

Lab 6: The bootstrap (I)

Due 2/24, 3 Questions

Last week we made a move from testing to estimation and introduced the bootstrap as a tool for assessing uncertainty. In this lab we will discuss some of the mechanics behind bootstrapping; in the next lab, we will examine built-in functions that perform the bootstrap with one command (as opposed to writing out the code to generate samples). This lab starts with a dangling thread, with a subject that has been waiting for a week or two to be fully described: The Q-Q plot.

1. Getting started

We will first load up data that we'll need for this lab.

```
source("http://www.stat.ucla.edu/~cocteau/stat13/data/fruit.R")  
  
ls()
```

This lab is a little, well, cluttered datawise. You should see the following separate data sets: `banana`, `broc`, `grapefruit`, `lemon`, `gs`, `lettuce`, `lime`, `orange`, `pepper`, `snow`, `spinach`, and `tomato` (`gs` stands for Granny Smith apples). Each holds measurements made on samples of produce collected from a chain of supermarkets in Atlanta, GA. So, the unit of observation is, literally, a piece of fruit. The two variables are `green` and `red`: `green` represents the intensity of light reflected off the samples in a subset of the green range (500-565nm); and `red` holds intensity measurements from the same sampled produce, but in a subset of the yellow/orange/red range (565-740nm).

The following commands will give you a basic sense of what's in these data.

```
names(banana)  
dim(banana)  
  
names(orange)  
dim(orange)
```

You should see that the data sets contain different numbers of items. These data were collected during the second phase of the fruit scanning project mentioned in Lecture 10; the sample sizes are more or less related to the color variation researchers found in different groups of produce.

2. Quantiles

We have been using normal Q-Q plots (what your book calls normal probability plots) for some time now in lecture, but haven't really let you kick the tires on the tool. It's time. We have already seen the idea behind a quantile (or percentile in your book's terminology). The median is the point that separates your data in half (if you have an even number of samples, say). It is also known as the 50th percentile or the 0.5 quantile. Similarly, the lower quartile is the point below which 25% of your data lie; this is also known as the 25th percentile or the 0.25 quantile. Finally, the upper quartile is the point below which 75% of the data lie; it is also known as the 75th percentile or the 0.75 quantile.

In general, for any $0 \leq q \leq 1$, we can define the q quantile, x_q , to be the point that separates the data into two parts: the proportion of data less than or equal to x_q is q and the proportion strictly greater than x_q is $1 - q$. Your book considers just 100 values of q (0.01, 0.02, ..., 0.99, 1.0) and calls the points percentiles (for obvious reasons). The idea of a quantile is more general in that q can take on any value between 0 and 1.

To try this out, consider the following sequence of commands.

```
hist(banana$green)
median(banana$green)
quantile(banana$green,0.5)
quantile(banana$green,c(0.25,0.5,0.75))
```

The last command will return a vector of length three. By “concatenating” the proportions 0.25, 0.5 and 0.75, we have created a vector of three items that R then uses as input to the `quantile` function. You can compare these numbers to the endpoints of a boxplot

```
boxplot(banana)
```

or you can add these three to your histogram.

```
hist(banana$green)
x = quantile(banana$green,c(0.25,0.5,0.75))
abline(v=x)
```

As we saw in class, a normal quantile-quantile or Q-Q plot (again, a normal probability plot according to your text) compares quantiles of your data to those computed for the normal distribution (the bell shape). In this case, we replace counting points (as we do to determine quantiles for a data set) with computing areas under the normal curve. The command is given below.

```
qqnorm(orange$green)
qqline(orange$green)
```

The second command adds a line to the plot to help your eye a little. As we discussed in lecture, if our data points are distributed around the median in proportions that match the way the normal curve specifies areas around its peak, then we should get a straight line. This is the easiest way to see that something does or does not follow a normal law.

In some cases, we will see departures from a straight line at the ends of a Q-Q plot; that is, in the tails of the distribution. For example, in the Q-Q plot of “greenness” of the orange samples, we see a tipped U shape. At the right, the data pull away from the line and move above it; this means that the tails of the data are “heavier” than the normal would want – that there is more data farther out than a bell shape would dictate. On the left side, we see the data also pulling away from the line and moving above it; this means that the left tail is not long enough – that there is more data closer to the center of the distribution than the bell shape would dictate.

Now look at broccoli.

```
qqnorm(broc$green)
qqline(broc$green)

hist(broc$green)
```

Question 1: Use the histogram and Q-Q plot above to describe whether or not the distribution of broccoli “greenness” follows a normal distribution. If not, explain why.

Question 2: Consider two other kinds of produce and either the red or green variable. For each, make a histogram and a Q-Q plot and explain whether or not the data appear bell-shaped and why.

3. A first pass at the bootstrap

Recall from Lecture 11 that we could use the bootstrap to provide an estimate of the sampling distribution of a statistic we're interested in, providing us with a fairly simple (at least conceptually) way to assess an estimate's accuracy. Let's consider a simple case, the mean. Suppose we would like to estimate the average greenness of grapefruits sold by a particular chain of markets in Atlanta. This will be the population and we would like to say something about the average greenness of this population, a population parameter. Our data are a sample of 767 grapefruit taken from this chain of markets. The mean greenness in our sample is 1.98 (compared to an average greenness of 3.67 for broccoli, say). The sample mean, 1.98, is an estimate of the population mean.

```
mean(grapefruit$green)
mean(broc$green) # just for comparison
```

Of course we are not interested in the 767 grapefruits in our sample, but rather, we would like to know what these data say about the greenness of the larger population of grapefruit at this chain of markets. To assess the accuracy of our estimate of 1.98, we will perform a basic bootstrap (re)sampling procedure. The code below draws bootstrap samples of size 767 from our original data (sampling with replacement), each time forming a bootstrap replicate of the mean.

```
nsim = 5000
bootmns = rep(0,5000)

for(i in 1:nsim)
{
  boots = sample(grapefruit$green,767,replace=T)
  bootmns[i] = mean(boots)
}
```

This code looks very much like an example from the last lab; recall our sequence of commands to simulate the sampling distribution for the odds ratio. Again, we have a loop and then a series of commands that are performed with each iteration. In this case, we are drawing new bootstrap samples each time and computing the bootstrap replicate of our original estimate, the mean.

The distribution of the bootstrap replicates is an estimate of the sampling distribution of our estimator, the mean greenness. In lecture, we saw that in many cases, this distribution is bell-shaped. What do you think?

```
hist(bootmns)
qqnorm(bootmns)
qqline(bootmns)
```

We can estimate the standard error for the mean of our sample of grapefruit by taking the standard deviation of the bootstrap replicates. The standard error provides us with a sense of the accuracy of our estimate, in this case 1.98. Assuming that the bootstrap replicates have a bell-shaped distribution, we can form a (roughly) 95% confidence interval by taking 1.98 plus or minus two estimated standard errors.

```
se = sd(bootmns)
1.98-2*se
1.98+2*se
```

And remember from lecture, that we could also form a 95% confidence interval using the quantiles of the bootstrap replicates. If the bootstrap replicates have a bell-shaped distribution, these two approaches should give similar results.

```
quantile(bootmns,c(0.025,0.975))
```

Question 3: Select a (as in one) different variety of produce to work with. Compute the sample mean of either its redness or greenness and estimate its standard error using the bootstrap resampling procedure above. Comment on whether the bootstrap replicates (an estimate of the sampling distribution of the sample mean) is bell-shaped and form the two kinds of confidence intervals examined above. Do they agree? Why?