How To Remove Observations and Access Subsets of Data

To this point in this course, we have run analyses on entire variables, such as calculating the mean of a variable or plotting the bivariate relationship between two quantitative variables. However, often we want to manipulate the variable in such a way that we are only running the analysis on a subset of the data. We may, for example, want to remove certain cases. We also may want to compare the values of one variable across different groups, such as the height of males and females and their distributions.

To do this, we must understand how the R program accesses subsets of data. It does so by specifying the conditions for analysis in brackets [ ].

Example #1:
Let's say that we were interested in the average height of students. As a first step, we simply calculate the mean for all students as a whole:

```
> mean(height, na.rm=T)
[1] 68.92958
```

But, now you decide that you want to compare the average height of male students and the average height of female students. This is done as follows:

```
>mean(height [gender==”Female”], na.rm=T)
[1] 65.75758

>mean(height [gender==”Male”], na.rm=T)
[1] 71.68421
```

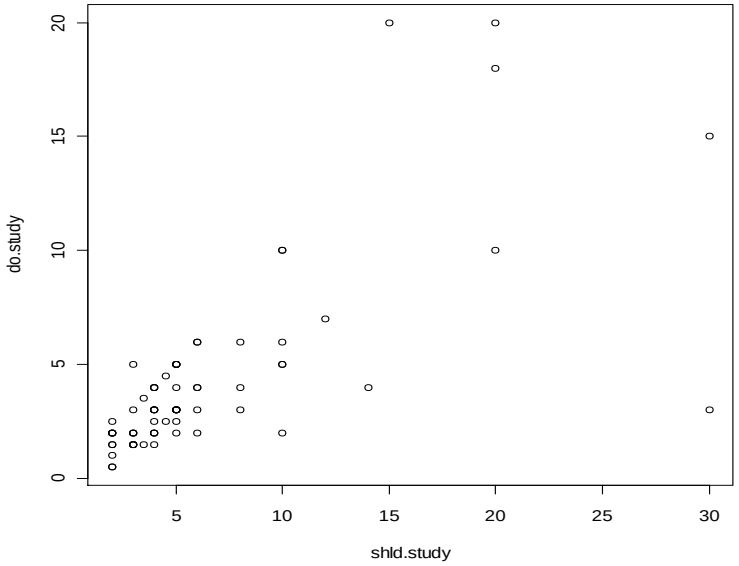
Notice a few things:

1. The brackets go immediately after the variable of analysis, in this case, height.
2. The two equal signs (i.e., = =) means “equal to” and you must have both of them. It will not work with just one equal sign.
3. Because gender is a categorical variable, the category names must be in parentheses. This is not the case if we are specifying groups based on a quantitative variable.
4. As is always the case, the labels are case sensitive: gender = = “male” without a capital M will not return the correct answer.
5. To see labels, the easiest way is to look at the data set with the fix( ) command, e.g., fix(POL201), find the variable and note the labels and their punctuation.

Example #2:
Let's say we are interested in plotting the relationship between the amount of time students report studying and the amount of time they think they *should* study for each course per week.

We do so as follows:

>plot(do.study ~ shld.study)



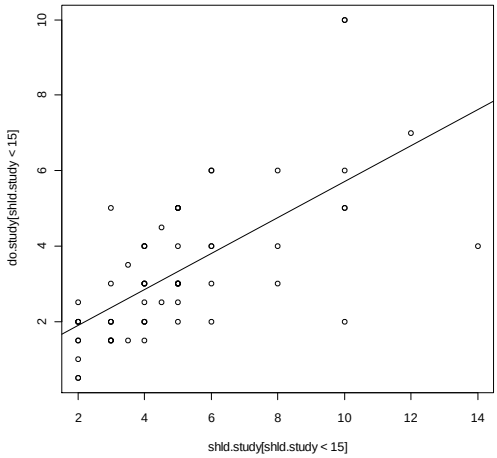
Now, looking at the above plot and thinking about the questions, we should find something is a bit fishy—namely, I highly doubt that any student really thinks they should have to study 30 hours per week *for each class*. Indeed, it appears that these students did not read the question correctly and may have assumed they were answering for all their courses combined. If so, this is a problem for understanding the true relationship between the amount of time students actually study and the amount of time they think they should study.

Thus, we may opt to remove these aberrant cases from the analysis. To do so, we must specify a condition in brackets that would remove those cases. In this case, I am going to assume that anyone who states that he or she studies more than 15 hours per week per course misread the question (Note that this is an arbitrary assumption. I could be incorrect, in which case I would want to keep them in the analysis!). We do this as follows:

>plot(do.study [shld.study < 15] ~ shld.study [shld.study < 15])

And to add the regression line:

>abline(lm(do.study [shld.study < 15] ~ shld.study [shld.study < 15]))



Again, notice a few things:

1. The qualifying condition in brackets must be used for both variables.
2. It also must be used when entering the follow-up regression line with the abline ( ) command.
3. Unlike for categorical variables, you do not put quotation marks around the qualifying value of interest.

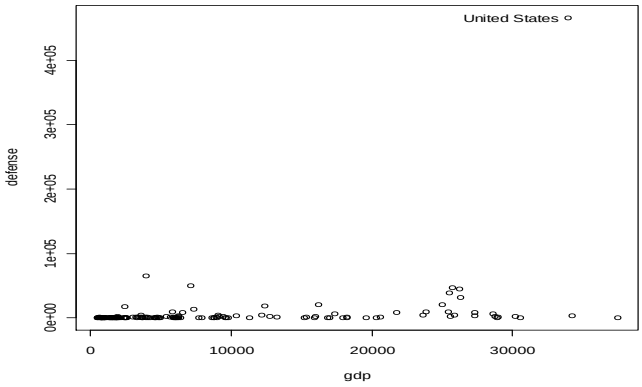
Example #3:

Sometimes we want to remove a specific single case from an analysis. There are a couple ways to do this. The first is with the "!=" operator, which means "not equal to".

For example, looking at the ICC data set, which can be downloaded as follows:

>ICC=read.csv("http://www.sfu.ca/~sweldon/ICC.csv")

We can examine the relationship between GDP per Capita and Defense spending. After attaching the data set, we enter the following command:
>plot(defense ~ gdp)



Notice that there is one extreme case, the United States, which spends more than the rest of the world *combined* on its military. This, in itself, is interesting, but equally important here is that it is also difficult to get a sense of the real relationship between GDP and Defense spending. Remember also that outliers can exert an extreme pull on the mean and distort results. Thus, we might want to remove the US from the graph.

>plot(defense [country != "United States"] ~ gdp [country != "United States"])

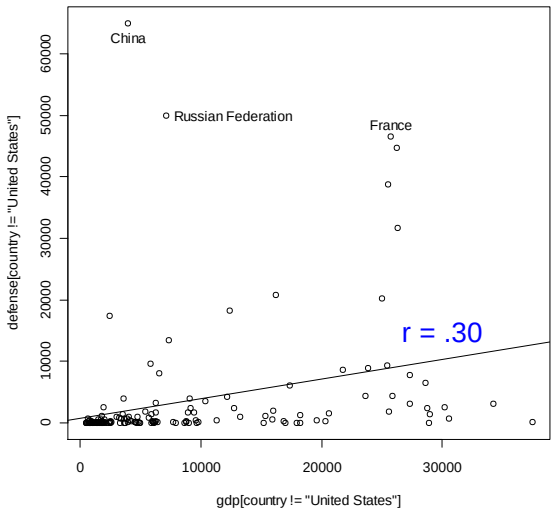
And to add the regression line:

>abline(lm(defense [country != "United States"] ~ gdp [country != "United States"]))

Similarly, to calculate the Pearson R correlation:

>cor.test(defense [country != "United States"], gdp [country != "United States"])

And, we get:



Pearson's product-moment correlation

data: defense[country != "United States"] and gdp[country != "United States"]
t = 3.8593, df = 149, p-value = 0.0001689
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.1488963 0.4399974
sample estimates:
cor

0.3014553

Entering Text into Scatterplots

By the way, to get the "r = .30" to appear on the graph, you need to use the text command, which takes the form: text(x , y, labels = " "), where x is the value on the x-axis and y is the value on the y-axis where you want the text to be placed. In this case,

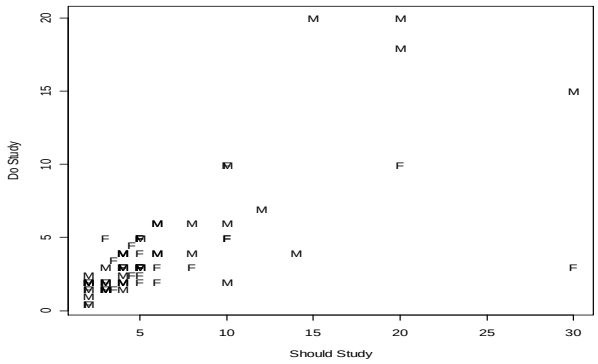
> text(30000, 15000, labels = "r = .30", col="blue", cex=2)

Example #4: How to run scatterplots broken down by group?

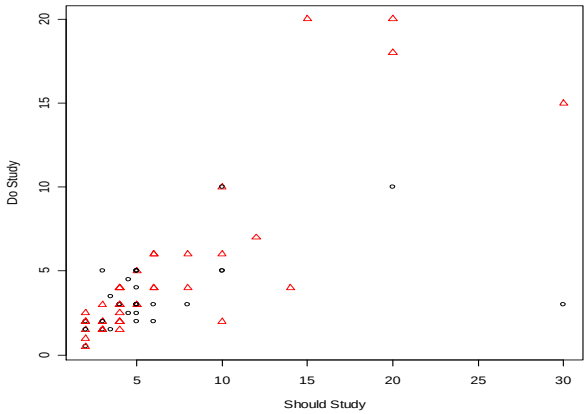
Often we want to further refine our scatterplots to compare two or more groups. For example, let's say we want to know how the relationship between the time students think they should study and the time that they actually do study varies between males and females.

Here are two basic ways of doing this:

> plot(do.study ~ shld.study, pch = as.character(gender))



> plot(do.study ~ shld.study, pch = as.numeric(gender), col = as.numeric(gender))



You should notice that both commands change the plotting character with the "pch = " argument so that it is possible to distinguish between males and females.

The first command uses it in combination with the as.character command to plot the characters M and F to correspond to Males and Females, whereas the second one uses it combination with the as.numeric command to plot colored triangles and circles to correspond to the points on the graph. The first is more useful here in this example. The latter is useful in a wider variety of situations.

To add respective regression lines for each gender, we must specify these conditions in our abline commands:

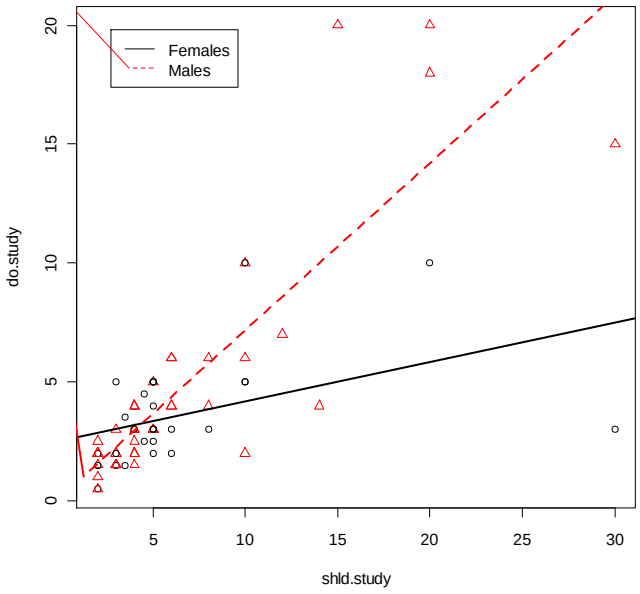
```
> abline(lm(do.study [gender == "Female"] ~ shld.study [gender == "Female"]),
+ col="black", lwd = 2, lty = 1)

> abline(lm(do.study [gender == "Male"] ~ shld.study [gender == "Male"]),
+ col="red", lwd = 2, lty = 2)
```

Note that "lwd = " changes the thickness of the plot line, while "lty = " changes the type of line that is plotted.

And to add a *legend* to the graph so that readers can differentiate between the two groups:

```
> legend(locator(1), legend=c("Females", "Males"), lty = c(1,2), col = c("black", "red"))
```

Note, just like the "identify ( )" command, you must click on your graph to display the legend.



Comparison Operators:
In specifying subsets of data for analysis, there are 6 types of comparison operators that can be used, several of which you saw above. These are:

- = = "is equal to" (yes, 2 equal signs)
- ! = "is not equal to"
- < "is less than"
- <= "is less than or equal to"
- > "is greater than"
- >= "is greater than or equal to"

Combining Expressions:

Though this can get messy, sometimes we want to specify more than one condition in plotting data. To do this we must combine expression with our logical operators.

For example, to plot the relationship between actual study time and should study time just for women and removing those who potentially answered the question for all of their classes, rather than each class, we need to do the following:

```
> plot(do.study [gender == "Female" & shld.study < 15] ~
+ shld.study [gender == "Female" & shld.study < 15] )
```

Notice that we use "&" in order combine two conditions. It stands for "and".

We also may want to combine groups. We can do this with the | symbol, which is the symbol above the "Enter key" and below the "Backspace key" (at least on my computer). This symbol stands for "or".

For example, let's say that we want to run a boxplot on the GPA of *all* smokers. Remember that currently the data set differentiates between Social and Regular smokers.

```
>boxplot(gpa [smoke == "Socially" | smoke == "Regularly"])
```

A Final Word of Caution:

Accessing subsets of data is one of the more difficult tasks to master in R. In order to really master this, you must develop a thorough understanding of how R is organized, how it accesses data, and how this varies across types of situations and commands. I have given you very simple examples here and these are the building blocks of mastering this task. Learn to carefully inspect your work and you will get better at recognizing problems and making sure you have done tasks correctly.