

Lab 1: Introduction to Data

Throughout this lab, you will be generating simple graphical and numerical summaries of a dataset collected by the Centers for Disease Control and Prevention (CDC). The Behavioral Risk Factor Surveillance System (BRFSS) is an annual telephone survey of 350,000 people in the United States. As its name implies, the BRFSS is designed to identify risk factors in the adult population and report emerging health trends. For example, respondents are asked about their diet and weekly physical activity, their HIV/AIDS status, possible tobacco use, and even their level of healthcare coverage. The BRFSS Web site (<http://www.cdc.gov/brfss>) contains a complete description of the survey, the questions that were asked and even research results that have been derived from the data.

We will focus on a random sample of 20,000 people from the BRFSS survey conducted in 2000. While there are over 200 questions or variables in this dataset, we are only going to work with a small subset. During the first part of the lab, we will walk you through the commands to recreate the graphical and numerical summaries and have you discuss what you see.

Getting started

We begin by loading the dataset of 20,000 observations into the R workspace. After launching RStudio, enter the following command.

```
source("http://www.openintro.org/stat/data/cdc.R")
```

The dataset `cdc` that shows up in your workspace is a *data matrix*, with each row representing a *case* and each column representing a *variable*. R calls this data format a *dataframe*, which is the term that we'll be using throughout the labs.

To find the names of the variables, type the command

```
names(cdc)
```

This will return the names `genhlth`, `exerany`, `hlthplan`, `smoke100`, `height`, `weight`, `wtdesire`, `age`, and `gender`. Each one of these variables corresponds to a question that was asked in the survey. For example, for `genhlth`, respondents were asked to evaluate their general health, responding either excellent, very good, good, fair or poor; `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 as well as their desired weight, `wtdesire`, `age` in years, and `gender`.

Exercise 1 How many cases are there in this dataset? How many variables? For each variable, identify its data type (eg. categorical, discrete).

We can have a look at the first few entries (rows) of our data with the command

```
head(cdc)
```

and similarly we can look at the last few by typing

```
tail(cdc)
```

You could also look at *all* of the dataframe at once by typing its name into the console, but that might be unwise here. We know `cdc` has 20,000 rows, so viewing the entire dataset would mean flooding your screen. It's better to take small peeks at the data with `head()`, `tail()` or the subsetting techniques that you'll learn in a moment.

Summaries and tables

The BRFSS questionnaire is a massive trove of information. A good first step in any analysis is to distill all of that information into a few summary statistics and graphics. As a simple example, the command `summary()` returns a six number summary: minimum, first quartile, median, mean, second quartile, and maximum. For `weight` this is

```
summary(cdc$weight)
```

R also functions like a very fancy calculator. If you wanted to compute the interquartile range for the respondents' weight, you would look at the output from the summary command above and then enter

```
190.0 - 140.0
```

R also has built-in functions to compute these summary statistics one by one. For instance, to calculate the mean and variance of `weight`, you type

```
mean(cdc$weight)
var(cdc$weight)
```

There is even a shortcut for the median

```
median(cdc$weight)
```

While it makes sense to describe a quantitative variable like `weight` in terms of these statistics, what about categorical data? We would instead consider the sample frequency or relative frequency distribution. The command `table()` does this for you by counting the number of times each kind of response was given. For example, to see the number of people who have smoked 100 cigarettes in their lifetime, you could type

```
table(cdc$smoke100)
```

or instead look at the relative frequency distribution by typing

```
table(cdc$smoke100)/20000
```

Again, since R is a big calculator, we can divide the totals from `table(cdc$smoke100)` by 20,000, the sample size. Finally, a barplot is as simple as

```
barplot(table(cdc$smoke100))
```

Notice what we've done here! We've formed a table from the `smoke100` variable with `table(cdc$smoke100)` and we've provided that output to the command `barplot()` that, in turn, makes a barplot of the counts in the table. This is an important idea: R commands can be nested. You could also break this into two steps by typing the following:

```
smoke <- table(cdc$smoke100)
barplot(smoke)
```

Here, we've made a new object, a table, called `smoke` (the contents of which we can see by typing `smoke` into the console) and then used it in as the input for `barplot()`. The special symbol `<-` performs an *assignment*, taking the output of one line of code and saving it into an object in your workspace. This is another important idea that we'll return to later.

Exercise 2 Create a numerical summary for `height` and `age` and compute the inter-quartile range for each. Compute the relative frequency distribution for `gender` and `exerany`. How many males are in the sample? What proportion of the sample reports being in excellent health?

The `table()` command can be used to tabulate any number of variables that you provide. That is, if we want to examine which participants have smoked broken down by gender, we could use the following.

```
table(cdc$gender, cdc$smoke100)
```

Here, we see column labels of 0 and 1; 0 meaning they have not smoked 100 cigarettes in their lifetime, and 1 meaning they have. The rows refer to gender. To create a mosaic plot of this table, we would enter the following command.

```
mosaicplot(table(cdc$gender, cdc$smoke100))
```

And we could have done this in two steps, as we did with the barplot example above.

Exercise 3 What does the mosaic plot reveal about smoking habits and gender?

Interlude: How R thinks about data

We mentioned that R likes to store data in dataframes, which you might think of as a type of spreadsheet. Each row is a different observation (a different respondent) and each column is a different variable (the first is `genhlth`, the second `exerany` and so on). We can see the size of the dataframe next to the object name in the workspace or we can type

```
dim(cdc)
```

which will return the number of rows and columns. Now, if we want to access a subset of the full dataframe, we can use row-and-column notation. For example, to see the weight of the 567th respondent, use the format

```
cdc[567,6]
```

which means we want the element of our dataset that is in the 567th row (meaning the 567th person or observation) and the 6th column (meaning weight). We know that `weight` is the 6th variable because it is the 6th entry in the list of variable names

```
names(cdc)
```

To see the weights for the first 10 respondents we can type

```
cdc[1:10,6]
```

In this expression, we have asked just for rows in the range 1 through 10. R uses the “:” to create a range of values, 1:10 expanding to 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. You can see this by entering

```
1:10
```

Finally, if we want all of the data for the first 10 respondents, you type

```
cdc[1:10,]
```

By leaving out an index or a range (we didn’t type anything between the comma and the square bracket), we get all the columns. Similarly, if we leave out an index or range for the rows, we would access all the observations, not just the 567th, or rows 1 through 10. Try the following to see all of the weights for all 20,000 respondents fly by on your screen

```
cdc[,6]
```

Again, this is asking for all the data from variable or column 6, `weight`. An alternative method to access the weight data is by referring to the name. Previously, we typed `names(cdc)` and saw all the variables contained in the dataset. We can use any of these to select items in our dataset.

```
cdc$weight
```

The dollar-sign tells R to look in dataframe `cdc` for the column called `weight`. Since that's a single vector, we can subset it with just a single index inside square brackets. We see the weight for the 567th respondent by typing

```
cdc$weight[567]
```

Similarly, for just the first 10 respondents

```
cdc$weight[1:10]
```

Both row-and-column notation and dollar-sign notation are widely used, which one you choose to use depends on your personal preference. While we will try to teach you only as much about the R language as you absolutely need, we thought we should spend some time with these various ways to access information in a dataset. It's a fundamental concept that we'll use frequently throughout the labs.

A little more on subsetting

While we have seen how to pull out the first 10 respondents from a dataset or maybe just a single variable, we often want to extract portions of a dataset based on the values themselves. We'll do this in stages. First, consider expressions like

```
cdc$gender == "m"
```

or

```
cdc$age > 30
```

If you type these, you will see a series of TRUE and FALSE values. There should be 20,000 of them, and each is true or false depending on whether the corresponding respondent is over age 30 or is a male.

Now, suppose we want to extract just the data for the men in the sample, or just for those over 30. We can use the R command `subset` to do that for us. For example, the command

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

will give us the complete data for just the men, and create a new dataset called `mdata`. In addition to finding it in your workspace alongside its dimensions, you can take a peek at the first several rows as usual

```
head(mdata)
```

Notice it contains all the same variables but just under half the rows. It is also possible to tell R to not keep all the variables, but just a few. We'll get to that later. For now, the important thing is that we can carve up the data based on values of one or more variables.

As an aside, you can use several of these conditions together with & and |; the & is read “and” so that

```
m_and_over30 <- subset(cdc, cdc$gender == "m" & cdc$age > 30)
```

will give you the data for men over the age of 30; while | is read “or” so that

```
m_or_over30 <- subset(cdc, cdc$gender == "m" | cdc$age > 30)
```

will take people who are either men or over the age of 30 (why that's an interesting group is hard to say, but right now the mechanics of this are the important thing). In principle, you can have as many “and” and “or” clauses as you like when forming a subset.

Exercise 4 What is the R command that will create a new object called `under23_and_smoke` that contains all observations of respondents under the age of 23 that have smoked 100 cigarettes in their lifetime?

Quantitative data

With our subsetting tools in hand, we'll now return to the task of the day: making basic summaries of the BRFSS questionnaire. We've already looked at categorical data such as `smoke` and `gender` so now let's turn our attention to quantitative data. Two common ways to visualize quantitative data are with boxplots and histograms. We can construct a boxplot for a single variable with the following command.

```
boxplot(cdc$height)
```

You can compare the locations of the components of the box by examining the summary statistics.

```
summary(cdc$height)
```

Confirm that the median and upper and lower quartiles reported in the numerical summary match those in the graph. The purpose of a boxplot is to provide a thumbnail sketch of a variable for the purpose of comparing across several categories. So we can, for example, compare the heights of men and women with

```
boxplot(cdc$height ~ cdc$gender)
```

The notation here is new. The `~` can be read “versus” or “as a function of”. So we're asking R to give us a boxplots of heights where the groups are defined by gender.

Next let's consider a new variable that doesn't show up directly in this data set: Body Mass Index (BMI). BMI is a weight to height ratio and can be calculated as.

$$BMI = \frac{\text{weight (lb)}}{\text{height (in)}^2} * 703^\dagger$$

The following two lines first make a new object called `bmi` and then creates boxplots of these values, defining groups by the variable `cdc$genhlth`.

```
bmi <- (cdc$weight / cdc$height^2) * 703
boxplot(bmi ~ cdc$genhlth)
```

Notice that the first line above is just some arithmetic, but it's applied to all 20,000 numbers in the `cdc` dataset. That is, for each of the 20,000 participants, we take their weight, divide by their height-squared and then multiply by 703. The result is 20,000 BMI values, one for each respondent. This is one reason why we like R; it lets us perform computations like this using very simple expressions.

Exercise 5 What does this boxplot show? Pick another categorical variable from the dataset and see how it relates to BMI. List the variable you chose, why you might think it would have a relationship to BMI and indicate what the figure seems to suggest.

Finally, let's make some histograms. We can have a look at the histogram for the age of our respondents with the command

```
hist(cdc$age)
```

Histograms are generally a very good way to see the shape of a single distribution, but that shape can change depending on how the data is split between the different bins. You can control the number of bins by adding an argument to the command. In the next two lines, we first make a default histogram of `bmi` and then one with 50 breaks.

```
hist(bmi)
hist(bmi, breaks = 50)
```

Note that you can flip between plots that you've created by clicking the forward and backward arrows in the lower right region of RStudio, just above the plots. How do these two histograms compare?

At this point, we've done a good first pass at analyzing the information in the BRFSS questionnaire. We've found an interesting association between smoking and gender and we can say something about the relationship between people's assessment of their general health and their

[†]703 is the approximate conversion factor to change units from metric (meters and kilograms) to imperial (inches and pounds)

own BMI. We've also picked up essential computing tools - summary statistics, subsetting, and plots - that will serve us well throughout this course.

On Your Own

1. Make a scatterplot of weight versus desired weight. Describe the relationship between these two variables.
2. Let's consider a new variable: the difference between desired weight (`wtdesire`) and current weight (`weight`). Create this new variable by subtracting the two columns in the dataframe and assigning them to a new object called `wdiff`.
3. What type of data is `wdiff`? If an observation `wdiff` is 0, what does this mean about the person's weight and desired weight. What if `wdiff` is positive or negative?
4. Describe the distribution of `wdiff` in terms of its center, shape, and spread, including any plots you use. What does this tell us about how people feel about their current weight?
5. Using numerical summaries and a side-by-side box plot analyze if men tend to view this differently than women.
6. Now it's time to get creative with the code: Find the mean and standard deviation of `weight` and determine what percent of the weights are within one standard deviation of the mean.
7. What concepts from the textbook are covered in this lab? What concepts, if any, are not covered in the textbook? Have you seen these concepts elsewhere, e.g. lecture, discussion section, previous labs, or homework problems? Be specific in your answer.

Notes

This lab was adapted for OpenIntro by Andrew Bray and Mine Çetinkaya-Rundel from a lab written by Mark Hansen of UCLA Statistics.