STAT 231: Problem Set 4A

Mike Santos

due by 5 PM on Monday, September 21

In order to most effectively digest the textbook chapter readings – and the new R commands each presents – series A homework assignments are designed to encourage you to read the textbook chapters actively and in line with the textbook's Prop Tip of page 33:

"**Pro Tip**: If you want to learn how to use a particular command, we highly recommend running the example code on your own"

A more thorough reading and light practice of the textbook chapter prior to class allows us to dive quicker and deeper into the topics and commands during class. Furthermore, learning a programming lanugage is like learning any other language – practice, practice, practice is the key to fluency. By having two assignments each week, I hope to encourage practice throughout the week. A little coding each day will take you a long way!

Series A assignments are intended to be completed individually. While most of our work in this class will be collaborative, it is important each individual completes the active readings. The problems should be straightforward based on the textbook readings, but if you have any questions, feel free to ask me!

Steps to proceed:

1. In RStudio, go to File > Open Project, navigate to the folder with the course-content repo, select the course-content project (course-content.Rproj), and click "Open"
2. Pull the course-content repo (e.g. using the blue-ish down arrow in the Git tab in upper right window)
3. Copy ps4A.Rmd from the course repo to your repo (see page 6 of the GitHub Classroom Guide for Stat231 if needed)
4. Close the course-content repo project in RStudio
5. Open YOUR repo project in RStudio
6. In the ps4A.Rmd file in YOUR repo, replace "YOUR NAME HERE" with your name
7. Add in your responses, committing and pushing to YOUR repo in appropriate places along the way
8. Run "Knit PDF"
9. Upload the pdf to Gradescope. Don't forget to select which of your pages are associated with each problem. *You will not get credit for work on unassigned pages (e.g., if you only selected the first page but your solution spans two pages, you would lose points for any part on the second page that the grader can't see).*

1. Web scraping

a.

In Section 5.5.1, the `rvest` package is used to scrape a Wikipedia page. BUT **WAIT**! While we techinically might be able to scrape a webpage, that doesn't necessarily mean we are allowed to. **ETHICS ALERT!** Before scraping a web page, you should always check whether doing so is allowed. If you're unsure of the permissions for a particular domain, you can use the handy `paths_allowed` function within the `robotstext` package.

Check the permissions for the Wikipedia page using the code below. Uncomment the code. You should get a response "TRUE", indicating that a bot has permissions to access this page.

```
# you may need to install the robotstext package if you're on your machine and haven't used it before
library(robotstxt)
paths_allowed("https://en.wikipedia.org/wiki/Mile_run_world_record_progression")
```

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

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

b.

Now, follow the code in the textbook to scrape the tables from the Wikipedia page on "Mile run world record progression" (e.g., pages 118-120). Use `length(tables)` to identify how many tables are in the object you created called `tables`. How many tables are there? (Note: the Wikipedia page has been updated since the first edition of the textbook was printed, so the number should be different than that in the textbook!)

> ANSWER: There appear to be 12 tables.

```
#disaster
url = read_html("https://en.wikipedia.org/wiki/Mile_run_world_record_progression")
tables = url %>% html_nodes(css="table")
length(tables)
```

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

c.

Next, look at the Wikipedia page: https://en.wikipedia.org/wiki/Mile_run_world_record_progression. The table toward the bottom titled "Women Indoor IAAF era" shows four records – one for Mary Decker, two for Doina Melinte, and one for Genzebe Dibaba.

Suppose we want to work in R with this "Women Indoor IAAF era" table. From your `tables` object created in part b, create a dataframe called `women_indoor` that includes this "Women Indoor IAAF era" table. You can use the same code as used in the textbook to create `Table3` and `Table4`, except you'll need to update the number within the double brackets to correspond to the correct table. You'll likely need to look at a number of different tables in the `tables` object before finding which one corresponds to the "Women Indoor IAAF" table. Print the table. Who holds the indoor one-mile world record for IAAF women, and what was her time?

> ANSWER: The indoor one-mile world record for IAAF women is held by Genezebe Dibaba and her time was 4:13.31

```r
women_indoor = html_table(tables[[10]])
women_indoor
```

```
##      Time        Athlete   Nationality              Date
## 1  4:20.5    Mary Decker United States February 19, 1982
## 2 4:18.86  Doina Melinte       Romania February 13, 1988
## 3 4:17.14  Doina Melinte       Romania  February 9, 1990
## 4 4:13.31 Genzebe Dibaba      Ethiopia February 17, 2016
##                             Venue
## 1       San Diego  United States
## 2 East Rutherford  United States
## 3 East Rutherford  United States
## 4               Stockholm  Sweden
```

```r
women_indoor$type = ("Indoor")
```

d. Lastly:

- 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` 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)
- arrange `women` by ascending time, and identify the fastest world record

Is the fastest record from an indoor or outdoor event?

> ANSWER: The fastest time was 4:12.33 by Sifan Hassan, thus it was from an outdoor event.

```r
women_outdoor = html_table(tables[[8]])
women_outdoor$type = ("Outdoor")
women = bind_rows(women_outdoor, women_indoor)
min(women$Time)
```

```
## [1] "4:12.33"
```