Homework 5

Statistical Computing, 36-350

Due Wednesday Oct 7, 2015

Your homework must be submitted in R Markdown format. We will not (indeed, cannot) grade homeworks in other formats. Your responses must be supported by both textual explanations and the code you generate to produce your result. (Just examining your various objects in the "Environment" section of R Studio is insufficient—you must use scripted commands.)

Background: In the previous lectures and lab, we began to look at user-written functions. For this assignment we will continue with a look at fitting models by optimizing error functions, and making user-written functions parts of larger pieces of code.

In lecture, we saw how to estimate the parameter a in a nonlinear model,

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

by minimizing the mean squared error

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - y_0 N_i^a)^2.$$

We did this by approximating the derivative of the MSE, and adjusting a by an amount proportional to that, stopping when the derivative became small. In this assignment, we will estimate a using a built-in R function; it uses a fancier version of the same idea.

Because the model is nonlinear, there is no simple formula for the parameter estimates in terms of the data. Also unlike linear models, there is no simple formula for the standard errors of the parameter estimates. We will therefore use a technique called *the jackknife* to get approximate standard errors.

Here is how the jackknife works:

- Get a set of n data points and get an estimate $\hat{\theta}$ for the parameter of interest θ.
- For each data point i, remove i from the data set, and get an estimate $\hat{\theta}_{(-i)}$ from the remaining $n-1$ data points. The estimates $\hat{\theta}_{(-i)}$ are sometimes called the "jackknife estimates".
- Find the mean $\overline{\theta}$ of the n values of $\hat{\theta}_{(-i)}$
- The jackknife variance of $\hat{\theta}$ is

  $$\frac{n-1}{n}\sum_{i=1}^{n}(\hat{\theta}_{(-i)} - \overline{\theta})^2 = \frac{(n-1)^2}{n}\text{var}[\hat{\theta}_{(-i)}]$$

  where var stands for the sample variance. (*Challenge*: can you explain the factor of $(n-1)^2/n$? *Hint*: think about what happens when n is large so $(n-1)/n \approx 1$.)
- The jackknife standard error of $\hat{\theta}$ is the square root of the jackknife variance.

You will estimate the power-law scaling model, and its uncertainty, using the data alluded to in lecture, available in the file http://www.stat.cmu.edu/~ryantibs/homework/statcomp/homework/gmp.dat, which contains data for 2006.

```
gmp = read.table("http://www.stat.cmu.edu/~ryantibs/statcomp/homework/gmp.dat")
gmp$pop = round(gmp$gmp/gmp$pcgmp)
```

Optimize away

1. First, plot the data as in lecture, with per capita GMP on the y-axis and population on the x-axis. Add the curve function with the default values provided in lecture. Add two more curves corresponding to $a = 0.1$ and $a = 0.15$; use the `col` option to give each curve a different color (of your choice).
2. Write a function, called `mse()`, which calculates the mean squared error of the model on a given data set. `mse()` should take four arguments: a, y_0, a numerical vector containing the values of N, and a numerical vector containing the values of Y. The function should return a single numerical value. y_0 should have a default value of 6,611, and the latter two arguments should have as the default values the columns `pop` and `pcgmp` (respectively) from the `gmp` data frame from lecture. Your function may not use `for()` or any other loop. Check that, with the default data, you get the following values.

```
> mse(0.15)
[1] 207057513
> mse(0.10,5000)
[1] 298459914
```

3. There is no question 3.
4. R has several built-in functions for optimization, which we will meet as we go through the course. One of the simplest is `nlm()`, or nonlinear minimization. `nlm()` takes two required arguments: a function, and a starting value for that function. Run `nlm()` three times with your function `mse()` and three starting values for a as in `nlm(mse,0.125)`.

What do the quantities `minimum` and `estimate` represent? What values does it return for these?

(Note: when you ran `nlm()`, you may have received a wanning message, to the effect of "NA/Inf replaced by maximum positive value". This is OK.)

5. Let's check to see that the result from `nlm()` makes sense. Evaluate `mse()` at 100 points between $a = 0.10$ and $a = 0.15$ and make a plot of MSE versus a. Are the values you found in question 4(a) consistent with your plot?
6. Using `nlm()`, and the `mse()` function you wrote, write a function, `plm()`, which estimates the parameter a of the model by minimizing the mean squared error. It should take the following arguments: an initial guess for a; a value for y_0; a vector containing the N values; a vector containing the Y values. All arguments except the initial guess for a should have suitable default values. It should return a list with the following components: the final guess for a; the final value of the MSE. Your function must call those you wrote in earlier questions (it should not repeat their code), and the appropriate arguments to `plm()` should be passed on to them.

(Note: if `nlm()` is throwing warnings of the type described above, then you could use something like `suppressWarnings()`, wrapped around your call to `nlm()`, in order to quiet it down.)

What parameter estimate do you get when starting from $a = 0.15$ and $y_0 = 6611$? From $a = 0.10$ and $y_0 = 6000$? If these are not the same, why do they differ? Which estimate has the lower MSE?

The Jackknife

7. Here you will convince yourself the jackknife can work.
    a. Calculate the mean per-capita GMP across cities, and the standard error of this mean, using the built-in functions `mean()` and `sd()`, and the formula for the standard error of the mean you learned in your intro stats class (or looked up on Wikipedia...).

b. Write a function which takes in an integer `i`, and calculate the mean per-capita GMP for every city *except* city number `i`.

c. Using this function, create a vector, `jackknifed.means`, which has the mean per-capita GMP where every city is held out in turn. (You may use a `for` loop or `sapply()`.)

d. Using the vector `jackknifed.means`, calculate the jackknife approximation to the standard error of the mean. How well does it match your answer from part (a)?

8. Write a function, `plm.jackknife()`, to calculate jackknife standard errors for the parameter a. It should take the same arguments as `plm()`, and return the standard error. This function should call your `plm()` function repeatedly. What standard error do you get, with the starting values $a = 0.15$ and $y_0 = 6611$?

Bonus

9. The file http://www.stat.cmu.edu/~ryantibs/statcomp/homework/gmp-2013.dat contains measurements for 2013. Load it, and use `plm()` and `plm.jackknife()` to estimate the parameters of the model for 2013, and their standard errors. Have the parameters of the model changed significantly?