

# Scraping HTML

Carolina A. de Lima Salge  
Assistant Professor  
Terry College of Business  
University of Georgia

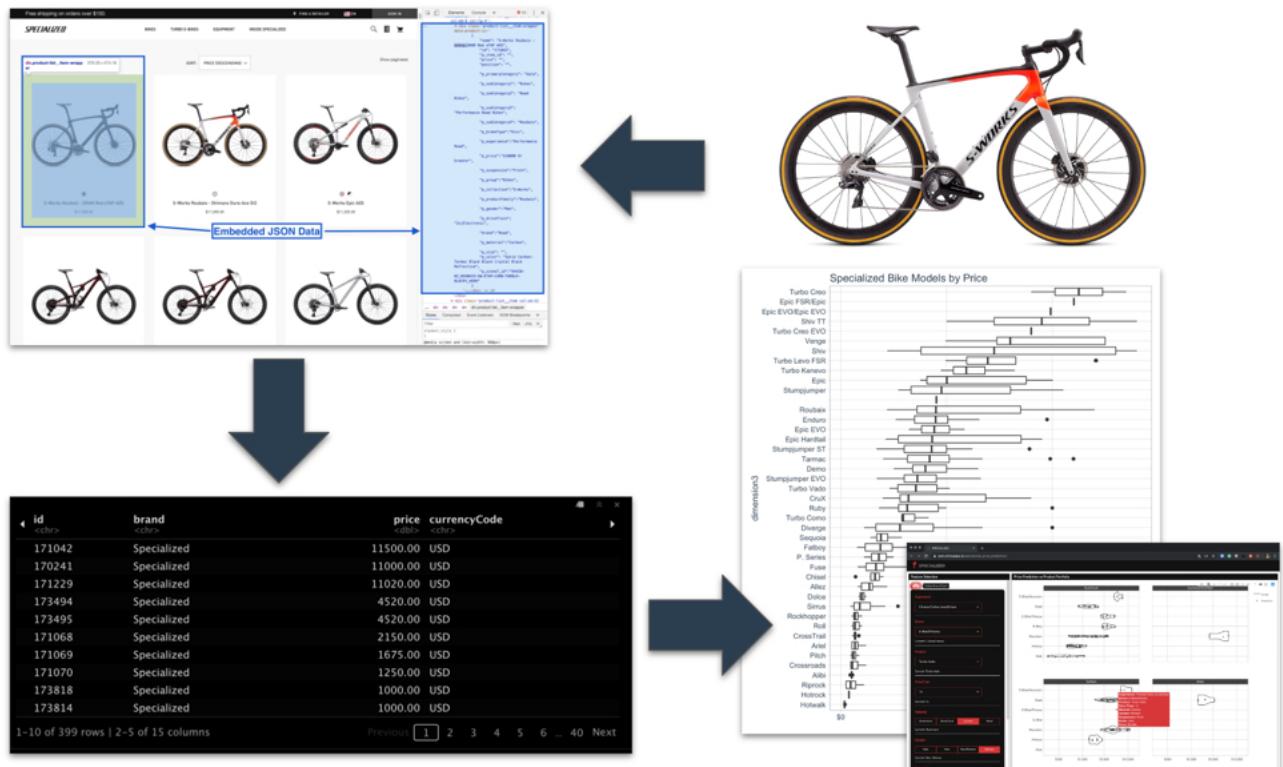
*Business Intelligence*  
Spring 2021



Terry College of Business  
UNIVERSITY OF GEORGIA



A screenshot of a web browser displaying the rvest documentation page on tidyverse.org. The page shows several examples of R code for extracting data from Star Wars articles. The code uses the rvest package to parse HTML and extract specific text blocks, demonstrating how to handle nested structures and extract content like titles and subtitles.



# Web Scraping

Let's say, **data** is vital for your e-commerce company – you can see the data that you want but the question is: *How will you download it in a usable format?*

- Most would copy and paste manually
- Not feasible for websites with lots of pages
- Solution? 

Web scraping is a process of automating the extraction of data in an efficient and fast way. With the help of web scraping, you can extract data from any website, no matter how large is the data, on your computer.



# Web Scraping

You copied and pasted some data, but how to convert or save it in a format of your choice?

- Ways to save as CSV, Excel, etc
- Able to import, analyze, and use the data

Web scraping simplifies the process of extracting data, speeds it up by automating it and creates easy access through familiar formats



# Web Scraping

Three common ways to scrape data

- Basic (HTML)
- APIs (Twitter, Yelp)
- Advanced (Javascript)
  - R selenium: <https://rpubs.com/johndharrison/RSelenium-Basics>



# Web Scraping

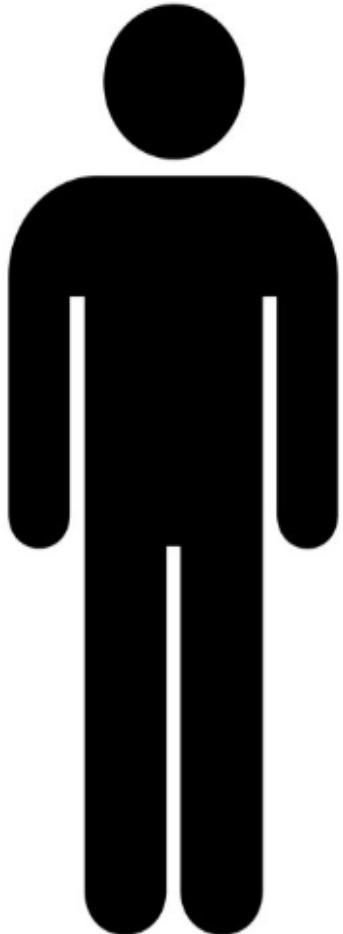
Three common ways to scrape data

- Basic (HTML)
- APIs (Twitter, Yelp)
- Advanced (Javascript)
  - R selenium: <https://rpubs.com/johndharrison/RSelenium-Basics>



# Basic HTML

<head>



</head>

<body>

</body>

Get smart about your upgrades | See all →



See who's at the door



Light up your home



Streaming made simple



Helpful speakers



Internet-savvy TVs



Control your comfort



Around-the-clock security

Today's Deals - All With Free Shipping | See all →

The screenshot shows the developer tools of a web browser with the URL [www.ebay.com](http://www.ebay.com). The 'Elements' tab is selected, displaying the HTML structure of a product page. The 'Sources' tab shows the raw JavaScript and CSS code. The 'Breakpoints' sidebar on the left lists various breakpoints defined in the code. The main pane displays the HTML code, which includes a conditional comment for Internet Explorer 9, a script block for font customization, and a link element for a preload image.

```
<!--[if IE 9]> <html class="ie9"> <![endif]-->
<!--[if (gt IE 9) || !(IE)]><!-->
<!--[endif]-->
<head>
<!--
    font-marketsans body {
        font-family: "Market Sans", Arial, sans-serif;
    }
-->
</style>
<script>
(function() {
    var useCustomFont = ('fontDisplay' in document.documentElement.style) ||
        (localStorage && localStorage.getItem('ebay-font'));
    if (useCustomFont) {
        document.documentElement.classList.add('font-marketsans');
    }
})()
</script>
<link rel=preload as=image href=https://ir.ebaystatic.com/pictures/av/pics/s_1x2.gif>
```

All HTML documents must start with a document type declaration: `<!DOCTYPE html>`  
The HTML document itself begins with `<html>` and ends with `</html>`  
The visible part of the HTML document is between `<body>` and `</body>`



# HTML for Web Scraping

```
<!DOCTYPE html>  
<html>  
  <body>
```

stuff we don't care about

```
    <h1>My First Heading</h1>
```

```
    <p>My first paragraph.</p>
```

stuff we want

```
  </body>  
</html>
```

tags hold stuff  
we want



# Finding Elements Inside HTML Code

```
<!DOCTYPE html>
```

```
<html>
```

```
  <body>
```

p .alpha

```
    <h1>My First Heading</h1>
    <p class="alpha">My first paragraph.</p>
    <p class="beta">My second paragraph.</p>
```

```
  </body>
```

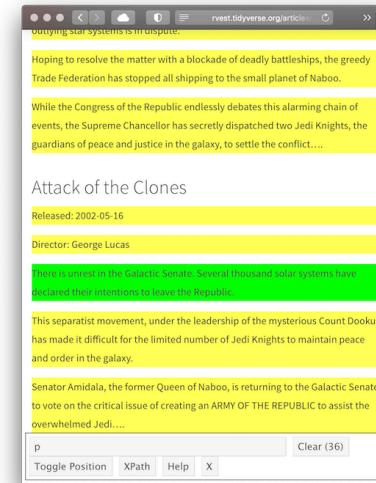
```
</html>
```

Use *selectors* to find elements based primarily on either tag type (e.g. p, h1) or attributes (e.g. class)



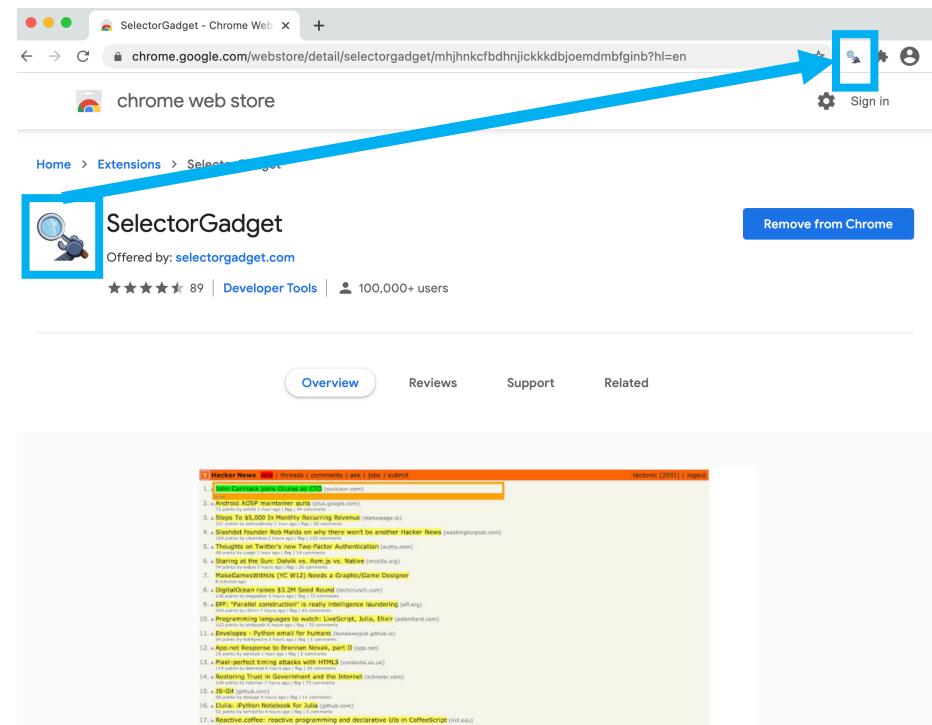
# Web Scraping HTML in R

- Get HTML document
- Find and extract text from HTML document
- Create table based on vectors



# Quick Exercise

Open the Google Chrome web browser and install **SelectorGadget**



<https://chrome.google.com/webstore/detail/selectorgadget/mhjhjnkcfdhnjickkkdbjoemdmbfginb?hl=en>



# SelectorGadget

A tool that makes it easy to find elements inside HTML code

```
<!DOCTYPE html>
<html>
  <body>
```

p .alpha → <h1>My First Heading</h1>  
 <p class="alpha">My first paragraph.</p>
 <p class="beta">My second paragraph.</p>

```
        </body>
      </html>
```



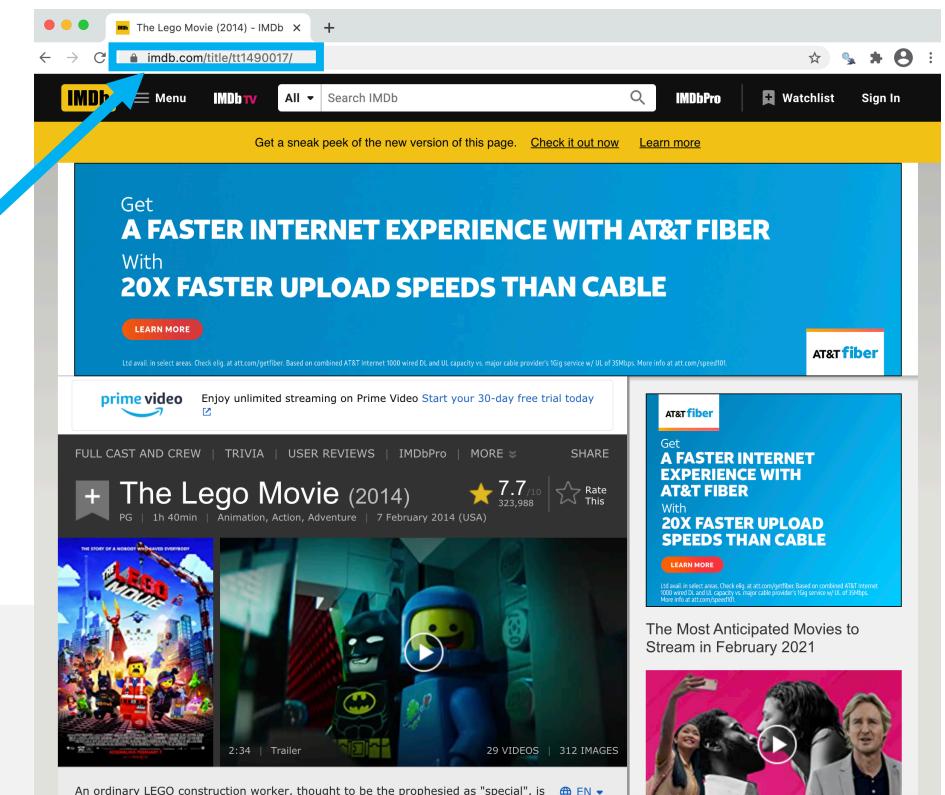
Use *selectors* to find elements based primarily on either tag type (e.g. p, h1) or attributes (e.g. class)

# Scraping Movie Data (IMDb)

Website we want to  
scrape data from

```
library(rvest)  
library(tidyverse)
```

```
lego_movie <- read_html("http://www.imdb.com/title/tt1490017/")
```



# Scraping Movie Data (IMDB)

rvest & tidyverse  
code

Base R code

```
rating <- lego_movie %>%  
  html_nodes("strong span") %>%  
  html_text() %>%  
  as.numeric()  
  rating  
  
cast <- lego_movie %>%  
  html_nodes(".primary_photo+ td a") %>%  
  html_text() %>%  
  trimws()  
  cast
```

Tags holding stuff we want

Pipe function



# Scraping Movie Data (IMDB)

```
poster <- lego_movie %>%  
  html_nodes(".poster img") %>%  
  html_attr("src")  
poster
```

```
lego <- tibble(rating = rating,  
               cast = cast,  
               poster = poster)  
lego
```

```
write_csv(lego, "lego.csv")
```



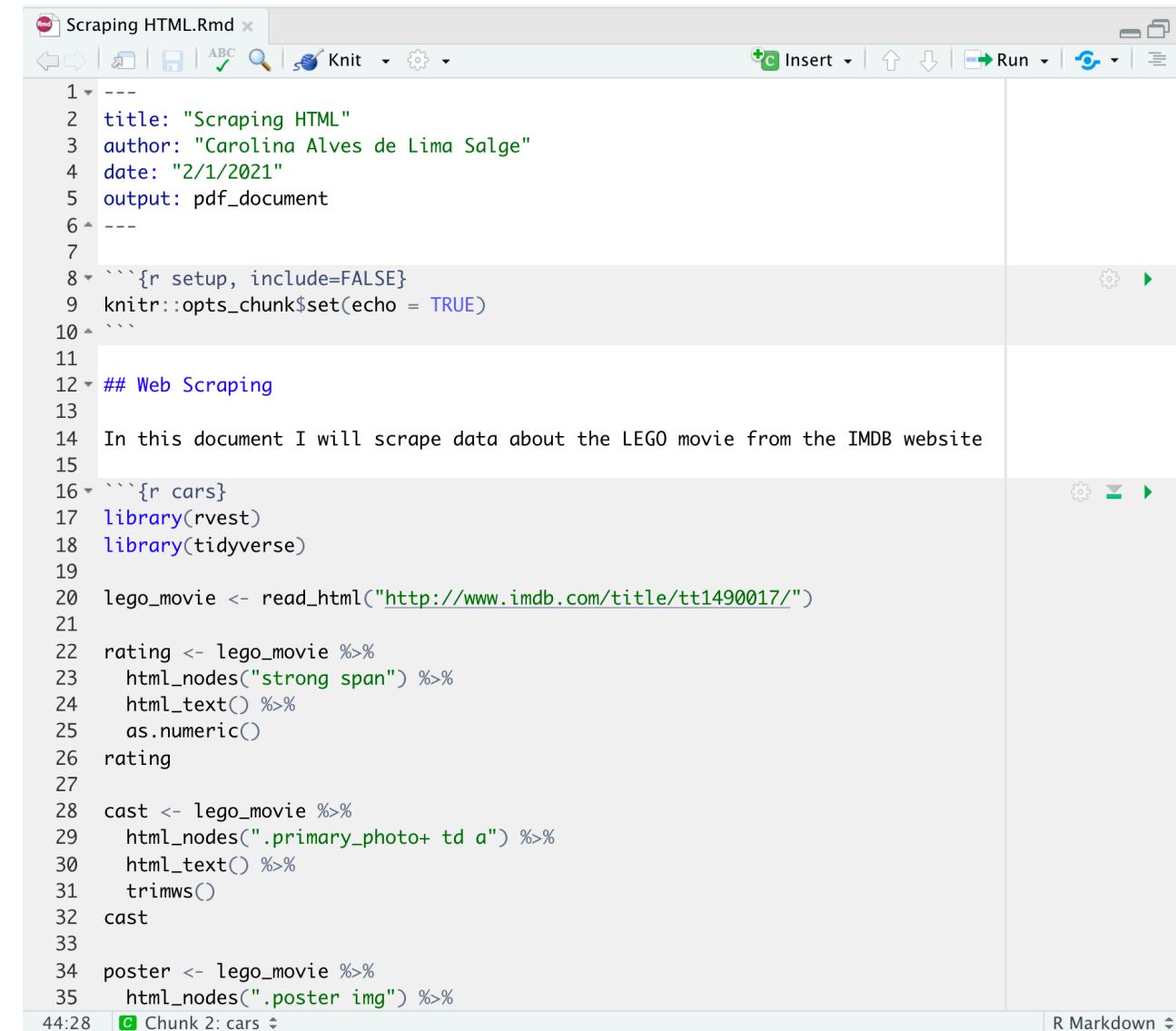


# **Ready to scrape some data?**



# In-Class Exercise

- Open a Rmd file
- Copy and paste code from slides 13 to 15
  - Change the code to add the path to the folder you want to save the data scraped  
[`write_csv(lego, "lego.csv")`]
- Knit to Word/PDF



The screenshot shows the RStudio interface with an R Markdown file open. The title bar says "Scraping HTML.Rmd". The code editor contains the following R code:

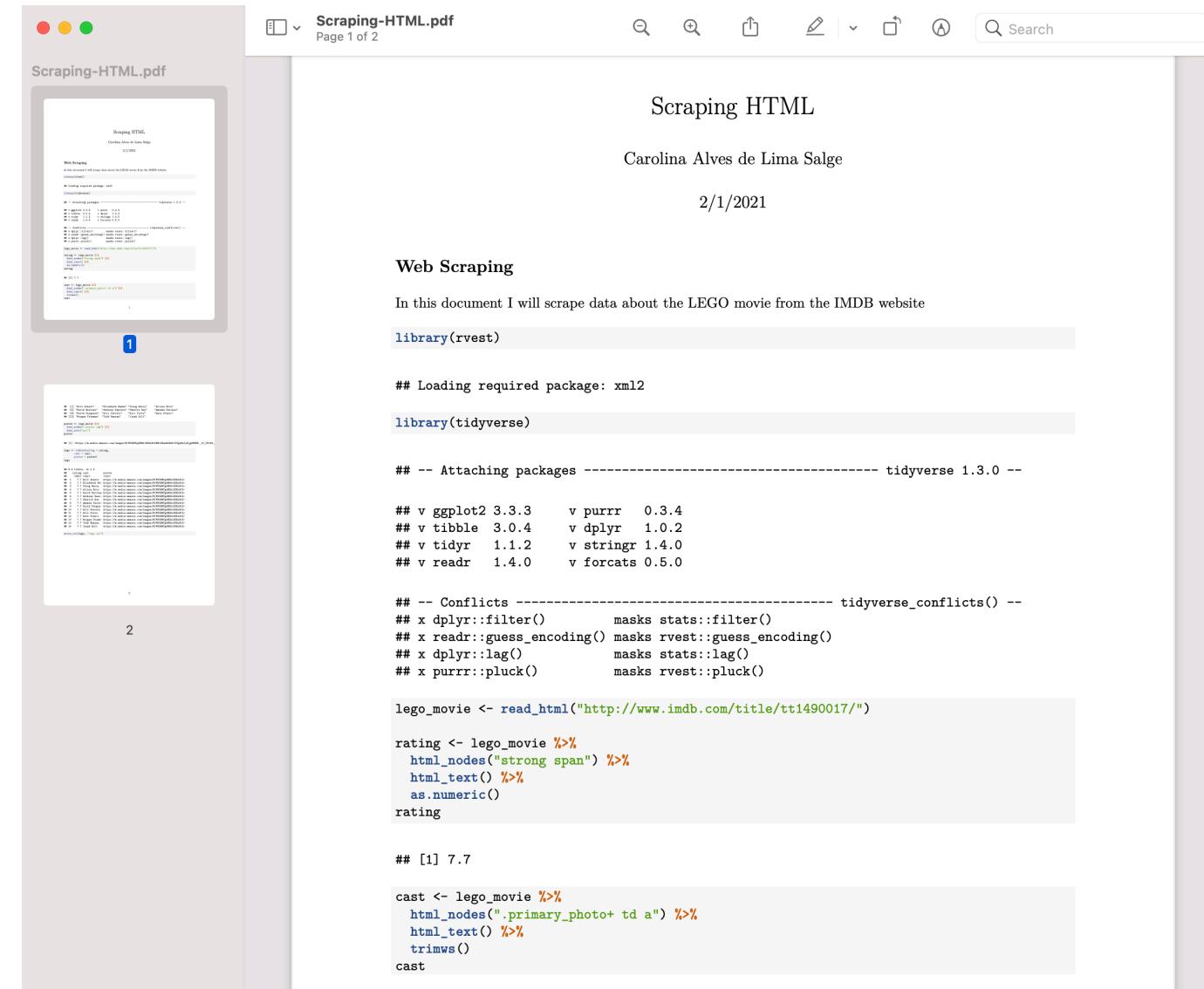
```
1 ---  
2 title: "Scraping HTML"  
3 author: "Carolina Alves de Lima Salge"  
4 date: "2/1/2021"  
5 output: pdf_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ```  
11  
12 ## Web Scraping  
13  
14 In this document I will scrape data about the LEGO movie from the IMDB website  
15  
16 ```{r cars}  
17 library(rvest)  
18 library(tidyverse)  
19  
20 lego_movie <- read_html("http://www.imdb.com/title/tt1490017/")  
21  
22 rating <- lego_movie %>%  
23   html_nodes("strong span") %>%  
24   html_text() %>%  
25   as.numeric()  
26 rating  
27  
28 cast <- lego_movie %>%  
29   html_nodes(".primary_photo+ td a") %>%  
30   html_text() %>%  
31   trimws()  
32 cast  
33  
34 poster <- lego_movie %>%  
35   html_nodes(".poster img") %>%
```

The status bar at the bottom left shows "44:28" and "Chunk 2: cars". The status bar at the bottom right shows "R Markdown".



# In-Class Exercise

- Open a Rmd file
- Copy and paste code from slides 13 to 15
  - Change the code to add the path to the folder you want to save the data scraped  
[`write_csv(lego, "lego.csv")`]
- Knit to Word/PDF



Scraping HTML

Carolina Alves de Lima Salge

2/1/2021

### Web Scraping

In this document I will scrape data about the LEGO movie from the IMDB website

```
library(rvest)

## Loading required package: xml2

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter()    masks stats::filter()
## x readr::guess_encoding() masks rvest::guess_encoding()
## x dplyr::lag()       masks stats::lag()
## x purrr::pluck()     masks rvest::pluck()

lego_movie <- read_html("http://www.imdb.com/title/tt1490017/")

rating <- lego_movie %>%
  html_nodes("strong span") %>%
  html_text() %>%
  as.numeric()
rating

## [1] 7.7

cast <- lego_movie %>%
  html_nodes(".primary_photo+ td a") %>%
  html_text() %>%
  trimws()
cast
```



# At-Home Exercise 1

Scrape the title, price, condition, and photo link for all items in page 1, put the data together in a tibble, and save the tibble as CSV

[https://www.ebay.com/sch/i.html?\\_from=R40&\\_nkw=mac+laptop&\\_sacat=0&rt=nc&\\_pgn=1](https://www.ebay.com/sch/i.html?_from=R40&_nkw=mac+laptop&_sacat=0&rt=nc&_pgn=1)

The screenshot shows the eBay search results for 'mac laptop'. The search bar at the top contains 'mac laptop'. Below the search bar, there are filters for 'Category' (All, Computers/Tablets & Networking, Laptops & Netbooks, Apple Laptops, PC Laptops & Netbooks, Home & Garden, eBay Motors, Business & Industrial, Crafts, Cell Phones & Accessories, Clothing, Shoes & Accessories, Show More), 'Price' (Under \$200.00, \$200.00 to \$300.00, Over \$300.00), and 'Release Year' (2019, 2018, 2017, 2015, 2012, 2011, 2010, 2015, 2017, 2018). The main results section displays two sponsored listings:

- MACBOOK PRO 15 LAPTOP | INTEL CORE 2.5GHZ | 500GB | WARRANTY MAC SUPPORT - CHARGER/CASE - FREE SHIPPING - 3 YR WARRANTY**  
Pre-Owned · 15.4 in  
**\$458.50**  
Was: \$917.00 50% off  
Buy It Now  
Free shipping  
Free returns  
**Almost gone**  
**14 sold**
- Apple MacBook 13" Laptop UPGRADED 8GB+1TB HD \*\* MAC OS X High Sierra \*\***  
15 colors 3D Carbon Fiber to choose\* Warranty\* Grade C  
Refurbished  
**\$194.95 to \$294.95**  
Buy It Now  
Free shipping  
Free returns  
**69+ sold**



# At-Home Exercise 2 (More Challenging)

Scrape the title, price, condition, and photo link for all items in **ALL pages** (1 through 7), put the data together in a tibble, and save the tibble as CSV

[https://www.ebay.com/sch/i.html?\\_from=R40&\\_nkw=mac+laptop&\\_sacat=0&rt=nc&\\_pgn=1](https://www.ebay.com/sch/i.html?_from=R40&_nkw=mac+laptop&_sacat=0&rt=nc&_pgn=1)

The screenshot shows an eBay search results page for "mac laptop". The search bar at the top contains "mac laptop". Below the search bar, there are filters for "All Listings", "Accepts Offers", "Auction", "Buy It Now", "Condition", "Shipping", and "Best Match". The results section displays 339 results for "mac laptop". A sponsored listing for a "MACBOOK PRO 15 LAPTOP" is shown with a price of \$458.50 (Was: \$917.00). Another sponsored listing for an "Apple MacBook 13" Laptop is shown with a price range from \$194.95 to \$294.95. The left sidebar includes categories like All, Computers/Tablets & Networking, Laptops & Netbooks, Apple Laptops, PC Laptops & Netbooks, Home & Garden, eBay Motors, Business & Industrial, Crafts, Cell Phones & Accessories, Clothing, Shoes & Accessories, and a "Show More" link. The right sidebar includes filters for "Release Year" (2009-2018) and "Screen Size" (12-12.9 in).



# Resources for Exercise 2

<https://rpubs.com/DrRobbieBaldock/690979>

<https://www.r-bloggers.com/2020/07/tutorial-web-scraping-of-multiple-pages-using-r/>

<https://towardsdatascience.com/coupling-web-scraping-with-functional-programming-in-r-for-scale-1bc4509eef29>

<https://www.business-science.io/code-tools/2019/10/07/rvest-web-scraping.html>



# *Thank You!*



Terry College of Business  
UNIVERSITY OF GEORGIA