Exercise 1: Diagram and model hypotheses in R

DATA 306

This exercise has two parts. In the first part, you will follow along with my code and explanations and give some interpretations. In part 2, you will customize my code with some small modifications.

We are starting this semester's work in the middle. In this exercise, you will try out some basic regression modeling in R. Normal statistics/data analysis courses reserve this for the last half of the semester. But don't panic! All you need to do is follow along and/or slightly tweak the R code I supply here.

I have two reasons for starting in the middle. One is to demonstrate how easy it is to push the buttons that will run "data analysis and models" for you *as long as you know what buttons to push*. The other is showing how difficult it can be to understand what the outputs *mean* and *where to go next*. Each of these reasons has its own set of risks and rewards. We will be considering and reconsidering them throughout the course.

Part 1: Follow along

The data we'll analyze is something I cobbled together from `https://opendata.cityofnewyork.us/`. It combines three sources of data on the NYC public school system: (1) teacher, student, and parent perceptions of student belonging at school (from three separate surveys); (2) class size data; and (3) various school demographics. The result is `NYCschools_r1.csv`, a table with 24 variables (columns) for 469 NYC public schools (rows).

This exercise introduces a lot of new functions in R. Any time you need help remembering what something does, it's a good idea to use the most important function of all: `?`, AKA the "help" function.

Using the `?` function is easy. If I want to know what `str()` does, I type `?str`. The help page will appear, explaining the function, its arguments, related functions, references, and (most helpful) examples. Get in the habit of using `?` *before* you do a Google search, ChatGPT prompt, etc. If nothing else, the help pages will give you the vocabulary you need to ask the right questions.

1. We have to put the data somewhere easy to find, then tell R where it is. Let's say you have a nicely organized folder called `DATA306`, such as the one we set up in class. If you are on a MacOS, I will assume you saved it on your Desktop. If you are on a Windows OS, I will assume you saved it in the `Documents` folder.

   Inside the `DATA306` folder, let's further assume you have another folder called `data` where you saved your `NYCschools_r1.csv` dataset when you downloaded it from the course webpage (again, we set this up in class). With this scheme, the steps on the "file path" to the NYC schools data are the following:

   (a) Go to the `Desktop` (Mac OS) or `Documents` (Windows)

   (b) Open the `DATA306` folder
   (c) Open the `data` folder inside `DATA306`
   (d) Open `NYCschools_r1.csv` in R

   To communicate this file path to your computer, write out the name of each folder (AKA directory) and separate them with slashes, starting with the highest-level directory. So the path above is:

   `~/Desktop/DATA306/data/NYschools_r1.csv`

   in MacOS. Or, for Windows users:

   `~/DATA306/data/NYschools_r1.csv`

   As discussed in class, in MacOS, the tilde (~) means "user's home directory." In Windows, it usually means "user's Documents directory." To find out what it means on your machine, write the following line:

   ```
   setwd("~"); getwd()
   ```

   It's worth finding out what the tilde means because it will save you keystrokes. Keystrokes are little moments in your life you will never get back.

   Now that we know the file path, you're ready to configure your own setup (if you have not done so already):

   - make a `DATA306` folder
   - save `NYCschools_r1.csv` in a folder called `data` within the `DATA306` folder
   - In R, create a new script called `e1_lastnameFirstname_r1.R` and save it in a new folder called `scripts` within the `DATA306` folder[1]

2. Within the script, use `#` to start what are called "comment" lines. Make some comment lines for your name, the date, and other metadata for the script. Something like the following will work:

   ```
   ## Firstname Lastname
   ## Exercise 1: Diagram and test hypotheses in R
   ## DATA 306
   ## 9/15/2023
   ### Part 1: Following along
   ```

   Remember, when you write scripts your most important audience is future you.

3. Once you have your directories set up and `NYCschools_r1.csv` downloaded to the right place, we can load the NYC schools dataset into R. I've broken this down into three steps.

   (a) First, ask R about your present working directory with the `getwd()` function:

[1]I will assume this file structure in the rest of the scripts I provide this semester. If you prefer a different structure, you will need to modify the file paths you write in your scripts.

```
getwd()
```

Note that the `getwd()` function does nothing more than tell you where R is currently configured to import and export files.

(b) Next, make a comment below the command explaining what it does. For example:

```
getwd()  # Returns the present working directory
```

Keep notes in this way throughout your scripts in this course. In general, the more notes you take the more useful these scripts will be to you later. Try to imagine what kinds of notes your future self will need to understand the script.

Be nice to your future self. Someday, your future self will be you.

(c) Now set the working directory like this (uncomment the appropriate line):

```
## setwd("~/Desktop/DATA306") # MacOS
## setwd("~/DATA306") # Windows OS
```

(d) If everything looks good, you can load the data into R.

The following code does several things. It loads data from the `.csv` file. It also uses the function `<-` to *assign* the data as a *value* in an object *named* `school`:

```
school <- read.csv("data/NYCschools_r1.csv")
```

Give it a try. Feel free to name your object something other than `school` (just make sure it's something easy to remember, and make sure to modify the code below as appropriate). Make some brief comments to yourself explaining what's going on.

4. The data records 25 "features" (AKA variables, AKA columns) of 469 schools (AKA cases, AKA rows). Some of you will feel lost without being able to "see" the data as you would in a spreadsheet application like Excel. The following steps outline some of the ways you can take a first look at the data in R. Try them out in your script, taking care to include notes explaining each step to yourself. If you want, you can open the data in a spreadsheet application and compare what you see there (warning: *do not* save the file in the spreadsheet application).

   (a) The function `str()` gives the object's structure. If it's a `data.frame` object, it gives the column names, data types, and the values in the first few rows. Try it.

   ```
   str(school)
   ```

   (b) I prefer `str()`, but `head()` is a more popular way of looking at the data. It gives the first six rows of an object (by default). There is also a `tail()` function. Check it out.

   ```
   head(school)
   ## head(school, 10) # if you want to look at the first 10 rows
                        ## instead of the first 6
   ```

(c) `str()` can't give you all the information you need for analysis. Most important, R has no function that will tell you what `resp_rate_parent` or `outsiders_student` means.

Typically, you would find this kind of information in what is called a *codebook* or *data dictionary*. These often take the form of technical reports or manuals that list the variables, how they were collected, and even some summary statistics. Problematically, https://opendata.cityofnewyork.us/ does not always include this kind of information, or does not include it in very much detail.

If we were using this data for a serious research study, we would need to make some calls or write some emails asking for more information (like the specific questions the survey researchers asked and how they asked them). For now, we will accept what we have as "given." You'll have to just take my word for it. For now.

For the purposes of the exercise, each variable in our dataset is defined as the following:

- `X` is the *index* value of each observation. It was automatically generated when I built the dataset. I left it for us to play with later.
- `ID` and `name` give unique identifiers for each school.
- `total_enrollment` gives the total headcount of students in the school.
- All columns from `female` through `white` are demographic variables giving the proportion of students identifying/identified (we don't know which) in each category for each school.
- `ESL` is the proportion of students enrolled in English language learning services in each school.
- `poverty` is the proportion of students below the poverty line in each school.
- The proportion of students in fifth grade or below is `elementary` school. The proportion in sixth through eighth grade is `middle` school. The proportion in ninth grade and above is `high` school.
- `mean_class_size` gives the average size of classes in each school.
- `resp_rate_parent`, `resp_rate_student`, and `resp_rate_teacher` give the "response rate" of each group for the question about belongingness. A response rate is the proportion of people who responded to a survey out of the total number of people who were asked to respond.
- `outsiders_parent`, `outsiders_student`, and `outsiders_teacher` indicates the proportion of each group that *disagrees* or *strongly disagrees* with statements indicating students feel they "belong" or "feel respected and listened to" at their school.

5. As you may recall from your previous stats or data course(s), histograms are the classic way to see the distribution of a continuous variable. It's not a bad place to start. Here's how to do it in R:

```
hist(school$mean_class_size)
```

We used the `hist()` command to take a look at `total_enrollment`.

What's the `$`? This is how we *extract* a column from a `data.frame` object.[2]

[2]Technically, it's how we "select a single element of a list." The documentation at `?$` is worth a read.

You can extract columns in several ways using square brackets or the dollar sign. The following will put three identical histograms side-by-side in one plot window (you can change the dimensions of the plotting window if you can't see them very well):

```
par(mfrow = c(1, 3)) # This command sets three plots on one row
hist(school[["mean_class_size"]])
hist(school[ , "mean_class_size"])
hist(school$mean_class_size)
```

If you want to know more about the differences between `[[]]`, `[]`, and `$`, try `?$`.

6. We don't need three identical histograms. Below, see how to rewrite the last code block so that it produces a histogram, a boxplot, and a stripchart. Notice how I add stuff inside the parentheses for the boxplot and stripchart. These are *arguments*. Arguments are separated by commas.

```
par(mfrow = c(1, 3))

hist(school[["mean_class_size"]])

boxplot(school[ , "mean_class_size"], horizontal = TRUE)
                                        # horizontal = TRUE puts the boxplot on its si
                                        # The default (horizontal = FALSE) is vertical
                                        # TRUE is a special word in R indicating a boo
                                        ## does not need quotes around it.

stripchart(school$mean_class_size, method = "jitter")
                                        # Jitter randomly assigns points to locations
                                        ## so you can see them more easily.
                                        # Remove the "method" argument (and the comma)
                                        ## what happens!
                                        # The method argument can take several preset
                                        ## so requires quotes.
```

Write down an interpretation of this output in a sentence or two in comments under the plotting code. What do you notice about the distribution of mean class size? Is there anything about this distribution that we should take notice of before we proceed? How should we interpret the "strip chart" — labelled a "one-dimensional scatter plot" in the documentation at `?stripchart` — in the third panel?

7. Numerical summaries are straightforward in R.

```
summary(school) # produces summaries of all columns in a data.frame
```

What does this add to your interpretation from above? Write another (commented) interpretation of 1-2 sentences below your `summary()` code.

8. Change `summary(school)` so that it returns a numerical summary for one and only one column. Based on the output, respond to the following questions in comment form:

   (a) What is the "center" of the data?

   (b) How does the data vary about this center?

9. What we just did are sometimes called "univariate" and "descriptive" analyses. They are highly useful, easily understood, and provide a great deal of information. Insofar as they use "data analysis" at all, organizations very often do so on the basis of these basic and easily understandable artifacts.

   Using the summaries above, write a sentence giving important piece of information to a caregiver enrolling a child in the NYC school system for the first time.

10. Univariate statistics should not be discounted, but there are many situations in which it is also appropriate to analyze how distributions *move together*, or how they *covary*. We may, for example, wish to understand the situations in which a student feels they belong in their school or not (see part 2 below). We will take a simpler example for Part 1 of this exercise: the relationship between average class size and total enrollment.

    Understanding the relationship between class size and enrollment is trickier than it may appear. For example, we need to decide if we are interested in establishing a *causal* relationship, or something else.[3]

    The idea of causality is complicated, but for now we can think about it in very simple terms: a relation in which changes in one phenomenon can be attributed to changes in another phenomenon. Rain falls, rivers run.

    If we assume it makes sense to investigate a possible causal relationship between how many students are enrolled and the average class size within an NYC school, then we need to identify which variable is the "outcome" and which is the "predictor." We need to decide *what* causes *what*. Does the average size of classes depend on school size? Or vice versa?

    It can be useful to draw a picture. Even a mind-numbingly simple one. Here's how to do it in R.

```
## install.packages("DiagrammeR") # uncomment this first to install the package,
### then re-comment it
library(DiagrammeR)
grViz("digraph {'School size' -> 'Class size'}")
```

    The `DiagrammeR` package makes it possible to write Graphviz code in R. Graphviz uses the "dot" language, which is everything wrapped in quotes in the lines above. The output will appear in your web browser as an HTML file. You can replace the labels I wrote with your own, or you could decide that I have the causal order wrong.

    You should specify what *kind* of relationship you expect between the two variables. We might start by hypothesizing the *direction* of the relationship. For example, it might make sense

[3]There are many other relationships we could be interested in aside from causal ones. Two variables can be related through the action of a third, for example. Or we may be interested in finding variables that track the movements of other variables without being causally related (as strategies for measurement, for example). Perhaps we are not interested in thinking in terms of "variables" at all. The list goes on.

to guess that as school size increases, average class size increases. This is also known as a positive relationship.

We can express the *direction* of the hypothetical causal relationship with a label. For the hypothesis that class size *increases* as a function of school size (a positive relationship):

```
grViz("digraph {'School size' -> 'Class size' [label = '+']}")
```

You can replace the + sign with a - sign if you want to adopt the opposite hypothesis (if you think class size declines as school size increases). Notice that I added a space before the negative sign in the code below to make it easier to see.

```
grViz("digraph {
        # rankdir = 'LR' # uncomment this to make the
                         ## diagram flow from left to right
         'School size' -> 'Class size' [label = ' -']
       }")
```

With these diagrams, we have nailed down a *specific* and *testable* relationship. Such a statement (either in visual or verbal form) is called a *hypothesis*. Reasoning with hypotheses is often called *deductive* reasoning. It's not the only way to do data analysis, but it's a conventional place to start. It became the dominant method for formal scientific reasoning sometime in the last century.

Within this style of reasoning, we cannot *establish* causality with observational data. We must instead be content with *accumulating evidence against* the idea that there is *not* a causal relationship. As the evidence builds, we can begin to have some *confidence* that our hypothesis does indeed describe something about the phenomena we are trying to explain.

Pause for a moment to think about this: in deduction, we presume that our analytical object is innocent of our accusations. We nonetheless proceed to pile up evidence against their innocence (if we can). We never really *prove* them guilty so much as *disprove* their innocence. It's OK if this seems odd to you. Just let it sink in.

11. As you may recall from a previous course, we can look at relations between two continuous variables directly with a scatterplot. You can produce a two-dimensional scatterplot with the following:

    ```
    plot(school$total_enrollment, school$mean_class_size)
    ```

    What does this tell us about the relationship between enrollment (the number of students in a school) and mean class size? Write a commented sentence or two interpreting this plot. Does the evidence from this sample of schools support the relationship you hypothesized above? Why or why not?

12. Let's do some modeling. A bivariate (bi = two, variate = variables) linear regression model in R takes only a few keystrokes:

    ```
    school_lm <- lm(mean_class_size ~ total_enrollment, data = school)
    ```

To see most of the statistics we will need, we can apply the `summary()` function to the `lm()` object we just named `school_lm`.

```
summary(school_lm)
```

We will spend the rest of the semester learning how to interpret this output! Nonetheless, some of it may be legible to some of you from earlier coursework:

- If you remember the "correlation coefficient," then you should be able to interpret the `Multiple R-squared` statistic, which is equivalent to it: total enrollment explains about 14.5% of the variation in mean class size.
- The intercept is where the model begins on the y-axis. It is akin to the mean, but more accurately interpreted as *mean class size when overall enrollment is zero* (if that makes no sense to you, good! It is meaningless in this case).
- The coefficient `total_enrollment` indicates how much `mean_class_size` changes with each additional student in `total_enrollment`. Note that the coefficient is small (0.004) and positive. This makes sense because average class sizes are small relative to total enrollments.
- The `Pr(>|t|)` column indicates the size and direction of the coefficient is probably not a result of random chance. The t value is large at 8.9. The probability of observing a coefficient that extreme is less than 2 with 15 zeroes in front of it (`<2e-16`).

13. There is much more to explain here, but we will leave it for later. For now, it helps to take a look at how this model looks compared to our data:

```
plot(school$mean_class_size ~ school$total_enrollment)
abline(school_lm, col = "blue", lwd = 2)
               ## col means color, lwd means line width
```

The blue line gives the predicted mean class size for a given total enrollment. There are several important observations to make about this model.

First, we should recognize that, like all models, this one is wrong. Then we should ask if it is wrong in a *useful* way. What, if anything, does the model indicate about our hypothesis of a positive relationship between total enrollment and mean class size? Write a sentence about this as a comment below the plotting code.

You may have noticed that the cloud of points has a gentle curve to it between about 200 and 700 on the X-axis. Our *linear* regression implies a *linear relationship* between these two variables, and so does not do a good job capturing this curvature. To illustrate the curve, and the difference between it and our model, we can add what is called a LOESS smoother to our plot.

The `scatter.smooth()` function is probably one of the most straightforward ways of doing this.

```
scatter.smooth(school$mean_class_size ~ school$total_enrollment,
               span = 2/3, degree = 2, # these are defaults
               ## play with them to see what changes
               lpars = list(col = "red", lwd = 2))
                                        # lpars means line parameters
abline(school_lm, col = "blue", lwd = 2)
```

Part 2: Do your thing

1. Write a comment that looks like this:

   ```
   ## Part 2: An analysis of student belonging and [covariate]
   ```

   . . . where `[covariate]` is a variable that you think may have a meaningful relation to `outsiders_student`

2. Repeat steps 6, 8, 10, 12, and 13 with the following modifications:

   (a) Use `outsiders_student` as your focal variable in step 6.

   (b) Use another variable of your choosing. Please choose something we did not look at already in Part 1. That would be boring.

   (c) You may decide if `outsiders_student` is an outcome or predictor variable, but you must have some reasonable explanation for your choice. Justify your choice in a sentence in your repetition of step 10.

   (d) Change all interpretations to suit your new outputs.

   (e) Write a few sentences about what you *think* you found out about the relation of interest.