Homework 0

Due Thursday September 7

Homework Instructions: This homework is due by 11:58pm of the assigned due date and must be handed in as a pdf file via Canvas. No late homeworks are allowed unless there is a note from a medical professional.

The goal of this homework is to introduce you to R and RStudio, which you'll be using throughout the course both to learn the statistical concepts discussed in class 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.

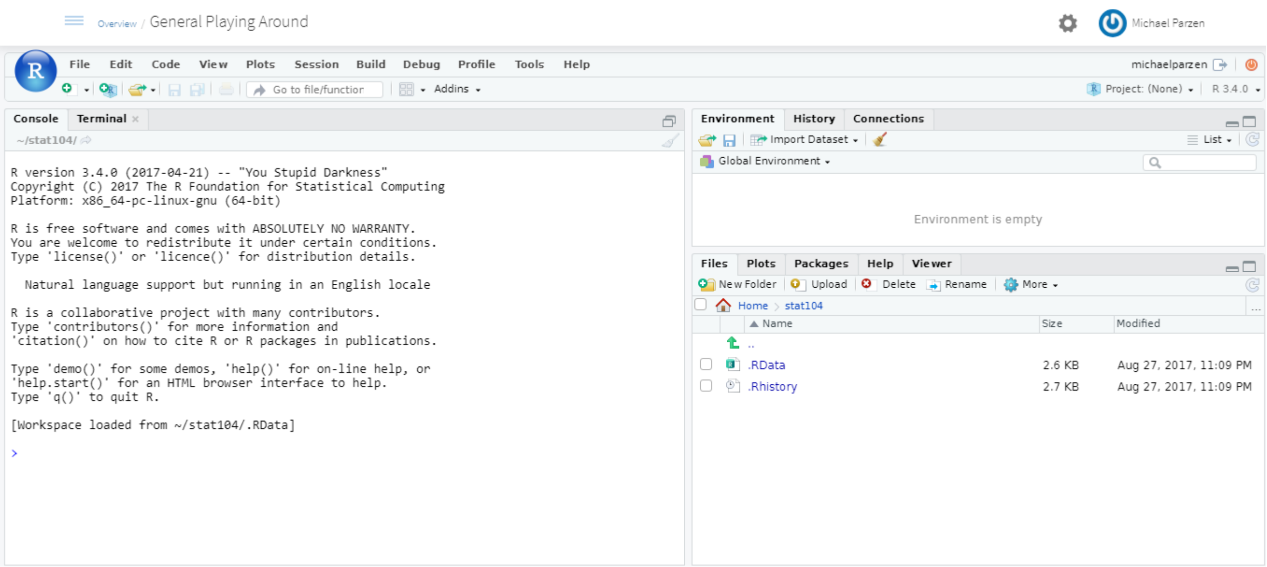
As the course progresses, you are encouraged to explore beyond what the homeworks dictate; a willingness to experiment will make you a much better user of statistics. Before we get to that stage, however, you need to build some basic fluency in R. Today we begin with the fundamental building blocks of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

Caution

Please read the Introduction to R and RStudio guide found on the class website before starting this assignment.

The Rstudio Layout

Recall from the Introductory Guide the initial set up of RStudio:


Overview / General Playing Around
Michael Parzen
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
michaelparzen
Project: (None)
R 3.4.0
Console
Terminal
~/stat104/
Environment History Connections
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Upload Delete Rename More
Home > stat104

```
R version 3.4.0 (2017-04-21) -- "You Stupid Darkness"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/stat104/.RData]

>
```

▲ Name	Size	Modified
..		
.RData	2.6 KB	Aug 27, 2017, 11:09 PM
.Rhistory	2.7 KB	Aug 27, 2017, 11:09 PM

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 (you might have to click on the Plots tab).

The panel on the left is where the action happens. Its 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 youre 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.

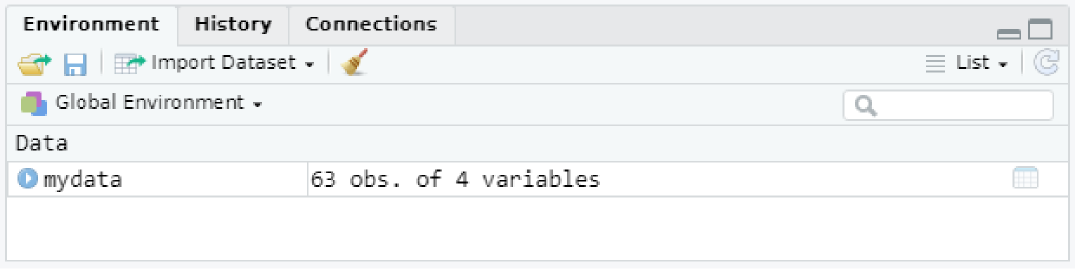
Reading in Data

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

Enter the following command into Rstudio

```
mydata=read.csv("http://tinyurl.com/birthdata1")
```

This command instructs R to fetch some data from the internet and store it in a matrix called **mydata**. The data consists of 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 mydata that has 63 observations on 3 variables.



```
mydata          63 obs. of 4 variables
```

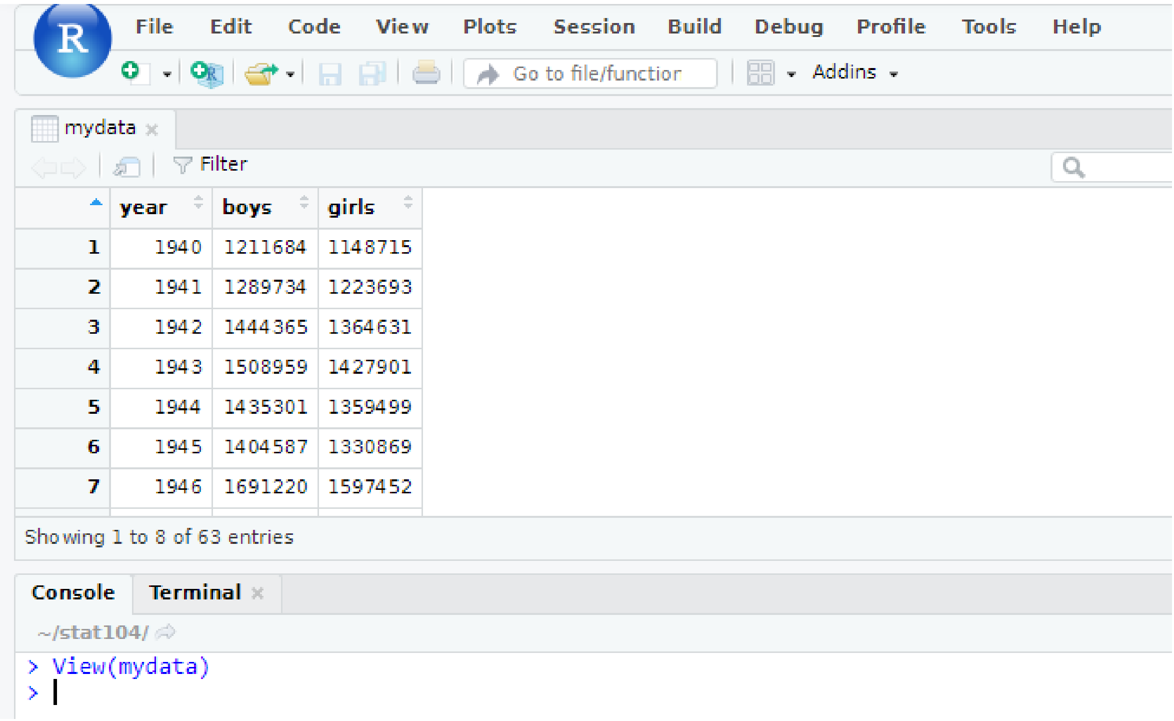
The data set refers to the number of male and female births in the United States for about a 60 year period. We can take a look at the data in several ways.

You can examine the data withe the `View` command.

Enter the following command into Rstudio

```
View(mydata)
```

A new upper left window opens up to show us the data.



	year	boys	girls
1	1940	1211684	1148715
2	1941	1289734	1223693
3	1942	1444365	1364631
4	1943	1508959	1427901
5	1944	1435301	1359499
6	1945	1404587	1330869
7	1946	1691220	1597452

Showing 1 to 8 of 63 entries

```
~/stat104/
> View(mydata)
> 
```

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 using the `dim` command.

Enter the following command into Rstudio

```
dim(mydata)
```

This command should output `[1] 63 3`, indicating that there are 63 rows and 3 columns (well get to what the `[1]` means in a bit), just as it says next to the object in your workspace.

The `names` command lets us see the names of the variables in the dataset.

Enter the following command into Rstudio

```
names(mydata)
```

You should obtain the following output

```
> names(mydata)
[1] "year"  "boys"  "girls"
```

As you interact with R, you will create a series of objects. Sometimes you load them as we have done here, and sometimes 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.

Exercise 1 What is the total number of observations in this dataset?

Exercise 2 What range years are included in this dataset?

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.

Some Exploration

Lets 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 of the form `dataframe$varname`. As an example, the command below will only show the number of boys born each year.

Enter the following command into Rstudio

```
mydata$boys
```

You should obtain the following output

```
> mydata$boys
 [1] 1211684 1289734 1444365 1508959 1435301 1404587 1691220 1899876 1813852 1826352
[11] 1823555 1923020 1971262 2001798 2059068 2073719 2133588 2179960 2152546 2173638
[21] 2179708 2186274 2132466 2101632 2060162 1927054 1845862 1803388 1796326 1846572
[31] 1915378 1822910 1669927 1608326 1622114 1613135 1624436 1705916 1709394 1791267
[41] 1852616 1860272 1885676 1865553 1879490 1927983 1924868 1951153 2002424 2069490
[51] 2129495 2101518 2082097 2048861 2022589 1996355 1990480 1985596 2016205 2026854
[61] 2076969 2057922 2057979
```

Exercise 3 What command would you use to extract just the counts of girls born?

(a) mydata$boys
(b) mydata$girls
(c) girls
(d) mydata[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 many built mathematical and statistical functions built in. We can take the mean (average) of the boy births as follows.

Enter the following command into Rstudio

```
mean(mydata$boys)
```

You should obtain the following output

```
> mean(mydata$boys)
[1] 1885600
```

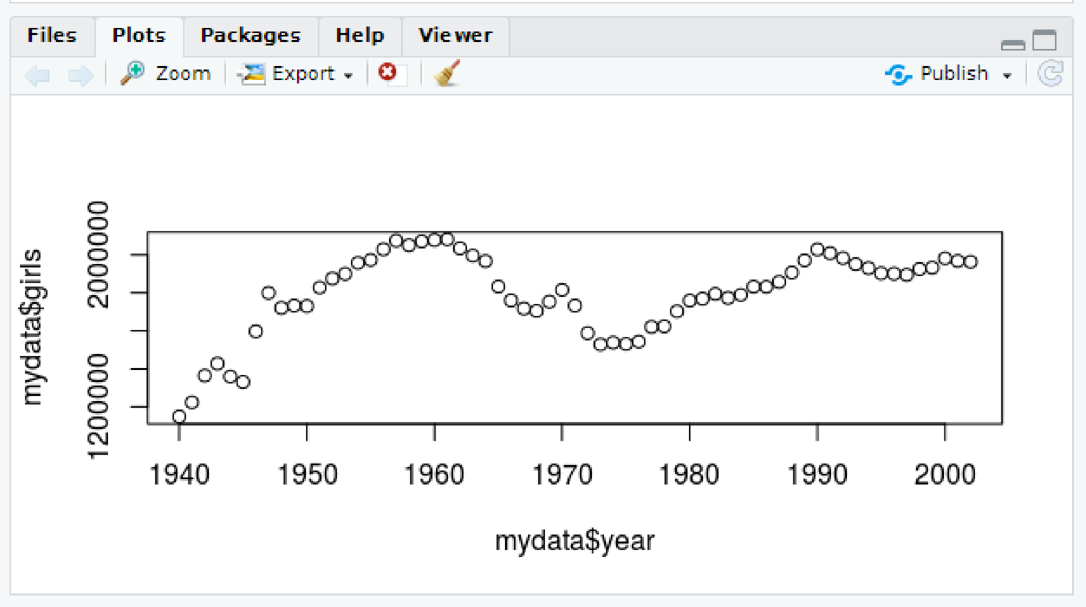
Exercise 4 What is the mean (average) of the girl births?

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

Enter the following command into Rstudio

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

You should obtain the following output under the Plots tab of the lower right panel of RStudio.

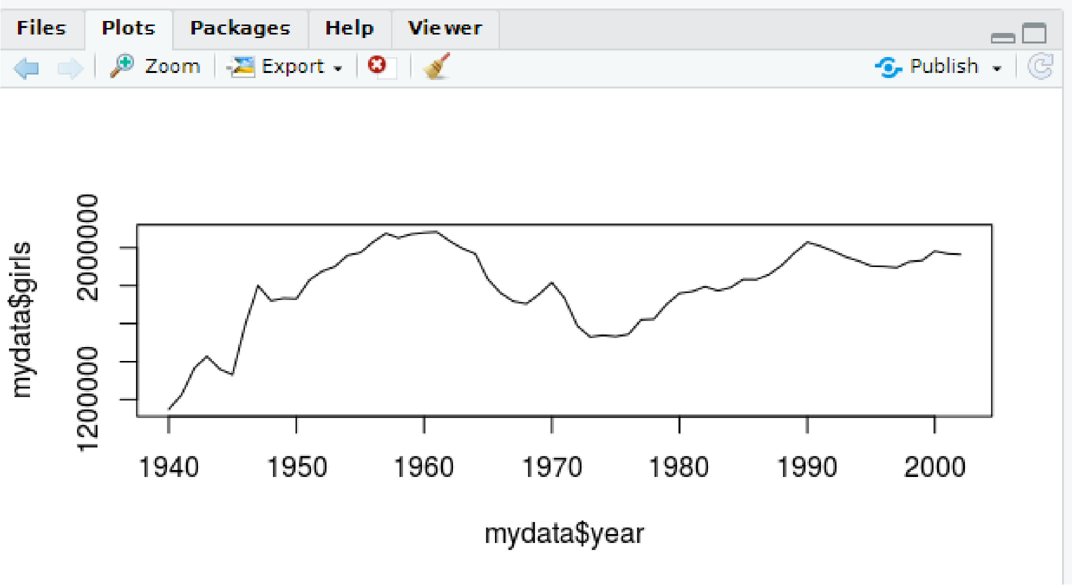


By default, R creates a scatterplot with each x,y pair indicated by an open circle. Notice that the command above again looks like a function, this time with two arguments separated by a comma. The first argument in the plot function specifies the variable for the x-axis and the second for the y-axis.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

Enter the following command into Rstudio

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

You should obtain the following output under the Plots tab of the lower right panel of RStudio.



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 youre interested in. Try the following.

Enter the following command into Rstudio

```
?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.

Exercise 5 Is there an apparent trend in the number of girls born over the years? How would you describe it? Answer with just a few sentences.

Exercise 6 Create a similar time plot for the boy data using connected lines. See the Introduction to R Guide to learn how to cut and paste graphs into Word.

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.

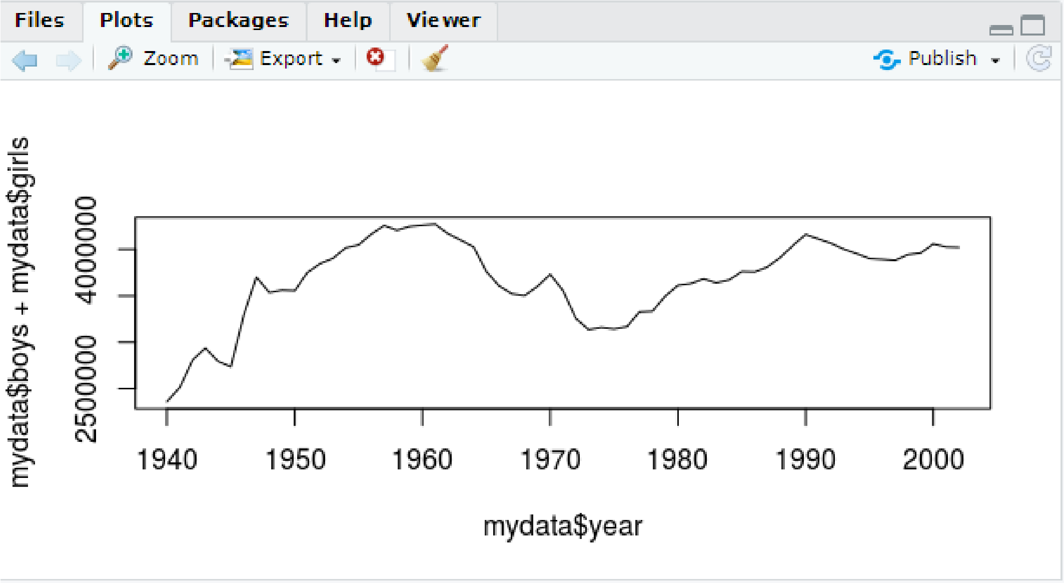
Enter the following command into Rstudio

```
mydata$boys + mydata$girls
```

What you will see are 63 numbers (in that packed display, because we arent looking at a data frame here), each one representing the sum were 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 as follows.

Enter the following command into Rstudio

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



Exercise 7 What is the minimum of the total number of births? Which year did this

happen? (the min function could be helpful here).

Exercise 8 What is the maximum of the total number of births? Which year did this happen? (the max function could be helpful here).

One can easily define new variables in R. Consider the following

Enter the following command into Rstudio

```
totalbirths=mydata$boys + mydata$girls
length(totalbirths)
```

```
> totalbirths=mydata$boys + mydata$girls
> length(totalbirths)
[1] 63
```

This creates a new variable which is a vector of length 63 containing the total births for each year.

We can compute the percentage of newborns that are boys with the following expression.

Enter the following command into Rstudio

```
perboys=mydata$boys/(mydata$boys+mydata$girls)
```

Exercise 9 What are the max and min of the proportion of boys born over time?

Exercise 10 Make a plot of the proportion of boys over time.

Exercise 11 Briefly comment on the plot of the proportion of boys over time.

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 following expression.

Enter the following command into Rstudio

```
mydata$boys > mydata$girls
```

You should obtain the following output

```
> mydata$boys > mydata$girls

 [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[18] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[35] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[52] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

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, weve 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.

Exercise 12 Which statement 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.

Exercise 13 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.

Exercise 14 Calculate absolute differences between number of boys and girls born in each year (you may need the R command `abs`) and determine which year out of the present data had the biggest absolute difference in the number of girls and number of boys born.