Homework 5: Rancorous Debugging

36-350, Fall 2011

Due at 11:59 pm on Tuesday, 4 October 2011

INSTRUCTIONS: Submit a single plain text file, whose name clearly includes both your Andrew ID and the assignment number. Inside the file, clearly indicate which parts of your responses go with which problems. Raw R output is not acceptable, and will be marked down accordingly; you are communicating with a human being, and need to write in a human language. Files in other formats (Word, PDF, etc.) will receive a grade of zero. Homework not submitted through Blackboard will receive a grade of zero.

Direct objective: Practice with debugging and testing code. *Indirect objectives:* Measures of association other than correlation coefficients.

In introductory statistics classes, you learned that the correlation between two random variables X and Y is

$$\rho(X,Y) = \frac{\mathrm{Cov}[X,Y]}{\sigma(X)\sigma(Y)}$$

where $\sigma(X)$ is of the standard deviation of X, and $\mathrm{Cov}[X,Y]$ is the covariance between the variables.

The correlation is OK for measuring linear relationships between two variables. It works badly with non-linear relationships. For instance, the correlation between x and x^2 is not 1, though one is an exact, deterministic function of the other. To alleviate this, we sometimes use the *rank* correlation between X and Y. This means replacing the actual values of x_i and y_i with their ranks in the sample (1 for the smallest, 2 for the next-smallest, and so on to n for the largest), and taking the correlation among their ranks[1]. The rank correlation between X and Y runs from $+1$, when Y always increases as X increases, to -1, when Y always shrinks as X grows.

Here is some code for calculating rank correlation[2]. It contains a bug.

```
rankcor <- function(x,y) {
  n <- length(x)
  stopifnot(length(y)==n)
  x.ranks <- rank(x)
```

[1] Ordinary and rank correlation are sometimes called "Pearson" and "Spearman" correlation, respectively, after their inventors.

[2] R contains a function, `cor`, which can calculate both ordinary and rank correlation, among others. In general, you should use it, but it is off limits for this assignment.

```
  y.ranks <- rank(y)
  mean.x <- mean(x.ranks)
  mean.y <- mean(y.ranks)
  covariance.term <- cov(x.ranks-mean.x,y-mean.y)
  sd.x <- sd(x.ranks)
  sd.y <- sd(y.ranks)
  rank.cor <- covariance.term/(sd.x*sd.y)
  return(rank.cor)
}
```

This function is online at `http://www.stat.cmu.edu/~cshalizi/statcomp/hw/05/hw-05.R`.

1. Explain, in words, how `rankcor` is supposed to work. (10)

2. Run this function through the following case:

   ```
   x <- c(2,4,6,8)
   y <- c(64,36,16,4)
   rankcor(x,y)
   ```

   What value do you get for the rank correlation? (5)

3. Describe how you can know that this value must be wrong. There are at least two ways to show this. (10)

4. Find a case where the code works properly. Explain how you know that the output is correct, and how you found this case. (10)

5. Write a function, `test.rankcor`, which automatically checks for proper behavior in both cases, that in question 2 as well as the one you found in question 4. It should return `TRUE` if both tests are passed, or `FALSE`, with warnings about which tests failed. This function should *not* presume that `x` and `y` from question 2 are part of the global environment. (15)

6. Find the bug. Carefully explain your reasoning. (10)

7. Fix the bug. Show that your modification passes both test cases. (10)

8. Explain why the mean rank of a vector of n observations is always $\frac{n}{2} + 0.5$. (Assume that ties are broken arbitrarily, so no two observations have the same rank.) Re-write the code to use this fact. Show that the code still passes the tests. (10)

9. Explain why it is not necessary to subtract the mean rank before calculating the covariance at all. Re-write the code accordingly, and show that it still works. (10)

10. Find a formula for the standard deviation of the ranks in a vector of n observations. Use it to replace the calls to `sd`. Verify that the code still works. (10)