Lab 2: Flow Control and the Urban Economy

36-350, Statistical Computing

Friday, 7 September 2012

Agenda: Manipulating data objects; understanding flow control; iterative approximation; turning math into code.

Instructions: Save all your answers in a single plain text file (Word files will not be graded), and upload it to Lore. When a question asks you to do something, give the command you use to do it. For questions which ask you to explain, write a short explanation in coherent, complete sentences. (You will be graded on your written explanation, not what you might say to the TA.) For figures and plots, include the command that you used to make the plot in your write-up, and show the figure to the TA. (Uploading the figure is optional.)

In your intro. stats. class, you saw the linear regression model

$$Y = b_0 + b_1 X + \text{noise} \tag{1}$$

and learned how to fit it by least squares. In the linear model, there is an exact formula for the best estimate of the parameters b_0 and b_1. This is not usually the case with non-linear regression models (they "have no closed-form solution"), but we can estimate them by iteration.

Recently, it has been proposed[1] that there is a non-linear relation between the number of people N in a city, and its economic output per person Y (per capita "gross metropolitan product"):

$$Y = y_0 N^a + \text{noise} \tag{2}$$

This is called a "power law" model.

We will use data from the 366 metropolitan area of the US (as of 2006) to estimate a, by minimizing the **mean squared error** (MSE),

$$\frac{1}{366}\sum_{i=1}^{366}(Y_i - y_0 N_i^a)^2 \tag{3}$$

over all a. We will see later in the class how we could estimate both parameters at once, but for now we'll treat y_0 as known, and in particular equal to \$6611.

[1] by Geoffrey West and collaborators; there's an entertaining on-line video of him giving a talk about the work to a TED conference, if you're into that sort of thing.

1. Using the `read.table` command, load the `gmp.dat` file[2] into a data frame called `gmp`. (5)
2. The last two columns give, for each city, the total gross metropolitan product (YN) and the per-capita metropolitan product (Y). Use this to make a vector which has the population (N) for each city. (5)
3. Add the vector of populations to the `gmp` data frame as a new column, named `pop`. (5)
4. Calculate the MSE of the power-law model, when $y_0 = 6611$ (in dollars) and $a = 0.15$. (10)
5. Use `seq()` to make a sequence of values of a, from 0.10 to 0.15, in steps of 0.005. (5)
6. Calculate the MSE for each value of a in that sequence, and store all the MSEs in a vector. (*Hint*: There are many ways to do this, including a `for` loop.) (15)
7. Plot the square roots of the MSEs vs. a. What are the units of the root mean squared error? What value of a, among those you looked at, leads to the least squared error? (15)
8. Explain what the following code[3] does. (20)

```
maximum.iterations <- 100
deriv.step <- 1/1000
step.scale <- 1e-12
stopping.deriv <- 1/100
iteration <- 0
deriv <- Inf
a <- 0.15
while ((iteration < maximum.iterations) && (deriv > stopping.deriv)) {
  iteration <- iteration + 1
  mse.1 <- mean((gmp$pcgmp - 6611*gmp$pop^a)^2)
  mse.2 <- mean((gmp$pcgmp - 6611*gmp$pop^(a+deriv.step))^2)
  deriv <- (mse.2 - mse.1)/deriv.step
  a <- a - step.scale*deriv
}
list(a=a,iterations=iteration,converged=(iteration < maximum.iterations))
```

9. What value of a does the code estimate? What is the root mean squared error at this value of a? How well does this agree with what you expect from the plot you made for question 7? (10)
10. Re-run this code, but change the initial value of a from 0.15, to your estimate from question 9. What happens? Why? (10)

[2] `http://www.stat.cmu.edu/~cshalizi/statcomp/labs/02/gmp.dat`
[3] `http://www.stat.cmu.edu/~cshalizi/statcomp/labs/02/lab-02.R`