**SEIS 631 Foundations of Data Analysis**

**Assignment 2**

In this assignment, you will gain insight into public health by generating simple graphical and numerical summaries of a data set collected by the Centers for Disease Control and Prevention (CDC). As this is a large data set, along the way you’ll also learn the indispensable skills of data processing and subsetting.

**Getting started**: 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, including the research questions that motivate the study and many interesting results 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 variables in this data set, we will work with a small subset. We begin by loading the data set of 20,000 observations into the R workspace. After launching RStudio, first make sure you are in the correct working directory and then enter either of the following two commands to load the data:

**load("cdc.RData")** or  **source("cdc.R")**

The data set 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 *data frame*, which is a term that we will be using henceforth as well. To view the names of the variables, type the command

**names(cdc)**

This returns 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. The exerany variable indicates whether the respondent exercised in the past month (1) or did not (0). Likewise, hlthplan indicates whether the respondent had some form of health coverage (1) or did not (0). The smoke100 variable indicates whether the respondent had smoked at least 100 cigarettes in her lifetime. The other variables record the respondent’s height in inches, weight in pounds as well as their desired weight, wtdesire, age in years, and gender.

|  |
| --- |
| **Question 1: How many cases (i.e. observations) and how many variables (i.e. attributes/features) are there in this data set?** |

|  |
| --- |
| **Question 2**: **What type of variable is genhlth (represented in the dataset)?** |

|  |
| --- |
| **Question 3: What type of variable is weight?** |

|  |
| --- |
| **Question 4**: **What type of variable is smoke100?** |

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 data frame 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 data set would mean flooding your screen. It’s better to take small peeks at the data with head, tail or the subsetting techniques that we will see next.

**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 function **summary**  
returns a numerical summary: minimum, first quartile, median, mean, second quartile, and maximum. For weight variable this is,

**summary(cdc$weight)**

R also functions like a 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 - 140**

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

**mean(cdc$weight)  
var(cdc$weight)  
median(cdc$weight)**

While it makes sense to describe a numerical variable like weight in terms of these statistics, what about categorical data? We would instead consider the sample frequency or relative frequency distribution. The function **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, type

**table(cdc$smoke100)**

or instead look at the relative frequency distribution by typing

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

Notice how R automatically divides all entries in the table by 20,000 in the command above. This is similar to something we observed in the last assignment; when we multiplied or divided a vector with a number, R applied that action across entries in the vectors. As we see above, this also works for tables. Next, we make a bar plot of the entries in the table by putting the table inside the **barplot** command.

**barplot(table(cdc$smoke100))**

Notice what we’ve done here! We’ve computed the table of cdc$smoke100 and then immediately applied the graphical function, barplot. 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.

|  |
| --- |
| **Question 5**: **Create a numerical summary for gender. How many males are in the sample?** |

|  |
| --- |
| **Question 6: Compute the relative frequency distribution of genhlth. 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. For example, to examine which participants have smoked across each gender, we could use the following

**gender\_smokers = table(cdc$gender, cdc$smoke100)**

Here, first create a new object, a table, called gender smokers (the contents of which we can see by typing gender smokers into the console). Recall that 1 indicates that a respondent has smoked at least 100 cigarettes. The rows refer to gender. To create a mosaic plot of this table, we would enter the following command

**mosaicplot(gender\_smokers)**

|  |
| --- |
| **Question 7: What does the mosaic plot reveal about smoking habits and gender?** |

**How R thinks about data:** We mentioned that R stores data in data frames, 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 data frame 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 data frame, we can use row-and-column notation. For example, to see the sixth variable of the 567th respondent, use the format

**cdc[567, 6]**

which means we want the element of our data set that is in the 567th (meaning the 567th person or observation) and the 6th column (in this case, 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, so 1:10 expands 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, 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. When starting out in R, this is a bit counterintuitive. As a rule, we omit the column number to see all columns in a data frame. 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 the weights for all 20,000 respondents fly by on your screen,

**cdc[, 6]**

Recall that column 6 represents respondents’ weight, so the command above reported all of the weights in the data set. An alternative method to access the weight data is by referring to the name. Previously, we typed names(cdc) to see all the variables contained in the cdc data set. We can use any of the variable names to select items in our data set.

**cdc$weight**

The dollar-sign tells R to look in data frame 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]**

The command above returns the same result as the cdc[1:10,6] command. Both row-and-column notation and dollar-sign notation are widely used, which one you choose to use depends on your personal preference.

**A little more on subsetting:** It’s often useful to extract all individuals (cases) in a data set that have specific characteristics. We accomplish this through *conditioning* commands. First, consider expressions like

**cdc$gender = = "m"**

or

**cdc$age > 30**

These commands produce a series of TRUE and FALSE values. There is one value for each respondent, where TRUE indicates that the person was male (via the first command) or older than 30 (second command).

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 function **subset** to do that for us. For example, the command

**mdata = subset(cdc, cdc$gender = = "m")**

will create a new data set called mdata that contains only the men from the cdc data set. (Note the double equal sign!). 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)**

This new data set contains all the same variables but just under half the rows. It is also possible to tell R to keep only specific variables, which is a topic we’ll discuss in the future. 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. The | character is read “or” so that

**m\_or\_over30 = subset(cdc, cdc$gender == "m" | cdc$age > 30)**

will take people who are men or over the age of 30. In principle, you may use as many “and” and “or ” clauses as you like when forming a subset.

|  |
| --- |
| **Question 8**: **Create a new object called under23 and smoke that contains all observations of respondents under the age of 23 that have smoked at least 100 cigarettes in their lifetime. Write the command you used to create the new object?** |

|  |
| --- |
| **Question 9: How many observations are in the subset under23 and smoke that you created in the previous exercise, i.e. how many people in the sample are under the age of 23 and have smoked at least 100 cigarettes in their lifetime?** |

**Numerical data:** With our subsetting tools in hand, we’ll now try to get 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 numerical data. Two common ways to visualize numerical data are with box plots and histograms. We can construct a box plot 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 ~ character can be read “versus” or “as a function of”. So we’re asking R to give us a box plots 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:



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

The following two lines first make a new object called bmi and then creates box plots 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 data set. 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.

|  |
| --- |
| **Question 10: Which of the following is *false* based on the box plot of BMI vs. general health?** (a) The median BMI is roughly 25 for all general health categories, and there is a slight increase in median BMI as general health status declines (from excellent to poor). (b) The IQR increases slightly as general health status declines (from excellent to poor). (c) Among people with excellent health, there are some with unusually low BMIs compared to the rest of the group. (d) The distributions of BMIs within each health status group is left skewed. |

|  |
| --- |
| **Question 11:** **Pick another categorical variable from the data set 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 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.

|  |
| --- |
| **Question 12: In the last assignment, when exploring how percentages of boys and girls born vary over time (two numerical variables) we used a scatterplot. Using the same tools (the plot function) make a scatterplot of weight versus desired weight. Based on the plot you made, which of the following is true about the relationship between weight and desired weight?** (a) moderately weak negative linear (b) moderately weak positive linear (c) moderately strong positive linear (d) moderately weak negative linear |

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 own BMI. We’ve also picked up essential computing tools – summary statistics, subsetting, and plots.

Save your script as a .R file by moving your cursor into your script, then selecting File-Save, or ctl+S.

**Submission:**

Submit the following two files (no need to compress into a folder):

* **Pdf or word document** with answers to all the questions asked in the assignment.
* **.R file** with relevant code. It is useful to add comments within the code with a pound sign.