Box-Jenkins Methodology: Linear Time Series Analysis Using R

Melody Ghahramani

Mathematics & Statistics

January 29, 2014
Outline

- Reading in time series (ts) data.
- Exploratory tools for ts data.
- Box-Jenkins Methodology for linear time series.

Figure: George E.P. Box
The Nature of Linear TS Data for Box-Jenkins

The data need to be:

- Continuous

- Or, be count data that can be approximated by continuous data
The data need to be:

- Continuous

Or, be count data that can be approximated by continuous data
  - eg. Monthly sunspot counts
The Nature of Linear TS Data for Box-Jenkins

The data need to be:

- Continuous

Or, be count data that can be approximated by continuous data
  - eg. Monthly sunspot counts

- Regularly spaced
The data need to be:

- Continuous

- Or, be count data that can be approximated by continuous data
  - eg. Monthly sunspot counts

- Regularly spaced
  - eg. daily, weekly, quarterly, monthly, annually
We will be using the **astsa** package written by David Stoffer and the **stats** package.

See *Time Series Analysis and Its Applications: With R Examples* by Shumway and Stoffer.

Many other time series packages are available in CRAN for estimating linear ts models.

A comprehensive link to ts analysis (not just linear ts analysis) can be found here:

http://cran.r-project.org/web/views/TimeSeries.html
co2dat = read.table("C:/R-seminar/co2-monthly.txt", header=T)
co2dat[1:15,]

<table>
<thead>
<tr>
<th>date</th>
<th>x</th>
<th>dec.date</th>
<th>average</th>
<th>interpolated</th>
<th>trend</th>
<th>days</th>
</tr>
</thead>
<tbody>
<tr>
<td>1958</td>
<td>3</td>
<td>1958.208</td>
<td>315.71</td>
<td>315.71</td>
<td>314.62</td>
<td>-1</td>
</tr>
<tr>
<td>1958</td>
<td>4</td>
<td>1958.292</td>
<td>317.45</td>
<td>317.45</td>
<td>315.29</td>
<td>-1</td>
</tr>
<tr>
<td>1958</td>
<td>5</td>
<td>1958.375</td>
<td>317.50</td>
<td>317.50</td>
<td>314.71</td>
<td>-1</td>
</tr>
<tr>
<td>1958</td>
<td>7</td>
<td>1958.542</td>
<td>315.86</td>
<td>315.86</td>
<td>314.98</td>
<td>-1</td>
</tr>
<tr>
<td>1958</td>
<td>8</td>
<td>1958.625</td>
<td>314.93</td>
<td>314.93</td>
<td>315.94</td>
<td>-1</td>
</tr>
<tr>
<td>1958</td>
<td>9</td>
<td>1958.708</td>
<td>313.20</td>
<td>313.20</td>
<td>315.91</td>
<td>-1</td>
</tr>
<tr>
<td>1958</td>
<td>10</td>
<td>1958.792</td>
<td>-99.99</td>
<td>312.66</td>
<td>315.61</td>
<td>-1</td>
</tr>
<tr>
<td>1958</td>
<td>11</td>
<td>1958.875</td>
<td>313.33</td>
<td>313.33</td>
<td>315.31</td>
<td>-1</td>
</tr>
<tr>
<td>1958</td>
<td>12</td>
<td>1958.958</td>
<td>314.67</td>
<td>314.67</td>
<td>315.61</td>
<td>-1</td>
</tr>
<tr>
<td>1959</td>
<td>1</td>
<td>1959.042</td>
<td>315.62</td>
<td>315.62</td>
<td>315.70</td>
<td>-1</td>
</tr>
<tr>
<td>1959</td>
<td>2</td>
<td>1959.125</td>
<td>316.38</td>
<td>316.38</td>
<td>315.88</td>
<td>-1</td>
</tr>
<tr>
<td>1959</td>
<td>3</td>
<td>1959.208</td>
<td>316.71</td>
<td>316.71</td>
<td>315.62</td>
<td>-1</td>
</tr>
<tr>
<td>1959</td>
<td>4</td>
<td>1959.292</td>
<td>317.72</td>
<td>317.72</td>
<td>315.56</td>
<td>-1</td>
</tr>
<tr>
<td>1959</td>
<td>5</td>
<td>1959.375</td>
<td>318.29</td>
<td>318.29</td>
<td>315.50</td>
<td>-1</td>
</tr>
</tbody>
</table>
Creating ts data in R

```r
co2 =
  ts(co2dat$interpolated, frequency=12, start=c(1958,3))
```

```r
> co2
   Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
1958 315.7 317.4 317.5 317.1 315.8 314.9 313.2 312.6 313.3 314.6 314.7
1959 315.6 316.3 316.7 317.7 318.2 318.1 316.5 314.8 313.4 314.3 315.5
1960 316.4 316.9 317.6 319.0 320.0 319.1 318.2 315.9 314.1 313.6 315.0
1961 316.9 317.0 318.5 319.4 320.6 319.7 318.5 316.7 314.8 313.8 316.1
1962 317.9 318.6 319.7 320.6 321.0 320.6 319.5 317.4 316.2 315.4 316.9
1963 318.7 319.0 319.8 321.3 322.5 321.4 319.7 316.2 315.9 317.1 318.3
1964 319.5 320.0 320.7 321.7 322.2 321.8 320.4 318.7 316.7 316.7 318.7
1965 319.4 320.4 320.8 322.1 322.1 321.8 321.3 318.8 317.8 317.3 318.8
1966 320.6 321.5 322.3 323.8 324.0 323.7 322.9 320.3 319.6 318.1 321.0
1967 322.0 322.5 323.0 324.4 325.0 324.0 322.5 320.9 319.3 319.3 320.7
1968 322.5 323.1 323.8 325.0 325.7 325.3 324.1 322.0 320.4 320.2 321.3
```
Creating ts data in R

- Sometimes the time series data set that you have may have been collected at regular intervals that were less than one year, e.g. monthly or quarterly.

- In this case, you can specify the number of times that data was collected per year by using the `frequency` parameter in the `ts()` function.

- For monthly ts data, set `frequency = 12`; for quarterly ts data, you set `frequency = 4`. 
Creating ts data in R

- Sometimes the time series data set that you have may have been collected at regular intervals that were less than one year, eg. monthly or quarterly.

- In this case, you can specify the number of times that data was collected per year by using the `frequency` parameter in the `ts()` function.

- For monthly ts data, set `frequency=12`; for quarterly ts data, you set `frequency=4`.

- You can also specify the first year that the data was collected, and the first interval in that year by using the `start` parameter in the `ts()` function.

- For example, if the first data point corresponds to the second quarter of 1986, you would set `start=c(1986,2)`.
Plotting ts data in R:

```r
plot(co2, xlab='Year', ylab='Parts per million', main='Mean Monthly Carbon Dioxide at Mauna Loa')
```
Plotting time series data in R:

```r
plot(co2, xlab='Year', ylab='Parts per million', main='Mean Monthly Carbon Dioxide at Mauna Loa')
```
Climate Change’s Psychological Milestone

Turning 400 is a lot worse than turning 40.

By Catherine Brahic | Posted Sunday, June 2, 2013, at 8:15 AM

The Keeling Curve as it surpasses 400 ppm.
Courtesy of the Scripps Institution of Oceanography at UCSD.
Assumption Needed for Box-Jenkins Model Fitting:

- Need (weakly) stationary ts: (i) constant mean, (ii) covariance is a function of lag only.
- Note: (ii) implies that variance is a constant also.
- Graphically, we look for constant mean and constant variance.
Assumption Needed for Box-Jenkins Model Fitting:

- Need (weakly) stationary ts: (i) constant mean, (ii) covariance is a function of lag only.
- Note: (ii) implies that variance is a constant also.
- Graphically, we look for constant mean and constant variance.
- If constant mean and variance are observed, we proceed with model fitting.
- Otherwise, we explore transformations of the ts such as differencing and fit models to the transformed data.
- We first explore fitting a class of models known as Integrated autoregressive moving average models (ARIMA($p, d, q$)).
Suppose we want to simulate from the following stationary processes:

#AR(1)
out1=arima.sim(list(order=c(1,0,0),ar=.9), n=100)

#MA(1)
out4=arima.sim(list(order=c(0,0,1), ma=-.5), n=100)

#ARMA(1,1)
out6=arima.sim(list(order=c(1,0,1), ar=0.9, ma=-.5), n=100)
Plots of Some Stationary Processes:

\begin{verbatim}
par(mfrow=c(3,1))

plot(out1, ylab="x",
     main=(expression(AR(1)~~~phi==+.9)))

plot(out4, ylab="x",
     main=(expression(MA(1)~~~theta==-.5)))

plot(out6, ylab="x", main=(expression(AR(1)
     ~~~phi==+.9~~~MA(1)~~~theta==-.5)))
\end{verbatim}
Plots of Some Stationary Processes (Cont’d):

AR(1) $\phi = + 0.9$

MA(1) $\theta = - 0.5$

AR(1) $\phi = + 0.9$, MA(1) $\theta = - 0.5$
install.packages("astsa")
require(astsa)

acf2(out1,48)  #prints values and plots

acf2(out4,48)

acf2(out6,48)
Model Identification of Simulated AR(1) Series:

Series: out1

LAG
ACF

LAG
PACF

Melody Ghahramani (U of Winnipeg)
R Seminar Series
January 29, 2014
Model Identification of Simulated MA(1) Series:

![Graph showing ACF and PACF for Series: out4]

ACF and PACF plots for Series: out4, showing the autocorrelation and partial autocorrelation at different lags.
Model Identification of Simulated ARMA(1,1) Series:

Series: `out6`

ACF

LAG

PACF

LAG
Plots of Theoretical ACF and PACF of an AR(2) Process:
Model Identification of ARMA($p, q$) Processes:

<table>
<thead>
<tr>
<th></th>
<th>AR($p$)</th>
<th>MA($q$)</th>
<th>ARMA($p, q$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF</td>
<td>Tails off</td>
<td>Cuts of after lag $q$</td>
<td>Tails off</td>
</tr>
<tr>
<td>PACF</td>
<td>Cuts off after lag $p$</td>
<td>Tails off</td>
<td>Tails off</td>
</tr>
</tbody>
</table>
Transforming ts data in R:

- ARMA models assume the process is weakly stationary.
- A ts plot can reveal lack of stationarity for example if:
  - there is a trend term, eg. linear, quadratic
  - the variance is not constant over time
Transforming ts data in R:

- ARMA models assume the process is weakly stationary.
- A ts plot can reveal lack of stationarity for example if:
  1. there is a trend term, eg. linear, quadratic
Transforming ts data in R:

- ARMA models assume the process is weakly stationary.
- A ts plot can reveal lack of stationarity for example if:
  1. there is a trend term, eg. linear, quadratic
  2. the variance is not constant over time
Transforming ts data in R:

- ARMA models assume the process is weakly stationary.
- A ts plot can reveal lack of stationarity for example if:
  1. there is a trend term, eg. linear, quadratic
  2. the variance is not constant over time
- Then, we need to transform the ts prior to fitting an ARMA($p$, $q$) model.
Transforming ts data in R:

Data with Trends

**Linear Trends:**
- Take a first difference: $w_t = \nabla y_t = y_t - y_{t-1}$. Then fit an ARMA model to $w_t$.
- Detrending: Fit $y_t = \beta_0 + \beta_1 \times t + a_t$. Then use residuals to fit an ARMA model.
Transforming ts data in R: Data with Trends

Linear Trends:

- Take a first difference: \( w_t = \nabla y_t = y_t - y_{t-1} \). Then fit an ARMA model to \( w_t \).
- Detrending: Fit \( y_t = \beta_0 + \beta_1 \times t + a_t \). Then use residuals to fit an ARMA model.

Quadratic Trends:

- Take a second difference:
  \[
  v_t = \nabla^2 y_t = \nabla (\nabla y_t) = w_t - w_{t-1} = y_t - 2y_{t-1} + y_{t-2}.
  \]
  Then fit an ARMA model to \( v_t \).
- Detrending: Fit \( y_t = \beta_0 + \beta_1 \times t + \beta_2 \times t^2 + a_t \). Then use residuals to fit an ARMA model.
TS Data with Trend:
Global Temperature Data (Source: Shumway & Stoffer)
ACF of TS Data with Trend and after Transformations:
Global Temperature Data (Source: Shumway & Stoffer)
TS Data with Non-constant Variance & Trend:
Johnson & Johnson Quarterly Earnings (Source: Shumway & Stoffer)
#install.packages("astsa")
#require(astsa)
data(jj)
par(mfrow=c(3,1))
plot(jj,xlab='Quarter',ylab='',main="Quarterly Earnings")

plot(log(jj),xlab='Quarter',ylab='',main="Log of Quarterly Earnings")

plot(diff(log(jj)),xlab='Quarter',ylab='',main="First Difference of Log of Quarterly Earnings")
ARIMA\((p, d, q)\) Modelling in R:
Using the \texttt{stats} package

\begin{verbatim}
arima(x, order = c(0, 0, 0),
    seasonal = list(order = c(0, 0, 0), period=NA),
    xreg = NULL, include.mean = TRUE,
    transform.pars = TRUE,
    fixed = NULL, init = NULL,
    method = c("CSS-ML", "ML", "CSS"),
    n.cond, optim.method = "BFGS",
    optim.control = list(), kappa = 1e6)
\end{verbatim}

- There are some issues with this function; see David Stoffer’s webpage for more details.
- Recommended: Use \texttt{sarima} of the \texttt{astsa} package; diagnostic plots are automatically produced.
- Note: \texttt{sarima} is a front end for \texttt{arima} function.
ARIMA($p, d, q$) Example:
Recruitment Series from astsa package:

The series represents the number of new fish from 1950-1987 ($n = 453$). The data are monthly.

data(rec)
plot(rec)
ARIMA($p$, $d$, $q$) Example:
Recruitment Series from astsa package:

```r
mean(rec)
[1] 62.26278

acf2(as.vector(rec), 48)

recruit.out = arima(rec, order=c(2, 0, 0))
```
ARIMA($p$, $d$, $q$) Example:
Recruitment Series Model Identification:

[Graph showing ACF and PACF plots for the series \textit{recruit}]

Melody Ghahramani (U of Winnipeg)
ARIMA($p, d, q$) Example:
Recruitment Series from *astsa* package (Cont’d):

> recruit.out

Call:
`arima(x = rec, order = c(2, 0, 0))`

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ar2</th>
<th>intercept</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.3512</td>
<td>-0.4612</td>
<td>61.8585</td>
</tr>
<tr>
<td>s.e.</td>
<td>0.0416</td>
<td>0.0417</td>
<td>4.0039</td>
</tr>
</tbody>
</table>

$\sigma^2$ estimated as 89.33:
log likelihood = -1661.51, aic = 3329.02
ARIMA($p$, $d$, $q$) Example:
Recruitment Series from astsa package (Cont’d):

The intercept in the `arima` function is really an estimate of the mean
(sort of).
The fitted model is

$$Y_t - 61.86 = 1.35(Y_{t-1} - 61.86) - 0.46(Y_{t-2} - 61.86) + \widehat{\alpha}_t.$$  

Now compare with

`sarima(rec,2,0,0)`
ARIMA($p, d, q$) Estimation Using `sarima` From `astsa`:

```
sarima(xdata, p, d, q, P = 0, D = 0, Q = 0, S = -1, details = TRUE, 
tol = sqrt(.Machine$double.eps), 
no.constant = FALSE)
```

The `no.constant` option:
- controls whether or not `sarima` includes a constant in the model.
- In particular, if there is no differencing ($d = 0$ and $D = 0$) you get the mean estimate.
- If there is differencing of order one (either $d = 1$ or $D = 1$, but not both), a constant term is included in the model.
- These two conditions may be overridden (i.e., no constant will be included in the model) by setting this to `TRUE`; e.g., `sarima(x,1,1,0,no.constant=TRUE)`.
Otherwise, no constant or mean term is included in the model.

The idea is that if you difference more than once \((d+D > 1)\), any drift is likely to be removed.

A possible work around if you think there is still drift when \(d+D > 1\), say \(d=1\) and \(D=1\), then work with the differenced data, e.g., \texttt{sarima(diff(x),0,0,1,0,1,1,12)}. 
ARIMA($p, d, q$) Estimation Using `sarima`

Recruitment Series (Cont’d)

Partial output from `sarima`:

```
sarima(rec, 2, 0, 0)
```

Call:
```
stats::arima(x = xdata, order = c(p, d, q),
seasonal = list(order = c(P, D, Q), period = S),
xreg = xmean, include.mean = FALSE,
optim.control = list(trace = trc,
REPORT = 1, reltol = tol))
```

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ar2</th>
<th>xmean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coef.</td>
<td>1.3512</td>
<td>-0.4612</td>
<td>61.8585</td>
</tr>
<tr>
<td>s.e.</td>
<td>0.0416</td>
<td>0.0417</td>
<td>4.0039</td>
</tr>
</tbody>
</table>
ARIMA($p, d, q$) Estimation Using \texttt{sarima}

Recruitment Series Partial Output (Cont’d)

\[ \sigma^2 \text{ estimated as } 89.33: \]
\[ \log \text{ likelihood} = -1661.51, \ aic = 3331.02 \]

\$AIC$
\[ [1] \ 5.505631 \]

\$AICc$
\[ [1] \ 5.510243 \]

\$BIC$
\[ [1] \ 4.532889 \]
ARIMA($p$, $d$, $q$) Example:
Recruitment Series from \texttt{astsa} package (Cont’d):

The following function (Yule-Walker estimator) from the \texttt{astsa} package gives the correct estimator of the mean.

\begin{verbatim}
rec.yw = ar.yw(rec,order=2)
names(rec.yw)
rec.yw$x.mean #estimate of mean
rec.yw$ar #autoregressive coefficients
sqrt(diag(rec.yw$asy.var.coef)) #se's of autoreg. param. estim's
\end{verbatim}

The fitted model is

\[ Y_t - 62.26 = 1.35(Y_{t-1} - 62.26) - 0.46(Y_{t-2} - 62.26) + \hat{a}_t. \]

See also \texttt{ar.mle}.
Once the model is fit, we need to examine its adequacy via residual analysis.

The model may need to be re-estimated.

Upon settling on an adequate model, we use it to forecast into the (not so distant) future.

Let’s see how residual analysis and forecasting are done in R using a more interesting model.
U.S. GNP Series:

- In this example, we consider the analysis of $Y_t$, the quarterly U.S. GNP series from 1947(1) to 2002(3), $n = 223$ observations.
- The data are real U.S. gross national product in billions of chained 1996 dollars and have been seasonally adjusted.
- The data were obtained from the Federal Reserve Bank of St. Louis (http://research.stlouisfed.org/) by Shumway & Stoffer.
U.S. GNP Series (Cont’d):
Clearly the GNP series is nonstationary.
The first difference $\nabla Y_t$ is highly variable.
U.S. GNP Series (Cont’d):

The growth series $\nabla \log(Y_t)$ is stationary.
U.S. GNP Series (Cont’d):
Model Identification of Growth Series

Series: as.vector(gnpgr)
data(gnp)
plot(gnp)
title('Quarterly U.S. GNP from 1947(1) to 1991(1)')
acf2(as.vector(gnp), 50)
plot(diff(gnp))
title('First Difference of U.S. GNP from
1947(1) to 1991(1)')
gnpgr = diff(log(gnp)) # growth rate
plot(gnpgr)
title('First difference of the U.S. GNP data')
acf2(as.vector(gnpgr), 24)
U.S. GNP Growth Series:
Estimation

\[ \text{ar.mod} = \text{sarima}(\text{gnpgr}, 1, 0, 0) \]
# AR(1); includes an intercept term

\[ \text{ar.mod$fit} \]

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>xmean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coeff.</td>
<td>0.3467</td>
<td>0.0083</td>
</tr>
<tr>
<td>s.e.</td>
<td>0.0627</td>
<td>0.0010</td>
</tr>
</tbody>
</table>

\[ \sigma^2 \text{ estimated as } 9.03e-05: \]
\[ \text{log likelihood } = 718.61, \text{ aic } = -1431.22 \]
U.S. GNP Growth Series:
Estimation (Cont’d)

\[
\text{ma.mod} = \text{sarima(gnpgr, 0, 0, 2)}
\]
#MA(2); includes an intercept term

\[
\text{ma.mod$fit}
\]

Coefficients:

\[
\begin{array}{ccc}
\text{ma1} & \text{ma2} & \text{xmean} \\
0.3028 & 0.2035 & 0.0083 \\
\end{array}
\]

s.e. 0.0654 0.0644 0.0010

\[
\text{sigma}^2\text{ estimated as } 8.919e-05:
\]
\[
\text{log likelihood } = 719.96,\quad \text{aic } = -1431.93
\]
Comparing AIC criteria, can select both models. Put $X_t = \nabla \log(Y_t)$.
The fitted AR(1) model is

$$X_t - 0.0083 = 0.347 (X_{t-1} - 0.0083) + \hat{a}_t$$

The fitted MA(2) model is

$$X_t - 0.0082 = \hat{a}_t + 0.303 \hat{a}_{t-1} + 0.204 \hat{a}_{t-2}$$
U.S. GNP Growth Series:
AR(1) Model Diagnostics

Standardized Residuals

ACF of Residuals

Normal Q–Q Plot of Std Residuals

p values for Ljung–Box statistic
Diagnostics

- Model diagnostics are produced automatically if you use `sarima` from the `astsa` package.
- The function `tsdiag` in the `stats` package produces INCORRECT p-values for the Ljung-Box statistics.
- See David Stoffer’s webpage on why the p-values produced are incorrect: http://www.stat.pitt.edu/stoffer/tsa3/Rissues.htm

Figure: Greta M. Ljung
Automatic ARIMA$(p, d, q)$ Model Selection in R:

- We may have several different candidate models to choose from.
- We select the model with minimum AIC or minimum BIC criterion.
- We can automate the process using the `auto.arima` function found in the `forecast` package.
- `auto.arima` outputs the same parameter estimates as `arima` from the `stats` package.
- CAUTION: Use `auto.arima` with care!
Automatic ARIMA($p, d, q$) Model Selection in R (Cont’d):

```r
install.packages("forecast")
library(forecast)

auto.arima(x, d=NA, D=NA, max.p=5, max.q=5,
            max.P=2, max.Q=2, max.order=5, start.p=2,
            start.q=2, start.P=1, start.Q=1,
            stationary=FALSE,
            seasonal=TRUE, ic=c("aicc","aic", "bic"),
            stepwise=TRUE, trace=FALSE,
            approximation=(length(x)>100 | frequency(x)>12),
            xreg=NULL, test=c("kpss","adf","pp"),
            seasonal.test=c("ocsb","ch"), allowdrift=TRUE,
            lambda=NULL, parallel=FALSE, num.cores=NULL)
```

CAUTION: Use `auto.arima` with care!
Automatic ARIMA($p$, $d$, $q$) Model Selection in R (Cont’d):

```r
armall = auto.arima(log(gnp),d=1,D=0,seasonal=FALSE)
> armall
Series: log(gnp)
ARIMA(2,1,2) with drift

Coefficients:
    ar1     ar2     ma1     ma2    drift
      1.3459  -0.7378 -1.0633  0.5620  0.0083
s.e.  0.1377  0.1543  0.1877  0.1975  0.0008

sigma^2 estimated as 8.688e-05: log likelihood=720.03
AIC=-1428.05  AICc=-1427.66  BIC=-1407.64
```
Model Selection for the GNP Growth Series:

```r
# Model Selection:

temp <- rbind(ar.mod$AIC, ar.mod$AICc, ar.mod$BIC)
temp2 <- rbind(ma.mod$AIC, ma.mod$AICc, ma.mod$BIC)
temp3 <- rbind(arma11$aic, arma11$aicc, arma11$bic)
out <- t(cbind(temp, temp2, temp3))
dimnames(out) <- list(c("AR(1)", "MA(2)", "ARMA(2,2)"),
                       c("AIC", "AICc", "BIC"))
round(out, 3)
```
Model Selection for the GNP Growth Series:

```r
> round(out,3)
  AIC   AICc   BIC
AR(1)  -8.294 -8.285 -9.264
MA(2)  -8.298 -8.288 -9.252
ARMA(2,2) -1428.054 -1427.664 -1407.638
```

- The information criteria for the AR and MA models were computed using `sarima`.
- The same criteria for the ARMA models are outputted from the `arima` function.
- For example, the AIC from `arima` is calculated using 
  \[-2 \log(likelihood)_k + 2k\], where \(k\) is the number of parameters in
  the model.
Model Selection

We use the information criteria defined as follows:

\[
\begin{align*}
\text{AIC} & = \log \hat{\sigma}_k^2 + \frac{n + 2k}{n} \\
\text{AICc} & = \log \hat{\sigma}_k^2 + \frac{n + k}{n - k - 2} \\
\text{BIC} & = \log \hat{\sigma}_k^2 + \frac{k \log n}{n}
\end{align*}
\]

where \( n \) is the length of the series and \( k \) is the number of parameters in the fitted model.
Model Selection for GNP Growth Series:

The information criteria are the following:

```r
> round(out,3)

     AIC  AICc  BIC
AR(1) -8.294 -8.285 -9.264
MA(2) -8.298 -8.288 -9.252
ARMA(2,2) -8.306 -8.295 -9.229
```

Either the AR(1) or the MA(2) model will do.
Let’s examine the residual analysis output once more.
ARIMA\((p, d, q) \times (P, D, Q)_S\) Modeling

- It may happen that a series is strongly dependent on its past at multiples of the sampling unit.
- For example, for monthly business data, quarters may be highly correlated.
- We can combine ‘seasonal models’ along with differencing, as well as the ARMA models to fit ARIMA\((p, d, q) \times (P, D, Q)_S\) models defined by

\[
\Phi(B^s)\phi(B) (1 - B^s)^D (1 - B)^d X_t = \Theta(B^s)\theta(B) w_t.
\]

- e.g. ARIMA\((0, 1, 1) \times (0, 1, 1)_{12}\) is

\[
(1 - B^{12})(1 - B) X_t = (1 + \Theta B^{12})(1 + \theta B) w_t
\]

Aside: Observe the MA parameters (plus or minus?)
Behavior of the ACF and PACF for Pure SARMA Models

<table>
<thead>
<tr>
<th></th>
<th>AR((P))_s</th>
<th>MA((Q))_s</th>
<th>ARMA((P, Q))_s</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF*</td>
<td>Tails off at lags ( k_s ), ( k = 1, 2, \ldots )</td>
<td>Cuts off after lag ( Q_s )</td>
<td>Tails off at lags ( k_s )</td>
</tr>
<tr>
<td>PACF*</td>
<td>Cuts off after lag ( P_s )</td>
<td>Tails off at lags ( k_s ), ( k = 1, 2, \ldots )</td>
<td>Tails off at lags ( k_s )</td>
</tr>
</tbody>
</table>

*The values at nonseasonal lags \( h \neq k_s \), for \( k = 1, 2, \ldots \), are zero.*
Johnson & Johnson Quarterly Earnings, revisited

Data in astsa package.

data(jj)
plot(jj)
title('Quarterly Earnings of Johnson & Johnson (J&J)')

#Transform data:
plot(diff(log(jj)),xlab='Quarter',ylab='',
     main="First Difference of Log of Quarterly Earnings")
JJ <- diff(log(jj)) #transformed series

#Model Identification
acf2(as.vector(JJ),max.lag=30)
J&J Model Identification
First difference of log-transformed series
Let’s take a seasonal difference (S=4).

Note: JJ is the first difference of log-transformed series.

```
JJ.dif <- diff(JJ,4)
acf2(as.vector(JJ.dif),max.lag=30)
```
Johnson & Johnson Model Identification (Cont’d)

A Seasonal Difference of first difference of log-transformed series; $S = 4$
logjj <- log(jj) #log-transform raw series
sarima(logjj, 1,1,1,1,1,0,4) #Candidate Model

Call:
stats::arima(x = xdata, order = c(p, d, q),
seasonal = list(order = c(P, D,Q), period = S),
optim.control = list(trace = trc, REPORT = 1,
reltol = tol))

Coefficients:

<table>
<thead>
<tr>
<th></th>
<th>ar1</th>
<th>ma1</th>
<th>sar1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.0141</td>
<td>-0.6700</td>
<td>-0.3265</td>
</tr>
</tbody>
</table>

s.e. 0.2221 0.1814 0.1320
sigma^2 estimated as 0.007913:
log likelihood = 78.46,
aic = -148.92

$AIC
[1] -3.767848

$AICc
[1] -3.73801

$BIC
[1] -4.681033
The non-seasonal AR term fails to be significant.
I refit the model without the non-seasonal AR term.
I also used `auto.arima` to see what model would be selected; a model with more parameters was selected.
I selected the $ARIMA(0, 1, 1) \times (1, 1, 0)_4$ model as it had the smaller AIC.

```
sarima(logjj, 0,1,1,1,1,0,4)
#Output omitted for brevity
```
J&J $ARIMA(0, 1, 1) \times (1, 1, 0)_4$ Model Diagnostics

Model is fit to log-transformed data
Johnson & Johnson Forecasting; four-steps ahead

Forecasts are for log-transformed data
Johnson & Johnson Forecasting; four-steps ahead
Forecasts are for log-transformed data

```
sarima.for(logjj,n.ahead=4, 0,1,1,1,1,0,4)
$pred
   Qtr1   Qtr2   Qtr3   Qtr4  
1981 2.910254 2.817218 2.920738 2.574797

$se
   Qtr1   Qtr2   Qtr3   Qtr4  
1981 0.08895758 0.09341102 0.09766159 0.10173473
```