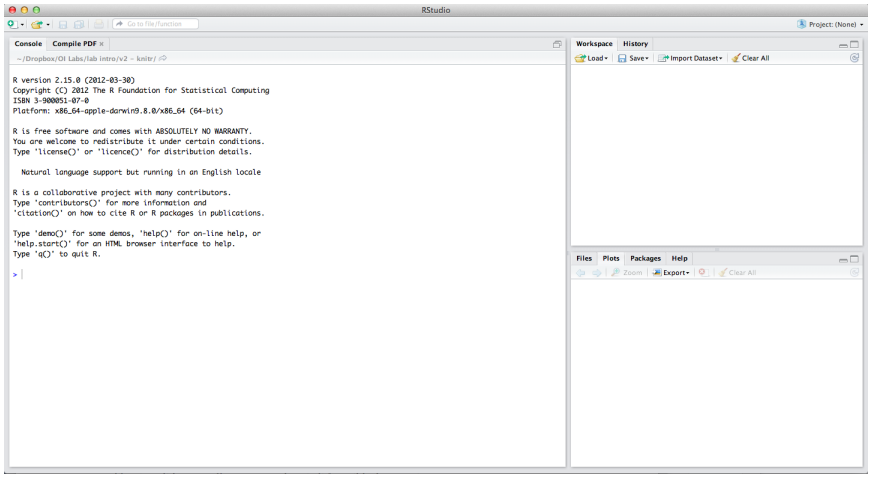
Complete all ***Exercises***, and submit answers to **Questions** on the Coursera platform. Note that the order of the choices in multiple choice questions may be different on the Coursera platform than the order in this document.

Lab 0 - Introduction to R and RStudio

The goal of this lab is to introduce you to R and RStudio, which you'll be using throughout the course both to learn the statistical concepts discussed in the texbook and also to analyze real data and come to informed conclusions. To straighten out which is which: R is the name of the programming language itself and RStudio is a convenient interface.

As the labs progress, you are encouraged to explore beyond what the labs dictate; a willingness to experiment will make you a much better programmer. Before we get to that stage, however, you need to build some basic fluency in R. Today we begin with the fundamental building blocks of R and RStudio: the interface, reading in data, and basic commands.



The panel in the upper right contains your *workspace* as well as a history of the commands that you've previously entered. Any plots that you generate will show up in the panel in the lower right corner.

The panel on the left is where the action happens. It's called the *console*. Everytime you launch RStudio, it will have the same text at the top of the console telling you the version of R that you're running. Below that information is the *prompt*. As its name suggests, this prompt is really a request, a request for a command. Initially, interacting with R is all about typing commands and interpreting the output. These commands and their syntax have evolved over decades (literally) and now provide what many users feel is a fairly natural way to access data and organize, describe, and invoke statistical computations.

To get you started, enter the following command at the R prompt (i.e. right after $>$ on the console). You can either type it in manually or copy and paste it from this document.

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

The data are stored in a data frame called `present`.

This command instructs R to access the OpenIntro website and fetch some data: the number of boys and girls born in the US each year. You should see that the workspace area in the upper righthand corner of the RStudio window now lists a data set called `present` that has 63 observations on 3 variables. As you interact with R, you will create a series of objects. Sometimes you load them as we have done here, and sometimes

Derivative of OpenIntro Lab 0.

you create them yourself as the byproduct of a computation or some analysis you have performed. Note that because you are accessing data from the web, this command (and the entire analysis) will work in a computer lab, in the library, or in your dorm room; anywhere you have access to the Internet.

Present Data

The present data set refers to the number of male and female births in the United States. The data set contains the data for all years from 1940 to 2002. We can take a look at the data by typing its name into the console.

```
present
```

What you should see are four columns of numbers, each row representing a different year: the first entry in each row is simply the row number (an index we can use to access the data from individual years if we want), the second is the year, and the third and fourth are the numbers of boys and girls born that year, respectively. Use the scrollbar on the right side of the console window to examine the complete data set.

Note that the row numbers in the first column are not part of the present data set. R adds them as part of its printout to help you make visual comparisons. You can think of them as the index that you see on the left side of a spreadsheet. In fact, the comparison to a spreadsheet will generally be helpful. R has stored the data in a kind of spreadsheet or table called a *data frame*.

You can see the dimensions of this data frame by typing:

```
dim(present)
```

This command should output `[1] 63 3`, indicating that there are 63 rows and 3 columns (we'll get to what the `[1]` means in a bit), just as it says next to the object in your workspace. You can see the names of these columns (or variables) by typing:

```
names(present)
```

Question 1 [MULTIPLE CHOICE] How many variables are included in this data set?

(a) 2

(b) 3

(c) 4

(d) 63

(e) 2002

Exercise What years are included in this dataset? *Hint:* View the `year` variable to answer this question.

You should see that the data frame contains the columns `year`, `boys`, and `girls`. At this point, you might notice that many of the commands in R look a lot like functions from math class; that is, invoking R commands means supplying a function with some number of arguments. The `dim` and `names` commands, for example, each took a single argument, the name of a data frame.

One advantage of RStudio is that it comes with a built-in data viewer. Click on the name `present` in the upper right window that lists the objects in your workspace. This will bring up an alternative display of the counts in the upper left window. You can close the data viewer by clicking on the "x" in the upper lefthand corner.

Some Exploration

Let's start to examine the data a little more closely. We can access the data in a single column of a data frame separately using a command like

```
present$boys
```

This command will only show the number of boys born each year.

> **Question 2** [MULTIPLE CHOICE] What command would you use to view just the counts of girls born?
>
> (a) present$boys
>
> (b) present$girls
>
> (c) girls
>
> (d) present[girls]
>
> (e) $girls

Notice that the way R has printed these data is different. When we looked at the complete data frame, we saw 63 rows, one on each line of the display. These data are no longer structured in a table with other variables, so they are displayed one right after another. Objects that print out in this way are called *vectors*; they represent a set of numbers. R has added numbers in [brackets] along the left side of the printout to indicate locations within the vector. For example, `1211684` follows `[1]`, indicating that `1211684` is the first entry in the vector. And if `[43]` starts a line, then that would mean the first number on that line would represent the 43rd entry in the vector.

R has some powerful functions for making graphics. We can create a simple plot of the number of girls born per year with the command

```
plot(x = present$year, y = present$girls)
```

By default, R creates a scatterplot with each x,y pair indicated by an open circle. The plot itself should appear under the "Plots" tab of the lower right panel of RStudio. Notice that the command above again looks like a function, this time with two arguments separated by a comma. The first argument in the plot function specifies the variable for the x-axis and the second for the y-axis. If we wanted to connect the data points with lines, we could add a third argument, the letter "l" for line.

```
plot(x = present$year, y = present$girls, type = "l")
```

Question 3 [MULTIPLE CHOICE] Is there an apparent trend in the number of girls born over the years? How would you describe it?

(a) There appears to be no trend in the number of girls born from 1940 to 2002.

(b) There is initially an increase in the number of girls born, which peaks around 1960. After 1960 there is a decrease in the number of girls born, but the number begins to increase again in the early 1970s. Overall the trend is an increase in the number of girls born in the US since the 1940s.

(c) There is initially an increase in the number of girls born. This number peaks around 1960 and then after 1960 the number of girls born decreases.

(d) The number of girls born has decreased over time.

(e) There is an initial increase in the number of girls born but this number appears to level around 1960 and not change since then.

You might wonder how you are supposed to know that it was possible to add that third argument. Thankfully, R documents all of its functions extensively. To read what a function does and learn the arguments that are available to you, just type in a question mark followed by the name of the function that you're interested in. Try the following.

```
?plot
```

Notice that the help file replaces the plot in the lower right panel. You can toggle between plots and help files using the tabs at the top of that panel.

Now, suppose we want to plot the total number of births. To compute this, we could use the fact that R is really just a big calculator. We can type in mathematical expressions like

```
1211684 + 1148715
```

to see the total number of births in 1940. We could repeat this once for each year, but there is a faster way. If we add the vector for births for boys and girls, R will compute all sums simultaneously.

```
present$boys + present$girls
```

What you will see are 63 numbers (in that packed display, because we aren't looking at a data frame here), each one representing the sum we're after. Take a look at a few of them and verify that they are right. Therefore, we can make a plot of the total number of births per year with the command

```
plot(present$year, present$boys + present$girls, type = "l")
```

This time, note that we left out the names of the first two arguments. We can do this because the help file shows that the default for `plot` is for the first argument to be the x-variable and the second argument to be the y-variable.

Question 4 [MULTIPLE CHOICE] In what year did we see the most total number of births in the U.S.?

(a) 1961

(b) 1960

(c) 1957

(d) 1991

Hint: You can refer to the help files or the R reference card (http://cran.r-project.org/doc/contrib/Short-refcard.pdf) to find helpful commands. For example, you might want to look into the `which.max` *function.*

Similarly to how we computed the proportion of boys, we can compute the ratio of the number of boys to the number of girls born in 1940 with

```
1211684/1148715
```

or we can act on the complete vectors with the expression

```
present$boys/present$girls
```

The proportion of newborns that are boys

```
1211684/(1211684 + 1148715)
```

or this may also be computed for all years simultaneously:

```
present$boys/(present$boys + present$girls)
```

Note that with R as with your calculator, you need to be conscious of the order of operations. Here, we want to divide the number of boys by the total number of newborns, so we have to use parentheses. Without them, R will first do the division, then the addition, giving you something that is not a proportion.

Question 5 [TRUE / FALSE] Now, make a plot of the proportion of boys over time, and based on the plot determine if the following statement is true or false: The proportion of boys born in the US has decreased over time.

- TRUE
- FALSE

Tip: If you use the up and down arrow keys, you can scroll through your previous commands, your so-called command history. You can also access it by clicking on the history tab in the upper right panel. This will save you a lot of typing in the future.

Finally, in addition to simple mathematical operators like subtraction and division, you can ask R to make comparisons like greater than, $>$, less than, $<$, and equality, $==$. For example, we can ask if boys outnumber girls in each year with the expression

```
present$boys > present$girls
```

This command returns 63 values of either `TRUE` if that year had more boys than girls, or `FALSE` if that year did not (the answer may surprise you). This output shows a different kind of data than we have considered so far. In the `present` data frame our values are numerical (the year, the number of boys and girls). Here, we've asked R to create *logical* data, data where the values are either `TRUE` or `FALSE`. In general, data analysis will involve many different kinds of data types, and one reason for using R is that it is able to represent and compute with many of them.

Question 6 [MULTIPLE CHOICE] Which of the following is true?

(a) Every year there are more girls born than boys.

(b) Every year there are more boys born than girls.

(c) Half of the years there are more boys born, and the other half more girls born.

Now try to answer the following questions without additional guidance on the coding, based on what you have learned so far.

Question 7 [MULTIPLE CHOICE] Make a plot that displays the boy-to-girl ratio for every year in the data set. What do you see?

(a) There appears to be no trend in the boy-to-girl ratio from 1940 to 2002.

(b) There is initially an increase in boy-to-girl ratio, which peaks around 1960. After 1960 there is a decrease in the boy-to-girl ratio, but the number begins to increase in the mid 1970s.

(c) There is initially a decrease in the boy-to-girl ratio, and then an increase between 1960 and 1970, followed by a decrease.

(d) The boy-to-girl ratio has increased over time.

(e) There is an initial decrease in the boy-to-girl ratio born but this number appears to level around 1960 and remain constant since then

Question 8 [MULTIPLE CHOICE] Calculate absolute differences between number of boys and girls born in each year, and determine which year out of the present data had the biggest absolute difference in the number of girls and number of boys born?

(a) 1963

(b) 1946

(c) 2002

(d) 1940

These data come from a report by the Centers for Disease Control (*http://www.cdc.gov/nchs/data/nvsr/nvsr53/nvsr53_20.pdf*). Check it out if you would like to read more about an analysis of sex ratios at birth in the United States.

That was a short introduction to R and RStudio, but we will provide you with more functions and a more complete sense of the language as the course progresses. Feel free to browse around the websites for R *http://www.r-project.org* and RStudio *http://rstudio.org* if you're interested in learning more, or find more labs for practice at *http://openintro.org*.

End of Lab Survey

The following questions are not graded, but your feedback is very much appreciated and immensely useful for the development of the course.

Question 9 [MULTIPLE CHOICE] This lab covered material that is covered in the class.

(a) Strongly Disagree

(b) Disagree

(c) Neutral

(d) Agree

(e) Strongly Agree

Question 10 [MULTIPLE CHOICE] The lab improved your understanding of these topics.

(a) Strongly Disagree

(b) Disagree

(c) Neutral

(d) Agree

(e) Strongly Agree

Question 11 [MULTIPLE CHOICE] The instructions were clear and it was easy to understand what was wanted.

(a) Strongly Disagree

(b) Disagree

(c) Neutral

(d) Agree

(e) Strongly Agree

Question 12 [MULTIPLE CHOICE] The data were relevant and interesting to me.

(a) Strongly Disagree

(b) Disagree

(c) Neutral

(d) Agree

(e) Strongly Agree

Question 13 [MULTIPLE CHOICE] The length of time took to complete lab.

(a) Less than 30 minutes

(b) Between 30 minutes and 1 hour

(c) Between 1 hour and 2 hours

(d) More than 2 hours