# Exam 3 - Data Science for the Social World

Michael G. Findley[*] Michael Denly[†]

## Instructions

This is a two-hour, open-book, open-internet exam. As soon as you click on the exam on Canvas is when your timer starts. If you submit the exam late, you will be penalized by 1 point for each minute late that you submit. For example, if you submit 5 minutes late, you will lose 5 points on your overall exam grade.

In terms of your submission, Canvas will only allow you to submit one file. That one file should be either a PDF file or Word document corresponding to the output of your `R` Markdown file. Kindly note that we will not accept Google Docs, and that you must write your exam using `R` Markdown. Students who use a regular `R` script and then copy/paste outputs onto a Word Document or PDF file will receive a 20-point penalty.

The last question of the exam asks you to submit a link to a GitHub repo with your `.Rmd`, `.dta`, and PDF file/Word Document. Failure to submit a link with all of these files to a working GitHub repo will result in an additional 15-point penalty. You may name the repo anything that you would like, but maybe something like “exam3” would be appropriate. Since GitHub provides time stamps for everything, we will be able to discern if you modify the files outside your two-hour exam window. In short, please respect the two-hour window.

Please work independently. You may *not* consult anyone in the class or outside the class for help, and you may not post the exam questions on the Stack Exchange, Google Groups, or any similar website. However, you may visit these websites or others. Please also do not discuss the questions or answers over WhatsApp, GroupMe, text message, or any other platform, especially because everyone will be taking the exams at different times. We will be monitoring accordingly, and anyone who violates any one of these policies will receive a zero on the exam.

Please annotate your `R` code chunks in your `R` Markdown file with comments, or make sure that the text surrounding it sufficiently explains what you are doing. Essentially, your `R` Markdown file should mimic that notes files that we submit on Canvas to accompany the

[*] Professor, Department of Government, UT Austin, mikefindley@utexas.edu

[†] PhD Candidate, Department of Government, UT Austin, mdenly@utexas.edu

video lectures. We will remove points when you do not provide clear comments or explanation to tell us exactly what you are doing with your code.

We have endeavored to make the exam self-explanatory, but feel free to email the instructors and the TA if you have questions. At least one of us will be available over email for the entire 4-hour exam. However, please email all three of us if you have a question (i.e., do not email only one or two of us), because we will be taking shifts.

One final note: if you have an SSD accommodation, please submit your exam under "Exam 3 (accommodation)". Otherwise, please submit on Canvas under "Exam 3".

And one final hint: use your time wisely. If you can't answer one question, move on to the next one, and come back to it once you are done with the ones that you can answer more quickly. Good luck!

# Questions

1. Clear the environment. [5 points]
2. Use the `tidycensus` package to (a) find the inequality Gini index variable explained on the last exam, (b) import in the state-level inequality Gini estimates for 2010 and 2015 in the five-year American Community Survey as a *single panel dataset*; (c) rename `estimate` as `gini` in your final data frame, which you should call `inequality_panel`; (d) rename `NAME` to `state` as well; (e) ensure that `inequality_panel` has a `year` variable so we can distinguish between the 2010 and 2015 `gini` index data; and (f) as a final step, run the `head()` command so we can get a quick peak at `inequality_panel` (Hint: you may need to import each year separately and then append the two data frames together.) [15 points]
3. Reshape the `inequality_panel` wide, such that the `gini` values for 2010 and 2015 have their own columns. Also, please keep both the `state` and `GEOID` variables. Call the resulting data frame `inequality_wide`. After you are done with the reshape, run the `head()` command so we can get a quick peak at the data. [5 points]
4. Reshape `inequality_wide` to long format. Once you are done, run the `head()` command so we can get a quick peak at the data. [5 points]
5. Show with some `R` code that `inequality_panel` and `inequality_long` have the same number of observations. [5 points]
6. Collapse the `inequality_long` data frame by `state`, such that you obtain a single mean `gini` score for each state for the years 2010 and 2015. When collapsing, also keep both the `GEOID` and `state` variables. Call your resulting data frame `inequality_collapsed`. [5 points]
7. Produce a map of the United States that colors in the state polygons by their mean `gini` scores from `inequality_collapsed`, using the WGS84 coordinate system. When

doing so, use the viridis color scheme. (Note: there are a few different ways to produce the map. We will accept any one of these ways, and the answer key will detail 3 ways. If you want to choose the method with the shape file, you can get the state-level shape file on the Census page). [10 points]

8. Use the WDI package to import in data on Gross Domestic Product (GDP) in current US dollars. When doing so, include all countries and only the years 2006 and 2007. Rename your GDP variable to gdp_current. [5 points]

9. Deflate gdp_current to constant 2010 or 2015 US dollars, and call the new variable gdp_deflated. In words, also tell us the base year that you picked and why. At the end, run a head() command to prove that everything works. [5 points]

10. In a Shiny app, what are the three main components and their subcomponents? [5 points]

11. Pull this .pdf file from Mike Denly's webpage. It is a report on governance in Armenia that Mike Denly and Mike Findley prepared for the US Agency for International Development (USAID). [5 points]

12. Convert the text pulled from this .pdf file to a data frame, using the , stringsAsFactors=FALSE option. Call the data frame armeniatext. [5 points]

13. Tokenize the data by word and then remove stop words. [5 points]

14. Figure out the top 5 most used word in the report. [5 points]

15. Load the Billboard Hot 100 webpage, which we explored in the course modules. Name the list object: hot100exam [5 points]

16. Use rvest to obtain identify all of the nodes in the webpage. [5 points]

17. Use Google Chrome developer to identify the necessary tags and pull the data on *Rank*, *Artist*, *Title*, and *Last Week*. HINT 1: In class we showed you how to get the first three of these. You simply need to add the *Last Week* ranking. HINT 2: You can navigate two ways. Hovering to find what you need or by doing Cmd+F / Ctrl+F and using actual data to find the location. HINT 3: You're looking to update the code based on the *way the information is in referenced*. Try out some different options and see what shows up in the environment. Keep trying until you see that you have a chr [1:100] with values that correspond to what is in the web page. [5 points]

Final question. Save all of the files (i.e. .Rmd, .dta, .pdf/Word Doc), push them to your GitHub repo, and provide us with the link to that repo. [no points; 15-point penalty for lack of submission (see above)]