Ch 5: ARIMA model

Contents

March 22, 2017

1

Non-stationary Time Series and Box-Jenkins Method

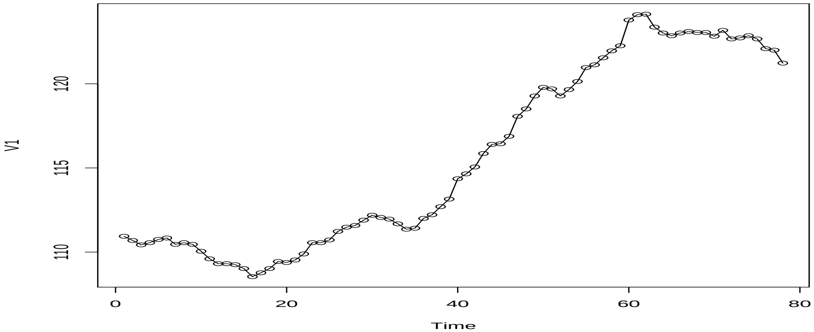
[ToC]

1.1 Non-Stationary Data

[ToC]

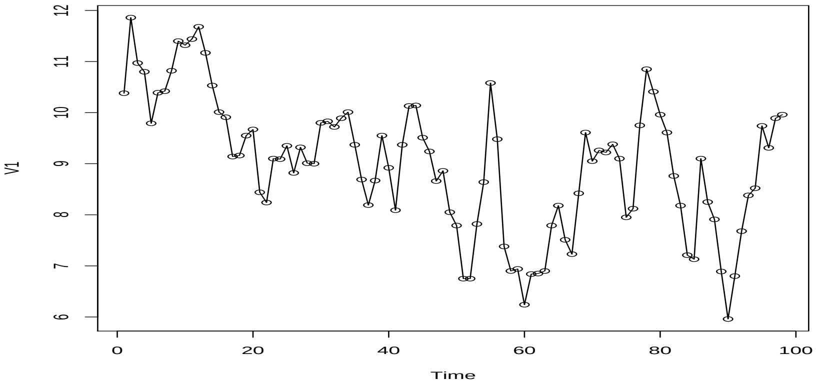
Dow Jones Index

From Aug. 28 to Dec. 18, 1972



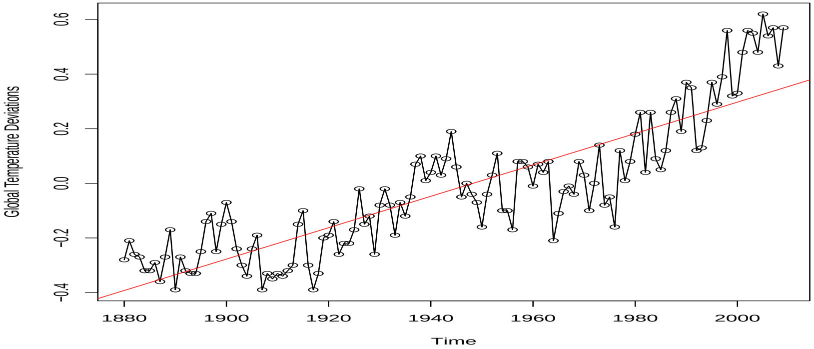
Lake Huron Data

Level of Lake Huron 1875-1972



Global Temperature Data

Global temparature data from Shumway



1.2 Tests for Stationarity

[ToC]

How do you check if a time seirs is stationary?

1. Visual inspection for constant mean, constant variance
2. KPSS test (H_0: Stationary)
3. ADF test (H_0: Not Stationary)
4. PP test (H_0: Not Stationary)
5. Fit AR(1), and see if $\hat{\phi}_1$ is significantly different from 1.

KPSS test

Kwiatkowski-Phillips-Schmidt-Shin (1992) test for

- Default choice in `auto.arima()`

$$\begin{cases} H_0 : X_t \text{ is trend stationary} \\ H_A : X_t \text{ is not stationary} \end{cases}$$

- Large p-value means "Stationary".
- Decompose the series as

$$X_t \quad = \quad T_t + W_t + Y_t$$

  where T_t is deterministic trend, W_t is random walk, and Y_t is stationary error.

- Lagrange multiplier test that the random walk has zero variance.

ADF test

Augumented Dickey-Fuller test

- "Unit-root" test
-

$$\begin{cases} H_0 : Y_t \text{ is not stationary} \\ H_A : Y_t \text{ is stationary} \end{cases}$$

  Is replaced by

$$\begin{cases} H_0 : Y_t \text{ has unit root} \\ H_A : Y_t \text{ does not have unit root} \end{cases}$$

- Small p-value rejects the null hypothesis of non-stationarity, and means "Stationary".

Fit AR(1) to a time series, and test if ϕ_1 is significantly different from 1, by

$$\frac{\hat{\phi}_1 - 1}{SE(\hat{\phi}_1)} \sim \mathcal{N}(0, 1)$$

PP test

Phillips-Perron test for the null hypothesis that x has a unit root.

- Same Hypothesis as ADF test
- Small p-value means "Stationary".
- The tests are similar to ADF tests, but they incorporate an automatic correction to the DF procedure to all ow for autocorrelated residuals.
- Calculation of the test statistics is complex.
- The tests usually give the same conclusions as the ADF tests

Three stationarity test

- P-value of (KPSS, ADF, PP)
  - (Large, small, small) $\rightarrow$ All three indicating stationarity.
  - (Small, large, large) $\rightarrow$ All three indicating non-stationarity.
  - (Large, Large, large) $\rightarrow$ Conflicing conclusions, inconclusive.

- `Stationarity.tests()` performs all of the three tests. It is located at the same place as `Randomness.tests()`. Command below will load it into R.

```
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")
```

Example:

```
  library(forecast)
  source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

#--- Test it on random sample ---
  X <- rnorm(100)
  plot(X, type="o")

  Stationarity.tests(X)

#--- Lake Huron Data ---
  X1 <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/lake.txt")
  X  <- ts(X1, start=1875, freq=1)
  plot(X, type="o")

  Stationarity.tests(X)

#--- Global Temp Data ---
#- install.packages("astsa")

  library(astsa)
  data(gtemp)
  plot(gtemp, type="o", ylab="Global Temperature Deviations")

  Stationarity.tests(gtemp)
```

Example:

```
#--- Australian Steel Data
  D  <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/Steel.csv")
  D1 <- ts(D[,2], start=c(1956,1), freq=12)
  D2 <- window(D1, start=c(1969, 12))

  plot(D1, type="o")
  Stationarity.tests(D1)

  plot(D2, type="o")
  Stationarity.tests(D2)

  D4 <- window(D2, end=c(1981, 12))
  D5 <- window(D2, start=c(1984, 1))

  Stationarity.tests(D4)
  Stationarity.tests(D5)
```

1.3 Box-Jenkins Method

[ToC]

Backward and Difference Operator

- Define the backward operator B,

$$BX_t = X_{t-1}.$$

- Then define **difference operator** $\bigtriangledown$ (del),

$$\bigtriangledown = (1 - B).$$

- For example,

$$\begin{aligned} \bigtriangledown X_t &= (1 - B)X_t = X_t - X_{t-1} \\ \bigtriangledown^2 X_t &= (1 - B)(1 - B)X_t = X_t - 2X_{t-1} + X_{t-2} \end{aligned}$$

Differencing Linear Trend

- Suppose your time series have a line plus a zero-mean staionary noise.

$$Y_t = a + bt + X_t$$

That means

$$Y_{t-1} = a + b(t-1) + X_{t-1}$$

$$\bigtriangledown Y_t = Y_t - Y_{t-1} = b + X_t - X_{t-1}$$

- If X_t is stationary, then $X_t - X_{t-1}$ is also stationary. Thus we have now

$$\bigtriangledown Y_t = \mu + M_t$$

We can try to model this with ARMA(p,q) with intercept $\mu = b$.

Differencing Quadratic Trend

If m_t is quadratic, then

$$
\begin{aligned}
Y_t &= a + bt + ct^2 + X_t \\
Y_{t-1} &= a + b(t-1) + c(t-1)^2 + X_{t-1} \\
Y_{t-2} &= a + b(t-2) + c(t-2)^2 + X_{t-2}.
\end{aligned}
$$

That means

$$
\begin{aligned}
\bigtriangledown^2 Y_t &= Y_t - 2Y_{t-1} + Y_{t-2} \\
&= 2c + X_t - 2X_{t-1} + X_{t-2}
\end{aligned}
$$

If X_t is stationary, so is $X_t - 2X_{t-1} + X_{t-2}$.

If m_t is polynomial of deg k, with coefficients $c_0, c_1, \ldots$ then applying k power of difference operator will remove the trend.

$$\bigtriangledown^k Y_t = k! c_k + \bigtriangledown^k X_t.$$

Then you will end up with some statonary series $\bigtriangledown^k X_t$ with constant trend $k! c_k$.

Example: Linear Trend

```
t   <- 1:100
Y   <- 3 - .1*t + arima.sim(n=100, list(ar = c(.7,.2)))

plot(Y,type="o")         #- Y is simulated data

Y2 <- diff(Y)            #- take the difference

plot(Y2, type="o" )
```

Example: Quadratic Trend

```
t   <- 1:100
tsq <- t^2
Y   <- 3 - .5*t + .1*tsq + arima.sim(n=100, list(ar = c(.7,.2)))*10

plot(Y,type="o")

plot(diff(Y),type="o")

plot(diff(diff(Y)),type="o")
```

1.4 Random Trends

[ToC]

Suppose we have a model with trend

$$Y_t = M_t + X_t,$$

where M_t is a random walk:

$$M_t = \sum_{i=1}^{t} e_i \quad \text{where} \quad e_i \sim_{iid} \mathcal{N}(0,1).$$

Random Walk

$$M_t = \sum_{i=1}^{t} e_i \qquad \text{where} \qquad e_i \sim_{iid} \mathcal{N}(0, \sigma^2).$$

With iid errors $(e_1, e_2, e_3, \ldots)$ random walk is generated as

$$\begin{aligned}
M_1 &= e_1 \\
M_2 &= e_1 + e_2 \\
M_3 &= e_1 + e_2 + e_3 \\
M_4 &= e_1 + e_2 + e_3 + e_4 \\
&\vdots
\end{aligned}$$

We can use `cumsum()` function in R to do this easily.

Mean and Var of RW

$$M_t = \sum_{i=1}^{t} e_i \qquad \text{where} \qquad e_i \sim_{iid} \mathcal{N}(0, \sigma^2).$$

Mean is

$$E(M_t) \;=\; E\left(\sum_{i=1}^{t} e_i\right) \;=\; \sum_{i=1}^{t} E(e_i) \;=\; 0$$

Variance is

$$V(M_t) \;=\; V\left(\sum_{i=1}^{t} e_i\right) \;=\; \sum_{i=1}^{t} V(e_i) \;=\; \sigma^2 t$$

Example: RW

```
e = rnorm(100)      #- 100 random number from N(0,1)
M = cumsum(e)
plot(M,type="o")

plot(M,type="l", ylim=c(-40,40))



e = rnorm(100)      #- copy and paste these 3 lines many times
M = cumsum(e)
lines(M)
```

Random Walk Difference

Since

$$M_t = \sum_{i=1}^{t} e_i,$$

we have

$$\bigtriangledown M_t \quad = \quad M_t - M_{t-1} \quad = \quad \sum_{i=1}^{t} e_i - \sum_{i=1}^{n-1} e_i \quad = \quad e_t$$

And we know that $e_t \sim \mathcal{N}(0, \sigma)$.

So if M_t is Random Walk, then $\bigtriangledown M_t$ should look like iid $\mathcal{N}(0, \sigma)$ noise.

Example: RW

```
e = rnorm(100)       #- 100 random number from N(0,1)
M = cumsum(e)
plot(M,type="o")

layout(c(1,2)); plot(diff(M), type="o"); acf(diff(M))
hist(diff(M))
```

Random Walk with a Drift

Random Walk with drift δ is

$$M_t = \sum_{i=1}^{t} e_i \qquad \text{where} \qquad e_i \sim_{iid} \mathcal{N}(\delta, 1).$$

That means if M_t is Random Walk with drift, then $\bigtriangledown M_t$ should look like iid $\mathcal{N}(\delta, \sigma)$ noise.

Example: RW with drift

```
e = rnorm(100, .1, 1)      #- 100 random number from N(.1,1)
M = cumsum(e)
plot(M,type="o")

plot(M,type="l", ylim=c(-40,40))


e = rnorm(100, .1, 1)      #- copy and paste these 3 lines many times
M = cumsum(e)
lines(M)
```

Random Walk as Trend

If $Y_t = M_t + X_t$, then

$$\begin{aligned} Y_t &= \sum_{i=1}^{t} e_i + X_t \\ Y_{t-1} &= \sum_{i=1}^{t-1} e_i + X_{t-1} \end{aligned}$$

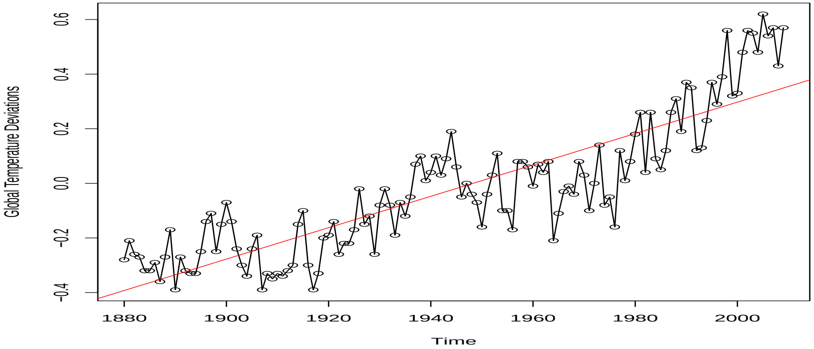
That means

$$\bigtriangledown Y_t = Y_t - Y_{t-1} = e_t + X_t - X_{t-1}$$

Since X_t, e_t are stationary, so is $e_t + X_t - X_{t-1}$.

Example: Temperature Data

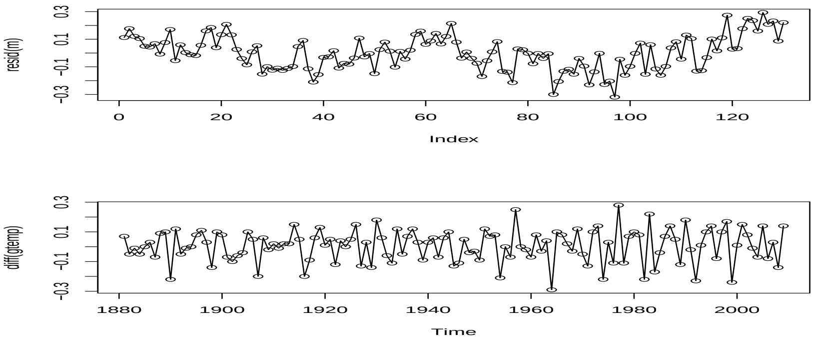
Global temparature data from Shumway



```
library(astsa)
data(gtemp)

plot(gtemp, type="o", ylab="Global Temperature Deviations")

fit <- lm(gtemp~time(gtemp));

layout(c(1,2))
plot(fit$residuals, type="o"); plot(diff(gtemp), type="o")

layout(c(1,2))
acf(fit$residuals);  acf(diff(gtemp))
```

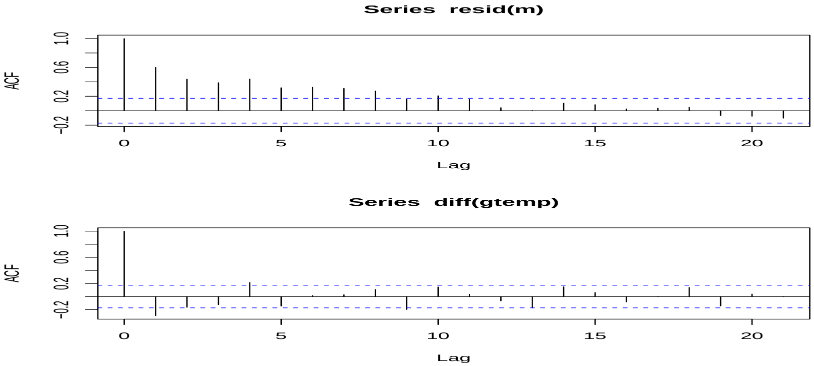
Example: Temperature Data

Residuals from regression (Top) vs Differencing with Lag 1 (Bottom)



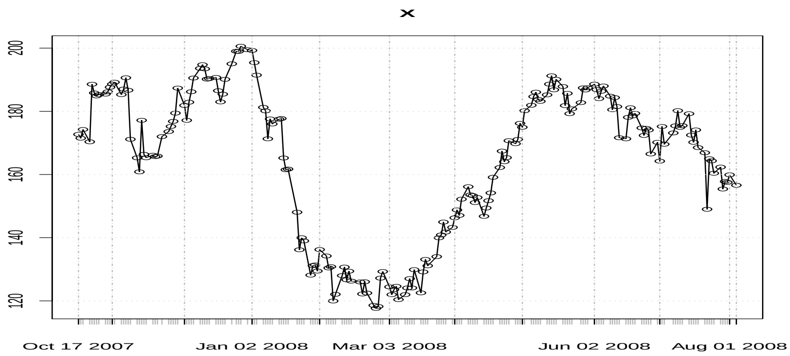
Example: Temperature Data

ACF of residuals (Top) vs ACF of Differences (Bottom)



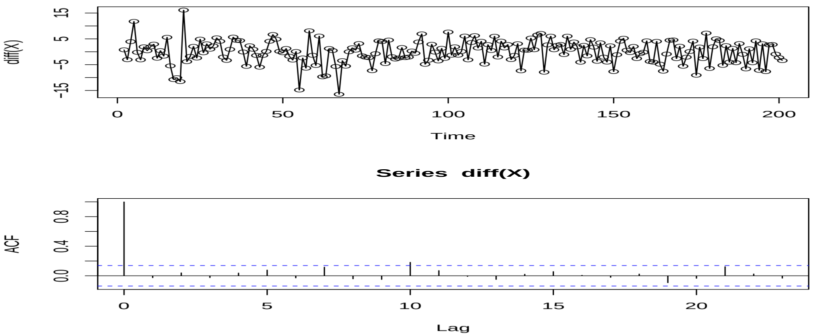
Example: AAPL

Daily adjusted close price of Apple, Inc. Between 10/17/2007 to 8/4/2008 from Yahoo! finance website



Example: AAPL

Difference at lag 1 (Top) and ACF of the difference (Bottom)



```r
install.packages("quantmod")  #- install if first time

library(quantmod)

getSymbols("AAPL")            #- download from Yahoo!

X <- as.ts(AAPL$AAPL.Adjusted[400:200])
plot(X, type="o")

layout(c(1,2))
plot(diff(X), type="o"); acf(diff(X))
```

1.5 Summary 1:

[ToC]

- Three popular test for stationarity is KPSS, ADF, and PP test.
- Tests are availabe in R, can be implemented with below command.

```
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")
Stationarity.tests(X)
```

- Classical method tries to identify trend, and tries to removeit by polynomial regression.
- Box-Jenkins Method tries to take a difference between today's data and yesterday's by applying $\bigtriangledown = (1 - B)$ operator.
- applying B-J method will often times stationarize non-stationary time series.
- We can use ARMA(p, q) model to model the stationarized version of the series.
- If you use $\bigtriangledown$ once, and apply ARMA(2,3), then it is called ARIMA (2,1,3) model.

2

Examples of ARIMA fitting

[ToC]

2.1 ARIMA(p,d,q) model

[ToC]

- Defiend as

$$\nabla^d Y_t = X_t$$

$$\Phi(B)\, X_t = \Theta(B)\, e_t$$

where $e_t \sim WN(0, \sigma^2)$.

- So ARIMA(p,d,q) model means you difference the data d times, then you get ARMA(p, q).

2.2 Lake Huron Data

Level of Lake Huron 1875-1972

```
library(forecast)
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

D  <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/lake.txt")
D1 <- ts(D, start=1875, freq=1)


#--- 1. Direct ARMA fit ---
Fit1 <- auto.arima(D1, d=0)    # find best ARMA(p,q) by AICc
Fit1

Randomness.tests(Fit1$resid)


#--- 2. Fit linear trend ---
Reg2 <- lm(D1~time(D1))
summary(Reg2)

plot(D1, type="o"); abline(Reg2, col="red")

Fit2 <- auto.arima(Reg2$residuals, d=0)
Fit2
Randomness.tests(Fit2$resid)
```

```
#--- 3. Direct ARIMA fit ---
Fit3 <- auto.arima(D1)

Randomness.tests(Fit3$resid)


Stationarity.tests(D1)
plot(forecast(Fit2))
plot(forecast(Fit3))
```

Lake Huron Data

1.

$$Y_t \text{ is ARMA}$$

2.

$$Y_t = a + bt + X_t \qquad X_t \text{ is ARMA}$$

3.

$$\bigtriangledown Y_t \text{ is WN}$$
$$Y_t \text{ is ARIMA(0,1,0)}$$

2.3 Sheep Data

Annual sheep population (1000s) in England and Wales 1867 1939

```
D  <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/sheep.csv", header=T)
D1 <- ts(D[,2], start=c(1867), freq=1)

plot(D1)

#--- 1. Direct ARIMA fit ---
Fit1 <- auto.arima(D1)
Fit1

Randomness.tests(Fit1$resid)


#--- 2. Fit linear trend ---
Reg2 <- lm(D1~time(D1))
summary(Reg2)

plot(D1, type="o"); abline(Reg2, col="red")

Fit2 <- auto.arima(Reg2$residuals, d=0)
Fit2
Randomness.tests(Fit2$resid)
```

Sheep Data

1.

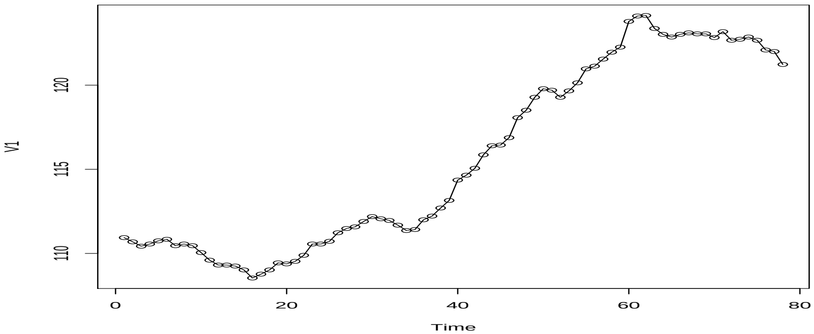
$$\begin{aligned} Y_t & \quad \text{is ARIMA(2,1,2) without drift} \\ Y_t &= a + X_t \\ \nabla Y_t &= X_t - X_{t-1} \text{ is ARMA(2,2) without drift} \end{aligned}$$

2.

$$Y_t = a + bt + X_t \qquad X_t \text{ is MA(3)}$$

2.4 Dow Jones Data

Dow Jones Index Aug. 28Dec. 18, 1972



```
D <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/dowj.csv")
X <- ts(D, start=c(1,1), freq=1)
plot(X, type='o')


delX <- diff(X)
plot(delX, type='o')

acf(delX)
pacf(delX)

library(forecast)
Fit1 <- auto.arima(delX, d=0)     #- Fit best ARMA to Y using  AIC.
Randomness.tests(Fit1$residuals)
Fit1

#-- Exactly same as below --
Fit2 <- Arima(delX, order=c(1,0,1), include.mean=FALSE)  #- same as demean=FALSE
Randomness.tests(Fit2$residuals)
Fit2
```

Current Model

- Differenced once, fit ARMA(1,1) with zero-mean.

$$
\begin{aligned}
\triangledown Y_t &= X_t \\
(1 - \phi_1 B)\, X_t &= (1 - \theta_1)\, e_t
\end{aligned}
$$

where $E(X_t) = 0$.

- This is same as

$$
(1 - \phi_1 B)\, (X_t - X_{t-1}) = e_t - \theta_1 e_{t-1}
$$

Constant terms after differencing

- After differencing, data may have non-zero mean.

$$\bigtriangledown Y_t = Y_t - Y_{t-1} = \mu + X_t$$

  where X_t is zero-mean ARMA.

- Differencing again does remove μ, but it will make X_t to be "over-differenced".

- Instead, we should just model $\mu + X_t$ with non-zero mean ARMA with intercept term.

```
plot(X, type='o')
plot(delX, type='o')

Fit2 <- Arima(delX, order=c(1,0,1), include.mean=FALSE)
Fit2
Randomness.tests(Fit1$residuals)

Fit3 <- Arima(delX, order=c(1,0,1), include.mean=TRUE)
Fit3
Randomness.tests(Fit3$residuals)
```

Current Models

1. $d = 1$, fit ARMA(1,1) with zero-mean.

$$\begin{aligned} \nabla Y_t &= X_t \\ (1 - \phi_1 B)\, X_t &= (1 - \theta_1)\, e_t \end{aligned}$$

2. $d = 1$ fit ARMA(1,1) with non-zero mean

$$\begin{aligned} \nabla Y_t &= \mu + X_t \\ (1 - \phi_1 B)\, X_t &= (1 - \theta_1)\, e_t \end{aligned}$$

1 was deemed better with AICc.

2.5 Drift vs Constant

[ToC]

- If original Y_t has a linear trend, it will show up after differencing.

$$\begin{aligned} Y_t &= a + bt + R_t \\ \bigtriangledown Y_t &= Y_t - Y_{t-1} = b + (R_t - R_{t-1}) = b + X_t \end{aligned}$$

- If original Y_t has a Random Walk with drift (cf. p.21), it will also show up after differencing.

$$\begin{aligned} Y_t &= W_t(\delta) + R_t \\ \bigtriangledown Y_t &= \delta + (R_t - R_{t-1}) = \delta + X_t \end{aligned}$$

- After differencing, drift term becomes constant.

```
Fit4 <- Arima(X, order=c(1,1,1), include.drift=T)
Fit4

Fit5 <- Arima(delX, order=c(1,0,1), include.drift=F)
Fit5

Fit6 <- Arima(delX, order=c(1,0,1), include.drift=T)
Fit6


# drift in Fit4 and intercept in Fit5 is the same constant
# drift in Fit6 is not.

# Note the numerical difference in AICc in Fit4 vs Fit5
```

3

ARIMA(p,d,q) Model Selection

[ToC]

3.1 Model Selection in ARIMA(p, d, q)

[ToC]

---

1. Use stationarity test to decide how many times to defference (d).
2. When AR(1) parameter is estimated, see if ϕ_1 is significantly different from 1.
3. Watch for sign of overdifferencing.

auto.arima()

Hyndman-Khandakar algorithm for automatic ARIMA modelling in `auto.arima`

1. The number of differences d is determined using repeated KPSS tests.
2. The values of p and q are then chosen by minimizing the AICc after differencing the data d times. Rather than considering every possible combination of p and q, the algorithm uses a stepwise search to traverse the model space.
   (a) The best model (with smallest AICc) is selected from the following four:
   ARIMA(2,d,2), ARIMA(0,d,0), ARIMA(1,d,0), ARIMA(0,d,1).
   If $d = 0$ then the constant c is included; if $d \geq 1$ then the constant c is set to zero. This is called the "current model".
   (b) Variations on the current model are considered:
   vary p and/or q from the current model by ± 1; include/exclude c from the current model. The best model considered so far (either the current model, or one of these variations) becomes the new current model.
   (c) Repeat Step above until no lower AICc can be found.

After auto.arima()

- Check parameter significance. If parameters are not significant, then remove from the model.
- For value of d selected by `auto.arima()`, take the difference by hand, and use `Stationarity.tests()` to see what other tests say.
- You can change stationarity test used by `auto.arima()`. Default is KPSS test. Type `?auto.arima` to see syntax of how to change default.

3.2 Overdifferencing

[ToC]

Suppose $\delta X_t = Y_t$

$$Y_t = e_t$$

Take $\bigtriangledown$ again, then you get

$$\bigtriangledown Y_t = e_t - e_{t-1}$$

Now we got MA(1) with $\theta_1 = 1$. That's not invertible.

Overdifferencing

Suppose

$$\bigtriangledown X_t \quad = \quad Y_t$$

$$Y_t = e_t - \theta_1 e_{t-1}$$

If you take $\bigtriangledown$ again,

$$\begin{aligned} \bigtriangledown Y_t &= (e_t - \theta_1 e_{t-1}) - (e_{t-1} - \theta_1 e_{t-2}) \\ &= (e_t - (1 + \theta_1)e_{t-1} + \theta_1 e_{t-2}). \end{aligned}$$

So $\bigtriangledown Y_t$ is MA(2) with

$$\Theta(z) = 1 - (1+\theta_1)z + \theta_1 z^2$$

Root is

$$\begin{aligned}
\frac{-b \pm \sqrt{b^2 \pm 4ac}}{2a} &= \frac{(1+\theta_1) \pm \sqrt{(1+\theta_1)^2 - 4\theta_1}}{2\theta_1} \\
&= \frac{(1+\theta_1) \pm \sqrt{1 - 2\theta_1 + \theta_1^2}}{2\theta_1} \\
&= \frac{(1+\theta_1) \pm \sqrt{(1-\theta_1)^2}}{2\theta_1} \\
&= \frac{2}{2\theta_1} \quad \text{or} \quad \frac{2\theta_1}{2\theta_1}
\end{aligned}$$

That's a unit root!

1. Testing Unit-Root in MA(q) polynomials $\rightarrow$ not fully resolved.

2. Test Unit-Root in MA(1) $\rightarrow$ see if $\hat{\theta}_1$ is significantly different from 1.

ARMA with same polynomials

Suppose you have ARMA(1,1) with

$$Y_t - .5Y_{t-1} = e_t - .5e_{t-1}$$

Then this is

$$\begin{aligned}(1 - .5B)Y_t &= (1 - .5B)e_t \\ Y_t &= e_t\end{aligned}$$

So Y_t is just a white noise.

Example: Monthly Oil Price

Cryer p88

```
#--- Load package TSA ---
  acf.orig <- acf    # keep the default acf()
  library(TSA)
  acf <- acf.orig

  library(forecast)
  source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

  data(oil.price)
  X <- oil.price
  plot(X, ylab='Price per Barrel',type='o')


#--- Take difference of lag 1 ---
  plot(diff(X),ylab='Change in Log(Price)',type='l')


#--- Take log difference of lag 1 ---
  plot(diff(log(X)),ylab='Change in Log(Price)',type='l')

  Stationarity.tests(diff(log(X)))
```

```
#--- Look for best ARIMA  (this gives us seasonal component) ---
  Fit1 <- auto.arima(log(X))
  Fit1


#--- Look for best ARIMA  (supress seasonal component) ---
  Fit2 <- auto.arima(log(X), seasonal=FALSE)
  Fit2
  Randomness.tests(Fit2$residuals)

  plot(forecast(Fit2))
```

Forcing some parameter to be 0

```
Arima(diff(log(X)),  c(2,0,4), fixed=c(NA,NA,NA,0,0,NA,NA))
```

3.3 Example: LArain data

```
library(forecast)
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

acf.orig <- acf;  library(TSA);  acf <- acf.orig

data(larain)

plot(larain, type="o")

Arima(larain, order=c(1,0,1))
```

Notes

- When orders of AR and MA mathces, watch out for equal value of parameters. It may indicate WN.
- If MA(1) gives you 1, it indicates the over-differencing.

3.4 Example: Color Data

```
library(forecast)
source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")
acf.orig <- acf
library(TSA)
acf <- acf.orig

data(color)
plot(color, type="o")

Fit1 <- auto.arima(color)

Stationarity.tests(color)
```

Notes

- By default, `auto.arima()` only uses KPSS to decide which d to use. Use $Staionarity.tests()$ and plots to make more informed desision.

3.5 Example: Global Temp Data

```
  source("http://gozips.uakron.edu/~nmimoto/477/TS_R-90.txt")

  D  <- read.csv("http://gozips.uakron.edu/~nmimoto/pages/datasets/gtemp.txt")
  D1 <- ts(D, start=c(1880), freq=1)
  plot(D1, type="o")


#--- fit linear trend inside auto.arima() ---
  Fit1 <- auto.arima(D1, xreg=time(D1))
  Fit1

  Randomness.tests(Fit1$resid)

  plot(forecasst(Fit1, xreg=2010:2019, h=10))


#--- fit linear trend by hand ---

  Reg1 <- lm(D1~time(D1));  Reg1

  Fit2 <- auto.arima(Reg1$resid);  Fit2

  Randomness.tests(Fit2$resid)

  plot(Reg1$resid, type="o")
```

```
  Stationarity.tests(Fit1$resid)


#--- Direct ARIMA fit ---
  Fit3 <- auto.arima(D1)
  Stationarity.tests(diff(D1))
  Randomness.tests(Fit3$resid)
```

Notes

- Parameter in AR(1) should be checked for being ± 1, which indicates non-stationarity.

4

ARIMA Forecasting

[ToC]

4.1 ARIMA forecasting

[ToC]

- Suppose $\bigtriangledown Y_t = X_t \sim$ARMA$(p, q)$.

- Since we know how to forecast ARMA, we know how to get

$$\hat{X}_n(h) = a_1 X_n + \cdots + a_n X_n$$

- How can we calculate $\hat{Y}_n(h)$ so that MSE,

$$E\left[\left(Y_{n+h} - \hat{Y}_n(h)\right)^2\right]$$

  is minimized?

- We have two vectors,

$$(1, Y_0, Y_1, \ldots, Y_n), \qquad (1, Y_0, X_1, \ldots, X_n)$$

- They actually span the same vector space, because

$$\begin{bmatrix} Y_1 - Y_0 = X_1 \\ Y_2 - Y_1 = X_2 \\ Y_3 - Y_2 = X_3 \\ \vdots \\ Y_n - Y_{n-1} = X_n \end{bmatrix} \iff \begin{bmatrix} Y_1 = X_1 + Y_0 \\ Y_2 = X_2 + Y_1 \\ Y_3 = X_3 + Y_2 \\ \vdots \\ Y_n = X_n + Y_{n-1} \end{bmatrix}$$

- Because $Y_{n+1} - Y_n = X_{n+1}$, we can rewrite MSE in Y_t as

$$E\left[\left(Y_{n+1} - \hat{Y}_n(1)\right)^2\right] = E\left[\left((X_{n+1} + Y_n) - \hat{Y}_n(1)\right)^2\right].$$

- Also, note that we can write

$$\begin{aligned} \hat{Y}_n(1) &= a_0' + a_1' Y_n + \cdots + a_n' Y_1 \\ &= a_0 + a_1 X_n + \cdots + a_n X_1 + b_0 Y_0 \\ &= \hat{X}_n(1) + b_0 Y_0. \end{aligned}$$

- That means

$$MSE = E\left[\left((X_{n+1} + Y_n) - (\hat{X}_n(1) + b_0 Y_0)\right)^2\right].$$

- If Y_0 is uncorrelated with $(X_1, X_2, \ldots, X_n)$, $b_0 = 0$, it drops out, and we get, given the observations $Y_1, \ldots, Y_n$,

$$
\begin{aligned}
MSE &= E\left[\left(X_{n+1} - \hat{X}_n(1) + Y_n\right)^2\right] \\
&= E\left[\left(X_{n+1} - \hat{X}_n(1)\right)^2 + 2\,Y_n\left(X_{n+1} - \hat{X}_n(1)\right) + Y_n^2\right] \\
&= E\left[\left(X_{n+1} - \hat{X}_n(1)\right)^2\right] + 2\,Y_n E\left[X_{n+1} - \hat{X}_n(1)\right] + Y_n^2 \\
&= E\left[\left(X_{n+1} - \hat{X}_n(1)\right)^2\right] + 0 + c.
\end{aligned}
$$

- We know that $\hat{Y}(1)$ is the minimizer of this.

- Therefore,

$$\hat{Y}_n(1) = Y_n + \hat{X}_n(1)$$

- Similarly,

$$\hat{Y}_n(h) = Y_n + \hat{X}_n(h)$$

- So forecast ARMA(p,q) as usual, then add it to the last observation X_n.