Time Series

John Tipton

August 24, 2015

```r
## Load R packages and define helper functions
library(ggplot2, quietly = TRUE)
library(reshape2, quietly = TRUE)

## Function to plot multiple ggplots on the same image
multiplot <- function(..., plotlist=NULL, cols) {
  require(grid)

    # Make a list from the ... arguments and plotlist
    plots <- c(list(...), plotlist)

    numPlots = length(plots)

    # Make the panel
    plotCols = cols                       # Number of columns of plots
    plotRows = ceiling(numPlots/plotCols) # Number of rows needed, calculated from # of cols

    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(plotRows, plotCols)))
    vplayout <- function(x, y)
        viewport(layout.pos.row = x, layout.pos.col = y)

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
        curRow = ceiling(i/plotCols)
        curCol = (i-1) %% plotCols + 1
        print(plots[[i]], vp = vplayout(curRow, curCol ))
    }

}
```

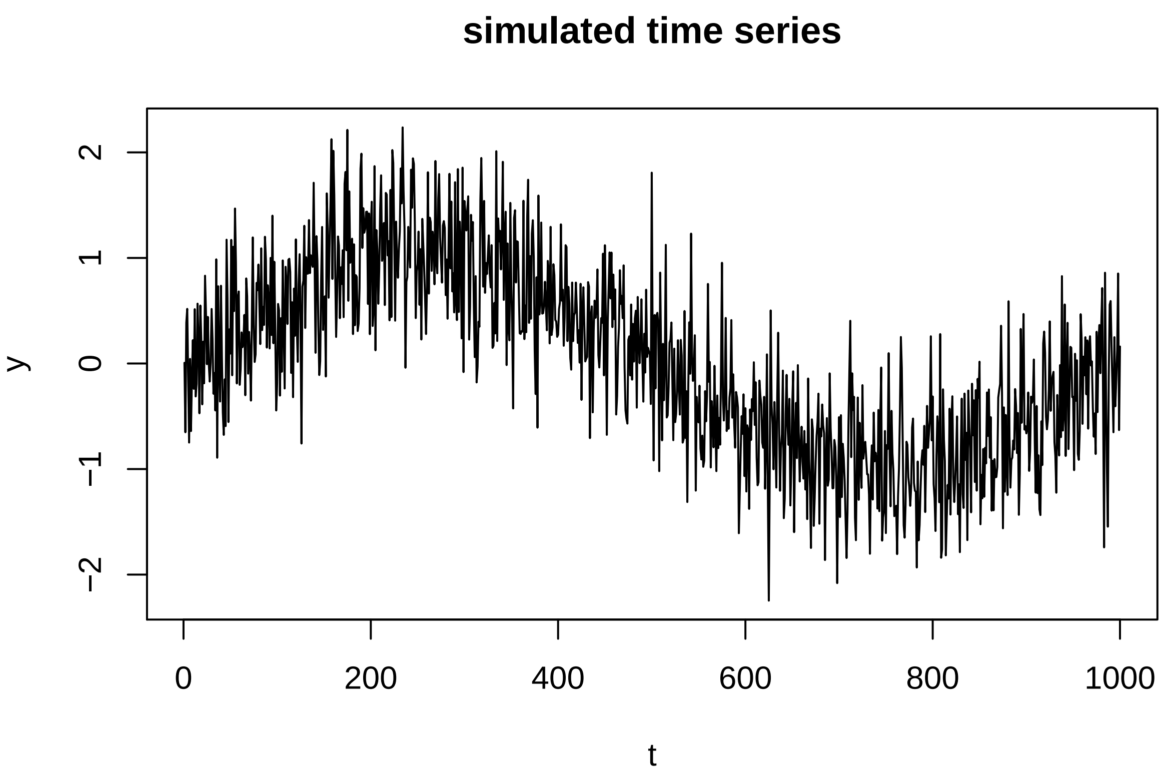
First, we start with the canonical difference equation for the time series autoregressive model of order 1 (AR(1))

$$y_t = \mu_t + \phi y_{t-1} + \epsilon_t \tag{1}$$

where the time series observations for times $t = 1, \ldots, T$ are given by the vector $\mathbf{y} = (y_1, \ldots, y_T)$ where y_t is the observation of the time serise at time t. The vector $\boldsymbol{\mu}$ is the temporal mean with μ_t representing the mean of the time series at time t. Often the mean is a trend or seasonal component like in the example below. The autoregressive parameter ϕ controls the strength of autocorrelation in the time series with $-1 < \phi < 1$ and the random error $\epsilon_t \sim N(0, \sigma^2)$ is independent for different times (i.e. the covariance $\mathrm{Cov}(\epsilon_t, \epsilon_{t+k}) = 0$ for $k \neq 0$).

Lets simulate some data here

```
N <- 1                                  ## pick the number of time series
t <- 1000                               ## pick a time series length
mu <- sin(2 * pi * (1:t)/t)
s <- 0.5                                ## pick standard deviation
phi <- 0.75                             ## pick autocorrelation parameter

##
## function to simulate time series
##

simTimeSeries <- function(t, N, mu, s, phi){
  y <- matrix(mu[1], t, N)                                   ## initialize container
  epsilon <- rnorm(N*(t-1), 0, s)                            ## independent random error
  y[2:t, ] <- mu[2:t] + phi * y[1:(t-1), ] + epsilon  ## autoregressive model
  return(y)
}

y <- simTimeSeries(t, N, mu, s, phi)
matplot(y, type="l", main="simulated time series", xlab="t")
```



The expected value $\mathrm{E}(y_t)$ of the time series at time t is

$$
\begin{aligned}
\mathrm{E}(y_t) &= \mathrm{E}(\phi y_{t-1}) + \mathrm{E}(\epsilon_t) \\
&= \phi \mathrm{E}(y_{t-1}) + 0
\end{aligned}
$$

where, assuming a constant mean μ accross time we have

$$
\begin{aligned}
\mathrm{E}(y_t) &= \phi \mathrm{E}(y_{t-1}) \\
\rightarrow \mathrm{E}(y_t)(1-\phi) &= 0 \\
\rightarrow \mathrm{E}(y_t) &= 0
\end{aligned}
$$

and assuming constant variance through time, the variance is

$$
\begin{aligned}
\mathrm{Var}(y_t) &= \mathrm{Var}(\phi y_{t-1} + \epsilon_t) \\
&= \mathrm{Var}(\phi y_{t-1}) + \mathrm{Var}(\epsilon_t) + 2\mathrm{Cov}(\phi y_{t-1}, \epsilon_t) \\
&= \phi^2 \mathrm{Var}(y_{t-1}) + \sigma^2 + 0
\end{aligned}
$$

Then using our modeling assumption $\mathrm{Var}(y_t) = \mathrm{Var}(y_{t-1})$,

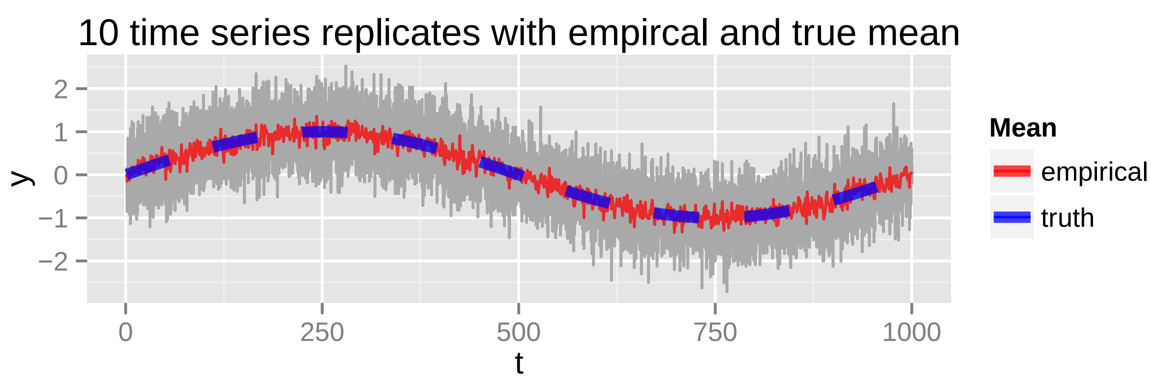
$$
\mathrm{Var}(y_t) - \phi^2 \mathrm{Var}(y_t) = \sigma^2
$$

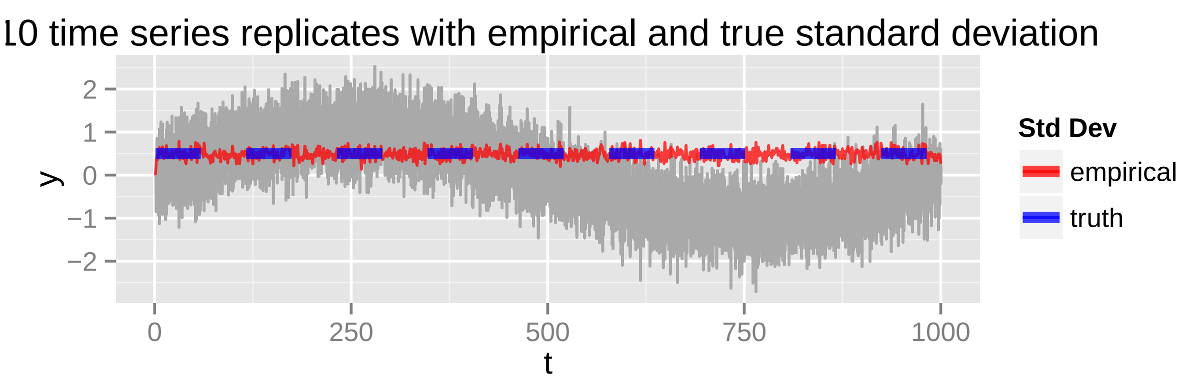
gives the solution $\mathrm{Var}(y_t) = \frac{\sigma^2}{1-\phi^2}$.

```
N <- 10                                 ## replicates N
y <- simTimeSeries(t, N, mu, s, phi)  ## simulate time series
## legend() add in legend for mean and y label
## Notice that the y axis is shrunk to 0

##

mean_y <- apply(y, 1, mean)    ##  calculate the mean
var_y <-apply(y, 1, var)       ##  calculate the variance
sd_y <- apply(y, 1, sd)        ##  calculate the standard deviation

time_data <- data.frame(y=y, t=1:t)
melt_time <- melt(time_data, id="t")
summary_data <- data.frame(mean_y=mean_y, var_y=var_y, sd_y=sd_y, mu=mu,
                        s=s, t=1:t)


## plot time series with mean and variance


plot_mean <- ggplot(data = melt_time, aes(y=value, x=t)) +
  geom_line(alpha=1, colour="darkgrey") +
  geom_line(data=summary_data, aes(y=mean_y, x=t, colour="empirical"),
            alpha=0.75) +
  geom_line(data=summary_data, aes(y=mu, x=t, colour="truth"), alpha=0.75,
            lty=2, lwd=2) +
```

```r
  scale_colour_manual("Mean", labels=c("empirical", "truth"),
                      values=c("empirical"="red","truth"="blue")) +
  scale_y_continuous("y") + scale_x_continuous("t") +
  ggtitle(paste(min(N, 10),
                "time series replicates with empircal and true mean"))

plot_sd <- ggplot(data = melt_time, aes(y=value, x=t)) +
  geom_line(alpha=1, colour="darkgrey") +
  geom_line(data=summary_data, aes(y=sd_y, x=t, colour="empirical"),
            alpha=0.75) +
  geom_line(data=summary_data, aes(y=s, x=t, colour="truth"), alpha=0.75,
            lty=2, lwd=2) +
  scale_colour_manual("Std Dev", labels=c("empirical", "truth"),
                      values=c("empirical"="red","truth"="blue")) +
  scale_y_continuous("y") + scale_x_continuous("t") +
  ggtitle(paste(min(N, 10),
                "time series replicates with empirical and true standard deviation"))


multiplot(plot_mean, plot_sd, cols=1)
```

```
## Loading required package: grid
```






```r
#   scale_fill_discrete(breaks=c(expression(hat(mu)), expression(mu)))
# matplot(y[, 1:min(N, 10)], type="l", col=adjustcolor("grey", alpha.f = 0.75),
#         main=paste(min(N, 10), " simulated time series", sep="") , xlab="t", ylab="y")
# matplot(mean_y, type="l", col=adjustcolor("red", alpha.f=0.75), lwd = 2, add = TRUE)
```

```
# lines(1:t, mu, col = adjustcolor("blue", alpha.f=0.75), lty=2, lwd = 3)
# legend("topright", c(expression(hat(mu)), expression(hat(mu))), lty=c(1,2),
#        col=c(adjustcolor("red", alpha.f=0.75),
#              adjustcolor("blue", alpha.f=0.75)), lwd=c(3,3), cex=0.5)
# matplot(y[, 1:min(N, 10)], type="l", col=adjustcolor("grey", alpha.f = 0.75),
#         main=paste(min(N, 10), " simulated time series", sep="") , xlab="t", ylab="y")
# matplot(sd_y, type = 'l', add = TRUE, col = 'red', lwd = 3)
# lines(1:t, rep(s, t), col =adjustcolor("blue", alpha.f=0.75), lty=2, lwd=3)
# legend("topright", c(expression(hat(sigma)), expression(hat(sigma))), lty=c(1,2),
#        col=c(adjustcolor("red", alpha.f=0.75),
#              adjustcolor("blue", alpha.f=0.75)), lwd=c(3,3), cex=0.5)
```

The covariance between observations $\mathrm{Cov}(y_t, y_{t+k})$ at times k lags apart (assuming without loss of generality that $k > 0$) is

$$
\begin{aligned}
\mathrm{Cov}(y_t, y_{t+k}) &= \mathrm{E}(y_t y_{t+k}) - \mathrm{E}(y_t)\mathrm{E}(y_{t+k}) \\
&= \mathrm{E}(y_t(\phi y_{t+k-1} + \epsilon_{t+k})) - 0 \\
&= \mathrm{E}(\phi y_t y_{t+k-1}) + \mathrm{E}(y_t \epsilon_{t+k}) \\
&= \mathrm{E}(\phi y_t y_{t+k-1}) + \mathrm{E}(y_t)\mathrm{E}(\epsilon_{t+k}) \\
&= \mathrm{E}(\phi y_t y_{t+k-1}) + 0 \\
&= \mathrm{E}(y_t(\phi y_{t+k-2} + \epsilon_{t+k-1})) \\
&= \vdots \\
&= \phi^k \mathrm{E}(y_t^2) \\
&= \phi^k \frac{\sigma^2}{1-\phi^2}.
\end{aligned}
$$

Thus, knowing the mean, variance, and covariance at each time t and each lag k, we can write the autoregressive model (**??**) as

$$
\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\eta} \tag{2}
$$

where $\mu = (0, \ldots, 0)$ and η $\mathrm{N}(\mathbf{0}, \mathbf{\Sigma})$ where

$$
\mathbf{\Sigma} = \frac{\sigma^2}{1-\phi^2} \left(\begin{pmatrix} 1 & \phi & \phi^2 & \phi^3 & \cdots & \phi^{T-1} \\ \phi & 1 & \phi & \phi^2 & \cdots & \phi^{T-2} \\ \phi^2 & \phi & 1 & \phi & \cdots & \phi^{T-3} \\ \phi^3 & \phi^2 & \phi & 1 & \cdots & \phi^{T-4} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi^{T-1} & \phi^{T-2} & \phi^{T-3} & \phi^{T-4} & \cdots & 1 \end{pmatrix} \right)
$$

Typically, the distributions of interest in a time series model are the forecast distribution (used for prediction) and the smoothing distribution (used for estimation of parameters). The forecast distribution at time $\tau + 1$ consists of knowledge of all of the observations of the time series up to the time τ $y_{1:\tau} = (y_1, \ldots, y_\tau)$ given by

$$[y_{\tau+1}|y_{1:\tau}] = [y_{\tau+1}|y_\tau]$$

by the Markov assumption in the autoregressive model. Then, the one step ahead expected forecast is

$$\begin{aligned}
\mathrm{E}(y_{\tau+1}|y_{1:\tau}) &= \mathrm{E}(y_{\tau+1}|y_\tau) \\
&= \mathrm{E}(\phi y_\tau + \epsilon_{\tau+1}|y_\tau) \\
&= \mathrm{E}(\phi y_\tau|y_\tau) + \mathrm{E}(\epsilon_{\tau+1}|y_\tau) \\
&\phi y_\tau + 0.
\end{aligned}$$

The k step ahead expected forecast is calculated by using a recursive formula of the equation above where $\mathrm{E}(y_{\tau+k}|y_{1:\tau}) = \phi^k y_\tau$.

Likewise, the one step ahead forecast variance is

$$\begin{aligned}
\mathrm{Var}(y_{\tau+1}|y_{1:\tau}) &= \mathrm{Var}(y_{\tau+1}|y_\tau) \\
&= \mathrm{Var}(\phi y_\tau + \epsilon_{\tau+1}|y_\tau) \\
&= \phi^2 \mathrm{Var}(y_\tau|y_\tau) + 2\mathrm{Cov}(\phi y_\tau, \epsilon_{\tau+1}|y_\tau) + \mathrm{Var}(\epsilon_{\tau+1}|y_\tau) \\
&= 0 + 0 + \sigma^2.
\end{aligned}$$

The k step ahead forecast variance can also be calcuated recursively giving $\mathrm{Var}(y_{\tau+k}|y_{1:\tau-1}) = \sum_{i=1}^{k} \phi^{2(i-1)}\sigma^2$.