Exam 3 PDF

Madison Pape

7/08/2021

## Question 1 - Clear the environment

rm(list = ls())  
  
library(pdftools)  
library(tidyr)  
library(tidytext)  
library(dplyr)  
library(stringr)  
library(ggplot2)

## Question 2 - Use the WDI package to download data on female labor force participation for all countries for the years 2010-2015. Save the data frame as female\_lfp. (Hint: you may/will need to Google the indicator.)

library(WDI)  
female\_lfp = WDI(country = "all",  
 indicator = "SL.TLF.CACT.FE.ZS",  
 start = 2010, end = 2015, extra = FALSE, cache = NULL)

## Question 3 - Rename the female labor force participation variable flfp.

flfp <- female\_lfp$SL.TLF.CACT.FE.ZS

## Question 4 - Collapse female\_lfp by the mean value for flfp for each country. When you do, keep the ISO-2 country code in your data frame as well as the country name. Name your resulting data frame collapsed\_flfp.

library(dplyr)  
collapsed\_flfp\_p1 <-   
 female\_lfp %>%  
 filter(between(year, 2010, 2015)) %>%  
 group\_by(country) %>%  
 summarize\_at(flfp, mean)

## Question 5 - Use R to show which countries have average female force participation rates for the 2010-2015 period that are less than 15%.

## Question 6

## Question 7

## Question 8

## Question 9 - In a Shiny app, what are the three main components and their subcomponents?

## Answer - The main components include the User Interface (UI), the Server, and the Output. The subcomponents of the User Interface include the Inputs and Outputs. The subcomponents of the Server include the directions which are stored in a object, the render() function, and the output object.

## Question 10 - Pull this pdf file from Mike Denly’s webpage. It is a report that Mike Denly and Mike Findley prepared for the US Agency for International Development (USAID).

library(pdftools)  
usaidpdf=pdf\_text(pdf = "https://pdf.usaid.gov/pdf\_docs/PA00TNMJ.pdf")  
usaidpdf

## Question 11 - Convert the text pulled from this.pdffile to a data frame, using the ,stringsAsFactors=FALSE option. Call the data frame armeniatext

armeniatext=as.data.frame(usaidpdf,  
 stringsAsFactors=FALSE)

## Question 12 - Tokenize the data by word and then remove stop words

library(tidytext)  
library(dplyr)  
armeniatext %>%  
 unnest\_tokens(word, text)  
   
data(stop\_words)  
armeniatext <- armeniatext %>%  
 anti\_join(stop\_words)

## Question 13 - Figure out the top 5 most used word in the report.

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

## Question 14 - Load the Billboard Hot 100 webpage, which we explored in the course modules. Name the list object:hot100exam

library(rvest)  
library(dplyr)  
library(ggplot2)  
hot100page <- "https://www.billboard.com/charts/hot-100"  
hot100exam <- read\_html(hot100page)

## Question 15 - Use rvest to obtain identify all of the nodes in the webpage.

body\_nodes <- hot100exam %>%   
 html\_node("body") %>%  
 html\_children  
   
body\_nodes %>%  
 html\_children()

## Question 16 - 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+Fand 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 achr [1:100] with values that correspond to what is in the web page.

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()  
  
lastweek <- hot100exam %>%  
 rvest::html\_nodes('body') %>%  
 xml2::xml\_find\_all("//span[contains(@class,  
 'chart-element\_\_meta.text--center.color--secondary.text- -last')]") %>%  
 rvest::html\_text()

## Question 17 - GitHub Repo Link

## link: <https://github.com/madisonpape/exam3>