),method='CSS'))
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value  Pr(>|z|)
## ar1        0.68286    0.17000  4.0168 5.899e-05 ***
## ma1       -0.22288    0.24651 -0.9042    0.3659
## intercept 75.39455    2.07262 36.3765 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Maximum likelihood estimates
ar(color,order.max=1,AIC=F,method='mle')
```

```
##
## Call:
## ar(x = color, order.max = 1, method = "mle", AIC = F)
##
## Coefficients:
##      1
## 0.5703
##
## Order selected 1  sigma^2 estimated as  24.83
```

```r
# Maximum likelihood estimates with significance tests
coeftest(arima(color,order=c(1,0,1),method='ML'))
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value  Pr(>|z|)
## ar1        0.67227    0.21455  3.1333  0.001728 **
## ma1       -0.14700    0.27416 -0.5362  0.591842
## intercept 74.17257    2.13617 34.7223 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Parameter estimates of an AR(3) model for the Canadian hare abundance series are shown below:

Coefficients:	**ar1**	**ar2**	**ar3**	**Intercept**[†]
	1.0519	–0.2292	–0.3931	5.6923
s.e.	0.1877	0.2942	0.1915	0.3371

```
sigma^2 estimated as 1.066: log-likelihood = -46.54, AIC = 101.08
```

p d q

```
data(hare)
coeftest(arima(sqrt(hare),order=c(3,0,0),method='ML'))
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value  Pr(>|z|)
## ar1         1.05190    0.18767  5.6051 2.081e-08 ***
## ar2        -0.22925    0.29419 -0.7793    0.4358
## ar3        -0.39304    0.19148 -2.0527    0.0401 *
## intercept   5.69227    0.33709 16.8866 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimates of the lag 1 and lag 3 autoregressive coefficients are significantly different from zero, as is the intercept term, but the lag 2 autoregressive parameter estimate is not significant.

Then the prediction model is written as

$$\sqrt{Y_t} - 5.6923 = 1.0519(\sqrt{Y_{t-1}} - 5.6923) - 0.2292(\sqrt{Y_{t-2}} - 5.6923) - 0.3930(\sqrt{Y_{t-3}} - 5.6923) + e_t$$

or

$$\sqrt{Y_t} = 3.25 + 1.0519\sqrt{Y_{t-1}} - 0.2292\sqrt{Y_{t-2}} - 0.3930\sqrt{Y_{t-3}} + e_t .$$

where Y_t is the hare abundance in year t in original terms.

We have specified an MA(1) model for the first difference of the logs of the oil prices. Estimates of θ are given below:

Coefficients:	ar1	ar2	ar3	Intercept†
	1.0519	–0.2292	–0.3931	5.6923
s.e.	0.1877	0.2942	0.1915	0.3371

```
sigma^2 estimated as 1.066: log-likelihood = -46.54, AIC = 101.08
```

p d q

```
data(oil.price)
coeftest(arima(log(oil.price),order=c(0,1,1),method='CSS'))
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ma1 0.273113   0.068106  4.0101 6.069e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(arima(log(oil.price),order=c(0,1,1),method='ML'))
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ma1 0.295600   0.069347  4.2626 2.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The method of moments estimate differs quite a bit from the others which are nearly equal given their standard errors of about 0.07.

A practical application

In this application, we will analyse stock and recruitment data for Klamath River Chinook Salmon between 1979-2000 from `ChinookKR` dataset of `FSAdata` package.

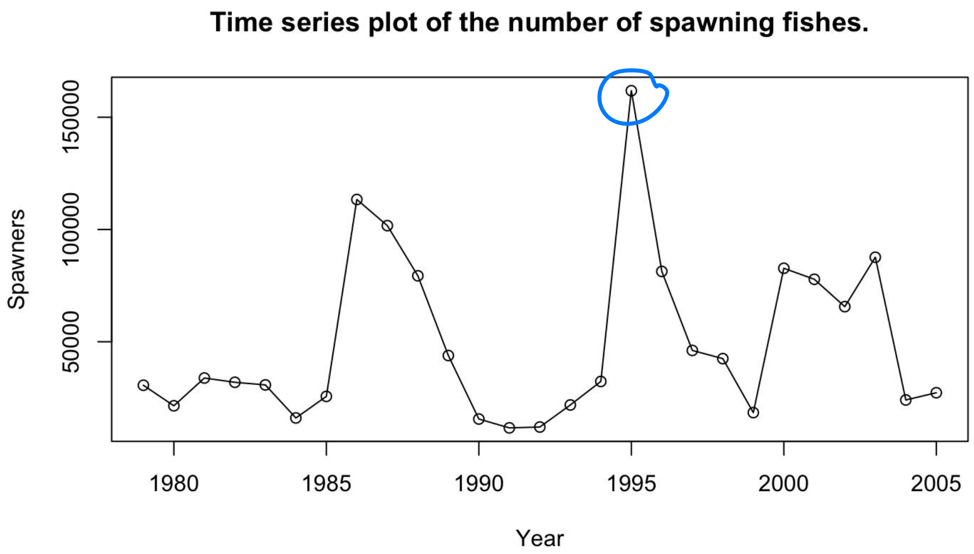
In this application, first, we will specify a set of possible models and estimate their parameters.

```
library(FSAdata)
```

```
## ## FSAdata v0.3.6. See ?FSAdata to find data for specific fisheries analyses.
```

```
data(ChinookKR)
chinook.spawners = ts(ChinookKR$spawners,start = 1979)
```

```
plot(chinook.spawners,ylab='Spawners',xlab='Year',type='o', main = "Time series plot of th
```



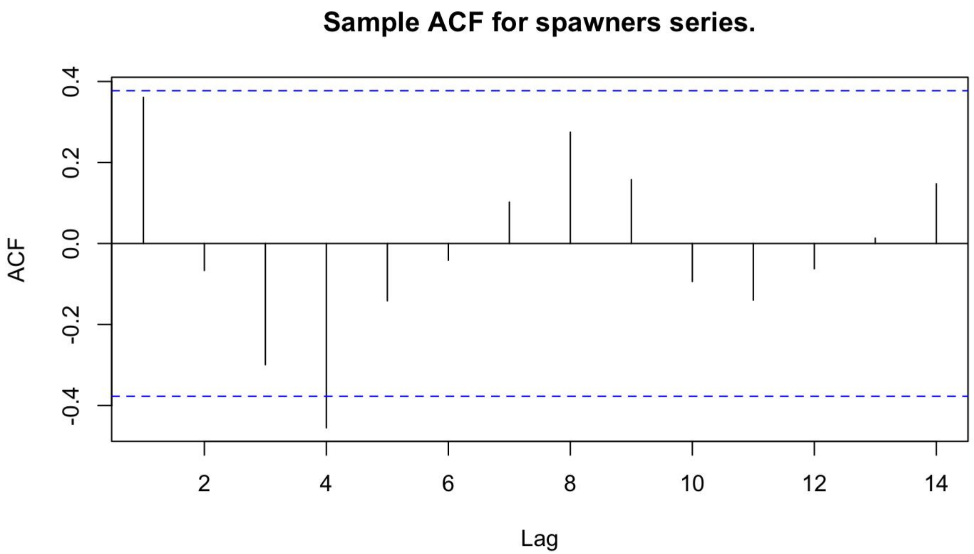
We observe succeeding observations implying the existence of autocorrelation.

Changing variance is considerable and no seasonality is obtained from this plot.

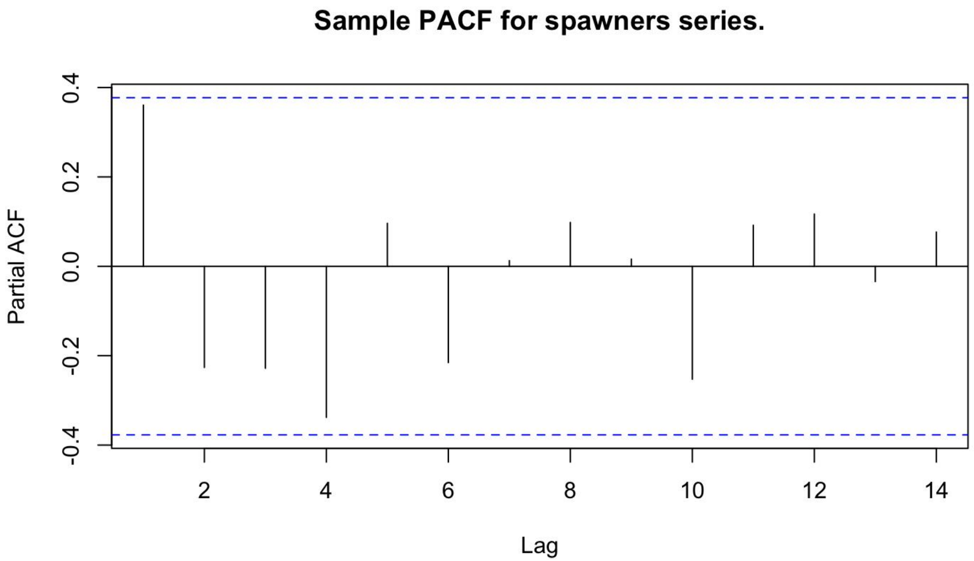
Existence of a trend is not obvious from this times series plot.

So we will display ACF and PACF plots.

```r
acf(chinook.spawners, main="Sample ACF for spawners series.")
```



```r
pacf(chinook.spawners, main="Sample PACF for spawners series.")
```



We will apply the ADF unit-root test to test the existence of non-stationarity with this series.

```
ar(diff(chinook.spawners))
```

```
##
## Call:
## ar(x = diff(chinook.spawners))
##
## Coefficients:
##        1        2        3        4
## -0.2809  -0.2625  -0.2319  -0.4607
##
## Order selected 4  sigma^2 estimated as  1.573e+09
```

Order for the test is obtained as 4

```r
adfTest(chinook.spawners, lags = 4, title = NULL,description = NULL)
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 4
##   STATISTIC:
##     Dickey-Fuller: -0.4568
##   P VALUE:
##     0.463
##
## Description:
##  Mon Apr 29 13:40:50 2019 by user:
```

We can conclude non-stationarity of the series at 5% level of significance.

We will go on with the first difference of actual observations.

```
spawners.diff = diff(chinook.spawners)
```

```r
ar(diff(spawners.diff))
```

```
##
## Call:
## ar(x = diff(spawners.diff))
##
## Coefficients:
##       1        2        3        4        5        6        7        8
## -1.1066  -1.1708  -1.1584  -1.3648  -1.0194  -0.9106  -0.7105  -0.4044
##
## Order selected 8  sigma^2 estimated as  2.249e+09
```

Order for the test is obtained as 8

```r
adfTest(spawners.diff, lags = 8, title = NULL,description = NULL)
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 8
##   STATISTIC:
##     Dickey-Fuller: -1.7789
##   P VALUE:
##     0.07444
##
## Description:
##  Mon Apr 29 13:40:50 2019 by user:
```

We can conclude non-stationarity of the first difference of the series at 5% level of significance. So, we will apply the second difference.

```r
spawners.diff2 = diff(chinook.spawners, difference = 2)
```

```
ar(diff(spawners.diff2))
```

```
##
## Call:
## ar(x = diff(spawners.diff2))
##
## Coefficients:
##        1         2
## -1.0026  -0.5501
##
## Order selected 2  sigma^2 estimated as  6.112e+09
```

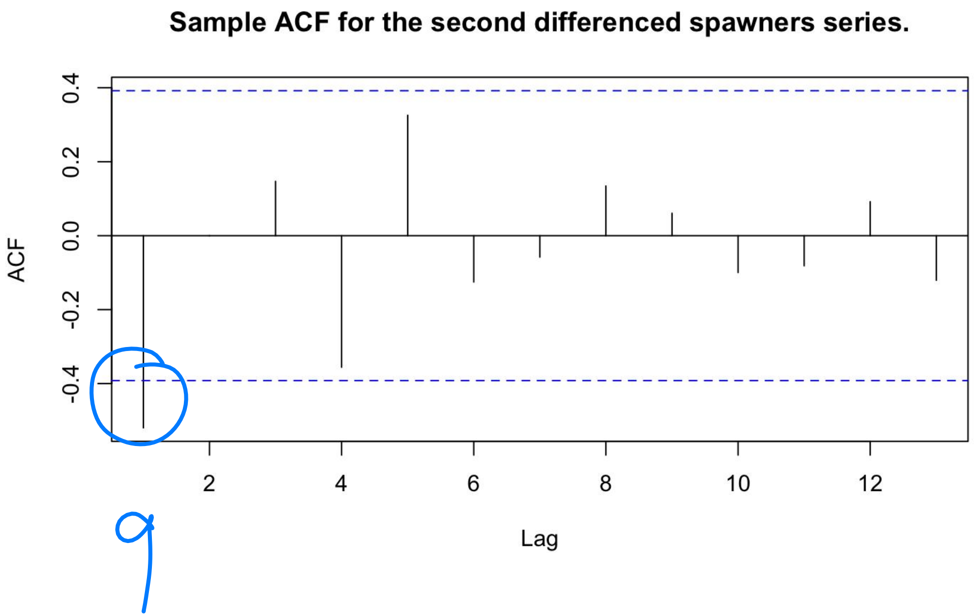
Order for the test is obtained as 2

```r
adfTest(spawners.diff2, lags = 2, title = NULL,description = NULL)
```

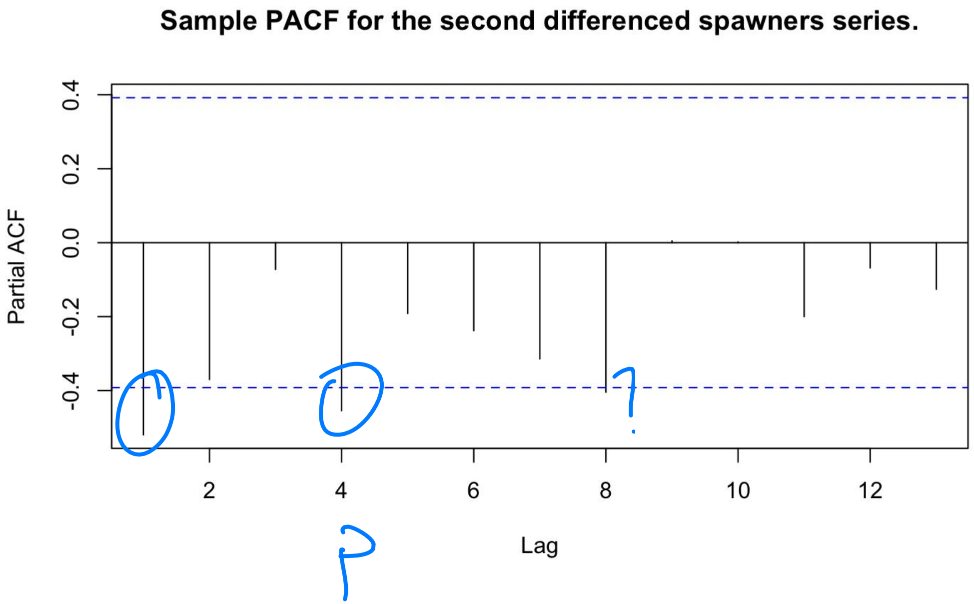
```
## Warning in adfTest(spawners.diff2, lags = 2, title = NULL, description =
## NULL): p-value smaller than printed p-value

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 2
##   STATISTIC:
##     Dickey-Fuller: -3.6501
##   P VALUE:
##     0.01
##
## Description:
##  Mon Apr 29 13:40:50 2019 by user:
```

```r
acf(spawners.diff2, main="Sample ACF for the second differenced spawners series.")
```



```
pacf(spawners.diff2, main="Sample PACF for the second differenced spawners series.")
```



From ACF and PACF, we see one significant autocorrelation in ACF and at least 2 significant autocorrelations in PACF. $d=2$

So, possible models from here are { ARIMA(1,2,1), ARIMA(2,2,1), ARIMA(3,2,1) }.

Now, we go on with EACF. Because of the size of the series, we should restrict the maximum number of AR and MA parameters in the EACF.

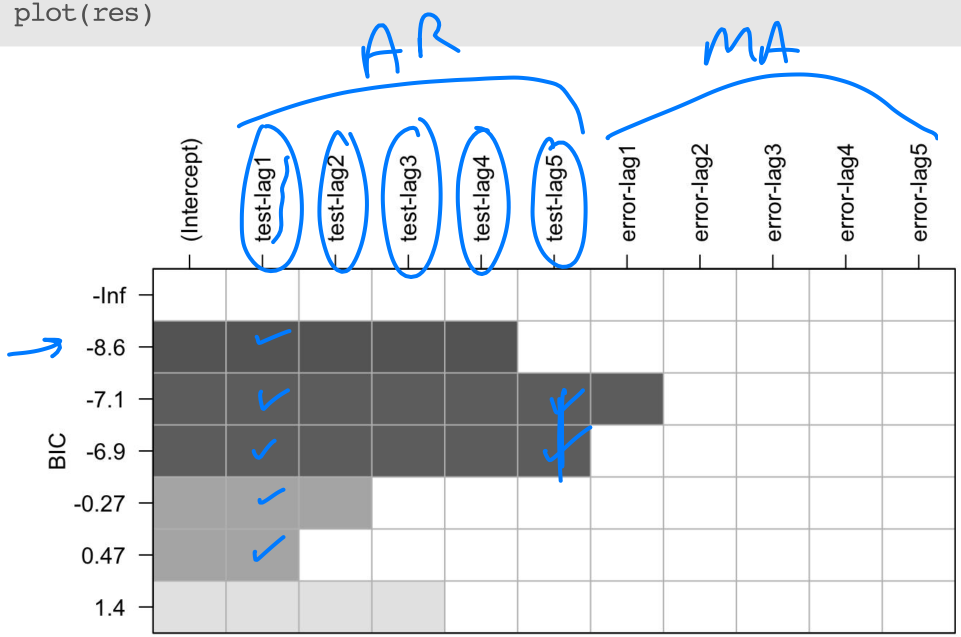
```r
eacf(spawners.diff2, ar.max = 5, ma.max = 5)
```

```
## AR/MA
##   0 1 2 3 4 5
## 0 x o o o o o
## 1 x o o o o o
## 2 o o o o o o
## 3 o x x o o o
## 4 o o o o o o
## 5 x o o o o o
```

The top left ‘o’ symbol in EACF is located at the intersection of AR = 0 and MA = 1. Then following the vertex downward, AR would be 1 and 2 as well.

The set of possible models becomes { ARIMA(1,2,1), ARIMA(2,2,1), ARIMA(3,2,1), ARIMA(0,2,1) }.

```r
options(warn=-1)
res = armasubsets(y=spawners.diff2,nar=5,nma=5,y.name='test',ar.method='ols')
plot(res)
```



From the BIC table, we read the models ARIMA(4,2,0), ARIMA(5,2,0), and ARIMA(5,2,1).

Then set of possible models becomes $\{$ ARIMA(1,2,1), ARIMA(2,2,1), ARIMA(3,2,1), ARIMA(0,2,1), ARIMA(4,2,0) , ARIMA(5,2,1), , ARIMA(5,2,0)$\}$.

Now, we will fit the models and find their parameter estimates and related significance tests.

```r
model.121 = arima(chinook.spawners,order=c(1,2,1),method='ML')
coeftest(model.121)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.13607    0.19635 -0.6930   0.4883
## ma1 -0.99998    0.11303 -8.8468   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model.221 = arima(chinook.spawners,order=c(2,2,1),method='ML')
coeftest(model.221)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value Pr(>|z|)
## ar1 -0.16104    0.19832 -0.8120   0.4168
## ar2 -0.12112    0.20293 -0.5968   0.5506
## ma1 -0.99998    0.11581 -8.6346   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model.321 = arima(chinook.spawners,order=c(3,2,1),method='ML')
coeftest(model.321)
```

```
##
## z test of coefficients:
##
##       Estimate Std. Error z value Pr(>|z|)
## ar1 -0.179040   0.199982 -0.8953   0.3706
## ar2 -0.141590   0.205502 -0.6890   0.4908
## ar3 -0.091369   0.200271 -0.4562   0.6482
## ma1 -0.999978   0.118428 -8.4437   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model.021 = arima(chinook.spawners,order=c(0,2,1),method='ML')
coeftest(model.021)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ma1 -1.00000    0.10784 -9.2734 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model.420 = arima(chinook.spawners,order=c(4,2,0),method='ML')
coeftest(model.420)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.75193    0.17421 -4.3162 1.588e-05 ***
## ar2 -0.58233    0.22009 -2.6458  0.008149 **
## ar3 -0.36890    0.21190 -1.7409  0.081696 .
## ar4 -0.42620    0.16707 -2.5510  0.010741 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10%

```r
model.521 = arima(chinook.spawners,order=c(5,2,1),method='ML')
coeftest(model.521)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.290245   0.205466 -1.4126    0.1578
## ar2 -0.273077   0.195383 -1.3976    0.1622
## ar3 -0.216880   0.188300 -1.1518    0.2494
## ar4 -0.445696   0.179249 -2.4865    0.0129 *
## ar5 -0.057397   0.199867 -0.2872    0.7740
## ma1 -0.999998   0.138694 -7.2101 5.591e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model.520 = arima(chinook.spawners,order=c(5,2,0),method='ML')
coeftest(model.520)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.83113    0.19699 -4.2190 2.453e-05 ***
## ar2 -0.65779    0.23604 -2.7868  0.005323 **
## ar3 -0.47162    0.24455 -1.9286  0.053787 .
## ar4 -0.54719    0.22060 -2.4805  0.013119 *
## ar5 -0.15699    0.19400 -0.8092  0.418386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When we include AR = 5 and MA = 1 in the model, AR = 5 coefficient becomes insignificant at 5% level of significance. Therefore, we include ARIMA(4,2,1) model as well.

```
model.421 = arima(chinook.spawners,order=c(4,2,1),method='ML')
coeftest(model.421)
```

```
##
## z test of coefficients:
##
##      Estimate Std. Error z value  Pr(>|z|)
## ar1 -0.25836    0.17337 -1.4902   0.13617
## ar2 -0.25500    0.18610 -1.3702   0.17062
## ar3 -0.19737    0.17604 -1.1211   0.26223
## ar4 -0.42764    0.16928 -2.5262   0.01153 *
## ma1 -0.99965    0.13497 -7.4064 1.298e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For now, we will consider AIC and BIC values of the models to decide the best one within the subset of possible models.

```
AIC(model.021), model..., . . . )
```

pdq

```
## [1] 611.0137
```

```
AIC(model.121)
```

```
## [1] 612.5445
```

```
AIC(model.221)
```

```
## [1] 614.1957
```

```
AIC(model.321)
```

```
## [1] 615.9907
```

```
AIC(model.421)
```

```
## [1] 612.8191
```

```
AIC(model.521)
```

```
## [1] 614.7374
```

```
AIC(model.420)
```

```
## [1] 618.1972
```

```
AIC(model.520)
```

```
## [1] 619.5581
```

```
AIC(model.021,k = log(28)) # Computes BIC
```

```
## [1] 613.6781
```

```
AIC(model.121,k = log(28)) # Computes BIC
```

```
## [1] 616.5411
```

```
AIC(model.221,k = log(28)) # Computes BIC
```

```
## [1] 619.5245
```

```
AIC(model.321,k = log(28)) # Computes BIC
```

```
## [1] 622.6517
```

```
AIC(model.421,k = log(28)) # Computes BIC
```

```
## [1] 620.8123
```

```
AIC(model.521,k = log(28)) # Computes BIC
```

```
## [1] 624.0628
```

```
AIC(model.420,k = log(28)) # Computes BIC
```

```
## [1] 624.8582
```

```
AIC(model.520,k = log(28)) # Computes BIC
```

```
## [1] 627.5514
```

We observe the smallest AIC with model ARIMA(0,2,1), which is 611.01, and smallest BIC with model ARIMA(0,2,1), which is 613.67.

Summary

In this module, we focused on the estimation of the parameters of ARIMA models. We considered

- the method of moments,
- least squares, and
- maximum likelihood

Approaches for the estimation of ARMA parameters. The estimators were illustrated both with simulated and actual time series data.

What's next?

In the next module, we will focus on testing the goodness of fit of the fitted models.

- analysis of residuals from the fitted model and
- analysis of overparameterized models.

References

Efron, B. and Tibshirani, R. J. (1993). An Introduction to the Bootstrap. New York: Chapman and Hall.

Davison, A. C. and Hinkley, D. V. (2003). Bootstrap Methods and Their Application, 2nd ed. New York: Cambridge University Press.

West, M. (1995). Time series decomposition and analysis in a study of oxygen isotope records. Working paper. ISDS Discussion Paper, 95-18, Duke University.