Stat 240 Week 11
Even Shinier

& More data from the web: json

Week 11

Dr. Dave Campbell

What is the lifetime of a university textbook?

How do we answer that question?

What kind of information do we need?

Webscraping

Best: find data already nicely formatted

Next Best: use API

Next Next Best: web scraping

Week 9: get a lot of course info
run a for loop over course numbers and sections
extract html

```
baseurl = "https://www.sfu.ca/outlines.html?"

year = "2017"

term = "spring"

dept="stat"

courseNo = "240"

section = "e100"

course_url1 = paste(baseurl,year,sep="")

(course_url = paste(course_url1,term,dept, courseNo,
section,sep="/"))
```

Week 9: get a lot of course info
run a for loop over course numbers and sections
extract html

```
<div id="course-outline-details">

  <div class="custom-header">

    <h1 id="name">Spring 2017 - STAT 240
<span>E100</span></h1>

          <h2 id="title">

        Introduction to Data Science

                (3)

              </h2>

          <h3 id="class-number">Class
Number: 6282</h3>

          <h3 id="delivery-
method">Delivery Method: In Person</h3>

        </div>

    <h2 class="overview"><a
name="overview">Overview</a></h2>

      <div class="ruled ruledMargin"></div>

      <div class="parsys_column">

        <div class="parsys_column -c0">

          <div class="text parbase section">

            <div class="overview-list">

              <ul class="course-times-line">

                  <li class="course-times">

    <h4>Course Times + Location:</h4>

    <p>Mo 4:30 PM &amp;ndash; 6:20 PM<br>AQ
5018, Burnaby</p>

                  </li>
```

html vs other

html: stable, designed for high volume, messy: has formatting and data

JSON: comes from an API, might be fragile, no formatting: just the data.

JavaScript Object Notation (JSON) formats

JSON data comes in key/value pairs

Think of it as named data frame columns or named vectors.

Our course schedule might be have several key/value pairs:

"startTime":"12:30", “days”:"Mo",

“endTime":"14:20",

“roomNumber":"7618","buildingCode":"EDB","campus":"Burnaby"

JavaScript Object Notation (JSON) formats

Data can be

Numbers: 1, 2, 3.1415

Strings: “Text in double quotes”

Boolean: TRUE

Array [“ordered”, “comma separated”, “enclosed in square brackets”,”any data type inside”]

Object {unordered, comma separated, collection of key: value pairs in curly brackets, any data types}

JavaScript Object Notation (JSON) formats

Our course schedule might be have several “key”:value pairs:

"startTime":"12:30", “days”:"Mo",

"endTime":"2:20",

“roomNumber":"7618","buildingCode":"EDB","campus":"Burnaby"

JavaScript Object Notation (JSON) formats

Our course info could be split into a hierarchical data structure with the top levels:

CourseSchedule

Course Instructor

Grading

Course Info

Text

JavaScript Object Notation (JSON) formats

Our course info could be split into a hierarchical data structure with the top levels:

CourseSchedule [start time, end time, day, room, …]

Course Instructor [name, office, email, phone,…]

Grading [assignments, midterms, final,…]

Course Info [pre-req, delivery method, title,description,…

Text [required, recommended,…]

Some course info in JSON:

“courseSchedule”:[{"startTime":"12:30", "startDate":"Thu Jan 03 00:00:00 PST 2019", "roomNumber":"7618", "days":"Mo","endDate":"Mon Apr 08 00:00:00 PDT 2019”,”sectionCode":"LEC","endTime":"14:20", **"isExam":false**,"buildingCode":"EDB","campus":"Burnaby"}],

“examSchedule":[{"startTime":"12:00","startDate":"Mon Apr 15 00:00:00 PDT 2019”,”roomNumber":"2502","days":"Mo","endDate":"Mon Apr 15 00:00:00 PDT 2019”, “endTime":"15:00", **“isExam":true**,"buildingCode":"WMC", "campus":"Burnaby"}, {"startTime":"12:00","startDate":"Mon Apr 15 00:00:00 PDT 2019”,”roomNumber":"3144","days":"Mo","endDate":"Mon Apr 15 00:00:00 PDT 2019”,”endTime":"15:00", **"isExam":true**,"buildingCode":"AQ","campus":"Burnaby"}]

The workflow:

```
library(jsonlite)

courseURL = "http://www.sfu.ca/bin/wcm/course-outlines?2019/spring/stat/240/d100"

course_info = fromJSON(courseURL)

class(course_info)  # find the data format & check if the API broke
```

```
attributes(course_info)  # Use this to find out
what the main pieces of the JSON contain

inst = as.data.frame(course_info$instructor)#
make a data frame
```

Some links:

http://www.sfu.ca/bin/wcm/course-outlines?

http://www.sfu.ca/bin/wcm/course-outlines?2014

http://www.sfu.ca/bin/wcm/course-outlines?2014/fall/

http://www.sfu.ca/bin/wcm/course-outlines?2014/fall/stat

http://www.sfu.ca/bin/wcm/course-outlines?2014/fall/stat/285

http://www.sfu.ca/bin/wcm/course-outlines?2014/fall/stat/285/d900

```
attributes(course_info)  # Use this to find out what the main pieces of the JSON contain

inst = as.data.frame(course_info$instructor)# make a data frame



courseURL2016 = "http://www.sfu.ca/bin/wcm/course-outlines?2016/fall/stat/285/d900"

course_info2016 = fromJSON(courseURL2016)

newinst = rbind(inst, course_info2016 $instructor)
```

Data Cleaning

Note that every time I modify the data I change the variable name. That lets me revert changes and diagnose problems.

Especially important for dealing with big data sets.