Exam 3

Antonio Tiu

7/9/2020

# 1. Clear environment

rm(list=ls())

# 2 Using tidycensus

library(tidycensus)  
  
census\_api\_key("5e97ead132046e45d7523a868bab0829571d2f75")

## To install your API key for use in future sessions, run this function with `install = TRUE`.

#inequality\_panel

# 3. Reshaping inequality\_panel wide then using head to peak the data

# 4. Reshape ineqaulity\_wide to long format

#long\_data\_frame <-  
 # wide\_by\_year%>%  
# pivot\_longer(cols =starts\_with("year"),  
# names\_prefix = "year\_",  
# values\_to = "current\_amount",  
# values\_drop\_na = FALSE)%>%  
# filter(!(current\_amount==0))

# 5. Showing that inequality\_panel and inequality\_long have the same data

# 6. Collapse inequality\_long data frame by state

#collapsed\_data <-  
 # alldata%>%  
 # group\_by(year, state\_name)%>%  
 # summarize(across(where(is.numeric), sum))%>%  
 # sumselect(-c("transaction\_id"))

# 7. Produce map of the U.S that colors in the state polygons by their mean gini scores

# 8. Use WDI package to import data

# 9. Deflate gdp\_current to constant 2010

library(WDI)  
deflator\_data =WDI(country = "all", indicator =c("NY.GDP.DEFL.ZS"),  
 start = 2010,  
 end = 2010,  
 extra = FALSE, cache = NULL)

# 10. What are the three main components and their subcomponents?

The three main components of Shiny app are the “ui”, “server”, and “shinyApp”.

# 11. Pull .pdf file

library(pdftools)

## Using poppler version 0.73.0

library(tidyverse)

## ── Attaching packages ───────────────────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.2 ✓ purrr 0.3.4  
## ✓ tibble 3.0.2 ✓ dplyr 1.0.0  
## ✓ tidyr 1.1.0 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.5.0

## ── Conflicts ──────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr)  
library(purrr)  
library(readr)  
  
GDP <- pdf\_text("usaid.pdf")%>%  
 readr::read\_lines()

# 12. Convert text pulled from the .pdf to a data frame

library(tm)

## Loading required package: NLP

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

#armeniatext <- function("usaid.pdf") {  
   
# dat = readPDF(control=list(text="-layout"))(elem=list(uri=file),   
 # language="en", id="id1")   
# dat = c(as.character(dat))  
  
# dat = dat[grep("^ {0,2}[0-9]{1,3}", dat)]  
  
# dat = gsub("^ ?([0-9]{1,3}) ?", "\\1|", dat)  
# dat = gsub("(, HVOL )","\\1 ", dat)  
# dat = gsub(" {2,100}", "|", dat)  
  
# excludeRows = lapply(gregexpr("\\|", dat), function(x) length(x)) != 6  
# write(dat[excludeRows], "rowsToCheck.txt", append=TRUE)  
  
   
# dat = dat[!excludeRows]  
  
   
# dat = read.table(text=dat, sep="|", quote="", stringsAsFactors=FALSE)  
 # names(dat) = c("RowNum", "Reference Entity", "Sub-Index", "CLIP",   
 # "Reference Obligation", "CUSIP/ISIN", "Weighting")  
# return(dat)  
#}

# 13. Tokenize the data by word and remove stop words

# armeniatext=armeniatext%>%  
# unnest\_tokens(word, text)  
  
#data(stop\_words)  
  
#armeniatext <- armeniatext%>%  
 # anti\_join(stop\_words)

# 14. Figure out top 5 most used word

# frequency <- armeniatext%>%  
 # count(word, sort = TRUE)  
  
#frequency

# 15. Load Billboard Hot 100 webpage

library(rvest)

## Loading required package: xml2

##   
## Attaching package: 'rvest'

## The following object is masked from 'package:purrr':  
##   
## pluck

## The following object is masked from 'package:readr':  
##   
## guess\_encoding

library(dplyr)  
  
hot100exam <- "https://www.billboard.com/charts/hot-100"  
hot100 <- read\_html(hot100exam)  
hot100

## {html\_document}  
## <html class="" lang="">  
## [1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=UTF-8 ...  
## [2] <body class="chart-page chart-page-" data-trackcategory="Charts-TheHot100 ...

# 16. Use rvest to obtain all of nodes in the webpage

body\_nodes <- hot100 %>%  
 html\_node("body") %>%  
 html\_children()  
body\_nodes

## {xml\_nodeset (36)}  
## [1] <div class="header-wrapper ">\n<header id="site-header" class="site-head ...  
## [2] <div class="site-header\_\_placeholder"></div>  
## [3] <script>\n var PGM = window.PGM || {};\n PGM.config = PGM. ...  
## [4] <div class="chart-piano-overlay\_\_attachment-point"></div>  
## [5] <main id="main" class="page-content"><div id="charts" data-page-title="T ...  
## [6] <div class="ad\_desktop dfp-ad dfp-ad-promo " data-position="promo" data- ...  
## [7] <div class="ad-container footerboard footerboard--bottom">\n <div cla ...  
## [8] <footer id="site-footer" class="site-footer"><div class="container foote ...  
## [9] <div class="biz-modal">\n <div class="biz-modal\_\_content">\n < ...  
## [10] <script>\n window.CLARITY = window.CLARITY || [];\n</script>  
## [11] <div class="ad\_clarity" data-out-of-page="true" style="display: none;">< ...  
## [12] <script>\n var darkMatterCMD = function() {\n this.darkMatterC ...  
## [13] <script src="https://www.billboard.com/assets/1593527595/js/vendors\_/art ...  
## [14] <script src="https://www.billboard.com/assets/1593527595/js/vendors\_/clo ...  
## [15] <script src="https://www.billboard.com/assets/1593527595/js/vendors\_/rea ...  
## [16] <script src="https://www.billboard.com/assets/1593527595/js/vendors\_/rea ...  
## [17] <script src="https://www.billboard.com/assets/1593527595/js/vendors\_/rea ...  
## [18] <script src="https://www.billboard.com/assets/1593527595/js/vendors\_/rea ...  
## [19] <script src="https://www.billboard.com/assets/1593527595/js/default\_/art ...  
## [20] <script src="https://www.billboard.com/assets/1593527595/js/default\_/rea ...  
## ...

# 17. Using Google Chrome to identify the necessary tags

rank <- hot100 %>%   
 rvest::html\_nodes('body') %>%  
 xml2:: xml\_find\_all("//span[contains(@class,   
 'chart-element\_\_rank\_\_number')]") %>%  
 rvest::html\_text()  
  
artist <- hot100 %>%   
 rvest::html\_nodes('body') %>%  
 xml2:: xml\_find\_all("//span[contains(@class,   
 'chart-element\_\_information\_\_artist')]") %>%  
 rvest::html\_text()  
   
title <- hot100 %>%   
 rvest::html\_nodes('body') %>%  
 xml2:: xml\_find\_all("//span[contains(@class,   
 'chart-element\_\_information\_\_song')]") %>%  
 rvest::html\_text()  
  
  
chart\_df <- data.frame(rank,artist,title)

# Link to Github repo

<https://github.com/antoniotiu21/exam3>