Prep3

Sebastian Montesinos

Due by midnight, Monday Feb. 28

Reminder: Prep assignments are to be completed individually. Upload a final copy of the .Rmd and renamed .pdf to your private repo, and submit the renamed pdf to Gradescope.

Reading

The associated reading for the week is Sections 6.4, 8.5-8.7, and 8.9-8.10. This reading explores data intake (getting data into R from a variety of data formats), and ethics issues.

Remember, I recommend you code along with the book examples. You can try out the code yourself - just be sure to load the mdsr package and any other packages referenced. You can get the code in R script files (basically, files of just R code, not like a .Rmd) from the book website.

1 - Data Intake Basics

part a - Many of our data sets have been provided as .csv files. What does .csv stand for?

Solution: CVS stands for comma separated values, a non-compressed, widely used format for data exchange.

part b - Name one web-related data-table friendly format.

Solution: One data-table friendly format this is web-related is HTML, or hypertext markup language.

part c - The *haven* package is designed to help import files from certain other apps into R. From its help file, list three apps it can import data files from.

Hint: You may need to install *haven* if you want to look this up within R. Searching it on the web is also fine for this problem.

Solution: Haven can import files from Stata, SAS, and SPSS.

part d - In Chapter 19, in our unit on text analysis, we will use the *aRxiv* package. From its help file, this package is an "Interface to the arXiv API". What does this mean?

Hint: arXiv is a website that hosts scholarly articles, often pre-prints, that are not peer-reviewed. (So, a later version of the paper might be peer-reviewed and published elsewhere.) The key to answering the question posed is the API part. Put another way, what does this package help you to do in relation to accessing the data stored by arXiv?

Solution: An API is a protocol for interacting with a program, so this package would likely provide access to public API for the arXiv website. This would provide us with the R functions we would need to access the actual data.

2 - Web scraping

In Section 6.4.1.2, the *rvest* package is used to scrape a Wikipedia page. BUT WAIT! While we may have the technical ability to scrape a webpage, that doesn't necessarily mean we are *allowed* to scrape it.

Before scraping a web page, you should always check whether doing so is allowed.

Sometimes this information is listed on the page or in an EULA.

If you're unsure of the permissions for a particular domain, you can use the handy `paths_allowed()` function within the *robotstxt* package.

> part a - Check the permissions for the Wikipedia page using the code below. If the code returns "TRUE", then that indicates a bot has permission to access the page. Do you (via R) have permission to access the page?

Solution: I do have permission to access this wikipedia page via R.

```
# Define url to use again
url <- "https://en.wikipedia.org/wiki/Mile_run_world_record_progression"

# Check bot permissions
paths_allowed(url)
```

```
##  en.wikipedia.org
```

```
## [1] TRUE
```

> part b - Now, use the code chunk below to follow along with the code in Section 6.4.1.2 to scrape the tables from the Wikipedia page on *Mile run world record progression*. Use `length(tables)` to identify how many tables are in the object you created called `tables`. How many tables are there?

Solution: There are 12 elements in the tables object I created.

```
url <- "http://en.wikipedia.org/wiki/Mile_run_world_record_progression"
tables <- url %>%
  read_html() %>%
  html_elements("table")
length(tables)
```

```
## [1] 12
```

Next, look at the Wikipedia page. We want to work with the table toward the bottom titled "Women Indoor IAAF era" that shows four records: one for Mary Decker, two for Doina Melinte, and one for Genzebe Dibaba.

> part c - From your `tables` object created in part b, create a dataframe called `women_indoor` that includes this "Women Indoor IAAF era" table data.

Hint: You can use the same code as used in the textbook to create the `amateur` and `records` tables, except you'll need to update the table number that's `plucked`.

Solution:

```
amateur <- tables %>%
  purrr::pluck(10) %>%
  html_table()
```

part d - Use `kable()` to display the table from part c. Who holds the indoor one-mile world record for IAAF women, and what was her time?

Solution:

```
amateur %>%
  kable(booktabs = TRUE)
```

Time	Athlete	Nationality	Date	Venue
4:20.5	Mary Decker	United States	February 19, 1982	San Diego United States
4:18.86	Doina Melinte	Romania	February 13, 1988	East Rutherford United States
4:17.14	Doina Melinte	Romania	February 9, 1990	East Rutherford United States
4:13.31	Genzebe Dibaba	Ethiopia	February 17, 2016	Stockholm Sweden

part e - Perform the necessary wrangling as directed to answer the question below.

Read through all the desired items before beginning!

- Create a dataframe called `women_outdoor` that contains the table for "Women's IAAF era" (starting with Anne Smith's record and ending with Sifan Hassan's record).
- Combine `women_indoor` and `women_outdoor` into one dataframe called `women_records` using the `bind_rows()` function.
- Include a variable called `Type` in this new dataframe to indicate whether a particular observation corresponds to an indoor record or an outdoor record (Hint: create `Type` separately in each dataframe before combining).
- Finally, arrange `women_records` by ascending time, drop the `Venue` variable, and display the table using `kable()`.
- Use your wrangled data set to answer this question: Who holds the fastest record, and was it from an indoor or outdoor event?

Solution:

3 - Ethics

As we wrap up the chapter on ethics, what are three major takeaways from Chapter 8 that had an impact on how you think about approaching your work as a budding data scientist?

Solution:

P.S. We're having an ethics discussion in class Tuesday, and the scraping material will be explored more on Thursday.