

Homework 3

Due: April 16, 2018 @ 11:59pm

Instructions

For this homework assignment, you will practice using the [SelectorGadget Chrome extension](#) to find the CSS selectors needed to scrape information from a webpage and use the `rvest` package to scrape data from the [official Mason Patriots sports website](#).

Obtain the [Github repository you will use to complete homework 3](#) that contains a starter RMarkdown file named `homework_3.Rmd`, which you will use to do your work and write-up when completing the questions below. Remember to fill in your name at the top of the RMarkdown document and be sure to save, commit, and push (upload) frequently to Github so that you have incremental snapshots of your work. When you're done, follow the [How to submit](#) section below to setup a Pull Request, which will be used for feedback.

Part 1 – SelectorGadget practice

One of the ways to target specific information on the webpage is through the use of CSS selectors, and becoming more comfortable with them will help you build more effective webscraping code. The [SelectorGadget Chrome extension](#) is a convenient tool for determining the CSS selectors you need for a webscraping task. For a refresher on how to use the extension, [review the SelectorGadget vignette](#).

For this part of the homework, use the SelectorGadget tool to figure out the CSS selectors needed to scrape specific data from a linked page. **You only need to report the CSS selectors needed to get the information and do not need to write `rvest` code to formally extract it**, although you are welcome to write code to test the CSS selectors if you like.

1. Using the SelectorGadget tool, find the CSS selectors for the following information on the IMDB page for the television show *The Office*, <https://www.imdb.com/title/tt0386676/>:
 - Number of episodes
 - Certificate (TV Rating)
 - First five plot keywords
 - Genres
 - Runtime
 - Country
 - Language
2. Using the SelectorGadget tool, find the CSS selectors for the following information on the data.gov *Data Catalog*, <https://catalog.data.gov/dataset>:
 - Number of datasets found
 - Dataset names (example: *Demographic Statistics By Zip Code*)
 - Dataset organization (example: *City of New York*)
 - Dataset description (example: "Demographic statistics broken down by zip code")
 - Dataset type (the ribbons on the upper-right of each row, for example: *Federal, City*)

Part 2 – Scraping Mason Patriots Scores

Webscrapers can be used for all kinds of purposes, such as building movie review databases, tracking prices for goods and services, and analyzing how a news story is reported on different news sites. Collecting sports data is another example, which can be used to [quantify how valuable players are](#) when [putting together a fantasy sports team](#). Actual sports teams also employ statistical methods when drafting players and developing strategies, with [Sabermetrics](#) (depicted in the movie *Moneyball*) being one of the better-known examples.

For this part of the homework, we will scrape the 2017-2018 season schedules and scores for the men's and women's basketball teams on the official [Mason Patriots sports site](#). For reference, the 2017-2018 schedule and scores page for the men's team should look like this:

The screenshot shows the '2017-18 Men's Basketball Schedule' page. At the top, there's a navigation bar with links like 'News', 'Roster', 'Coaches', 'Schedule/Results', 'Statistics', 'Game Notes (PDF)', 'Prospectus (PDF)', 'Practice Facility', and 'More Links+'. Below this is a 'Game of the Week 2017-2018 Season' section. The main content area features a table with overall record (16-17), PCT (.485), CONF (9-9), PCT (.500), STREAK (L1), HOME (10-7), AWAY (5-7), and NEUTRAL (1-3). Below the table, there are game entries for November 10 (vs LAFAYETTE), November 12 (at #16 LOUISVILLE), November 16 (vs BINGHAMTON), and November 18 (vs CSUN). Each game entry includes the date, time, location, opponent, and final score.

OVERALL	PCT	CONF	PCT	STREAK	HOME	AWAY	NEUTRAL
16-17	.485	9-9	.500	L1	10-7	5-7	1-3

Game of the Week 2017-2018 Season

2017-18 Men's Basketball Schedule

Print | Subscribe with... | All Games | 2017-18

DATE	TIME	LOCATION	OPPONENT	SCORE
NOV 10 (FRI)	7:00 P.M.	FAIRFAX, VA. / EAGLEBANK ARENA	VS LAFAYETTE	W, 67-65
NOV 12 (SUN)	2:00 P.M.	LOUISVILLE, KY. / TV: NBC SPORTS WASHINGTON/RSN	AT #16 LOUISVILLE	L, 61-72
NOV 16 (THU)	7:00 P.M.	FAIRFAX, VA. / EAGLEBANK ARENA	VS BINGHAMTON	W, 69-57
NOV 18 (SAT)	6:00 P.M.	FAIRFAX, VA. / EAGLEBANK ARENA	VS CSUN	W, 78-73 (OT)
NOV 21 (TUE)	6 P.M.	FAIRFAX, VA. / EAGLEBANK ARENA	VS CSUN	L, 61-72

and the 2017-2018 schedule and scores page for the women's team should look like this:

The screenshot shows the '2017-18 Women's Basketball Schedule' page. At the top, there's a navigation bar with links like 'News', 'Roster', 'Coaches', 'Schedule/Results', 'Statistics', 'Butler/DiDi', and 'More Links+'. Below this is a 'Game of the Week 2017-2018 Season' section. The main content area features a table with overall record (24-10), PCT (.706), CONF (11-5), PCT (.688), STREAK (L1), HOME (17-2), AWAY (6-7), and NEUTRAL (1-1). Below the table, there are game entries for November 10 (at NO. 24/23 UNIVERSITY OF MICHIGAN (FIRST ROUND)), November 14 (at LOYOLA UNIVERSITY (MD.)), and November 17 (vs SOUTHEAST MISSOURI STATE 82, ST. FRANCIS (PA.) 50). Each game entry includes the date, time, location, opponent, and final score.

OVERALL	PCT	CONF	PCT	STREAK	HOME	AWAY	NEUTRAL
24-10	.706	11-5	.688	L1	17-2	6-7	1-1

Game of the Week 2017-2018 Season

2017-18 Women's Basketball Schedule

Print | Subscribe with... | All Games | 2017-18

DATE	TIME	LOCATION	OPPONENT	SCORE
NOV 10 (FRI)	7 P.M.	ANN ARBOR, MI	AT NO. 24/23 UNIVERSITY OF MICHIGAN (FIRST ROUND)	L, 61-75
NOV 14 (TUE)	7 P.M.	BALTIMORE, MD	AT LOYOLA UNIVERSITY (MD.)	W, 80-72
NOV 17 (FRI)	12 P.M.	FAIRFAX, VA. / EAGLEBANK ARENA	VS SOUTHEAST MISSOURI STATE 82, ST. FRANCIS (PA.) 50	W, 80-72
NOV 17 (FRI)	2:30 P.M.	FAIRFAX, VA. / EAGLEBANK ARENA	VS SOUTHEAST MISSOURI STATE 82, ST. FRANCIS (PA.) 50	W, 80-72

The following questions will guide you through the process of scraping this data. You are encouraged to review the examples provided in the [Web scraping activity](#), the [Class 16 slides](#), and the [Class 19 slides](#) while completing this part of the homework assignment.

Men's basketball schedule and scores

3. To start, you need to load the men's basketball schedule and scores page into R. Do this using one line of code, and assign the accessed page data to a variable called `mens_bb`.
4. Mason's opponent for each game is listed on the left side of each row, just after a small box that says *VS* or *AT*. Use the SelectorGadget tool to determine the CSS selector needed to scrape this information, and then write the code that scrapes this information. Assign the scraped data to a variable called `mens_opponents`. If done right, `mens_opponents` should be a character vector containing 33 teams.
5. The location for each game is listed to the right of the opponent's name. For games played in the United States it lists the city and state, for example "Fairfax, VA" is the location for home games. Use the SelectorGadget tool to determine the CSS selector needed to scrape this information, and then write the code that scrapes this information. Assign the scraped data to a variable called `mens_locations`.
6. The date for each game is listed above the opponent's name, and has the format *Month Day (Day of the Week)*. For example, the first listed game has the date *Nov 10 (FRI)*. Use the SelectorGadget tool to determine the CSS selector needed to scrape this information, and then write the code that scrapes this information. Assign the scraped data to a variable called `mens_dates`.
7. The time for each game is listed to the right of the game date. For example, the first listed game has the time *7:00 P.M.* Use the SelectorGadget tool to determine the CSS selector needed to scrape this information, and then write the code that scrapes this information. Assign the scraped data to a variable called `mens_times`.
8. The score for each game is listed on the right side of each row in the format *Mason's score-Opponent's score*. For example, the first listed game has the score *67-65*. Use the SelectorGadget tool to determine the CSS selector needed to scrape this information, and then write the code that scrapes this information. Assign the scraped data to a variable called `mens_scores`. Your `mens_scores` vector should only have 33 pieces of data in it, if it has more or less then you need to try another CSS selector.
9. The **W** or **L** to the left of each game score indicates whether Mason won (**W**) or lost (**L**) the game. Use the SelectorGadget tool to determine the CSS selector needed to scrape this information, and then write the code that scrapes this information. You'll note that for each game you'll actually get `W, or L, .` Pipe (`%>%`) your scraped data into the `str_remove()` function and tell it to get rid of the comma. Assign the scraped data to a variable called `mens_win_loss`.
10. Use the `data_frame()` function to create a tibble containing your scraped data. The columns should have the following names and be in this order:
 - `date`
 - `time`
 - `opponent`
 - `location`
 - `score`
 - `win_loss`

Assign the tibble to a variable called `mens_df`. This table should have 33 rows.

Women's basketball schedule and scores

11. The code you created for scraping the men's basketball team schedule and score should also work on the page for the women's team with minimal changes. Copy the code you wrote in the blocks for the men's page and paste it here. **Change the prefix of the variable names you assign each output into from `mens_` to**

womens_, and the code so that it loads the women's schedule and scores page. The final result should be a tibble assigned to a variable called `womens_df`, which has the following columns in this order:

- `date`
- `time`
- `opponent`
- `location`
- `score`
- `win_loss`

This table should have 34 rows.

Quick data exploration

Collecting data doesn't serve much of a purpose if we don't explore or analyze it. Create the summary reports and visualizations requested below to help you better understand the data you just collected.

12. What was the average score for the men's team (Mason only) when they won a game and when they lost a game? What was the average score for the women's team (Mason only) when they won a game and when they lost a game?

Hint: To answer this, you will need to use the `separate()` function.

13. Plot the men's histogram of scores and the women's histogram of scores (just for the Mason teams, not the opponents), and then compare the two histograms. Which histogram is centered at a higher score? Which histogram has the larger spread? Are there any other notable differences?

How to submit

When you are ready to submit, be sure to save, commit, and push your final result so that everything is synchronized to Github. Then, navigate to **your copy** of the [Github repository](#) you used for this assignment. You should see your repository, along with the updated files that you just synchronized to Github. Confirm that your files are up-to-date, and then do the following steps:

1. Click the *Pull Requests* tab near the top of the page.
2. Click the green button that says "New pull request".
3. Click the dropdown menu button labeled "base:", and select the option `starting`.
4. Confirm that the dropdown menu button labeled "compare:" is set to `master`.
5. Click the green button that says "Create pull request".
6. Give the *pull request* the following title: `Submission: Homework 3, FirstName LastName`, replacing `FirstName` and `LastName` with your actual first and last name.
7. In the messagebox, write: `My homework submission is ready for grading @shuaibm @jkglass-brenner`.
8. Click "Create pull request" to lock in your submission.

Cheatsheets

You are encouraged to review and keep the following cheatsheets handy while working on this assignment:

- [RStudio cheatsheet](#)
- [RMarkdown cheatsheet](#)
- [RMarkdown reference](#)
- [ggplot2 cheatsheet](#)
- [Data transformation cheatsheet](#)
- [Data import cheatsheet](#)