Statistics 13, Lab 4

The bootstrap

1. Getting started

In this lab, we will examine the use of the bootstrap for assessing the accuracy of sample estimates. Our test data will be the CDC Behavioral Risk Factor Surveillance System you studied in a previous lab. At that time, we focused mainly on descriptive statistics and on telling a story about a particular sample. We looked at histograms and barplots and various numerical summaries to identify patterns in the data set. Our reasoning, however, never led us beyond the particular group of respondents. In this lab, we will turn from description to estimation. In particular, we will use the fact that the CDC survey is a random sample to make inferences about the population from which it was drawn, in this case all adults living in the United States.

As with Lab 2, we begin by loading the dataset of 20,000 observations into you R workspace. After starting R (by clicking on the RStudio icon in your dock, or on your desktop if you have downloaded a copy of R/RStudio onto a computer running some flavor of the Windows operating system), enter the following command.

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

You should see (possibly among other things) the data set `cdc`. Recall that it contains responses from 20,000 people to a series of 11 questions; therefore, our sample consists of 20,000 people, each observation is person, and for each person we have 11 variables. You can see the size of the data set and the names of the variables it contains by typing

```
dim(cdc)

names(cdc)
```

This should return the names `state`, `genhlth`, `physhlth`, `exerany`, `hlthplan`, `smoke100`, `height`, `weight`, `wtdesire`, `age`, and `gender`. In class, we described the questions leading to these variables: For example, `genhlth`, respondents were asked to evaluate their general health, giving a score from 1-5 (excellent = 1, very good = 2, good = 3, fair = 4 and poor = 5); `exerany`, this is 1 if the respondent exercised in the past month and 0 otherwise; `hlthplan`, this is a 1 if the respondent has some form of health coverage and 0 otherwise; `smoke100`, this is 1 if the respondent has smoked at least 100 cigarettes in their entire life and 0 otherwise; and finally, we have variables that record the respondent's height in inches, weight in pounds, age in years, and gender. Consult your lecture notes for a compete description of all the variables in our dataset. Also, the state variable is coded using the FIPS (Federal Information Processing Standards) code, described at `http://www.itl.nist.gov/fipspubs/fip5-2.htm` (scroll down the page a bit; you'll find a table of numeric codes and state names).

2. Quantiles

In the last few lectures, we have been using normal Q-Q plots as a means for assessing the shape of a distribution and how "normal" it appears. We've seen that this display can be a useful addition to our other summaries like boxplots and histograms that give us a sense of how values of a variable are distributed. In lecture we discussed briefly the idea behind quantiles and percentiles and your book precise contains definitions. Recall, for example, that 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. If q takes on one of the 100 values $0.01, 0.02, \ldots, 0.99, 1.0$ we call the points x_q percentiles (for obvious reasons). The idea of a quantile, however, 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(cdc$weight)

median(cdc$weight)

quantile(cdc$weight,0.5)

quantile(cdc$weight,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(cdc$weight)
```

or you can add these three to your histogram.

```
hist(cdc$weight)

x <- quantile(cdc$weight,c(0.25,0.5,0.75))

abline(v=x)
```

As we saw in class, a normal quantile-quantile or Q-Q plot compares quantiles of your data to those computed for the normal distribution (a 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. For any $0 \leq q \leq 1$, we can define the q quantile, x_q, to be the point that divides the real line into two parts: the area under the curve to the left of x_q is q and the area to the right is 1-q. A normal Q-Q plot in R, then, compares quantiles computed from your sample to those of the normal distribution. Looking at the histogram of respondents' weights,

describe the distribution.

```
hist(cdc$weight)
```

Now make a normal Q-Q plot.

```
qqnorm(cdc$weight)

qqline(cdc$weight)
```

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 it's 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 weight, we see a slightly 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 a little 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.

As we start to examine various statistics from these data, we found that it might be useful to separate the data into males and females.

```
males <- subset(cdc,gender=="m")

females <- subset(cdc,gender=="f")
```

Question 1: Use a histogram and a normal Q-Q plot to describe the distribution of male and female heights. Do they follow a normal distribution? If not, explain why.

3. A first pass at the bootstrap

Recall from lecture 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 proportion of adult males in the United States that want to lose weight. In technical terms, our population consists of adult males living in the United States from which the CDC survey is a random sample. We would like to say something about the population from studying our sample.

```
lose <- males$weight > males$wtdesire

head(lose)

mean(lose)
```

The first command creates a variable `lose` of `TRUE` and `FALSE` values (`TRUE` if a man's current weight is larger than his desired weight). We see the first few values of this vector with the second command, and form the proportion of men who want to lose weight in the sample with the final `mean` command. (When you need to compute with `TRUE` and `FALSE`, R will convert `TRUE` to 1 and `FALSE` to 0 so that the `mean` is summing up the number of `TRUE`s and dividing by the number of men in the sample, giving us the proportion we want.)

We see that 54.8% of male respondents in the survey want to lose weight. What can we say about this fraction in the population of all adult males in the United States? Following the strategy described in class we will "analyze as we randomized" and use the bootstrap to resample a number of times, providing us with an approximation to the sampling distribution of our estimate. In the code below, we will draw 5,000 bootstrap samples.

```
n <- nrow(males)

replicates <- rep(0,5000)

for(i in 1:5000){

   bootsample <- sample(lose,n,replace=TRUE)
   replicates[i] = mean(bootsample)
}
```

This code starts by figuring out how big our sample is (in this case the number of males). We wanted 5,000 bootstrap samples and so we then create a vector of that length (all zeroes) to hold the results (have a look at `replicates` after you issue the second command above by typing its name and hitting enter). Then, the line that starts `for(...)` is the beginning of a loop. It says you repeat the commands in the curly braces 5000 times, each time assigning a new value to i. The first time through the loop, $i = 1$, the second time $i = 2$ and the last time, $i = 5000$. For each iteration, we are generating a new bootstrap sample by sampling with replacement from our original data. (This step mimics what happens when we copy data to generate the bootstrap population if the real world population is large.) For each bootstrap sample we form the mean, which in this case is just the proportion of males in the bootstrap sample that want to lose weight. In the end `replicates` will have 5,000 bootstrap replicates, `replicates[1]` having the proportion associated with the first iteration, `replicates[2]` having the number from the second iteration and so on.

The distribution of the bootstrap replicates is an estimate of the sampling distribution of our estimator, the proportion of males who want to lose weight. In lecture, we saw that in many cases, this distribution is bell-shaped. What do you think?

```
hist(replicates)

qqnorm(replicates)
```

```
qqline(replicates)
```

We can estimate the standard error for the mean of our sample of males by taking the standard deviation of the bootstrap replicates. The standard error provides us with a sense of the accuracy of our estimate. Assuming that the bootstrap replicates have a bell-shaped distribution, we can form a (roughly) 95% confidence interval by taking 0.548 plus or minus two estimated standard errors.

```
se <- sd(replicates)

0.548-2*se

0.548+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(replicates,c(0.025,0.975))
```

What do you think? Do these two numbers look close to the $0.548 \pm 2se$ from the previous block of code?

Question 2: Repeat this process for females. Compute the proportion in our sample who want to lose weight and construct the two different confidence intervals for the proportion in the population. How do they compare? What does this say about adult females in the United States?

Now, if we want to estimate the average desired change in weight among adult males in the United States, we can focus on a slightly different statistic.

```
diffs <- males$weight - males$wtdesire

head(diffs)

mean(diffs)
```

The vector `diffs` now holds the differences between male respondents' weights and their desired weights. The average difference in our sample is 10.7lbs. What does this say about adult males in the U.S. population?

Question 3: Use the bootstrap to come up with a confidence interval for the average difference between current and desired weight among adult males in the United States. Have a look at your bootstrap replicates and tell us something about the sampling distribution of this estimate.