Time Series

John Tipton

August 24, 2015

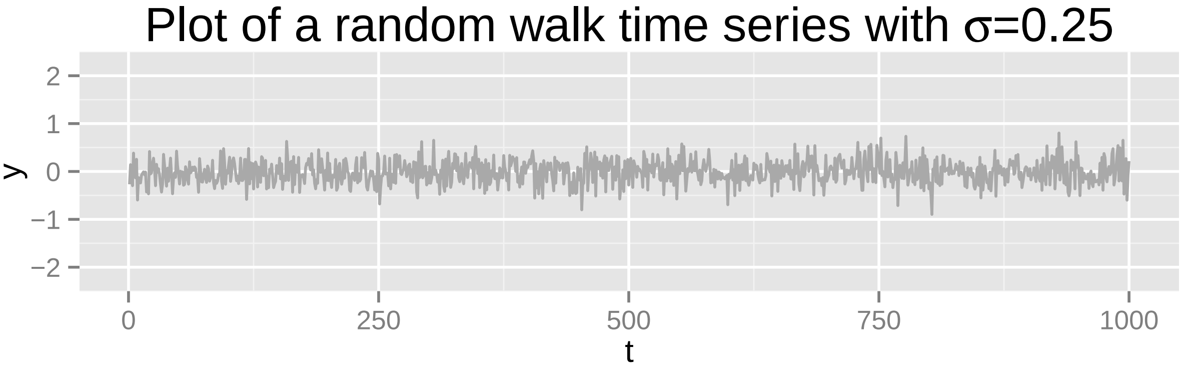
```r
## Load R packages and define helper functions
library(ggplot2, quietly = TRUE)
library(reshape2, quietly = TRUE)
library(grid, quietly = TRUE)
## Function to plot multiple ggplots on the same image
multiplot <- function(..., plotlist=NULL, cols) {
  require(grid)

    # 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 ))
    }
}

##
## function to simulate time series
##

simTimeSeries <- function(t, N, mu, s, phi){
  if(N == 1){
    y <- rep(0, t)                                  ## initialize container
    y[1] <- mu[1] + rnorm(1, 0, s)                  ## initailize time series at time 1
    for(i in 2:t){
      epsilon <- rnorm(1, 0, s)                     ## independent random error
      y[i] <- mu[i] + phi * y[i-1] + epsilon        ## autoregressive model
    }
  } else {
    y <- matrix(0, t, N)                            ## initialize container
    y[1, ] <- mu[1] + rnorm(N, 0, s)                ## initailize time series at time 1
    for(i in 2:t){
      epsilon <- rnorm(N, 0, s)                         ## independent random error
      y[i, ] <- mu[i] + phi * y[i-1, ] + epsilon        ## autoregressive model
    }
  }
  return(y)
}
```

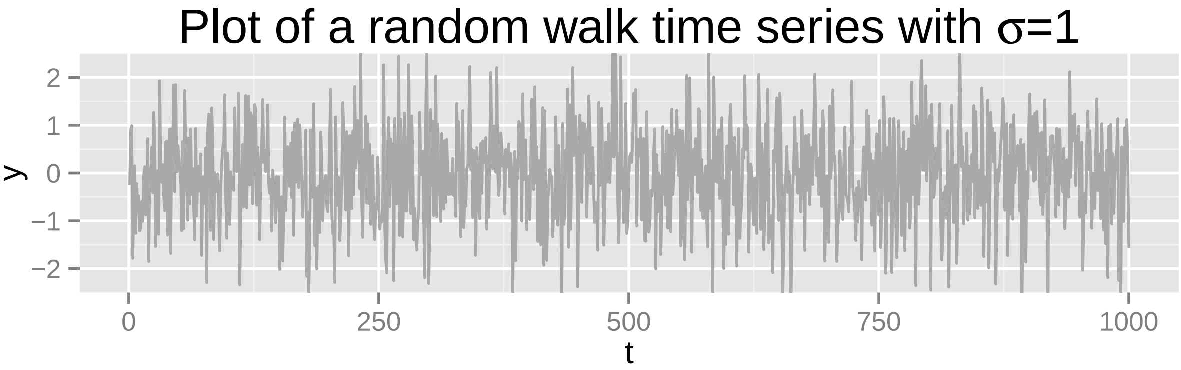
A time series is a series of observations y_t that occur over a period of time $t = 1, \ldots, T$. Depending on the problem being studied, these measurements could be made at all possible times, at regular intervals in time, or randomly through time. Thus, the analysis that is used must take into account the sampling methods used to obtain the data. The simplest time series is an uncorrelated random walk. In this time series, each observation y_t is just a random move from the previous location y_{t-1}. The random walk model is

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

where ϵ_t $\mathrm{N}(0, \sigma^2)$ is independent, uncorrelated error.

```
N <- 1                              ## pick the number of time series
t <- 1000                           ## pick a time series length
mu <- rep(0, t)                     ## pick a mean structure
s_low <- 0.25                       ## pick standard deviation
s_high <- 1                         ## pick standard deviation

y_s_low <- simTimeSeries(t, N, mu, s_low, phi=0)
y_s_high <- simTimeSeries(t, N, mu, s_high, phi=0)
plot_s_low <- ggplot(data=data.frame(y=y_s_low, t=1:t), aes(y=y, x=t)) +
  geom_line(alpha=1, colour="darkgrey") + coord_cartesian(ylim=c(-2.5, 2.5)) +
  ggtitle(substitute(paste("Plot of a random walk time series with ",
                           sigma, "=", sd), list(sd=s_low))) +
  theme(plot.title = element_text(size=18))
plot_s_high <- ggplot(data=data.frame(y=y_s_high, t=1:t), aes(y=y, x=t)) +
  geom_line(alpha=1, colour="darkgrey") + coord_cartesian(ylim=c(-2.5, 2.5)) +
  ggtitle(substitute(paste("Plot of a random walk time series with ",
                           sigma, "=", sd), list(sd=s_high))) +
  theme(plot.title = element_text(size=18))
multiplot(plot_s_low, plot_s_high, cols=1)
```





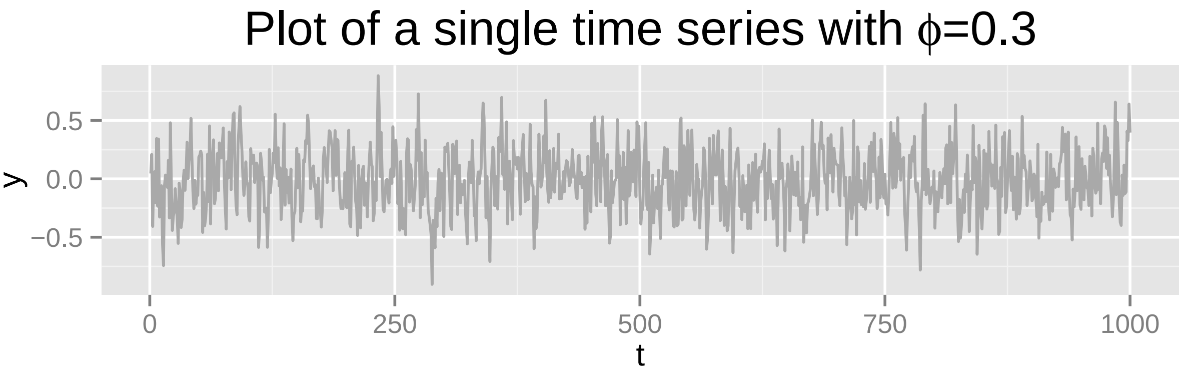
start with the canonical difference equation for the time series autoregressive model of order 1 (AR(1))

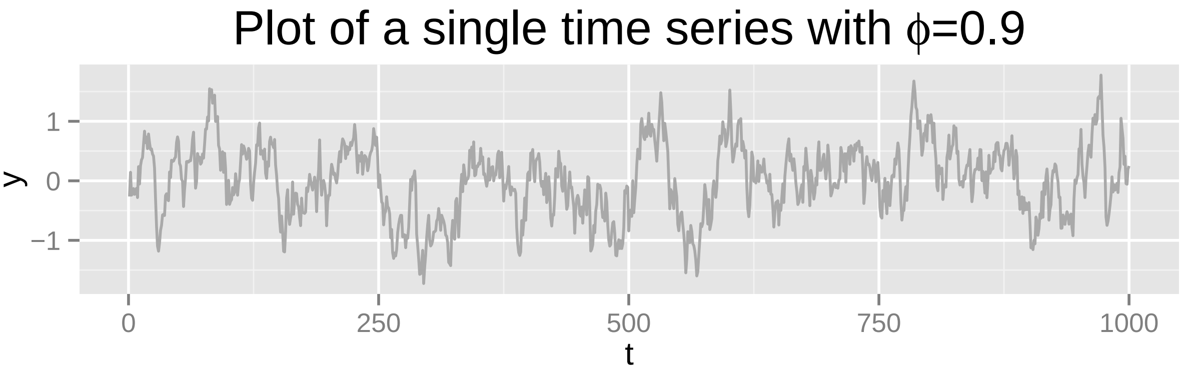
$$y_t = \mu + \phi y_{t-1} + \epsilon_t \tag{1}$$

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 representing the mean of the time series at time t. Often the mean is a trend or seasonal component. 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 $\text{Cov}(\epsilon_t, \epsilon_{t+k}) = 0$ 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 <- rep(0, t)                      ## pick a mean structure
s <- 0.25                            ## pick standard deviation
phi_low <- 0.30                      ## pick autocorrelation parameter
phi_high <- 0.90                     ## pick autocorrelation parameter

y_phi_low <- simTimeSeries(t, N, mu, s, phi=phi_low)
y_phi_high <- simTimeSeries(t, N, mu, s, phi=phi_high)
plot_phi_low <- ggplot(data=data.frame(y=y_phi_low, t=1:t), aes(y=y, x=t)) +
  geom_line(alpha=1, colour="darkgrey") +
  ggtitle(substitute(paste("Plot of a single time series with ", phi, "=", p), list(p=phi_low))) +
  theme(plot.title = element_text(size=18))
plot_phi_high <- ggplot(data=data.frame(y=y_phi_high, t=1:t), aes(y=y, x=t)) +
  geom_line(alpha=1, colour="darkgrey") +
  ggtitle(substitute(paste("Plot of a single time series with ", phi, "=", p), list(p=phi_high))) +
  theme(plot.title = element_text(size=18))
multiplot(plot_phi_low, plot_phi_high, cols=1)
```





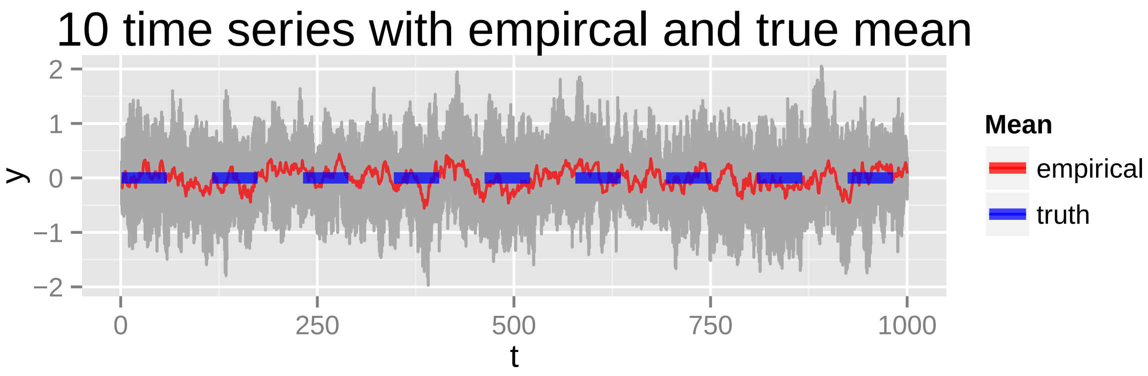
The expected value $\mathrm{E}(y_t)$ of the time series at time t is

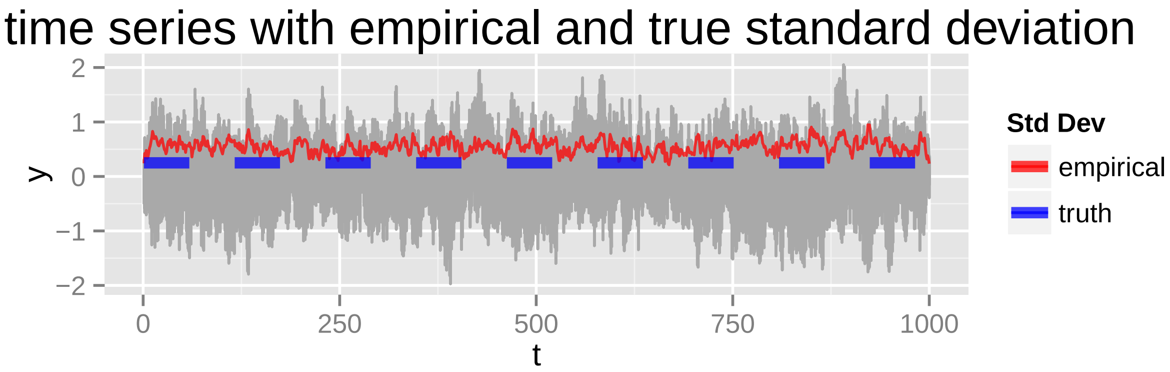
$$\begin{aligned}
\mathrm{E}(y_t) &= \mathrm{E}(\mu) + \mathrm{E}(\phi y_{t-1}) + \mathrm{E}(\epsilon_t) \\
&= \mu + \phi \mathrm{E}(y_{t-1}) + 0 \\
&= \mu + \phi(\mathrm{E}(\mu) + \mathrm{E}(\phi y_{t-2}) + \mathrm{E}(\epsilon_{t-1})) \\
&= \mu + \phi\mu + \phi^2\mu + \cdots \\
&= \frac{\mu}{1-\phi}
\end{aligned}$$

where the $\cdots$ form an infinite recursive sum. The autoregressive model assumes a constant variance through time (homoskedasticity). This means that for any times t and τ $\mathrm{Var}(y_t) = \mathrm{Var}(y_\tau)$. Therefore, the variance

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

```
N <- 10                               ## replicates N
phi <- 0.9                            ## autocorrelatin parameter
y <- simTimeSeries(t, N, mu, s, phi)  ## simulate time series

mean_y <- apply(y, 1, mean)           ##  calculate the empirical mean
var_y <-apply(y, 1, var)              ##  calculate the empirical variance
sd_y <- apply(y, 1, sd)               ##  calculate the empirical 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/(1-phi),
                        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) +
  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 with empircal and true mean")) +
  theme(plot.title = element_text(size=18))

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 with empirical and true standard deviation")) +
  theme(plot.title = element_text(size=18))

## Plot using multiplot
multiplot(plot_mean, plot_sd, cols=1)
```


10 time series with empircal and true mean

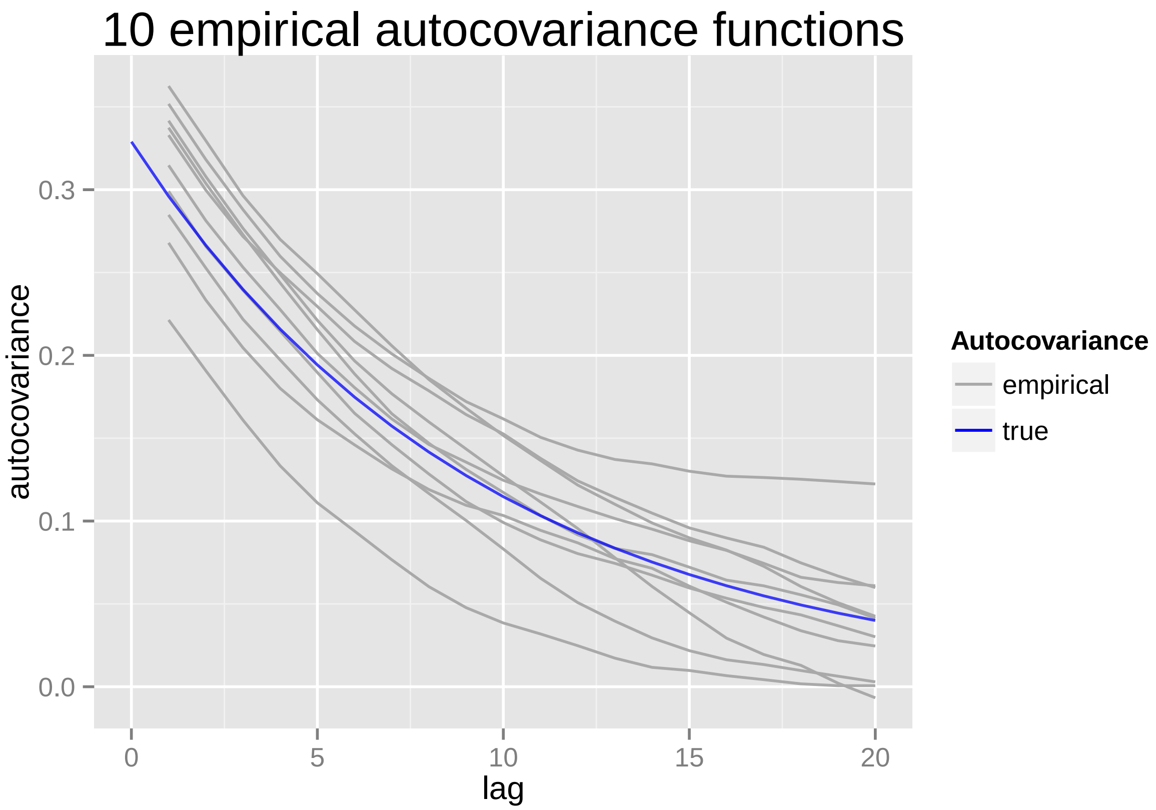

time series with empirical and true standard deviation

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

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

```
covariances <- matrix(0, min(t, 20), N)
for(i in 1:N){
  for(k in 1:min(t, 20)-1){
   covariances[k+1, i] <- cov(y[1:(t-k), i], y[1:(t-k) + k, i])
 }
}

cov_data <- data.frame(y=covariances, t=1:20)
melt_cov <- melt(cov_data, id="t")
```

```
ggplot(data = melt_cov, aes(y=value, x=t)) +
  geom_line(data=melt_cov, aes(y=value, x=t, group=variable, colour="empirical"), alpha=1) +
  geom_line(data=data.frame(y=s^2/(1-phi^2) * phi^(0:min(t,20)), x=0:min(t, 20)), aes(y=y, x=x, colour=
            alpha=0.75) +
  scale_colour_manual("Autocovariance", labels=c("empirical", "true"),
                      values=c("empirical"="darkgrey","truth"="blue")) +
  scale_y_continuous("autocovariance") + scale_x_continuous("lag") +
  ggtitle(paste(min(N, 10),
                "empirical autocovariance functions")) +
  theme(plot.title = element_text(size=18))
```



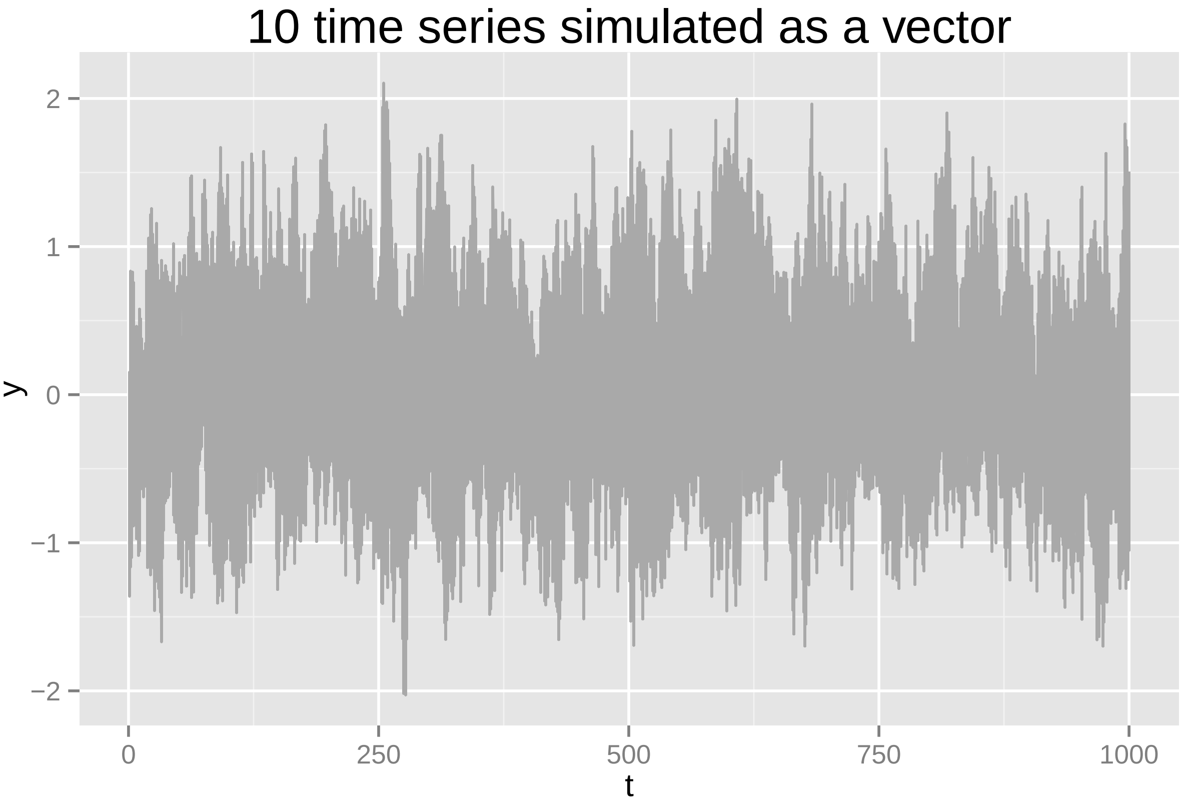
Thus, knowing the mean, variance, and covariance at each time t and each lag k, we can write the autoregressive model (**??**) as

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

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

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

```r
Sigma <- s^2 / (1 - phi^2) * toeplitz(phi^(0:(t-1)))
library(mvtnorm)
y_vec <- t(rmvnorm(N, mu, Sigma))

vec_data <- data.frame(y=y_vec, t=1:t)
melt_vec <- melt(vec_data, id="t")


ggplot(data = melt_vec, aes(y=value, x=t)) +
  geom_line(alpha=1, colour="darkgrey") +
  scale_y_continuous("y") + scale_x_continuous("t") +
  ggtitle(paste(min(N, 10),
                "time series simulated as a vector")) +
  theme(plot.title = element_text(size=18))
```



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 $\mathrm{Var}(y_{\tau+k}|y_{1:\tau-1}) = \sum_{i=1}^{k} \phi^{2(i-1)}\sigma^2$.