Analysis 2: Joining Cholesterol with Crimes and Web Scraping Wikipedia

Anthony

06/08/2021

Instructions

Overview: For each question, show your R code that you used to answer each question in the provided chunks. When a written response is required, be sure to answer the entire question in complete sentences outside the code chunks. When figures are required, be sure to follow all requirements to receive full credit. Point values are assigned for every part of this analysis.

Helpful: Make sure you knit the document as you go through the assignment. Check all your results in the created HTML or PDF file.

Submission: Submit via an electronic document on Sakai. Must be submitted as an PDF or an HTML file generated in RStudio.

Introduction

Does high cholesterol lead to high crime rates? Probably not, but web scraping will definitely lead to lower crime rates. This data analysis assignment is separated into three parts which cover material from the lectures on tidy data, joins, and webscraping. In Part 1, you will demonstrate the basic concept of joins by connecting relational data involving a cholesterol study. For this segment, `pivot_longer` and `pivot_wider` will be utilized to create a single tidy dataset ready for analysis. In Part 2, we will join all 5 datasets from the lecture series on web scraping. Part 3 will require an understanding of web scraping to import a table found on Wikipedia directly into R. The following R code reads in all datasets required for this assignment.

```
# Data for Part 1
CHOL1=read_csv("Cholesterol.csv")
CHOL2=read_csv("Cholesterol2.csv")

# Data for Part 2
VIOLENT=read_csv("FINAL_VIOLENT.csv")
ZIP=read_csv("FINAL_ZIP.csv")
STATE_ABBREV=read_csv("FINAL_STATE_ABBREV.csv")
CENSUS=read_csv("FINAL_CENSUS.csv")
S_VS_D=read_csv("FINAL_SAFE_VS_DANGEROUS.CSV")
```

Assignment

Part 1: Cholesterol Experiment

The data frame `CHOL1` contains experimental results from randomly assigning *18* people to one of two competing margarine brands "A" and "B". The cholesterol of these patients was measured once before using the margarine brand, once after *4* weeks with the margarine brand, and then again after *8* weeks with the margarine brand. Researchers want to see if there is benefit of these brands of margarine on reducing an

individual's cholesterol and want to determine if there is a statistically significant difference between the two competing brands.

```
CHOL1
```

```
## # A tibble: 18 x 5
##       ID Before After4weeks After8weeks Margarine
##    <dbl>  <dbl>       <dbl>       <dbl> <chr>
##  1     1   6.42        5.83        5.75 B
##  2     2   6.76        6.2         6.13 A
##  3     3   6.56        5.83        5.71 B
##  4     4   4.8         4.27        4.15 A
##  5     5   8.43        7.71        7.67 B
##  6     6   7.49        7.12        7.05 A
##  7     7   8.05        7.25        7.1  B
##  8     8   5.05        4.63        4.67 A
##  9     9   5.77        5.31        5.33 B
## 10    10   3.91        3.7         3.66 A
## 11    11   6.77        6.15        5.96 B
## 12    12   6.44        5.59        5.64 B
## 13    13   6.17        5.56        5.51 A
## 14    14   7.67        7.11        6.96 A
## 15    15   7.34        6.84        6.82 A
## 16    16   6.85        6.4         6.29 B
## 17    17   5.13        4.52        4.45 A
## 18    18   5.73        5.13        5.17 B
```

```
CHOL2
```

```
## # A tibble: 9 x 1
##   `Brand\tStatistic\tValue`
##   <chr>
## 1 "A\tServing\t14"
## 2 "A\tCalories\t70"
## 3 "A\tFat\t7"
## 4 "A\tSatFat\t2.5"
## 5 "A\tSodium\t130"
## 6 "B\tServing\t14"
## 7 "B\tCalories\t50"
## 8 "B\tFat\t6"
## 9 "B\tSatFat\t1.5"
```

Q1 ***(3 Points)***

Start by examing the tables `CHOL1` and `CHOL2` and answering the following questions with *Yes* or *No* responses.

Is the variable `ID` in `CHOL1` a primary key?

Answer *(1 Point)*: Yes

Is the variable, `Margarine` in `CHOL1` a primary key?

Answer *(1 Point)*: No

Is the variable, `Brand` in `CHOL2` a primary key?

Answer *(1 Point)*: No

Q2 ***(2 Points)***

In a new data frame called `CHOL1a` based on `CHOL1`, rename the variables `After4weeks` and `After8weeks` to nonsynctactic variable names `4` and `8`, respectively. Use `names(CHOL1a)` to display this modification.

```
#
CHOL1a <- CHOL1 %>%
  rename("4" = After4weeks, "8" = After8weeks)
names(CHOL1a)
```

```
## [1] "ID"        "Before"    "4"         "8"         "Margarine"
```

```
CHOL1a
```

```
## # A tibble: 18 x 5
##       ID Before   `4`   `8` Margarine
##    <dbl>  <dbl> <dbl> <dbl> <chr>
##  1     1   6.42  5.83  5.75 B
##  2     2   6.76  6.2   6.13 A
##  3     3   6.56  5.83  5.71 B
##  4     4   4.8   4.27  4.15 A
##  5     5   8.43  7.71  7.67 B
##  6     6   7.49  7.12  7.05 A
##  7     7   8.05  7.25  7.1  B
##  8     8   5.05  4.63  4.67 A
##  9     9   5.77  5.31  5.33 B
## 10    10   3.91  3.7   3.66 A
## 11    11   6.77  6.15  5.96 B
## 12    12   6.44  5.59  5.64 B
## 13    13   6.17  5.56  5.51 A
## 14    14   7.67  7.11  6.96 A
## 15    15   7.34  6.84  6.82 A
## 16    16   6.85  6.4   6.29 B
## 17    17   5.13  4.52  4.45 A
## 18    18   5.73  5.13  5.17 B
```

Q3 ***(4 Points)***

Use the `pivot_longer()` function or `gather()` function on `CHOL1a` to create a new numeric variable called `Week` that contains numeric values *4* or *8* and a new numeric variable called `Response` that contains the Cholesterol after the corresponding number of weeks. Create a new data frame called `CHOL1b` with these modifications and use `str(Chol1b)` to show that both variables have been created correctly and are indeed numeric (an integer variable is a specific type of numeric variable).

```
#
CHOL1a
```

```
## # A tibble: 18 x 5
##       ID Before   `4`   `8` Margarine
##    <dbl>  <dbl> <dbl> <dbl> <chr>
##  1     1   6.42  5.83  5.75 B
##  2     2   6.76  6.2   6.13 A
```

```
##  3     3   6.56  5.83  5.71 B
##  4     4   4.8   4.27  4.15 A
##  5     5   8.43  7.71  7.67 B
##  6     6   7.49  7.12  7.05 A
##  7     7   8.05  7.25  7.1  B
##  8     8   5.05  4.63  4.67 A
##  9     9   5.77  5.31  5.33 B
## 10    10   3.91  3.7   3.66 A
## 11    11   6.77  6.15  5.96 B
## 12    12   6.44  5.59  5.64 B
## 13    13   6.17  5.56  5.51 A
## 14    14   7.67  7.11  6.96 A
## 15    15   7.34  6.84  6.82 A
## 16    16   6.85  6.4   6.29 B
## 17    17   5.13  4.52  4.45 A
## 18    18   5.73  5.13  5.17 B
```

```
CHOL1b <- CHOL1a %>%
  pivot_longer(3:4, names_to = "Week", values_to = "Response")
CHOL1b
```

```
## # A tibble: 36 x 5
##       ID Before Margarine Week  Response
##    <dbl>  <dbl> <chr>     <chr>    <dbl>
##  1     1   6.42 B         4         5.83
##  2     1   6.42 B         8         5.75
##  3     2   6.76 A         4         6.2
##  4     2   6.76 A         8         6.13
##  5     3   6.56 B         4         5.83
##  6     3   6.56 B         8         5.71
##  7     4   4.8  A         4         4.27
##  8     4   4.8  A         8         4.15
##  9     5   8.43 B         4         7.71
## 10     5   8.43 B         8         7.67
## # ... with 26 more rows
```

```
str(CHOL1b)
```

```
## tibble [36 x 5] (S3: tbl_df/tbl/data.frame)
##  $ ID       : num [1:36] 1 1 2 2 3 3 4 4 5 5 ...
##  $ Before   : num [1:36] 6.42 6.42 6.76 6.76 6.56 6.56 4.8 4.8 8.43 8.43 ...
##  $ Margarine: chr [1:36] "B" "B" "A" "A" ...
##  $ Week     : chr [1:36] "4" "8" "4" "8" ...
##  $ Response : num [1:36] 5.83 5.75 6.2 6.13 5.83 5.71 4.27 4.15 7.71 7.67 ...
```

Q4 *(4 Points)*

Now working with `CHOL2`, we want to spread the variable `Statistic` across multiple columns. Do this in a new data frame called `CHOL2a` and use `print(CHOL2a)` to display the modified complete table.

```
#
names(CHOL2)
```

```
## [1] "Brand\tStatistic\tValue"
```

```
CHOL2a <- CHOL2 %>%
  separate("Brand\tStatistic\tValue", c("Brand", "Statistics", "Value"), "\t")
CHOL2a
```

```
## # A tibble: 9 x 3
##   Brand Statistics Value
##   <chr> <chr>      <chr>
## 1 A     Serving    14
## 2 A     Calories   70
## 3 A     Fat        7
## 4 A     SatFat     2.5
## 5 A     Sodium     130
## 6 B     Serving    14
## 7 B     Calories   50
## 8 B     Fat        6
## 9 B     SatFat     1.5
```

Q5 ***(3 Points)***

Start by examing the tables `CHOL1b` and `CHOL2a` and answering the following questions with *Yes* or *No* responses.

Is the variable `ID` in `CHOL1b` a primary key?

Answer *(1 Point)*: No

Is the variable, `Margarine` in `CHOL1b` a primary key?

Answer *(1 Point)*: No

Is the variable, `Brand` in `CHOL2a` a primary key?

Answer *(1 Point)*: No

Q6 ***(4 Points)***

Get the nutritional facts of the different margarine brands in `CHOL2a` into the experimental results found in `CHOL1b` using a join. Create a new data frame named `CHOL.COMBINED` and display the table using `head(CHOL.COMBINED)`. This final data frame should contain *36* observations and *10* variables.

```
#
CHOL.COMBINED <- CHOL1b %>%
  rename(Brand = "Margarine") %>%
  full_join(CHOL2a, by = "Brand") %>%
  pivot_wider(names_from = "Statistics", values_from = "Value")
head(CHOL.COMBINED)
```

```
## # A tibble: 6 x 10
##      ID Before Brand Week  Response Serving Calories Fat   SatFat Sodium
```

```
##   <dbl>  <dbl> <chr> <chr>    <dbl> <chr>   <chr>    <chr> <chr>  <chr>
## 1     1   6.42 B     4         5.83 14      50       6     1.5    <NA>
## 2     1   6.42 B     8         5.75 14      50       6     1.5    <NA>
## 3     2   6.76 A     4         6.2  14      70       7     2.5    130
## 4     2   6.76 A     8         6.13 14      70       7     2.5    130
## 5     3   6.56 B     4         5.83 14      50       6     1.5    <NA>
## 6     3   6.56 B     8         5.71 14      50       6     1.5    <NA>
```

```
dim(CHOL.COMBINED)
```

```
## [1] 36 10
```

Part 2: Linking Important Information to 2017 Violent Crimes Data

In the zipped folder, there are *5* CSV files. In this section, we are going to merge all of that data into one object called `FINAL.VIOLENT`.

Q1 *(2 Points)*

The dataset `S_VS_D` contains a variable `CLASS` where "S=Safe" and "D=Dangerous" according to the article *These Are the 2018 Safest and Most Dangerous States in the U.S* by Steve Karantzoulidis. We seek to compare the violent crime statistics for states not in this list. Use a filtering join to create a new data frame called `VIOLENT2` that only contains violent crime statistics from the states not represented in the data frame `S_VS_D`. Use `str(VIOLENT2)` to display the variables and the dimensions of `VIOLENT2`.

```
#

S_VS_D_2 <- S_VS_D %>%
  rename(State = "STATE")
VIOLENT2 <- VIOLENT %>%
  anti_join(S_VS_D_2, by = "State")
str(VIOLENT2)
```

```
## spec_tbl_df [68 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ State     : chr [1:68] "Arizona" "Arizona" "Arizona" "Arizona" ...
##  $ City      : chr [1:68] "Chandler" "Gilbert" "Glendale" "Mesa" ...
##  $ Population: num [1:68] 249355 242090 249273 492268 1644177 ...
##  $ Total     : num [1:68] 259.5 85.5 488.2 415.8 760.9 ...
##  $ Murder    : num [1:68] 2.01 2.07 4.81 4.67 9.55 ...
##  $ Rape      : num [1:68] 52.1 16.1 38.9 51.2 69.5 ...
##  $ Robbery   : num [1:68] 57 21.1 193 92.2 200.3 ...
##  $ Assault   : num [1:68] 148.4 46.3 251.5 267.7 481.6 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   State = col_character(),
##   ..   City = col_character(),
##   ..   Population = col_double(),
##   ..   Total = col_double(),
##   ..   Murder = col_double(),
##   ..   Rape = col_double(),
##   ..   Robbery = col_double(),
##   ..   Assault = col_double()
##   .. )
```

```
dim(VIOLENT2)
```

```
## [1] 68  8
```

Q2 *(4 Points)*

Start by creating a new data set called `VIOLENT3` based on `VIOLENT2` that fixes some problems in the variable `City`. Specifically, we would like to change "Louisville Metro" to "Louisville".

Next, create a new data frame named `VIOLENT4` that connects the population change and density measures from 2019 contained in `CENSUS` to the cities and states in `VIOLENT3`. Use `head(VIOLENT4)` to give a preview of the new merged dataset.

Finally, in a complete sentence, identify any location(s) (Cities and States) missing violent crime information.

Code and Output *(2 Points)*:

```
#
VIOLENT2$City[VIOLENT2$City == "Louisville Metro"] <- "Louisville"
VIOLENT3 <- VIOLENT2

VIOLENT4 <- CENSUS %>%
  rename(City = "Name") %>%
  full_join(VIOLENT3, by = "City") %>%
  select(-State.y, State = State.x)
VIOLENT4
```

```
## # A tibble: 315 x 10
##    City  State Change Density Population Total Murder  Rape Robbery Assault
##    <chr> <chr>  <dbl>   <dbl>      <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl>
##  1 New ~ New ~  -2.35   10676         NA   NA   NA      NA      NA       NA
##  2 Los ~ Cali~   0.07    3278    4007147  761.   7.01  61.3   270.     423.
##  3 Chic~ Illi~  -0.52    4576    2706171 1099.  24.1   65.1   439.     570.
##  4 Hous~ Texas   0.74    1405         NA   NA   NA      NA      NA       NA
##  5 Phoe~ Ariz~   4.49    1254    1644177  761.   9.55  69.5   200.     482.
##  6 Phil~ Penn~   1.02    4557    1575595  948.  20.1   75.0   382.     470.
##  7 San ~ Texas   3.67    1296         NA   NA   NA      NA      NA       NA
##  8 San ~ Cali~   1.22    1690    1424116  367.   2.46  39.2    99.0    226.
##  9 Dall~ Texas   1.93    1522         NA   NA   NA      NA      NA       NA
## 10 San ~ Cali~  -0.37    2223    1037529  404.   3.08  55.0   133.     213.
## # ... with 305 more rows
```

```
head(VIOLENT4, 10)
```

```
## # A tibble: 10 x 10
##    City  State Change Density Population Total Murder  Rape Robbery Assault
##    <chr> <chr>  <dbl>   <dbl>      <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl>
##  1 New ~ New ~  -2.35   10676         NA   NA   NA      NA      NA       NA
##  2 Los ~ Cali~   0.07    3278    4007147  761.   7.01  61.3   270.     423.
##  3 Chic~ Illi~  -0.52    4576    2706171 1099.  24.1   65.1   439.     570.
##  4 Hous~ Texas   0.74    1405         NA   NA   NA      NA      NA       NA
##  5 Phoe~ Ariz~   4.49    1254    1644177  761.   9.55  69.5   200.     482.
##  6 Phil~ Penn~   1.02    4557    1575595  948.  20.1   75.0   382.     470.
```

```
##  7 San ~ Texas   3.67    1296         NA   NA   NA     NA      NA       NA
##  8 San ~ Cali~   1.22    1690    1424116  367.   2.46  39.2    99.0    226.
##  9 Dall~ Texas   1.93    1522         NA   NA   NA     NA      NA       NA
## 10 San ~ Cali~  -0.37    2223    1037529  404.   3.08  55.0   133.     213.
```

```
VIOLENT_non_NA <- VIOLENT4 %>%
  filter(Population > 0| Total > 0 | Murder > 0 | Rape > 0 |  Robbery > 0 | Assault > 0)

VIOLENT_NA <-VIOLENT4 %>%
  anti_join(VIOLENT_non_NA, by = "City") %>%
  arrange(State)
VIOLENT_NA
```

```
## # A tibble: 244 x 10
##    City  State Change Density Population Total Murder  Rape Robbery Assault
##    <chr> <chr>  <dbl>   <dbl>      <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl>
##  1 Birm~ Alab~  -1.36     553         NA    NA     NA    NA      NA      NA
##  2 Hunt~ Alab~   3.98     363         NA    NA     NA    NA      NA      NA
##  3 Mont~ Alab~  -0.69     480         NA    NA     NA    NA      NA      NA
##  4 Mobi~ Alab~  -2.1      523         NA    NA     NA    NA      NA      NA
##  5 Tusc~ Alab~   0.85     545         NA    NA     NA    NA      NA      NA
##  6 Anch~ Alas~  -4.18      65         NA    NA     NA    NA      NA      NA
##  7 Tempe Ariz~   7.33    1890         NA    NA     NA    NA      NA      NA
##  8 Peor~ Ariz~   7.1      387         NA    NA     NA    NA      NA      NA
##  9 Surp~ Ariz~   7.84     507         NA    NA     NA    NA      NA      NA
## 10 Litt~ Arka~  -0.64     642         NA    NA     NA    NA      NA      NA
## # ... with 234 more rows
```

Answer *(2 Points)*: In the dataset VIOLENT_NA, various cities, especially those in Alabama, Texas, Arizona, California, Colorado, Connecticut, Florida, Illinois, are missing datavalues for crime.

Q3 *(6 Points)*

Either ambitiously using one step or less-ambitiously using multiple steps add the longitude and latitude information provided in `ZIP` to the cities and states in `VIOLENT4`. You will need to use `STATE_ABBREV` data to link these two data frames. Your final data frame named `FINAL.VIOLENT` should contain all of the information in `VIOLENT4` along with the variables `lat` and `lon` from `ZIP`. There should be **no** state abbreviations in `FINAL.VIOLENT` since this information is redundant. Use `str(FINAL.VIOLENT)` to demonstrate that everything worked as planned.

In `FINAL.VIOLENT` identify what cities are missing latitude and longitude. Closely, inspect both the `ZIP` and `VIOLENT4` data frames. Report the location(s) missing geographical information and explain in complete sentences why this happened.

Finally, challenge yourself and attempt to fix this problem in a new data frame called `FINAL.VIOLENT.FIX`. Use a combination of `str()` and `filter()` to only display the data in `FINAL.VIOLENT.FIX` for the location(s) that `FINAL.VIOLENT` was missing latitude and longitude. Do this in the second code chunk below.

Code and Output *(4 Points)*:

```
#
ZIP_v2 <- ZIP %>%
  rename(City = "city") %>%
  full_join(STATE_ABBREV, by = "state")
```

```
FINAL.VIOLENT <- VIOLENT4 %>%
  full_join(ZIP_v2, by = "City") %>%
  rename(State = "State.y", State_code = "state") %>%
  arrange(City)


Missing_Long_Lat <- FINAL.VIOLENT %>%
  filter(is.na(lon) | is.na(lat)) %>%
  arrange(City)
Missing_Long_Lat
```

```
## # A tibble: 12 x 14
##    City       State.x  Change Density Population Total Murder  Rape Robbery
##    <chr>      <chr>     <dbl>   <dbl>      <dbl> <dbl>  <dbl> <dbl>   <dbl>
##  1 "Coral Sp~ Florida    2.91    2171         NA    NA     NA    NA      NA
##  2 "Davie"    Florida    4.34    1176         NA    NA     NA    NA      NA
##  3 "Jurupa V~ Califor~   5.78     986         NA    NA     NA    NA      NA
##  4 "Miami Ga~ Florida   -2.61    2335         NA    NA     NA    NA      NA
##  5 "Miramar"  Florida    2.05    1855         NA    NA     NA    NA      NA
##  6 "New York~ New York  -2.35   10676         NA    NA     NA    NA      NA
##  7 "Overland~ Kansas     3.51    1005         NA    NA     NA    NA      NA
##  8 "Spokane ~ Washing~   0.04    1035         NA    NA     NA    NA      NA
##  9 "St. Loui~ Missouri  -3.48    1872         NA    NA     NA    NA      NA
## 10 "St. Pete~ Florida    1.62    1657         NA    NA     NA    NA      NA
## 11 "West Val~ Utah      -0.9     1472         NA    NA     NA    NA      NA
## 12 "Winston?~ North C~   2.33     722         NA    NA     NA    NA      NA
## # ... with 5 more variables: Assault <dbl>, State_code <chr>, lat <dbl>,
## #   lon <dbl>, State <chr>
```

Answer *(1 Points)*: In the Missing_Long_lat dataset, there are 13 observations with either missing lat or lon values, because the 13 cities in the VIOLENT4 dataset do not have a corresponding observation of the same City value – the primary key – in the ZIP dataset.

Code and Output *(1 Point)*:

```
#
str(FINAL.VIOLENT$lat)
```

```
##  num [1:30536] 40.9 31.6 32 30 34.5 ...
```

```
str(FINAL.VIOLENT$lon)
```

```
##  num [1:30536] -77.4 -85.2 -83.3 -92.2 -89.5 ...
```

```
FINAL.VIOLENT.FIX <- FINAL.VIOLENT %>%
  filter(is.na(lon) | is.na(lat)) %>%
  select(-"State") %>%
  rename(State = "State.x") %>%
  select(City, State, lat, lon, everything())
FINAL.VIOLENT.FIX
```

```
## # A tibble: 12 x 13
##    City     State    lat   lon Change Density Population Total Murder  Rape
##    <chr>    <chr>  <dbl> <dbl>  <dbl>   <dbl>      <dbl> <dbl>  <dbl> <dbl>
##  1 "Coral ~ Flori~    NA    NA   2.91    2171         NA    NA     NA    NA
##  2 "Davie"  Flori~    NA    NA   4.34    1176         NA    NA     NA    NA
##  3 "Jurupa~ Calif~    NA    NA   5.78     986         NA    NA     NA    NA
##  4 "Miami ~ Flori~    NA    NA  -2.61    2335         NA    NA     NA    NA
##  5 "Mirama~ Flori~    NA    NA   2.05    1855         NA    NA     NA    NA
##  6 "New Yo~ New Y~    NA    NA  -2.35   10676         NA    NA     NA    NA
##  7 "Overla~ Kansas    NA    NA   3.51    1005         NA    NA     NA    NA
##  8 "Spokan~ Washi~    NA    NA   0.04    1035         NA    NA     NA    NA
##  9 "St. Lo~ Misso~    NA    NA  -3.48    1872         NA    NA     NA    NA
## 10 "St. Pe~ Flori~    NA    NA   1.62    1657         NA    NA     NA    NA
## 11 "West V~ Utah      NA    NA  -0.9     1472         NA    NA     NA    NA
## 12 "Winsto~ North~    NA    NA   2.33     722         NA    NA     NA    NA
## # ... with 3 more variables: Robbery <dbl>, Assault <dbl>,
## #   State_code <chr>
```

Part 3: Web Scraping a Table From Wikipedia

Wikipedia contains a rough estimate of a billion tables. Search through Wikipedia pages and identify an article, completely unrelated to crimes data, that contains an HTML table that has at least *5* rows and *3* columns. You will be required to web scrape the table into a data frame or tibble into R. This portion will require a minor knowledge of the `rvest` package. Utilize information from the web scraping lectures and tutorials to assist you with this.

Q1 *(4 Points)*

What is the URL of the Wikipedia page you plan on webscraping (Knit the Document and Check the Hyperlink)?

Answer *(2 Points)*: Fastest recorded tennis serves

In *2* to *5* sentences, Identify and describe the specific table you plan on web scraping. State the variables in *1* of the sentences.

Answer *(2 Points)*: The table I plan to web scrape is highest recorded men's and women's tennis serves, but they have different tables, for men and women. There are multiple variables, such as Player(Player name), speed(in both mph and km/h), Event(Tournament that serve occured), Type(Singles or Doubles), and Round(which stage of the tournament). The list is potentially missing some entries, due to the fact that some serves are not verifiable using modern technology. Also, the serves listed are only the personal best for the players. Some players could have multiple entries, but are only limited to one–their personal best– on the table.

Q2 *(4 Points)*

Utilize the functions `read_html()` and `html_table()` to web scrape the specific table you described above. Internet access will be required for these functions to work. Create an R data frame named `DATA` which contains the information from the Wikipedia table. All code should be contained in the R code chunk below. Finally, use the `print()` function to display the table to demonstrate that everything worked as planned. The variable names and the content should match the table on the Wikipedia page you chose exactly. You are not required to perform any cleaning of this data. As long as the content of the table you describe matches `DATA`, then you are good. Don't worry if the table bleeds over multiple pages.

```
#
URL.SERVES = "https://en.wikipedia.org/wiki/Fastest_recorded_tennis_serves"
DATA = URL.SERVES %>%
          read_html() %>%
          html_nodes(css=".headerSort , td , .Template-Fact span") %>%
          html_text
DATA
```

```
##   [1] "1"
##   [2] " Sam Groth"
##   [3] "263.0 km/h (163.4 mph)"
##   [4] "2012 Busan Open Challenger Tennis\n"
##   [5] "Singles\n"
##   [6] "2R\n"
##   [7] "[9]"
##   [8] "2"
##   [9] " Albano Olivetti"
##  [10] "257.5 km/h (160.0 mph)"
##  [11] "2012 Internazionali Trofeo Lame Perrel-Faip"
##  [12] "Singles\n"
##  [13] "1R\n"
##  [14] "[10]"
##  [15] "3"
##  [16] " John Isner"
##  [17] "253.0 km/h (157.2 mph)"
##  [18] "2016 Davis Cup"
##  [19] "Singles\n"
##  [20] "1R\n"
##  [21] "[11]"
##  [22] "4"
##  [23] " Ivo Karlović"
##  [24] "251.0 km/h (156.0 mph)"
##  [25] "2011 Davis Cup\n"
##  [26] "Doubles\n"
##  [27] "1R\n"
##  [28] "[12]"
##  [29] " Jerzy Janowicz[note 1]"
##  [30] "251.0 km/h (156.0 mph)"
##  [31] "2012 Pekao Szczecin Open"
##  [32] "Singles\n"
##  [33] "1R\n"
##  [34] "[14]"
##  [35] "6"
##  [36] " Andy Roddick"
##  [37] "249.4 km/h (155.0 mph)"
##  [38] "2004 Davis Cup"
##  [39] "Singles\n"
##  [40] "SF\n"
##  [41] "[15]"
##  [42] " Milos Raonic"
##  [43] "249.4 km/h (155.0 mph)"
##  [44] "2012 SAP Open"
##  [45] "Singles\n"
##  [46] "SF\n"
```

```
##  [47] "[16]"
##  [48] "8"
##  [49] " Joachim Johansson"
##  [50] "244.6 km/h (152.0 mph)"
##  [51] "2004 Davis Cup\n"
##  [52] "Doubles\n"
##  [53] "1R\n"
##  [54] "[17]"
##  [55] " Ryan Harrison"
##  [56] "244.6 km/h (152.0 mph)"
##  [57] "2013 Western & Southern Open\n"
##  [58] "Singles\n"
##  [59] "2R\n"
##  [60] "[18]"
##  [61] " Feliciano López"
##  [62] "244.6 km/h (152.0 mph)"
##  [63] "2014 Aegon Championships\n"
##  [64] "Singles\n"
##  [65] "1R\n"
##  [66] "[19]"
##  [67] "11"
##  [68] " Marius Copil"
##  [69] "244.0 km/h (151.6 mph)"
##  [70] "2016 European Open"
##  [71] "Singles\n"
##  [72] "QF\n"
##  [73] "[20]"
##  [74] "12"
##  [75] " Hubert Hurkacz"
##  [76] "243.0 km/h (151.0 mph)"
##  [77] "2016 Davis Cup"
##  [78] "Singles\n"
##  [79] "1R\n"
##  [80] "[20]"
##  [81] "13"
##  [82] " Taylor Dent"
##  [83] "241.0 km/h (149.8 mph)"
##  [84] "2006 ABN AMRO World Tennis Tournament"
##  [85] "Singles\n"
##  [86] "1R\n"
##  [87] "[16]"
##  [88] "14"
##  [89] " Juan Martín del Potro"
##  [90] "240.0 km/h (149.1 mph)"
##  [91] "2017 Stockholm Open"
##  [92] "Singles\n"
##  [93] "F\n"
##  [94] "[21]"
##  [95] "15"
##  [96] " Greg Rusedski"
##  [97] "239.8 km/h (149.0 mph)"
##  [98] "1998 Newsweek Champions Cup"
##  [99] "Singles\n"
## [100] "SF\n"
```

```
## [101] "[22]"
## [102] "16"
## [103] " Dmitry Tursunov"
## [104] "237.0 km/h (147.3 mph)"
## [105] "2006 Davis Cup"
## [106] "Singles\n"
## [107] "SF\n"
## [108] "[citation needed]"
## [109] "citation needed"
## [110] " Jo-Wilfried Tsonga"
## [111] "237.0 km/h (147.3 mph)"
## [112] "2014 Rogers Cup"
## [113] "Singles\n"
## [114] "QF\n"
## [115] "[23]"
## [116] " Frances Tiafoe"
## [117] "237.0 km/h (147.3 mph)"
## [118] "2018 Estoril Open"
## [119] "Singles\n"
## [120] "F\n"
## [121] "[citation needed]"
## [122] "citation needed"
## [123] " Taylor Fritz"
## [124] "237.0 km/h (147.3 mph)"
## [125] "2020 US Open"
## [126] "Singles\n"
## [127] "3R\n"
## [128] "[24]"
## [129] "20"
## [130] " Fernando González"
## [131] "236.0 km/h (146.6 mph)"
## [132] "2007 Italian Open"
## [133] "Singles\n"
## [134] "SF\n"
## [135] "[25]"
## [136] "21"
## [137] " Gaël Monfils"
## [138] "235.0 km/h (146.0 mph)"
## [139] "2007 Legg Mason Tennis Classic"
## [140] "Singles\n"
## [141] "QF\n"
## [142] "[16]"
## [143] " Dušan Vemić"
## [144] "235.0 km/h (146.0 mph)"
## [145] "2008 Countrywide Classic"
## [146] "Singles\n"
## [147] "?\n"
## [148] "[16]"
## [149] " Marin Čilić"
## [150] "235.0 km/h (146.0 mph)"
## [151] "2016 Davis Cup"
## [152] "Singles\n"
## [153] "1R\n"
## [154] "[citation needed]"
```

```
## [155] "citation needed"
## [156] " Reilly Opelka"
## [157] "235.0 km/h (146.0 mph)"
## [158] "2019 Swiss Indoors"
## [159] "Singles\n"
## [160] "QF\n"
## [161] "[citation needed]"
## [162] "citation needed"
## [163] " Matteo Berrettini\n"
## [164] "235.0 km/h (146.0 mph)\n"
## [165] "2021 Mutua Madrid Open\n"
## [166] "Singles\n"
## [167] "F\n"
## [168] "\n"
## [169] "26\n"
## [170] " Ivan Ljubičić\n"
## [171] "234.0 km/h (145.4 mph)\n"
## [172] "2005 Mutua Madrileña Masters Madrid\n"
## [173] "Singles\n"
## [174] "F\n"
## [175] "[citation needed]"
## [176] "citation needed"
## [177] " Stan Wawrinka"
## [178] "246.0 km/h (152.9 mph)\n"
## [179] "2008 Beijing Olympics\n"
## [180] "Doubles\n"
## [181] "SF\n"
## [182] "\n"
## [183] "28"
## [184] " Grigor Dimitrov"
## [185] "233.4 km/h (145.0 mph)"
## [186] "2013 Aegon Championships\n"
## [187] "Singles\n"
## [188] "?\n"
## [189] "[citation needed]"
## [190] "citation needed"
## [191] " Viktor Troicki"
## [192] "233.4 km/h (145.0 mph)"
## [193] "2017 Davis Cup\n"
## [194] "Singles\n"
## [195] "?\n"
## [196] "[citation needed]"
## [197] "citation needed"
## [198] "30"
## [199] " Nicolás Jarry"
## [200] "233.0 km/h (144.8 mph)"
## [201] "2018 Davis Cup\n"
## [202] "?\n"
## [203] "1R\n"
## [204] "[26]"
## [205] "31\n"
## [206] " Fernando Verdasco\n"
## [207] "232.0 km/h (144.2 mph)"
## [208] "2009 French Open\n"
```

```
## [209] "?\n"
## [210] "?\n"
## [211] "[16]"
## [212] " Dominic Thiem"
## [213] "232.0 km/h (144.2 mph)"
## [214] "2017 Gerry Weber Open\n"
## [215] "?\n"
## [216] "?\n"
## [217] "[citation needed]"
## [218] "citation needed"
## [219] "33"
## [220] " Mardy Fish\n"
## [221] "231.7 km/h (144 mph)"
## [222] "2007 Pacific Life Open\n"
## [223] "Singles\n"
## [224] "1R\n"
## [225] "[citation needed]"
## [226] "citation needed"
## [227] " Alexander Zverev\n"
## [228] "231 km/h (144 mph)\n"
## [229] "2020 ATP Finals\n"
## [230] "Singles\n"
## [231] "RR\n"
## [232] "[citation needed]"
## [233] "citation needed"
## [234] " Marcin Matkowski"
## [235] "231.7 km/h (144 mph)"
## [236] "2009 ATP World Tour Finals\n"
## [237] "Doubles\n"
## [238] "?\n"
## [239] "[citation needed]"
## [240] "citation needed"
## [241] "36"
## [242] " Robin Söderling\n"
## [243] "230.1 km/h (143.0 mph)"
## [244] "2010 ATP World Tour Finals\n"
## [245] "Singles\n"
## [246] "RR\n"
## [247] "[citation needed]"
## [248] "citation needed"
## [249] " Nick Kyrgios"
## [250] "230.1 km/h (143.0 mph)"
## [251] "2019 Wimbledon"
## [252] "Singles\n"
## [253] "2R\n"
## [254] "[27]"
## [255] " Roger Federer"
## [256] "230.1 km/h (143.0 mph)"
## [257] "2010 Gerry Weber Open\n"
## [258] "?\n"
## [259] "?\n"
## [260] "[28]"
## [261] "39"
## [262] " Martin Verkerk\n"
```

```
## [263] "230.0 km/h (142.9 mph)"
## [264] "2003 Breil Milano Indoor\n"
## [265] "?\n"
## [266] "?\n"
## [267] "[citation needed]"
## [268] "citation needed"
## [269] " Nicolás Almagro\n"
## [270] "230.0 km/h (142.9 mph)\n"
## [271] "2016 Argentina Open\n"
## [272] "?\n"
## [273] "?\n"
## [274] "[citation needed]"
## [275] "citation needed"
## [276] "1\n"
## [277] " Georgina Garcia Pérez"
## [278] "220.0 km/h (136.7 mph)"
## [279] "2018 Hungarian Ladies Open"
## [280] "[30]"
## [281] "2\n"
## [282] " Aryna Sabalenka\n"
## [283] "214.0 km/h (133.0 mph)\n"
## [284] "2018 WTA Elite Trophy"
## [285] "[31]"
## [286] "3"
## [287] " Sabine Lisicki"
## [288] "210.8 km/h (131.0 mph)"
## [289] "2014 Stanford Classic"
## [290] "[32]"
## [291] "4\n"
## [292] " Brenda Schultz-McCarthy\n"
## [293] "209.2 km/h (130.0 mph)\n"
## [294] "2006 Cincinnati Masters (qualifiers)"
## [295] "[33]"
## [296] "5"
## [297] " Venus Williams"
## [298] "207.6 km/h (129.0 mph)"
## [299] "2007 US Open"
## [300] "[citation needed]"
## [301] "citation needed"
## [302] "6"
## [303] " Serena Williams\n"
## [304] "207.0 km/h (128.6 mph)"
## [305] "2013 Australian Open"
## [306] "[34]"
## [307] " Ivana Jorović"
## [308] "207.0 km/h (128.6 mph)"
## [309] "2017 Fed Cup"
## [310] "[citation needed]"
## [311] "citation needed"
## [312] "8"
## [313] " Julia Görges"
## [314] "203.0 km/h (126.1 mph)"
## [315] "2012 French Open"
## [316] "[citation needed]"
```

```
## [317] "citation needed"
## [318] " Caroline Garcia"
## [319] "203.0 km/h (126.1 mph)"
## [320] "2016 Fed Cup"
## [321] "[citation needed]"
## [322] "citation needed"
## [323] "10"
## [324] " Brenda Schultz-McCarthy"
## [325] "202.7 km/h (126.0 mph)"
## [326] "2007 Indian Wells Masters"
## [327] "[citation needed]"
## [328] "citation needed"
## [329] "11"
## [330] " Nadiya Kichenok"
## [331] "202.0 km/h (125.5 mph)"
## [332] "2014 Australian Open"
## [333] "[citation needed]"
## [334] "citation needed"
## [335] "12\n"
## [336] " Lucie Hradecká\n"
## [337] "201.2 km/h (125.0 mph)"
## [338] "2015 Wimbledon"
## [339] "[citation needed]"
## [340] "citation needed"
## [341] " Naomi Osaka"
## [342] "201.2 km/h (125.0 mph)"
## [343] "2016 US Open"
## [344] "[citation needed]"
## [345] "citation needed"
## [346] "14"
## [347] " Anna-Lena Grönefeld"
## [348] "201.1 km/h (125.0 mph)"
## [349] "2009 Indian Wells Masters"
## [350] "[citation needed]"
## [351] "citation needed"
## [352] "15"
## [353] " Ana Ivanovic"
## [354] "201.0 km/h (124.9 mph)"
## [355] "2007 French Open"
## [356] "[citation needed]"
## [357] "citation needed"
## [358] " Denisa Allertová"
## [359] "201.0 km/h (124.9 mph)"
## [360] "2015 Australian Open"
## [361] "[35]"
## [362] "17"
## [363] " Kristina Mladenovic"
## [364] "200.0 km/h (124.3 mph)"
## [365] "2009 French Open"
## [366] "[citation needed]"
## [367] "citation needed"
## [368] "Singles\nMen's champions\nChronological\nWomen's champions\nChronological\nFinals\nOpen Era\nM
## [369] "\nMen's champions\nChronological\nWomen's champions\nChronological\nFinals\nOpen Era\nMen's fi
## [370] "\nMen's champions\nWomen's champions\nMixed champions\nChampions by country\nBoys' champions\n
```

```
## [371] "Singles\nOpen Era records\nAll-time records\nATP records\nMasters records\nBig Titles champion
## [372] "\nOpen Era records\nAll-time records\nATP records\nMasters records\nBig Titles champions\nRank
## [373] "\nRankings\nWorld No. 1\nWorld Champions\nMasters champions\nBig Titles champions\nTop players
## [374] "\nOpen Era records\nAll-time records\nWTA records\nTier I/Premier champions\nRankings\nWorld N
## [375] "\nNotable rivalries\nOpen Era\nTitles leaders\nPlayers' achievements\nRanking per country\nTou
## [376] "\nMatches\nLongest\nShortest\nBagel\nGolden set\nTiebreaker\nServing\nSpeed\nAces\nDouble faul
```