

Functions and loops practice problem

Dave Millar
dave.millar@uwyo.edu
BOT 5790 - R for Ecologists
11/1/17

Bark beetle epidemics have resulted in widespread forest mortality throughout western North America. Once adult bark beetles emerge and attack a nearby tree, they introduce blue stain fungi to the xylem, effectively occluding it by spreading over time, which leads to hydraulic failure and ultimately death for the tree :(. The spread of this fungi within the xylem is a function of temperature-dependent growth rate. A blue stain fungal growth model can be fit to recently reported data (*Moore and Six 2015*), using a least-squares fit to a Gaussian function:

$$GR_i = a \cdot e^{-0.5 \cdot \left(\frac{AT_i - b}{c} \right)^2}$$

In the above equation, GR_i represents fungal growth rate ($\text{mm}^2 \text{ d}^{-1}$), AT_i represents mean daily air temperature ($^{\circ}\text{C}$), and a ($\text{mm}^2 \text{ d}^{-1}$), b ($^{\circ}\text{C}$), and c ($^{\circ}\text{C}$) are best-fit model coefficients. Fungal biomass (FBM) accumulation can then be simulated through the growing season using time series temperature data, where for each time step, FBM is equal to the sum of GR for that time step and all previous time steps following the onset of beetle attack.

The FBM model can then be used to model xylem sap flux decline in attacked trees over time, by using a least-squares fit to data that includes measured percent functional xylem and air temperature using the following sigmoid function:

$$XS_i = 100 \cdot \frac{1}{(1 + a_2 \cdot e^{b_2 \cdot FBM_i})}$$

The xylem scalar (XS_i) in the above equation represents percent functional xylem, and ranges from 100% (no impact) to 0% (full xylem occlusion) at any given time step, which can then be used to downscale sap flux over time. a_2 and b_2 best-fit model coefficients.

REFERENCES:

Moore, M. L., and D. L. Six (2015), Effects of temperature on growth, sporulation, and competition of mountain pine beetle fungal symbionts, *Microb. Ecol.*, 336-347, doi:10.1007/s00248-015-0593-8.

PROBLEMS:

1) Assume you want to include the above-mentioned functions as a submodel in a more complex ecosystem model. Write an R script that reads in two data sets (one for temperature-dependent growth for blue stain fungi, one based on sap flux decline for beetle-attacked trees) and determines best fit parameters for both equations using the `nls` function. Since this will be incorporated as a submodel into a more complex one, you'll want to contain everything into one function that can be called from a main script elsewhere.

The data set *blue_stain_temp_and_growth_rate.csv* contains temperature data (°C) in the column named “temp” and blue stain fungal growth rate (mm² d⁻¹) in the “growth” column (data from Moore and Six 2015). The data set *daily_at_and_perc_xylem.csv* contains dates in one column, daily mean air temperature (°C) in the column named “temp”, and percent functional xylem in the column named “perc_xylem”.

HINTS: Remember that variables calculated using one equation can be used as input for another, and also that **for** loops work great for iterative processes.

2) Visualize the XS model as a best-fit line, with the observed XS data on the same plot as points, both plotted against date, using ggplot2. Please include the function equation including the best-fit parameter values, along with the R² value in the plotting area.

The code chunk below is an example for using the `nls` (Nonlinear Least Squares) function in R.

First we'll just create some fake data to work with that has a bit of stochasticity to it. For now, just pretend these are response (`y_data`) and predictor (`x_data`) variable data you've collected during a study, and ignore the equation used to calculate `y_data`.

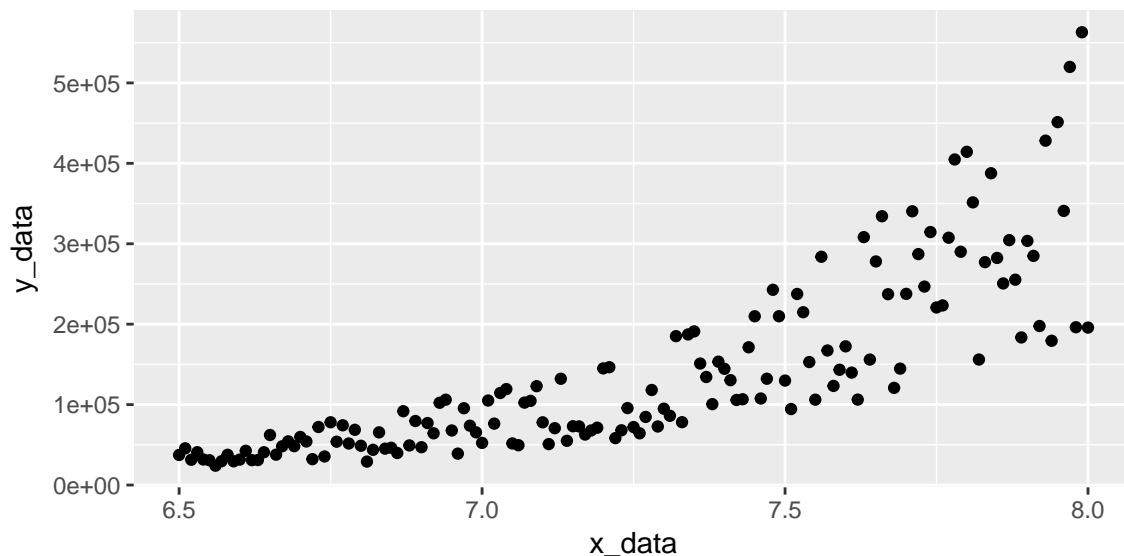
```
#clear everything out of memory
rm(list=ls())

library(ggplot2)

x_data <- seq(6.5,8,0.01)
y_data <- 5^(x_data)*runif(length(x_data),0.5,1.5)

fake_data <- cbind.data.frame(x_data,y_data)

# quick check
qplot(x_data,y_data)
```



Now, let's pretend after looking at the data using `qplot`, that we want to use a nonlinear equation to model `y_data` as a function of `x_data`, similar in structure to the equation in the code chunk above.

```
best_fit <- nls(y_data ~ a^x_data, data = fake_data,  
              start = list(a = 7.4))
```

```
best_fit
```

```
## Nonlinear regression model  
##   model: y_data ~ a^x_data  
##   data: fake_data  
##     a  
## 4.966  
## residual sum-of-squares: 4.844e+11  
##  
## Number of iterations to convergence: 6  
## Achieved convergence tolerance: 6.917e-06
```

```
summary(best_fit)
```

```
##  
## Formula: y_data ~ a^x_data  
##  
## Parameters:  
##   Estimate Std. Error t value Pr(>|t|)  
## a 4.96597    0.01765   281.4   <2e-16 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 56830 on 150 degrees of freedom  
##  
## Number of iterations to convergence: 6  
## Achieved convergence tolerance: 6.917e-06
```

```
a <- coef(best_fit)
```

```
a
```

```
##      a  
## 4.965971
```