**Basic models**

Stochastic process – Collection of random variables {Xt} indexed by t

Time series – Collection of random variables indexed according to the order they are obtained in time.

Let Xt be the random variable at time t

Then

X1 = random variable at time 1

X2 = random variable at time 2

A realization of the stochastic process is the observed values

The observed values are denoted by x1, x2, … .

Notice that lowercase letters are used to denote the observed value of the random variables.

NOTE: It is common to just use lowercase values for both the random variables and the realization at this level of a statistics course. The reason is because it should be clear from the context whether xt represents the random variable or the realization. Most time series textbooks do the same.

Example: White noise (white\_noise.R)

The term “white noise” comes from engineering.

The simplest kind of time series is a collection of independent and identically distributed random variables with mean 0 and constant variance.

This can be written as wt ~ independent (0,) for t = 1, …, n.

Very often, the probability distribution is assumed to be a normal probability distribution.

This can be written as wt ~ independent N(0,) for t = 1, …, n.

What does this mean?

* Each wt has a normal distribution with mean of 0 and a constant variance.
* w1, w2, …, wn are independent of each other

Given this set up, answer the following questions:

* What patterns are there over time (t)?
* How can the dependence among observations be used to help model the data?
* How can we “simulate” a white noise process using R?

Because each random variable is independent, we can simulate 100 observations from a normal distribution. I am going to use  = 1 here.

> set.seed(8128)

> w <- rnorm(n = 100, mean = 0, sd = 1)

> head(w)

[1] -0.10528941 0.25548490 0.82065388 0.04070997

[5] -0.66722880 -1.54502793

> dev.new(width = 6, height = 6, pointsize = 10)

> plot(x = w, ylab = expression(w[t]), xlab = "t", type

= "o", col = "red", main = expression(paste("White

noise where ", w[t], " ~ ind. N(0, 1)")),

panel.first = grid(col = "gray", lty = "dotted"))



> #Advantage of second plot is separate control over color

of points

> plot(x = w, ylab = expression(w[t]), xlab = "t", type =

"l", col = "red", main = expression(paste("White

noise where ", w[t], " ~ ind. N(0, 1)")),

panel.first = grid(col = "gray", lty = "dotted"))

> points(x = w, pch = 20, col = "blue")



Given this data set, answer the following questions:

* What patterns are there over time (t)?
* How can the dependence among observations be used to help model the data?

Suppose another white noise process is simulated. Below is a plot overlaying the two time series.

> set.seed(1298)

> w.new <- rnorm(n = 100, mean = 0, sd = 1)

> head(w.new)

[1] 1.08820292 -1.46217413 -1.10887422 0.55156914

[5] 0.70582813 0.05079594

> plot(x = w, ylab = expression(w[t]), xlab = "t", type =

"l", col = "red", main = expression(paste("White

noise where ", w[t], " ~ ind. N(0, 1)")), panel.first

= grid(col = "gray", lty = "dotted"), c(min(w.new,

w), max(w.new, w)))

> points(x = w, pch = 20, col = "blue")

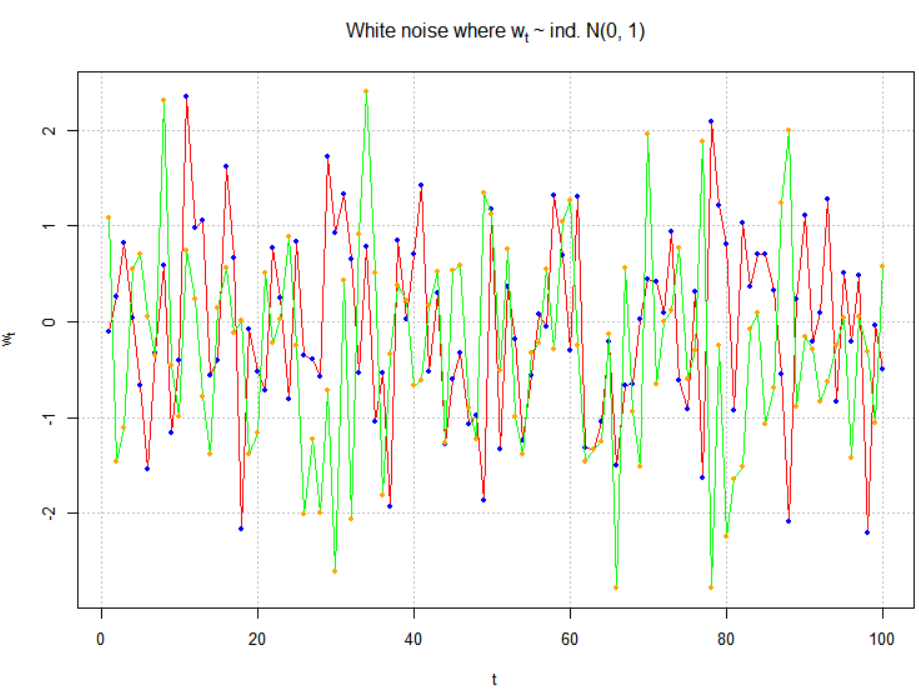
> lines(x = w.new, col = "green")

> points(x = w.new, pch = 20,col = "orange")

> legend(x = locator(1),legend=c("Time series 1", "Time

series 2"), lty=c(1,1), col=c("red", "green"),

bty="n")



We could also plot the two time series separately.

> par(mfrow = c(2,1))

> plot(x = w, ylab = expression(w[t]), xlab = "t", type =

"l", col = "red", main = expression(paste("White

noise where ", w[t], "~N(0, 1)")), panel.first =

grid(col = "gray", lty = "dotted"))

> points(x = w, pch = 20, col = "blue")

> plot(x = w.new, ylab = expression(w.new[t]), xlab =

"t", type = "l", col = "green", main =

expression(paste("White noise where ", w[t], " ~ ind.

N(0, 1)")), panel.first=grid(col = "gray", lty =

"dotted"))

> points(x = w.new, pch = 20, col = "orange")



Example: Moving average of white noise (moving\_average.R)

The previous time series had no correlation between the observations. One way to induce correlation is to create a “moving average” of the observations. This will have an effect of “smoothing” the series.

Let mt = (wt + wt-1 + wt-2)/3. This can be done in R using the following code:

> set.seed(8128)

> w <- rnorm(n = 100, mean = 0, sd = 1)

> head(w)

[1] -0.10528941 0.25548490 0.82065388 0.04070997

[5] -0.66722880 -1.54502793

> m <- filter(x = w, filter = rep(x = 1/3, times = 3),

method = "convolution", sides = 1)

> head(m)

[1] NA NA 0.32361646 0.37228292

[5] 0.06471168 -0.72384892

> tail(m)

[1] 0.3158762 -0.1803096 0.2598066 -0.6450531 -0.5879723

[6] -0.9120182

> (w[1]+w[2]+w[3])/3

[1] 0.3236165

> (w[98]+w[99]+w[100])/3

[1] -0.9120182

> plot(x = w, ylab = expression(paste(m[t], " or ", w[t])),

xlab = "t", type = "l", col = "red", panel.first =

grid(col = "gray", lty = "dotted"), lty = "dotted")

> points(x = w, pch = 20, col = "blue")

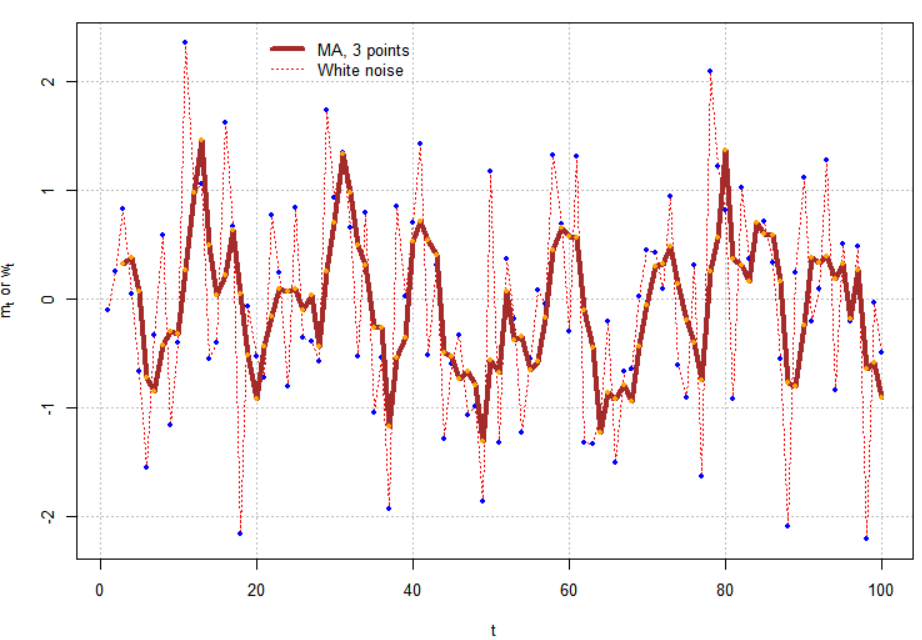
> lines(x = m, col = "brown", lty = "solid", lwd = 4)

> points(x = m, pch = 20, col = "orange")

> legend(x = locator(1), legend = c("MA, 3 points", "White

noise"), lty = c("solid", "dotted"), col=c("brown",

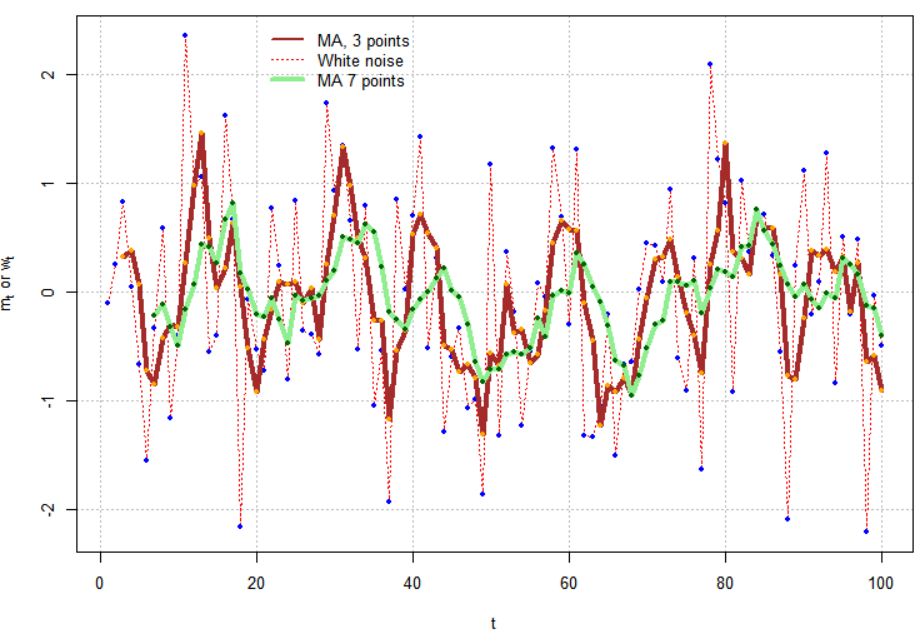
"red"), lwd = c(4,1), bty = "n")



Given these observed values of mt, answer the following questions:

* What patterns are there over time (t)?
* How can the correlation between observations be used to help model the data?

The plot below shows a 7-point moving average (see program for code).



Example: Autoregression (ar1.R)

An “autoregression” model uses past observations to predict future observations in a regression model.

Suppose the autoregression model is

xt = 0.7xt-1 + wt where wt ~ independent N(0,1) for t = 1, …, n.

Notice how similar this is to a regression model! Because there is one past period on the right hand side, this is often denoted as an AR(1) model where AR stands for “autoregressive”.

Therefore,

x2 = 0.7x1 + w2

x3 = 0.7x2 + w3

Obviously, there will be a correlation between the random variables.

Below is one way in R to simulate observations from this model.

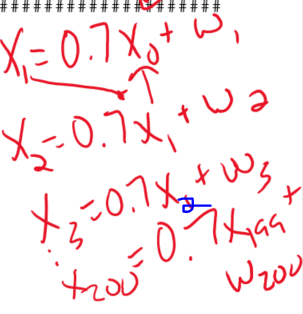
> set.seed(6381) #Different seed from white\_noise.R

> w <- rnorm(n = 200, mean = 0, sd = 1)

> head(w)

[1] 0.06737166 -0.68095839 0.78930605 0.60049855   
[5] -1.21297680 -1.14082872

During the video, the following was written in red. I have included in blue a correction for a small typo.



> #######################################################

> # autoregression

> #Simple way to simulate AR(1) data

> x <- numeric(length = 200)

> x.1 <- 0

> for(i in 1:length(x)) {

x[i] <- 0.7\*x.1 + w[i]

x.1 <- x[i]

}

> head(data.frame(x, w))

x w

1 0.06737166 0.06737166

2 -0.63379823 -0.68095839

3 0.34564730 0.78930605

4 0.84245166 0.60049855

5 -0.62326064 -1.21297680

6 -1.57711117 -1.14082872

> #Do not use first 100

> x <- x[101:200]

> plot(x = x, ylab = expression(x[t]), xlab = "t", type =

"l", col = c("red"), lwd = 1 , main =

expression(paste("AR(1): ", x[t] == 0.7\*x[t-1] +

w[t])), panel.first = grid(col = "gray", lty =

"dotted"))

> points(x = x, pch = 20, col = "blue")



Notes:

* Notice the syntax of the for() loop.
* See the first 6 rows of x and w right after the loop. Make sure you understand how the data was simulated!

The 1st value of x is

0.06737166 = 0.7×0 + 0.06737166

The 2nd value of x is

-0.6337982 = 0.7×(0.06737166) – 0.68095839

The 3rd value of x is

0.3456473 = 0.7×(-0.6337982) + 0.78930605

* Why are the first 100 observations discarded?
* If you have taken a course where AR() structures of a covariance matrix are discussed, what do you think the approximate correlation between xt and xt-1 is?

Here is an easier way to simulate observations from an AR(1). Note that this uses an Autoregressive Integrated Moving Average (ARIMA) structure that we will discuss later in the course. In this case, I use  = 10.

> set.seed(7181)

> x <- arima.sim(model = list(ar = c(0.7)), n = 100,

rand.gen = rnorm, sd = 10)

> plot(x = x, ylab = expression(x[t]), xlab = "t", type =

"l", col = "red", lwd = 1 ,main =

expression(paste("AR(1): ", x[t] == 0.7\*x[t-1] +

w[t])), panel.first=grid(col = "gray", lty =

"dotted"))

> points(x = x, pch = 20, col = "blue")



More notes:

* Both the moving average and autoregressive models will be discussed extensively later in the course.
* One can match up simulated data plots to actual data plots. This is done to develop an equation that reasonably mimics or “models” real data.