Lab 5: Testing Our Way to Outliers

36-350, Statistical Computing

Friday, 30 September 2011

Agenda: Debugging and testing; finding outliers; making code more general.

Instructions: Save all your answers in a single plain text file (Word files will not be graded), and upload it to Blackboard, using the page for this assignment. (There is no general digital dropbox any more.) 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.)

Identifying outliers in data is an important part of statistical analyses. One simple rule of thumb (due to John Tukey) for finding outliers is based on the quartiles of the data: the first quartile Q_1 is the value $\geq 1/4$ of the data, the second quartile Q_2 or the median is the value $\geq 1/2$ of the data, and the third quartile Q_3 is the value $\geq 3/4$ of the data. The interquartile range, IQR, is $Q_3 - Q_1$. Tukey's rule says that the outliers are values more than 1.5 times the interquartile range from the quartiles — either below $Q_1 - 1.5IQR$, or above $Q_3 + 1.5IQR$.

Consider the data values

```
x <- c(2.2, 7.8, -4.4, 0.0, -1.2, 3.9, 4.9, -5.7, -7.9, -4.9,  28.7,  4.9)
```

We will use these as part of writing a function to identify outliers according to Tukey's rule. Our function will be called `tukey.outlier`, and will take in a data vector, and return a Boolean vector, `TRUE` for the outlier observations and `FALSE` elsewhere.

1. Calculate the first quartile, the third quartile, and the inter-quartile range of `x`. Some built-in R functions calculate these; you cannot use them, but you could use other functions, like `sort` and `quantile`. (5)

2. Write a function, `quartiles`, which takes a data vector and returns a vector of three components, the first quartile, the third quartile, and the inter-quartile range. Show that it gives the right answers on `x`. (You do not have to write a formal test for `quartiles`.) (10)

3. Which points in `x` are outliers, according to Tukey's rule, if any? (5)

4. Write a function, `test.tukey.outlier`, which tests the function `tukey.outlier` against your answer in the previous question. This function should return `TRUE` if `tukey.outlier` works properly; otherwise, it can either return `FALSE`, or an error message, as you prefer. (You can do the next problem first, if you find that easier.) (10)

5. Write `tukey.outlier`, using your `quartiles` function. Show that it passes `test.tukey.outlier`. (10)

6. Which data values should be outliers in `-x`? (5)

7. Modify `test.tukey.outlier` to include a test for this case. (5)

8. Show that your `tukey.outlier` function passes the new set of tests, or modify it until it does. (10)

9. Tukey's test can also be used when we record multiple variables for each observation: a data point is an outlier if it is an outlier along any dimension. Consider

   ```
   y <- c(11.0, 14.0, 3.5, 52.5, 21.5, 12.7, 16.7, 11.7, 10.8, -9.2, 12.3, 13.8)
   z <- cbind(x,y)
   ```

   Treating each row of z as a data point, which ones are outliers according to Tukey's rule? (10)

10. Modify `test.tukey.outlier` to test `tukey.outlier` against this `z`. (10)

11. Modify `test.tukey.outlier` to check whether its input is a vector or a one-column array, or an array with multiple columns. In the vector or one-column case, it should work as before. In the multiple column case, it should apply the old procedure to each column separately, and then combine them appropriately. It should still return a vector of Boolean values in either case. Work on the code until it passes the modified `test.tukey.outlier`. (20)