SOCI 385 - Social Statistics

Fall 2019

Downloading and Cleaning GSS Data

1. The General Social Survey data is available through the National Opinion Research Center's Data Explorer. The website is: https://gssdataexplorer.norc.org. On that page, click the button to create an account. After submitting your information, you will get a confirmation email. Click on the link to complete registration.

2. On the GSS Data Explorer home page, click "Projects Overview" and "Create A Project." Think of the project like a folder; you will be able to name the extract later. A project name like "SOCI 385 Downloads" would make sense. Click continue. You can choose any privacy setting, then click continue. For the "File format extract type" select Excel Workbook (data + metadata), Stata (Control + System), and R Script. And then continue.

3. You should now see your project workspace dashboard with sections for variables, user searches, analyses, and extracts. In the variables window, click the green button with the white plus symbol.

4. You will now be on the "Search Data" page. Note that the `year` and `id` variables are already selected and in your cart. In the keyword box, you can enter the name of a variable or a description of the variable you want. Search for `hours worked` and press return.

5. If there is a match, you will see the variable name, the description of the variable, and the years in which that survey question was asked. Click on the variable name to see the full text of the survey question and the frequency table for each year. If this is the variable you want, click the cart icon to add it, and/or click the left arrow to go back to variable results.

6. You can continue searching for and adding variables. For this exercise, find and add the variables for survey respondents' hours worked last week, class, health condition, and sex.

7. When your cart has all the variables you want to download, return to your project workspace dashboard (you can click the link in the "My GSS" header). You should see the variables you added in the variables window. Scroll down to the extracts window and click "Create an extract." Give your extract a name to identify it within your project. Something like "assignment_6" would make sense here. Then click next.

8. You can drag the variables you want from your cart to the "Choose variables" window. It is always a good idea to include the `year` and `id` variables. We also want all the variables we added. After selecting variables, click next. We will skip the optional "Case Selection" window.

9. Now it is time to choose your output options. For this exercise, we only want responses from 2008 and 2018. To do so, click the "Select certain years" link and then check the 2008 and 2018 boxes. Under "File Format" make sure that there are checks next to Excel Workbook (data + metadata), Stata, and R Script. Then click "Create Extract."

10. Your extract will automatically start downloading. If you are downloading many variables (and/or variables from many years) the download may take a while. You will know the extract is ready when the spinning icon becomes a down arrow. Click on the arrow to download the extract and find it in your downloads folder. You may want to move the downloaded folder to your desktop or working directory. Unzip the folder. (If you are using Windows, see note below for how to unzip the folder.)

11. Your downloaded folder will have five files: a dat file, a dct file, a do file, an R file, and an excel file. Open the R file in R. The first line loads the `foreign` package. You may have to install it before running the rest of the file. ***Include the full path for the .dct and .dat files in lines 32 and 33.*** Then highlight all the code and click run.

12. If the file ran correctly, you should now have two data frames loaded into your workspace: one called GSS and one called GSS_ascii. They are exactly the same. Open the GSS data frame to see your four variables there.

13. There are a few things to fix before we are ready to do some analyses. We can do so by adding code to the R file we previously ran. This is a regular R script file (not a notebook) so we don't need to open and close each chunk. Just type your code.

First, let's make all the variable names lower case. We can use the `tolower()` function for this by combining it with the `names()` function:

```r
names(GSS) <- tolower(names(GSS))
```

If you want to change the name of a column, you can use the `rename` function, which states the *new* name first followed by the *existing* name:

```r
library(tidyverse) # rename requires tidyverse to be loaded!

GSS <- GSS %>%
  rename(id = id_,
         class = class_)
```

14. Now let's clean up the `hrs1` variable. While R identifies missing values with an NA, the GSS does not. So we have to use the GSS codebook to find the values that refer to missing data. The codebooks are in the excel file (the Codes tab) and the do file that we downloaded. I find the do file (which you can open in any text reader) to be a little cleaner, but the excel file might be less intimidating. It's your choice. Both codebooks are the same, and they both show that values of -1, 98, and 99 all refer to missing values for the `hrs1` variable. Let's change all these to NA using na_if:

```r
GSS <- GSS %>%
     mutate(hrs1 = na_if(hrs1, -1), hrs1 = na_if(hrs1, 98),
            hrs1 = na_if(hrs1, 99))
```

15. Now let's clean up the `class` variable. The initial step is to replace missing values with NAs. According to the codebook, values of 0, 5, 8, and 9 all refer to missing values for the `class` variable. We have already seen that we can do this with na_if. But when we have more than one value to replace with NA, it can be more efficient to use indexing and not deal with mutate or na_if.

```
GSS$class[GSS$class==0 | GSS$class>=5] <- NA
```

For the non-missing values, we also have to replace the numbers with characters. We do this by creating a factor variable with mutate. Use the GSS codebook here! The labels have to be in the same order as the numeric values to which they correspond.

```
GSS <- GSS %>%
     mutate(class = factor(class,
            labels = c("Lower class", "Working class",
                       "Middle class", "Upper class")))
```

16. The file is now clean and ready to use. You can continue to write your code in this R file. Alternatively (and perhaps preferably), you can save the cleaned up file as a separate csv file and then open it in an R notebook as usual. To save the file as a csv file use the `write.csv()` function. In this example, GSS is the data frame you want to save and inside the quotes include the full path - including the new file name - where you want the file saved. For example:

```
write.csv(GSS, "/users/lawrence/desktop/assignment_6.csv")
```

17. One thing to note when you open the file in a new R Notebook: the factor variable `class` will be sorted alphabetically so you will have to re-order it before starting:

```
GSS$class <- factor(GSS$class,
                    levels = c("Lower class", "Working class",
                               "Middle class", "Upper class"))
```

Unzipping Folders In Windows

If you are using a Windows operating system, there are a few additional steps that may be required to unzip the downloaded folder before you can access the required files. Locate the zipped folder that you want to extract files or folders from. Then press and hold (or rick-click) the folder, select "Extract All" and follow the instructions. Once you have access to all the files you downloaded, return to step #11 in these instructions to continue.