Ploting Example

Lukas Arnroth

4 augusti 2017

Generate Data

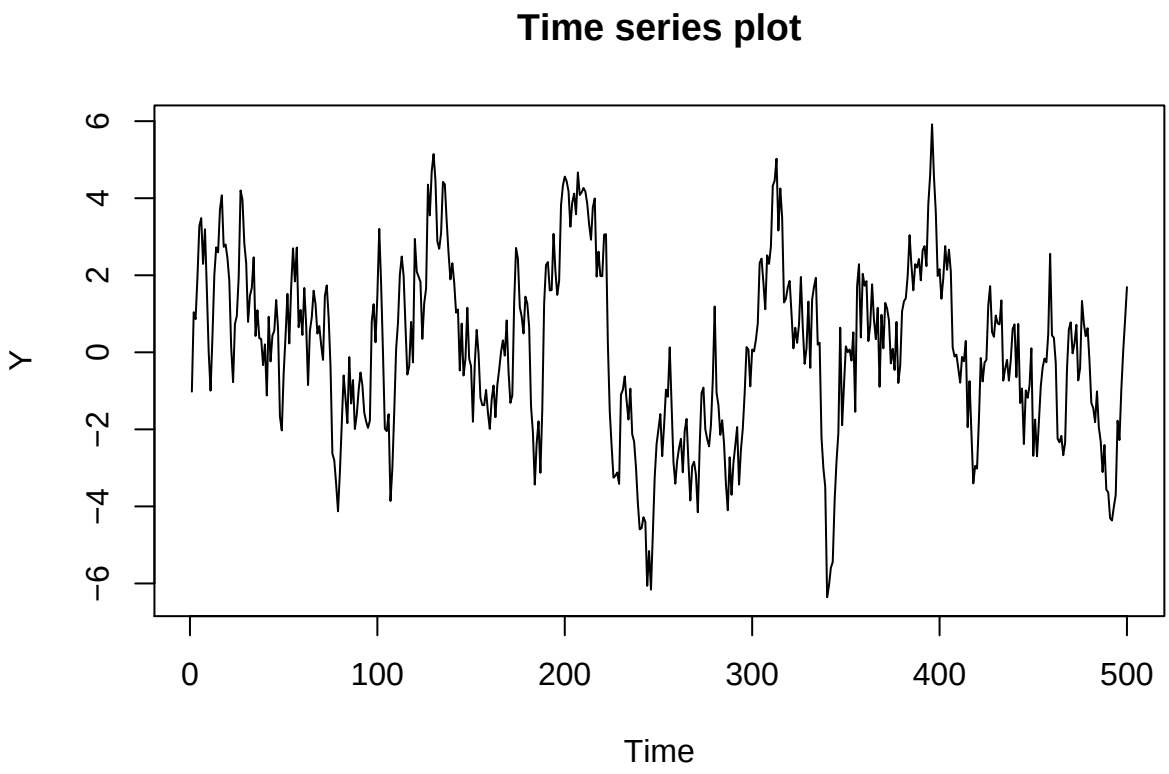
In this help document I will show how you should plot the result to minimize time spent on puzzling everything together. Lets begin by using the AR program to generate data.

```
# simulate AR(2) process
# there is a base function called arima.sim in R, which does simulations of time series.
# Problem is that it can't generate non-stationary AR processes so this one has
# to be written by hand

# Set parameters
phi_1 <- 0.9 # AR(1) parameter
phi_2 <- 0 # AR(2) parameter
t <- 5000 # t is the sample size (T is reserved as TRUE)

# Set seed for random number generation
set.seed(123456)
ar.sim <- arima.sim(model = list(ar = c(0.9)), n = 5000)

# variance of noise term
sigma2 <- 1

# initiate series. Let the first two elements of y_AR be innovations
y_AR <- c(rnorm(1)*sqrt(sigma2), rnorm(1)*sqrt(sigma2), rep(0, t-2))
e <- rnorm(t)*sqrt(sigma2) # residual


# realization
i <- 0
for(i in 3:t){
  y_AR[i] <- phi_1*y_AR[i-1] + phi_2*y_AR[i-2] + e[i]
}
# should be stored as time series object, makes for easier ploting.
y_AR <- as.ts(y_AR)
```

Plot the Result

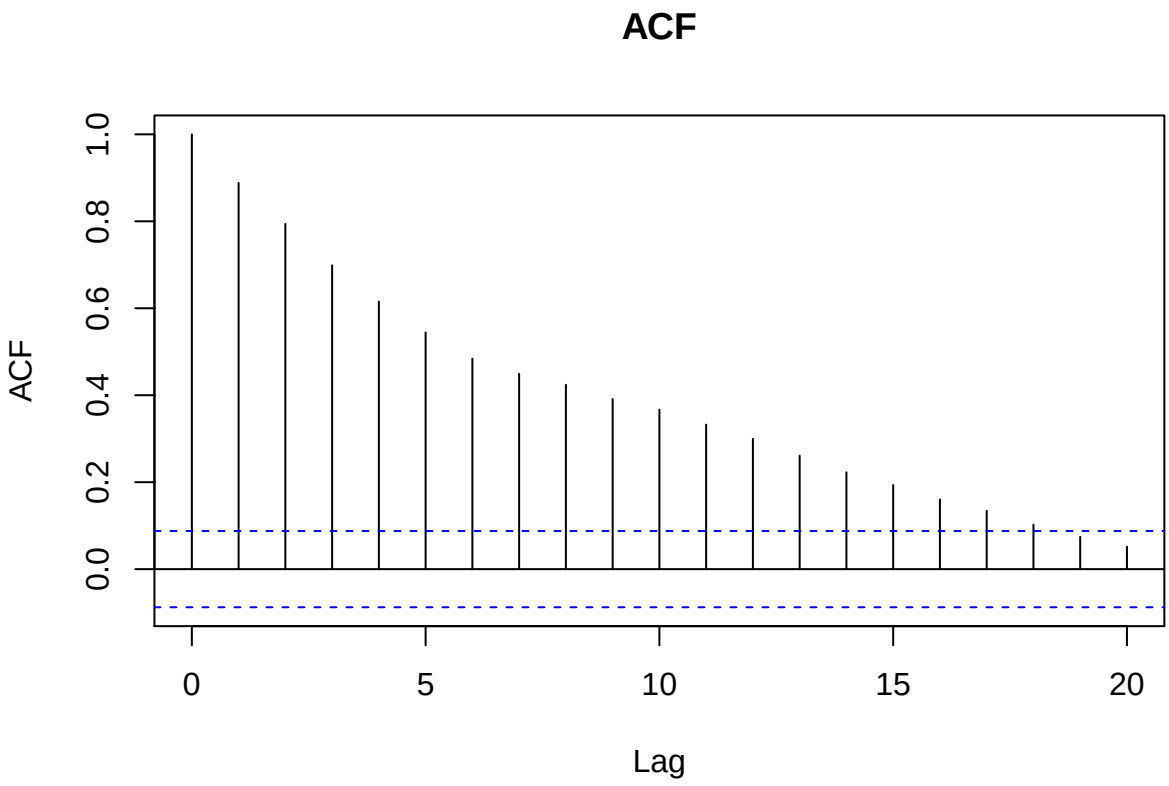
We will split up the plot window into three parts and fill these with plots. Instead of saving three seperate plots you only have to save one. We end this section with combining all the plots. Lets begin by actually creating the plots we want to combine. First up we want to plot the values. We will plot the first 500 observations of the time series we created at the end of the first section. When ploting a ts type object (ts standing for time series) we use the function ts.plot() as done below.

```
ts.plot(y_AR[1:500], ylab = "Y", main = "Time series plot")
```



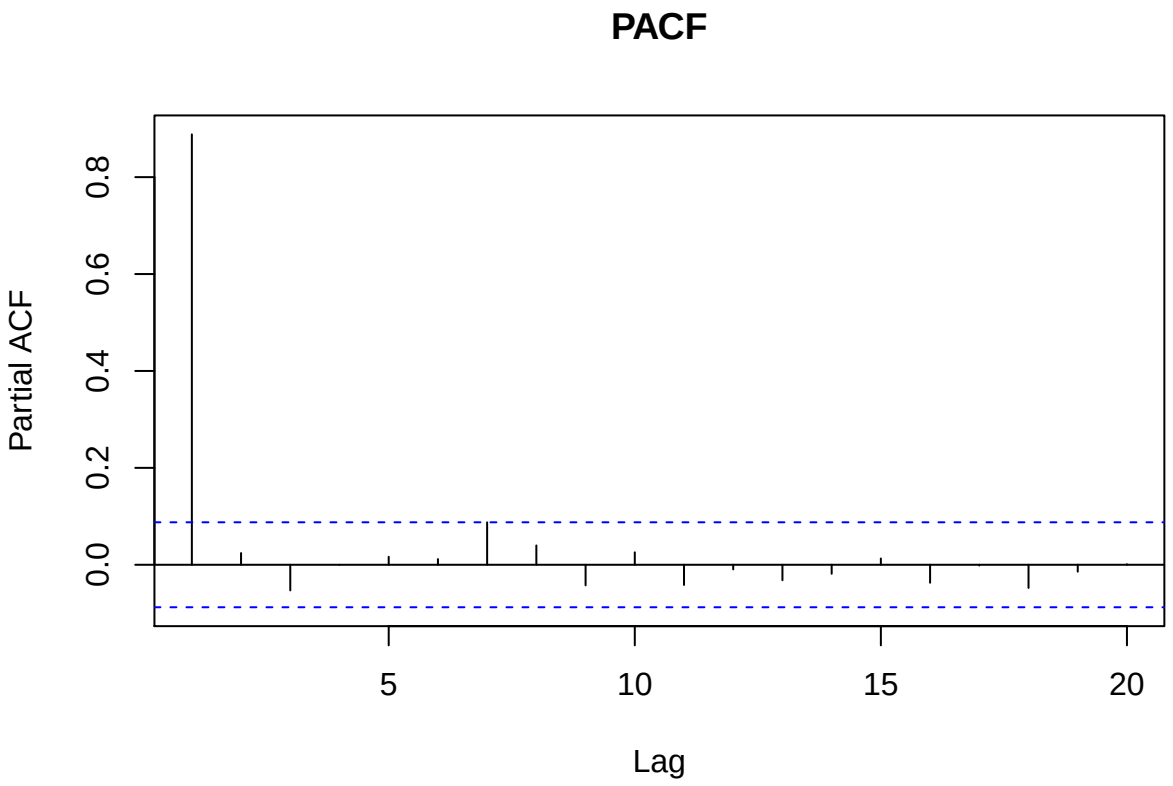
Next we plot the auto correlation. This is done using the acf() function and setting the parameter plot to TRUE. If this is not done you will instead get a vector of values. Note that type is set to "correlation"

```
acf(y_AR[1:500], lag.max = 20, type = "correlation", plot = T, main = "ACF")
```



When ploting the partial auto correlation we set type to “partial”.

```
acf(y_AR[1:500], lag.max = 20, type = "partial", plot = T, main = "PACF")
```



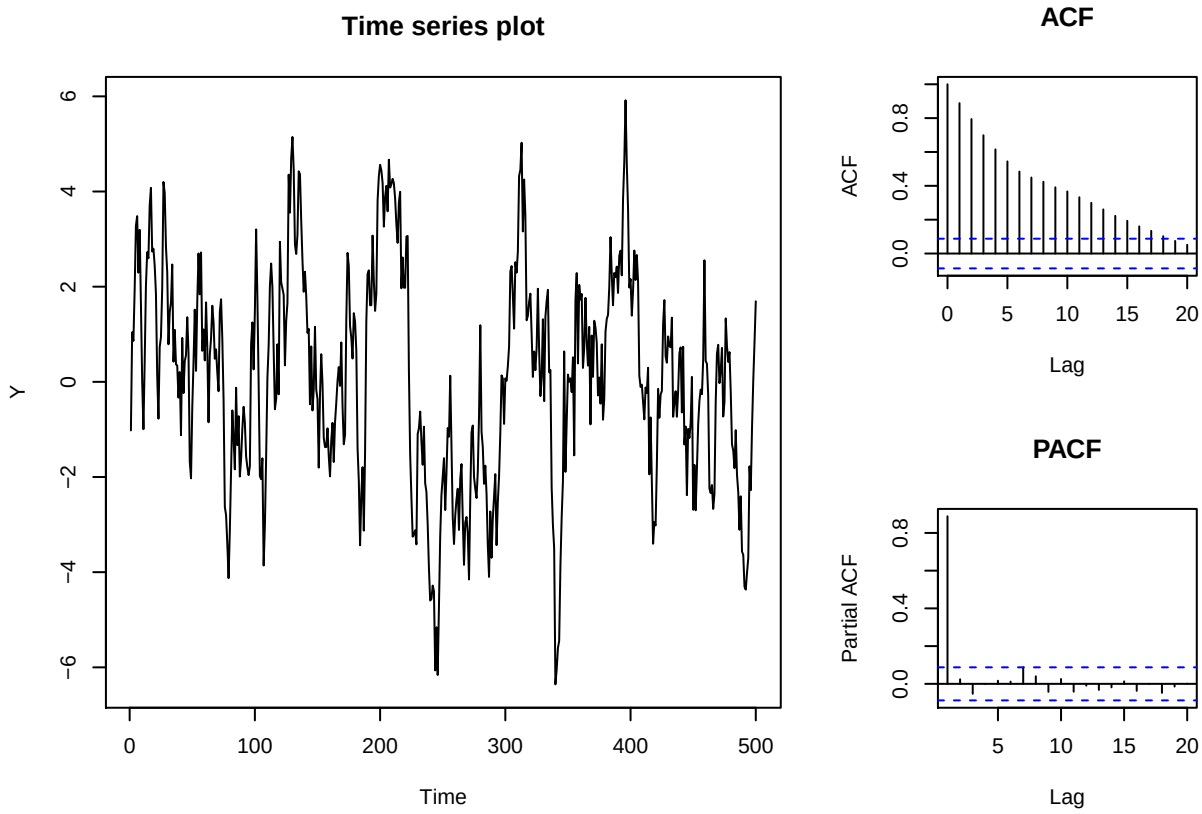
Combine Plots

There are different ways of doing this but we will use the layout() function as done below

```
layout(matrix(c(1, 1, 2,
                1, 1, 3),
              nrow=2, byrow=TRUE))
```

The layout function is supplied a matrix which has 2 rows and 3 columns. The numbers within $c()$ specifies the position of the plots. Plot 1 takes position row = 1,2 & column = 1,2. Plot 2 takes position row = 1 & column = 3. Plot 3 takes position row = 2 & column = 3.

```
layout(matrix(c(1, 1, 2,
                1, 1, 3),
              nrow=2, byrow=TRUE))

ts.plot(y_AR[1:500], ylab = "Y", main = "Time series plot")
acf(y_AR[1:500], lag.max = 20, type = "correlation", plot = T, main = "ACF")
acf(y_AR[1:500], lag.max = 20, type = "partial", plot = T, main = "PACF")
```



It's important to keep in mind that R remembers that you split up the plot window. Reseting it to default again is done as follows

```
par(mfrow = c(1,1))
```