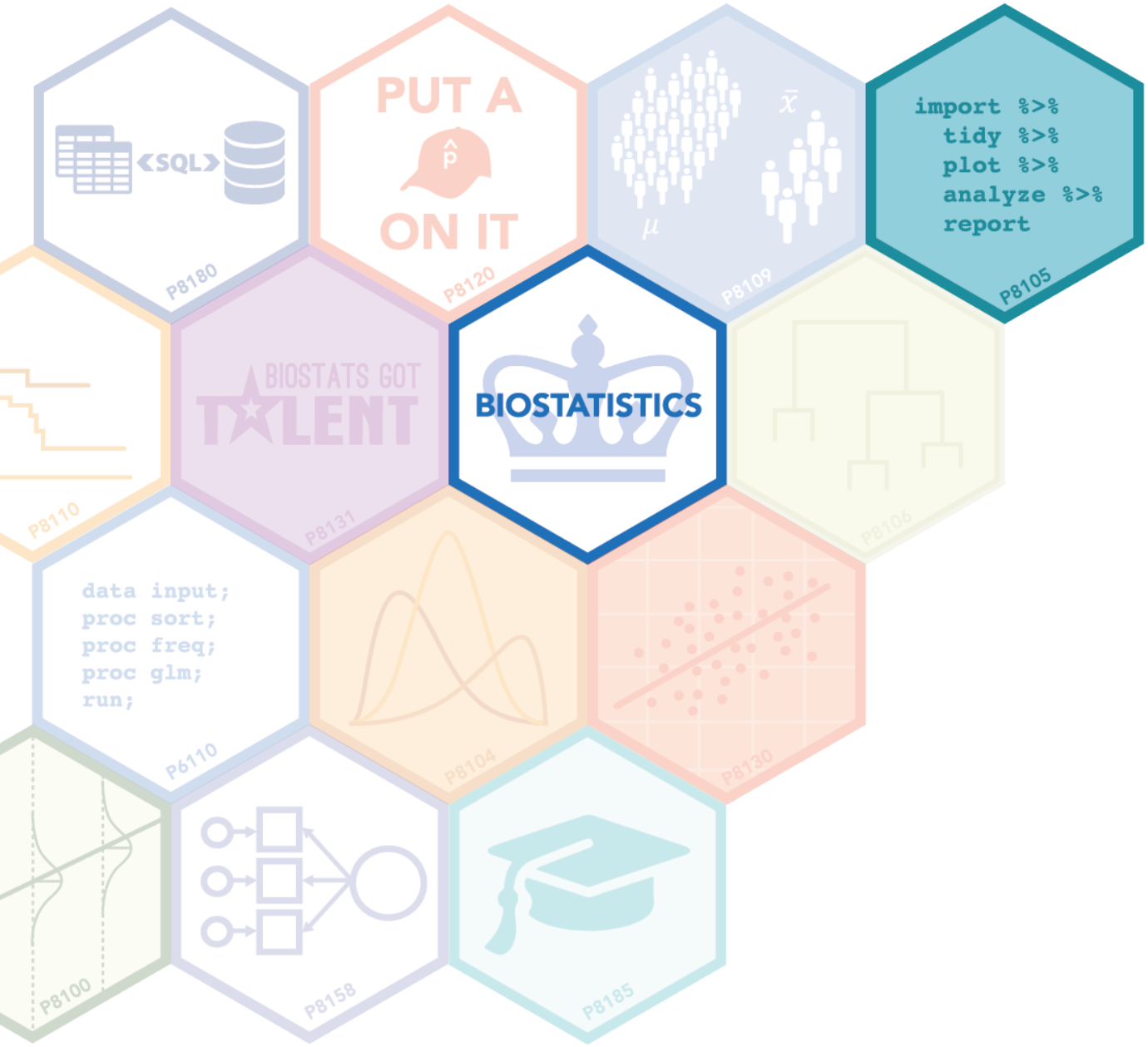



# READING DATA FROM THE WEB

Jeff Goldsmith, PhD
Department of Biostatistics

# Two major paths

- There's data included as content on a webpage, and you want to "scrape" those data
  - Table from Wikipedia
  - Reviews from Amazon
  - Cast and characters on IMBD

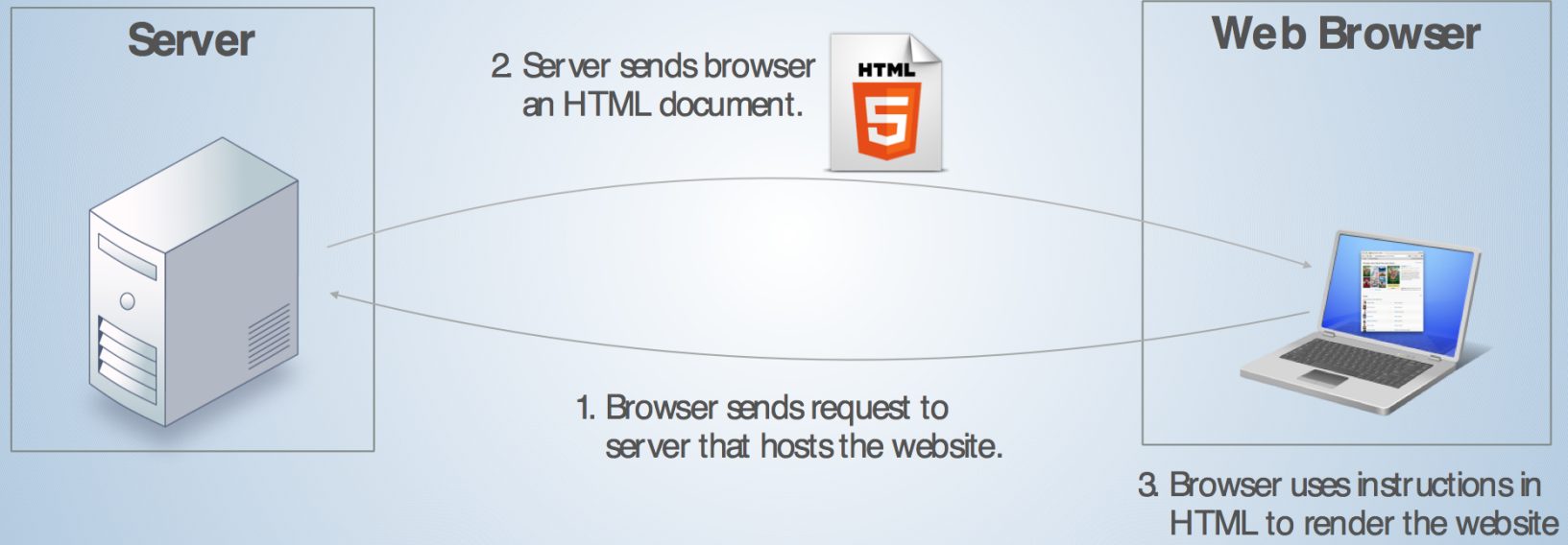
- There's a dedicated server holding data in a relatively usable form, and you want to ask for those data
  - Open NYC data
  - Data.gov
  - Star Wars API

# Scraping web content

- Webpages combine HTML (content) and CSS (styling) to produce what you see
- When you retrieve the HTML for a page with data you want, you've retrieved the data

- Also you have a lot of other stuff
- Challenge is extracting what you want from the HTML

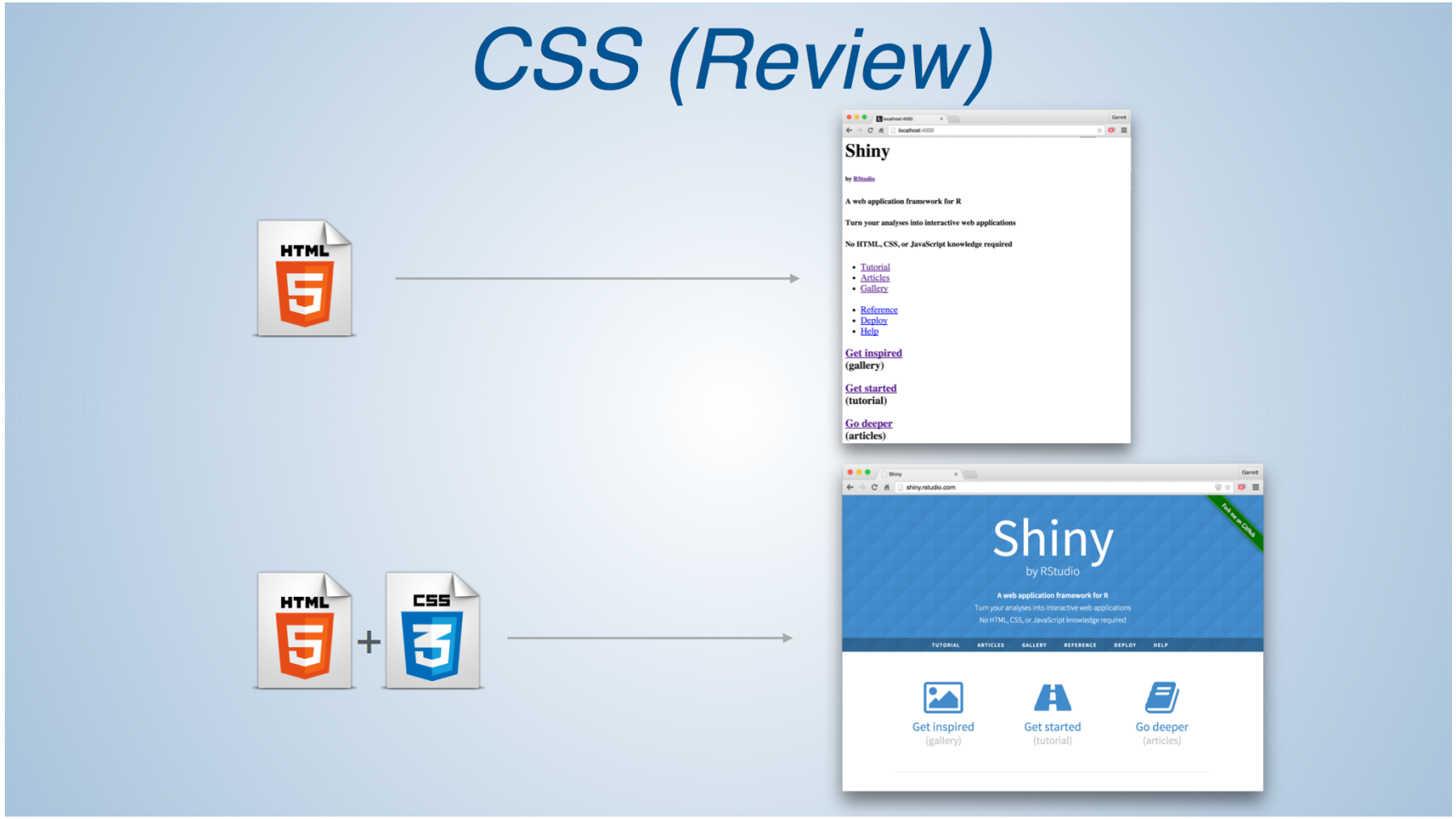
# HTML (Review)




https://github.com/ropensci/user2016-tutorial

Garrett Grolemund, “Extracting data from the web”

# CSS (Review)




https://github.com/ropensci/user2016-tutorial

Garrett Grolemund, “Extracting data from the web”

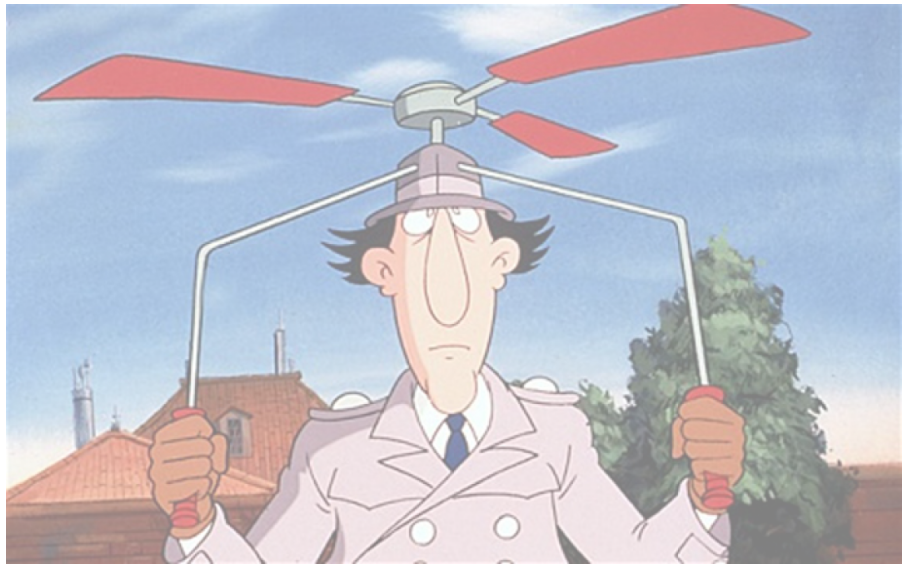
# CSS Selectors

- Because CSS controls appearance, CSS identifiers appear throughout HTML code
- HTML elements you care about frequently have unique identifiers

- Extracting what you want from HTML is often a question of specifying an appropriate CSS Selector

# Find the CSS Selector

- Selector Gadget is the most common tool for finding the right CSS selector on a page
  - In a browser, go to the page you care about
  - Launch the Selector Gadget
  - Click on things you want
  - Unclick things you don't
  - Iterate until only what you want is highlighted
  - Copy the CSS Selector



Inspector Gadget

# Scraping data into R


- `rvest` facilitates web scraping
- Workflow is:
  - Download HTML using `read_html()`
  - Extract nodes using `html_nodes()` and your CSS Selector
  - Extract content from nodes using `html_text()`, `html_table()`, etc

# APIs

- In contrast to scraping, **A**pplication **P**rogramming **I**nterfaces provide a way to communicate with software
- Web APIs may give you a way to request specific data from a server

- Web APIs aren't uniform
  - The Star Wars API is different from the NYC Open Data API
- This means that what is returned by one API will differ from what is returned by another API

# Getting data into R

- Web APIs are mostly accessible using HTTP (the same protocol that's used to serve up web pages)
- `httr` contains a collection of tools for constructing HTTP requests
- We'll focus on GET, which retrieves information from a specified URL
  - You can refine your HTTP request with query parameters if the API makes them available

# API data formats

- In “lucky” cases, you can request a CSV from an API
  - Sometimes you could download this by clicking a link on a webpage, but

    ```
    ### I went to <website> and clicked “download”
    ```

    isn’t reproducible

- In more general cases, you’ll get **J**ava**S**cript **O**bject **N**otation (JSON)
  - JSON files can be parsed in R using jsonlite

# Real talk about web data

- Data from the web is messy
- It will frequently take a lot of work to figure out
  - How to get what you want
  - How to tidy it once you have it