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 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

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 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
  - eg. daily, weekly, quarterly, monthly, annually

Time Series Packages Available on CRAN

- 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`

Reading ts data in R

```
co2dat= read.table("C:/R-seminar/co2-monthly.txt",
        header=T)
co2dat[1:15,]
```

	date	x	dec.date	average	interpolated	trend	days
1	1958	3	1958.208	315.71	315.71	314.62	-1
2	1958	4	1958.292	317.45	317.45	315.29	-1
3	1958	5	1958.375	317.50	317.50	314.71	-1
4	1958	6	1958.458	-99.99	317.10	314.85	-1
5	1958	7	1958.542	315.86	315.86	314.98	-1
6	1958	8	1958.625	314.93	314.93	315.94	-1
7	1958	9	1958.708	313.20	313.20	315.91	-1
8	1958	10	1958.792	-99.99	312.66	315.61	-1
9	1958	11	1958.875	313.33	313.33	315.31	-1
10	1958	12	1958.958	314.67	314.67	315.61	-1
11	1959	1	1959.042	315.62	315.62	315.70	-1
12	1959	2	1959.125	316.38	316.38	315.88	-1
13	1959	3	1959.208	316.71	316.71	315.62	-1
14	1959	4	1959.292	317.72	317.72	315.56	-1
15	1959	5	1959.375	318.29	318.29	315.50	-1

Creating ts data in R

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

```
> co2
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1958			315.71	317.45	317.50	317.10	315.86	314.93	313.20	312.66	313.33	314.67
1959	315.62	316.38	316.71	317.72	318.29	318.15	316.54	314.80	313.84	313.26	314.80	315.58
1960	316.43	316.97	317.58	319.02	320.03	319.59	318.18	315.91	314.16	313.83	315.00	316.19
1961	316.93	317.70	318.54	319.48	320.58	319.77	318.57	316.79	314.80	315.38	316.10	317.01
1962	317.94	318.56	319.68	320.63	321.01	320.55	319.58	317.40	316.26	315.42	316.69	317.69
1963	318.74	319.08	319.86	321.39	322.25	321.47	319.74	317.77	316.21	315.99	317.12	318.31
1964	319.57	320.07	320.73	321.77	322.25	321.89	320.44	318.70	316.70	316.79	317.79	318.71
1965	319.44	320.44	320.89	322.13	322.16	321.87	321.39	318.81	317.81	317.30	318.87	319.42
1966	320.62	321.59	322.39	323.87	324.01	323.75	322.39	320.37	318.64	318.10	319.79	321.08
1967	322.07	322.50	323.04	324.42	325.00	324.09	322.55	320.92	319.31	319.31	320.72	321.96
1968	322.57	323.15	323.89	325.02	325.57	325.36	324.14	322.03	320.41	320.25	321.31	322.84

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**.

Creating ts data in R

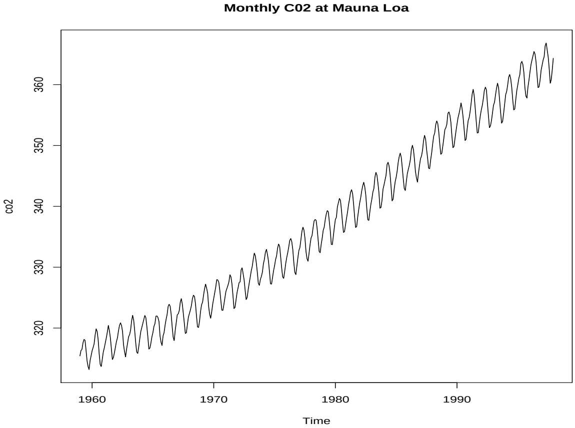
- Sometimes the time series data set that you have may have been collected at regular intervals that were less than one year,eg. monthly or quarterly.
- In this case, you can specify the number of times that data was collected per year by using the **frequency** parameter in the ts() function.
- For 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:

```
plot(co2,xlab='Year',ylab='Parts per million', main='Mean Monthly
Carbon Dioxide at Mauna Loa')
```

Plotting ts data in R:

plot(co2,xlab='Year',ylab='Parts per million', main='Mean Monthly Carbon Dioxide at Mauna Loa')



Time Series Data in the News:

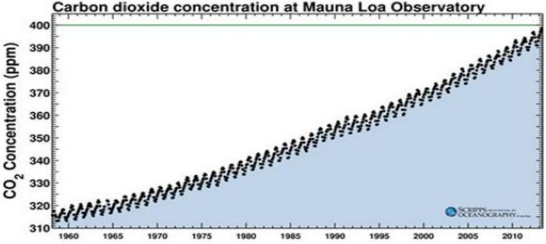
HOME / NEW SCIENTIST : STORIES FROM NEW SCIENTIST.

Article from **NewScientist**

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)).

Simulating ARIMA(p, d, q) Processes in R

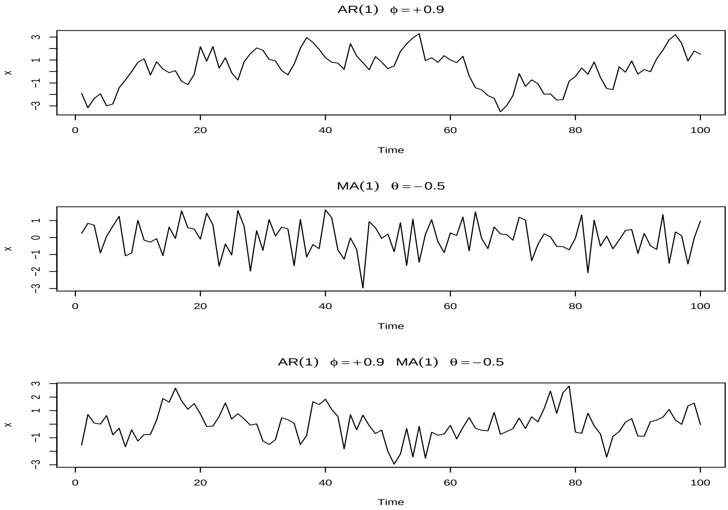
Suppose we want to simulate from the following stationary processes:

```
#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:

```
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)))
```

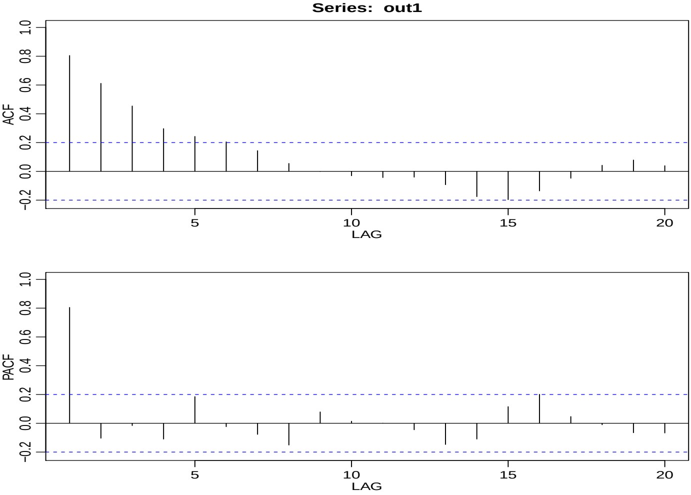
Plots of Some Stationary Processes (Cont'd):



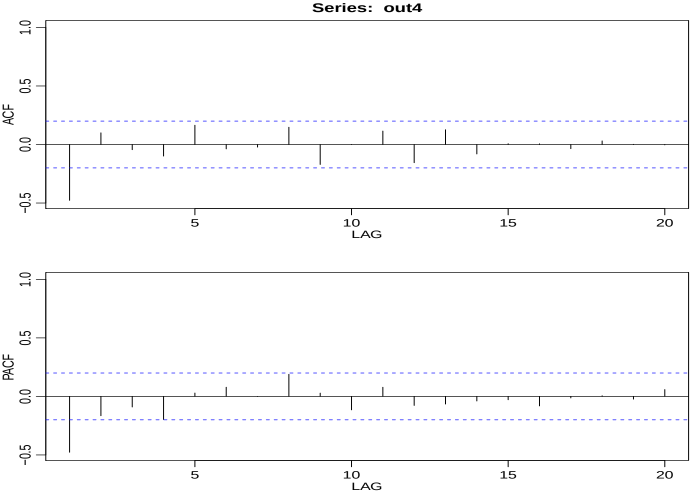
Model Identification of ARMA(p, q) Processes Using R:

```
install.packages("astsa")
require(astsa)

acf2(out1,48) #prints values and plots


acf2(out4,48)


acf2(out6,48)
```

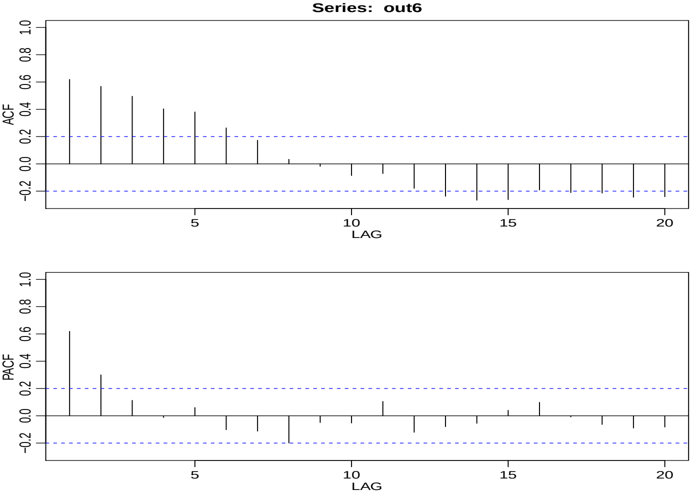
Model Identification of Simulated AR(1) Series:



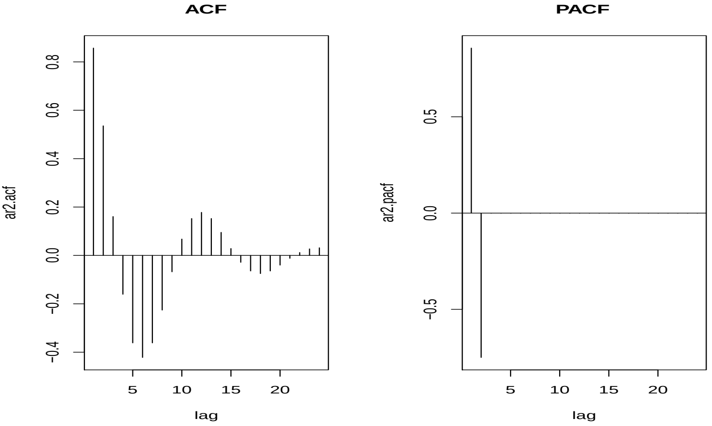
Model Identification of Simulated MA(1) Series:



Model Identification of Simulated ARMA(1,1) Series:



Plots of Theoretical ACF and PACF of an AR(2) Process:



Model Identification of ARMA(p, q) Processes:

	AR(p)	MA(q)	ARMA(p, q)
ACF	Tails off	Cuts of after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

Transforming ts data in R:

- ARMA models assume the process is weakly stationary.
- A ts plot can reveal lack of stationarity for example if:

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: $w_t = \bigtriangledown y_t = y_t - y_{t-1}$. 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.

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:

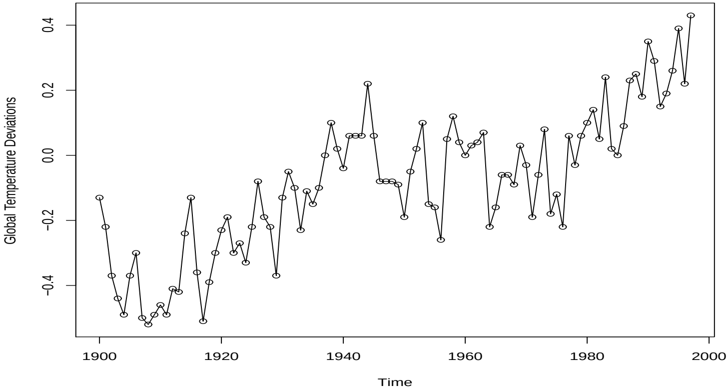
- Take a second difference:

$$v_t = \nabla^2 y_t = \nabla(\nabla y_t) = w_t - w_{t-1} = y_t - 2y_{t-1} + y_{t-2}.$$

  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.

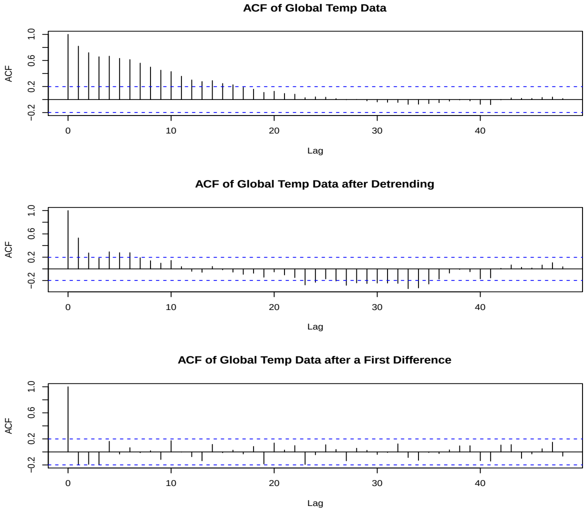
TS Data with Trend:

Global Temperature Data (Source: Shumway & Stoffer)



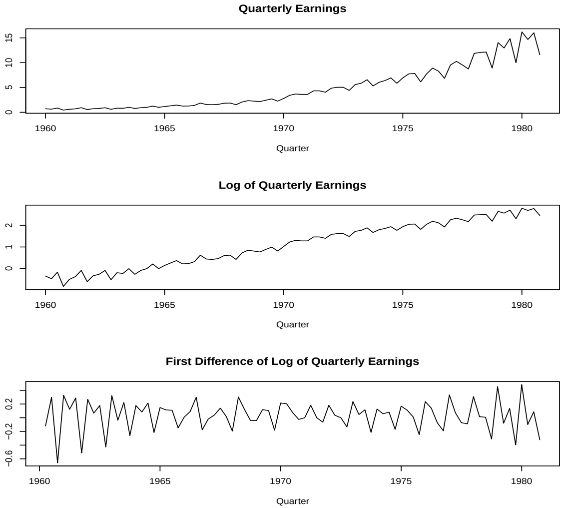
ACF of TS Data with Trend and after Transformations:

Global Temperature Data (Source: Shumway & Stoffer)



TS Data with Non-constant Variance & Trend:

Johnson & Johnson Quarterly Earnings (Source: Shumway & Stoffer)



Differencing and log-transformations in R:

Data 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 **stats** package

```
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)
```

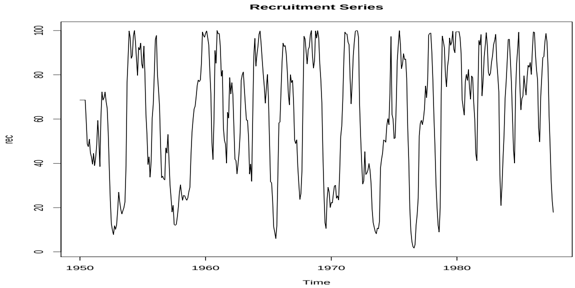
- There are some issues with this function; see David Stoffer's webpage for more details.
- Recommended: Use **sarima** of the **astsa** package; diagnostic plots are automatically produced.
- Note: **sarima** is a front end for **arima** function.

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)
```



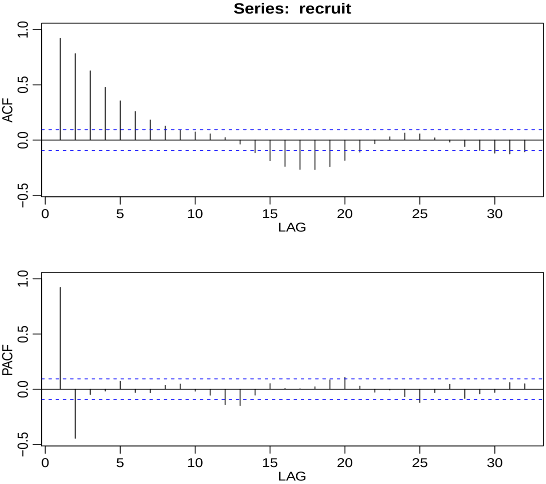
ARIMA(p, d, q) Example:

Recruitment Series from **astsa** package:

```
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:



ARIMA(p, d, q) Example:

Recruitment Series from **astsa** package (Cont'd):

```
> recruit.out

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

Coefficients:
         ar1      ar2  intercept
      1.3512  -0.4612    61.8585
s.e.  0.0416   0.0417     4.0039

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

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

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).

sarima (Cont'd)

- 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., 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:
         ar1      ar2    xmean
      1.3512  -0.4612  61.8585
s.e.  0.0416   0.0417   4.0039
```

ARIMA(*p*, *d*, *q*) Estimation Using **sarima**

Recruitment Series Partial Output (Cont'd)

```
sigma^2 estimated as 89.33:
log 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 **astsa** package (Cont'd):

The following function (Yule-Walker estimator) from the **astsa** package gives the correct estimator of the mean.

```
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
```

The fitted model is

$$Y_t - 62.26 = 1.35(Y_{t-1} - 62.26) - 0.46(Y_{t-2} - 62.26) + \widehat{a}_t.$$

See also **ar.mle**.

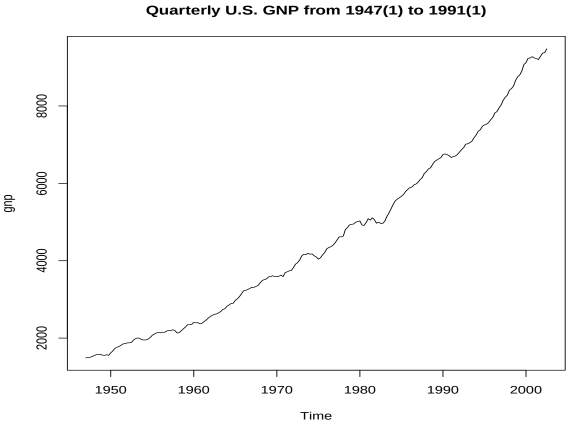
After ARIMA model Estimation...

- Once the model is fit, we need to examine is 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.

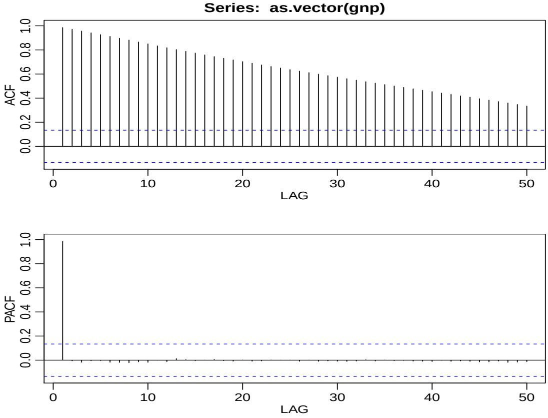
U.S. GNP Series:

- 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):

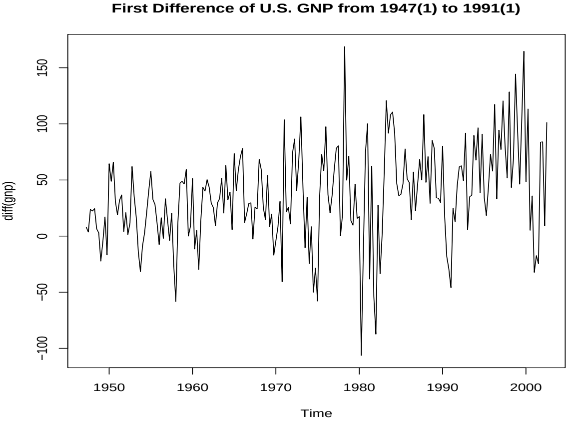


U.S. GNP Series (Cont'd):



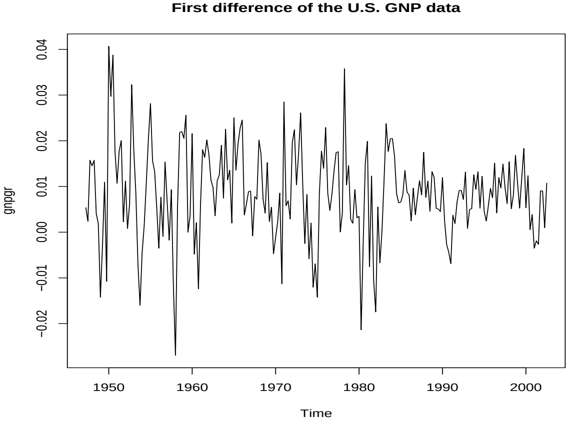
Clearly the GNP series is nonstationary.

U.S. GNP Series (Cont'd):



The first difference ∇Y_t is highly variable.

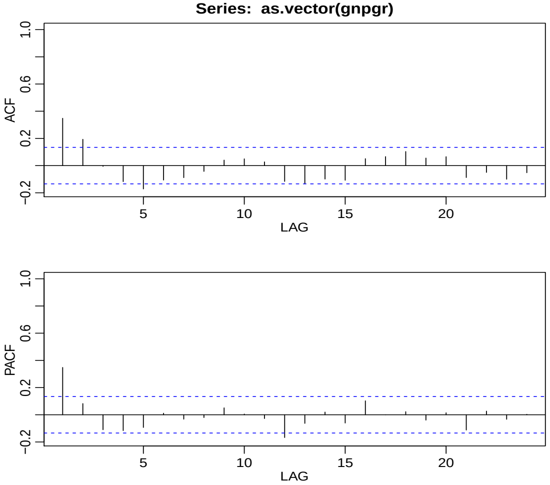
U.S. GNP Series (Cont'd):



The growth series $\nabla \log(Y_t)$ is stationary.

U.S. GNP Series (Cont'd):

Model Identification of Growth Series



U.S. GNP Series:

Model Identification

```
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

```
ar.mod = sarima(gnpgr, 1, 0, 0)
# AR(1); includes an intercept term

ar.mod$fit

Coefficients:
         ar1   xmean
      0.3467  0.0083
s.e.  0.0627  0.0010

sigma^2 estimated as 9.03e-05:
log likelihood = 718.61, aic = -1431.22
```

U.S. GNP Growth Series:

Estimation (Cont'd)

```
ma.mod = sarima(gnpgr, 0, 0, 2)
#MA(2); includes an intercept term

ma.mod$fit

Coefficients:
         ma1     ma2   xmean
      0.3028  0.2035  0.0083
s.e.  0.0654  0.0644  0.0010

sigma^2 estimated as 8.919e-05:
log likelihood = 719.96,  aic = -1431.93
```

Comparing AIC criteria, can select both models. Put $X_t = \nabla \log(Y_t)$. The fitted AR(1) model is

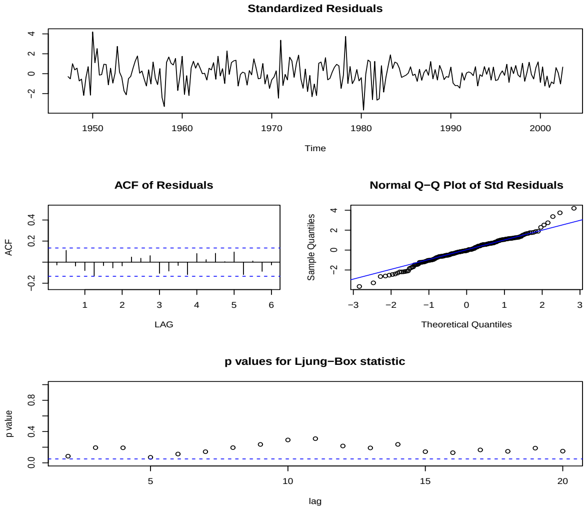
$$X_t - 0.0083 = 0.347\,(X_{t-1} - 0.0083) + \widehat{a}_t$$

The fitted MA(2) model is

$$X_t - 0.0082 = \widehat{a}_t + 0.303\,\widehat{a}_{t-1} + 0.204\,\widehat{a}_{t-2}$$

U.S. GNP Growth Series:

AR(1) Model Diagnostics



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):

```
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)
```

Automatic ARIMA(p, d, q) Model Selection in R (Cont'd):

```
arma11 = auto.arima(log(gnp),d=1,D=0,seasonal=FALSE)
> arma11
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:

```
#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:

```
> 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 $-2\log(\textit{likelihood})_k + 2\,k$, where k is the number of parameters in the model.

Model Selection

We use the information criteria defined as follows:

$$\begin{aligned}
\text{AIC} &= \log \widehat{\sigma}_k^2 + \frac{n+2k}{n} \\
\text{AICc} &= \log \widehat{\sigma}_k^2 + \frac{n+k}{n-k-2} \\
\text{BIC} &= \log \widehat{\sigma}_k^2 + \frac{k \log n}{n}
\end{aligned}$$

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:

```
> 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

	$AR(P)_s$	$MA(Q)_s$	$ARMA(P,Q)_s$
ACF*	Tails off at lags ks, $k = 1, 2, \ldots,$	Cuts off after lag Qs	Tails off at lags ks
PACF*	Cuts off after lag Ps	Tails off at lags ks $k = 1, 2, \ldots,$	Tails off at lags ks

*The values at nonseasonal lags $h \neq ks$, for $k = 1, 2, \ldots,$ are zero.

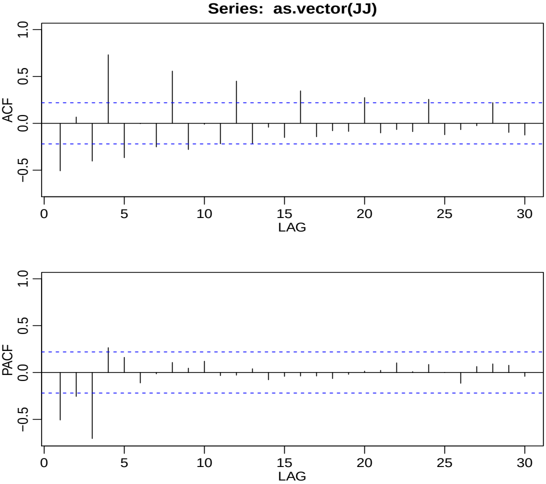
Johnson & Johnson Quarterly Earnings, revisited

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



Johnson & Johnson Model Identification (Cont'd)

First difference of log-transformed series

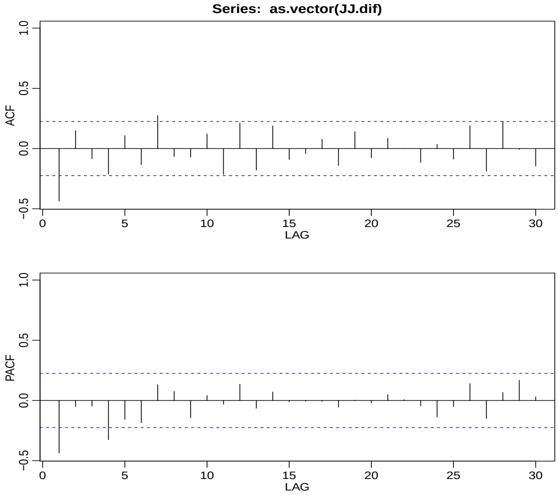
Let's take a seasonal difference (S=4).

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; $S = 4$



Johnson & Johnson Model Estimation

```
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:
         ar1      ma1     sar1
     -0.0141  -0.6700  -0.3265
s.e.  0.2221   0.1814   0.1320
```

Johnson & Johnson Model Estimation (Cont'd)

```
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
```

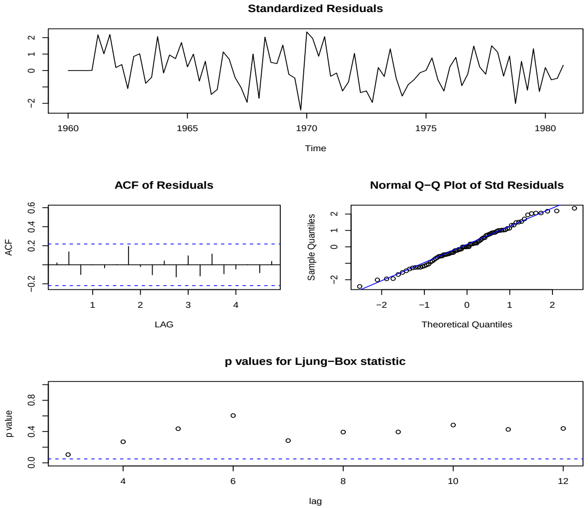
Johnson & Johnson Model Estimation (Cont'd)

- 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
```

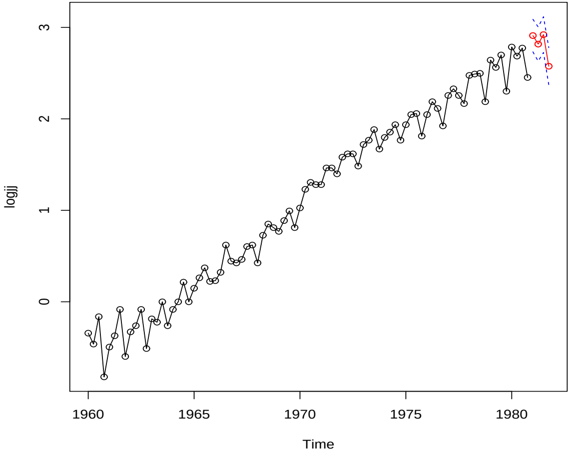
J&J $ARIMA(0,1,1) \times (1,1,0)_4$ Model Diagnostics

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
```