**Drift term**

Random walk model

A random walk model is simply an ARIMA(0,1,0):

(1 - B)xt = wt

⇒ xt = xt-1 + wt

where wt ~ ind. (0,)

The name comes about through xt being xt-1 plus a random “movement”.

A drift term δ can be added to the model,

xt = δ + xt-1 + wt

This can be rewritten then as a cumulative sum,

xt = δ + (δ + xt-2 + wt-1) + wt = 2δ + xt-2 + wt-1 + wt

 = 2δ + (δ + xt-3 + wt-2) + wt-1 + wt = 3δ + xt-3 + wt-2 +

 wt-1 + wt

xt = δt + 

One can see that the drift term creates a non-stochastic trend in the series (i.e., allows xt to drift away from 0). Notice the random walk model with a drift term is not stationary because the mean is a function of time.

Estimating the drift term

When arima()is used with differencing, the include.meanargument is set to FALSE. This is because differencing removes the trend and allows for the differenced series to have a mean of 0. Note that even if you say include.mean = TRUE, there will be no “mean” estimated. Here’s what R’s arima() help says for the include.mean argument,

|  |  |
| --- | --- |
| include.mean | Should the ARIMA model include a mean term? The default is TRUE for undifferenced series, FALSE for differenced ones (where a mean would not affect the fit nor predictions). |

As shown in the random walk model example, there may be times where you do want to estimate a drift term, like δ, still. This leads to an ARIMA model with the drift term,

(1-ϕ1B-ϕ2B2-…-ϕpBp)(1-B)dxt
 = δ + (1+θ1B+θ2B2+…+θqBq)wt

⇔ ϕ(B)(1-B)dxt = δ + θ(B)wt

Most time series textbooks will not include the constant term in the model when there is differencing. For example, Wei’s textbook says (referring to δ as θ0),

We assume that θ0 = 0, unless it is clear from the data or the nature of the problem that a deterministic [non-stochastic] trend is really needed.

This is why R does not include the possible estimation of δ in the arima()function.

If you did want to include δ, there are a few ways to estimate it.

1. Use xreg = 1:length(x) in arima() where x contains the data used with the arima().
2. Use the sarima() and sarima.for() functions in the astsa package for fitting the model and forecasting, respectively. The sarima()name comes from an extension of the ARIMA model that will be discussed later in our course to allow for seasonality.
3. Do all differencing needed BEFORE invoking arima(). Use d = 0 in arima() and include.mean = TRUE*.*

Below are examples of their implementations. Shumway and Stoffer also include an example when finding the best model for their GNP data.

Example: ARIMA(1,1,1) with ϕ1 = 0.7, θ1 = 0.4,  = 9, n = 200 (arima111\_sim.R, arima111.csv)

Of course, δ should be 0 for this data set since the data were simulated directly from an ARIMA(1,1,1). However, we can still investigate what would happen if δ was estimated.

First, the original model’s fit has been reproduced below.

> arima111 <- read.csv(file = "arima111.csv")

> x <- arima111$x

> mod.fit <- arima(x = x, order = c(1, 1, 1))

> mod.fit

Call:

arima(x = x, order = c(1, 1, 1))

Coefficients:

 ar1 ma1

 0.6720 0.4681

s.e. 0.0637 0.0904

sigma^2 estimated as 9.558: log likelihood = -507.68, aic = 1021.36

> #Forecasts 5 time periods into the future

> fore.mod <- predict(object = mod.fit, n.ahead = 5, se.fit

 = TRUE)

> fore.mod

$pred

Time Series:

Start = 201

End = 205

Frequency = 1

[1] -486.3614 -484.9361 -483.9784 -483.3348 -482.9023

$se

Time Series:

Start = 201

End = 205

Frequency = 1

[1] 3.091673 7.303206 11.578890 15.682551 19.534208

The estimated model is

(1 - 0.6720B)(1 - B)xt = (1 + 0.4681B)wt with  = 9.56

Next, the models are fit including an estimate of δ.

1. Using xreg = 1:length(x) in arima() where x contains the data

> mod.fit2 <- arima(x = x, order = c(1, 1, 1), xreg =

 1:length(x))

> mod.fit2

Call:

arima(x = x, order = c(1, 1, 1), xreg = 1:length(x))

Coefficients:

 ar1 ma1 1:length(x)

 0.6382 0.4826 -1.6847

s.e. 0.0671 0.0894 0.8815

sigma^2 estimated as 9.404: log likelihood = -506.03, aic = 1020.05

> fore.mod2 <- predict(object = mod.fit2, n.ahead = 5,

 se.fit = TRUE, newxreg = (length(x)+1):(length(x)+5))

> fore.mod2

$pred

Time Series:

Start = 201

End = 205

Frequency = 1

[1] -486.7318 -486.2243 -486.5100 -487.3020 -488.4169

$se

Time Series:

Start = 201

End = 205

Frequency = 1

[1] 3.066606 7.190241 11.284195 15.141661 18.709358

> mod.fit2$coef[3]/sqrt(mod.fit2$var.coef[3,3])

1:length(x)

 -1.911036

Note the newxreg option used in predict()and the values that I put into it. The estimated model is

(1 − 0.6382B)(1 − B)xt = -1.6847 + (1 + 0.4826B)wt with  = 9.404

A Wald test was performed for Ho:δ = 0 vs. Ha:δ ≠ 0, and, surprisingly, δ is marginally significant. This is an example of a type I error!

1. The astsa package

> library(package = astsa)

> mod.fit.SS <- sarima(xdata = x, p = 1, d = 1, q = 1)

> mod.fit.SS

$fit

Call:

arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S), xreg = constant, transform.pars = trans, fixed = fixed, optim.control = list(trace = trc,

REPORT = 1, reltol = tol))

Coefficients:

 ar1 ma1 constant

 0.6382 0.4826 -1.6847

s.e. 0.0671 0.0894 0.8815

sigma^2 estimated as 9.404: log likelihood = -506.03, aic = 1020.05

$degrees\_of\_freedom

[1] 196

$ttable

 Estimate SE t.value p.value

ar1 0.6382 0.0671 9.5105 0.0000

ma1 0.4826 0.0894 5.4008 0.0000

constant -1.6847 0.8815 -1.9110 0.0575

$AIC

[1] 5.12588

$AICc

[1] 5.126498

$BIC

[1] 5.192077



> names(mod.fit.SS)

[1] "fit" "degrees\_of\_freedom" "ttable"

[4] "AIC" "AICc" "BIC"

> mod.fit.SS$fit$coef

 ar1 ma1 constant

 0.6381503 0.4825974 -1.6846517

> mod.fit.SS$fit$var.coef

 ar1 ma1 constant

ar1 0.0045023544 -3.544068e-03 4.935338e-04

ma1 -0.0035440684 7.984740e-03 7.767717e-05

constant 0.0004935338 7.767717e-05 7.771103e-01

The plot that includes information about the residuals will be discussed later in the course. The estimated model is

(1 − 0.6382B)(1 − B)xt = -1.6847 + (1 + 0.4826B)wt with  = 9.404

The forecasts are shown below.

> save.for <- sarima.for(xdata = x, n.ahead = 5, p = 1, d

 = 1, q = 1)

> save.for

$pred

Time Series:

Start = 201

End = 205

Frequency = 1

[1] -486.7318 -486.2243 -486.5100 -487.3020 -488.4169

$se

Time Series:

Start = 201

End = 205

Frequency = 1

[1] 3.066606 7.190241 11.284195 15.141661 18.709358



The envelope surrounding the forecasts represent ±1 and ±2 multiplied by ****

1. Perform the differencing BEFORE invoking arima(), Use d = 0 in arima(), and include.mean = TRUE.

See the program for partial implementation. Methods #1 and #2 are easier, so these will be used in this course.