ExamThree

Claire Sheppard

7/9/2020

# EXAM THREE

This is Exam Three for Gov355M I have learned a great deal in this course. The time commitment was challenging and I struggled to accomplish everything in a timely manner due to commitments outside of class. But I definitely will continue to work with these tools, not only in my job at Capital One Auto Finance, but also for websites for my businesses. Thank you. - Claire Sheppard

## Question 1 Clear the environment. [5 points]

rm(list=ls(all=TRUE))

## Question 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]

# load tidycensus

library(tidyverse) library(tidycensus)

# (a) find the inequality Gini index variable explained

##on the last exam head(tidycensus, "inequality\_gini)

# (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; 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;

# (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.)

#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] library(WDI) head(WDI)

gdp\_current = WDI(country = “all”, indicator = c(“GDP”), # indicator from web start = 2006, end = 2007, extra = FALSE, cache = NULL)

#10 In a Shiny app, what are the three main components and their subcomponents? [5] A Shiny App has a User Interface Object, a Server function, and a Call to the ShinyApp function. The User Interface (UI) is created with an HTML function. Through the Server function, there is an input, output, and a session. From this point the UI and the Server are called in the shinyApp

#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] pdf is <https://pdf.usaid.gov/pdf_docs/PA00TNMG.pdf>

#First be sure libraries all loaded #install.packages library(pdftools) #pull from Mike Denly’s webpage mytext=pdf\_text(pdf=“<https://pdf.usaid.gov/pdf_docs/PA00TNMG.pdf>”)

mytext

#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] library(tidytext) armeniatext <- as.data.frame(mytext) armeniatext$page=c(1:65) colnames(armeniatext)[which(names(armeniatext) == “armeniatext”)] <- armeniatext

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

#in order to tokenize text into words: armeniatext <- armeniatext %>% unnest\_tokens(armeniatext,text)

#in order to get rid of stop words: data(stop\_words) armeniatext <- armeniatext %>% anti\_join(stop\_words)

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

armeniatext %>% count(word, sort = TRUE)

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

library(rvest) library(dplyr) library(ggplot2)

hot100page <- “<https://www.billboard.com/charts/hot-100>” hot100exam <- read\_html(hot100page )

hot100exam str(hot100exam) library(pdftools)

#16. Use rvest to obtain identify all of the nodes in the webpage. [5 points] rank <- hot100exam %>% rvest::html\_nodes(‘body’) %>% xml2::xml\_find\_all(“//span[contains(@class, ’chart-element\_\_rank\_\_number’)]”) %>% rvest::html\_text()

artist <- hot100exam %>% rvest::html\_nodes(‘body’) %>% xml2::xml\_find\_all(“//span[contains(@class, ’chart-element\_\_information\_\_artist’)]”) %>% rvest::html\_text()

title <- hot100exam %>% rvest::html\_nodes(‘body’) %>% xml2::xml\_find\_all(“//span[contains(@class, ’chart-element\_\_information\_\_song’)]”) %>% rvest::html\_text()

#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]

#on google chrome, ctrl+shift+c. then ctrl+f anything to see how it is coded #for example, in this table, CASE column and its entries are coded #under ’ td headers=‘…’ td data-th=“CASE” ‘. same goes for other columns. # I know this is the basis, just ran out of time. CASE <- ois %>% rvest::html\_nodes(’body’) %>% xml2::xml\_find\_all(“//td[contains(@data-th, 'CASE')]”) %>% rvest::html\_text()

DATE <- ois %>% rvest::html\_nodes(‘body’) %>% xml2::xml\_find\_all(“//td[contains(@data-th, 'DATE')]”) %>% rvest::html\_text()

“OFFICER: #/RACE/SEX” <- ois %>% rvest::html\_nodes(‘body’) %>% xml2::xml\_find\_all(“//td[contains(@data-th, 'OFFICER: #/RACE/SEX')]”) %>% rvest::html\_text()

SUMMARY <- ois %>% rvest::html\_nodes(‘body’) %>% xml2::xml\_find\_all(“//td[contains(@data-th, 'SUMMARY')]”) %>% rvest::html\_text()

“SUSPECT: RACE/GENDER” <- ois %>% rvest::html\_nodes(‘body’) %>% xml2::xml\_find\_all(“//td[contains(@data-th, 'SUSPECT: RACE/GENDER')]”) %>% rvest::html\_text()

“GREENE COUNTY PROSECUTING DISPOSITION” <- ois %>% rvest::html\_nodes(‘body’) %>% xml2::xml\_find\_all(“//td[contains(@data-th, 'GREENE COUNTY PROSECUTING DISPOSITION')]”) rvest::html\_text()

#Final question. Save all of the files (i.e. .Rmd, .dta, .pdf/Word Doc), push

My GitHub is