ACQUIRING EXTERNAL DATA WITH R

Mark Chisholm

11 July 2016

Agenda

- The data landscape
- Using R to access external data
- Illustrative example
- Conclusion

The Data Landscape

The Data Landscape

- Underwriters want information that will help them answer questions such as:
    - How can we offer relevant insurance products for our clients?
    - How much in premium would we need to charge in order to break even in the long run?
    - How many years will it take for all claims arising from a group of policies to settle?
- Actuaries and data scientists that work closely with underwriters can help answer these questions

The Data Landscape

- Data can be limited in specialty insurance
  - Policies might not be homogenous
  - Claims can arise from unforeseen risks
  - May not capture enough attributes about the contract, insured, claim, etc.

- Lack of data: “not enough rows” (small sample size)
  “not enough columns” (not enough variables)
  “not enough granularity” (not enough levels)

The Data Landscape

- Some possible solutions to the issue of lack of data
  - Capture selected information going forward in corporate systems
  - Assemble the information from other internal sources
  - Purchase data from vendors
  - Acquire the data from sources on the internet

The Data Landscape

- Acquire the data from sources on the internet
  - Some web sites that are relevant to insurance will provide data under their terms & conditions
  - In some cases it could be relatively easy to download (for example, an FTP site)
  - Others may provide data, but will require navigating a web interface
    - Some web sites are easier to grab information from than others

The Data Landscape

- Acquire the data from sources on the internet
    - Numerous tools available in many languages which can extract external data
    - Packages to accomplish these tasks are available in R

Using R to Access External Data

Using R to Access External Data

XL CATLIN

- What makes up a web site?

```
343                             <li><a href="/about-xl/what-we-believe/our-culture">
344                                 <div>Culture</div>
345                             </a></li>
346                             <li><a href="/about-xl/what-we-believe/corporate-social-responsibility">
347                                 <div>Responsibility</div>
348                             </a></li>
349                             <li><a href="http://xlgroup.com/about-xl/what-we-believe/sponsorship">
350                                 <div>Sponsorship</div>
351                             </a></li>
352                     </ul>
353             </div>
354             <div class="Our-Strengths submenu">
355                     <ul>
356                             <li><a href="/about-xl/our-strengths/global-capabilities">
357                                 <div>Global Capabilities</div>
358                             </a></li>
359                             <li><a href="/about-xl/ratings">
360                                 <div>Ratings</div>
361                             </a></li>
362                     </ul>
363             </div>
364             <div class="Media submenu">
365                     <ul>
366                             <li><a href="/press">
367                                 <div>Press Releases</div>
368                             </a></li>
369                             <li><a href="/press">
370                                 <div>Fact Sheets</div>
```

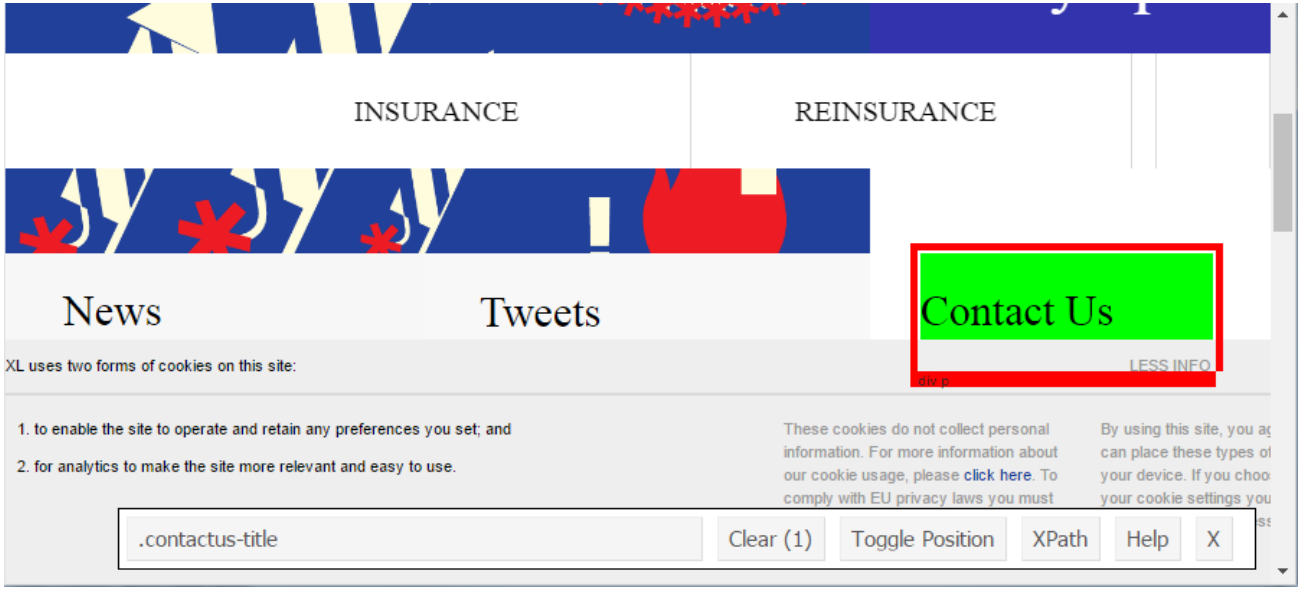
Using R to Access External Data

- Primarily make use of these tools:
  - Selector Gadget helps identify elements to extract in a web page
  - rvest then extracts the elements, and R can then be used to further process the data or make it available in shiny
  - RSelenium is a useful tool for manipulating web pages with more complex layouts

Using R to Access External Data

- How can you identify elements in a web page?
  - Selector Gadget

https://cran.r-project.org/web/packages/rvest/vignettes/selectorgadget.html



Using R to Access External Data

- Common web browser tasks using rvest and RSelenium
  - Navigating to a web page
  - Follow links
  - Extract information from a table
  - Extract information from free text
  - Manipulate pages with “complex” user interfaces

Using R to Access External Data

- Common web browser tasks using rvest and RSelenium
  - Navigating to a web page

```
library(rvest)

# Pull all the web site data from the link
webPage <- read_html("<< URL of web page >>")
```

  - Follow links

```
# Extract the link
webLink<-webPage %>%
  html_node("<< Use a reference to a link provided by Selector Gadget >>") %>%
  html_attr("href")

# Go to the next page and extract the relevant table
webPage2<-read_html(webLink)
```

Using R to Access External Data

- Common web browser tasks using rvest and RSelenium
  - Extract information from a table

```
# Extract info as a table
extractedTable<-webPage %>%
  html_nodes("table") %>%
  .[[1]] %>%
  html_table()
```

  - Extract information from free text

```
# Extract text using a location from Selector Gadget
textData<-webPage %>%
  html_node("div.Rap12-Subtitle:nth-child(7)") %>%
  html_text()
```

Using R to Access External Data

- Common web browser tasks using rvest and RSelenium
  - Manipulate pages with “complex” user interfaces
    - Selenium is a useful tool for creating instances of a web browser
      Primarily used to test UIs, but can be useful in extracting data

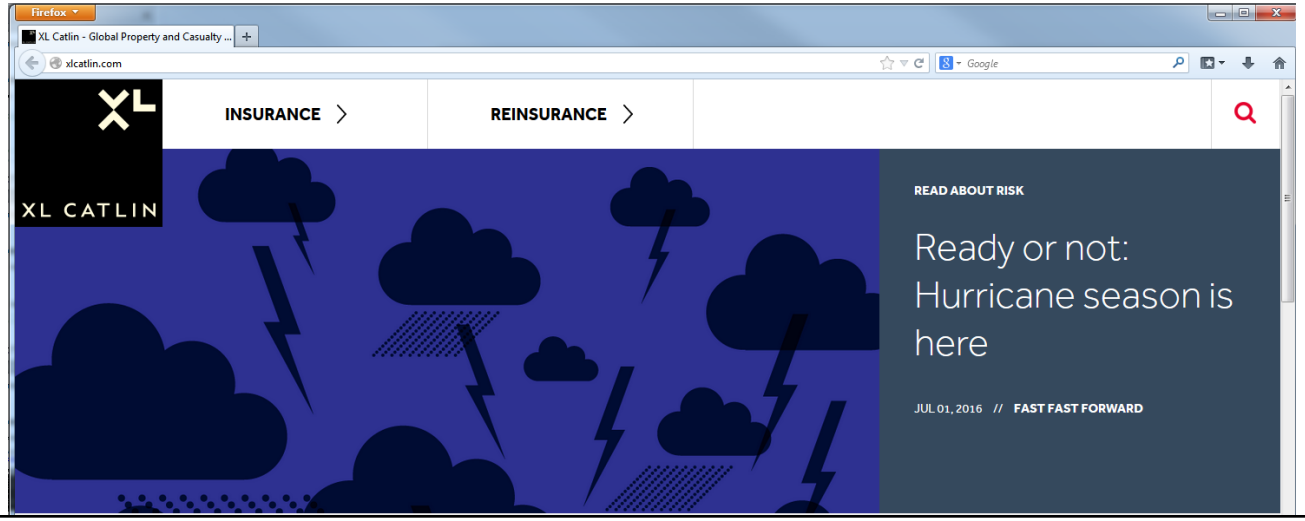
http://www.seleniumhq.org/download/

```
C:\Selenium>java -jar selenium-server-standalone-2.52.0.jar

12:24:06.711 INFO - Launching a standalone Selenium Server
12:24:06.963 INFO - Java: Oracle Corporation 23.1-b03
12:24:06.964 INFO - OS: Windows 7 6.1 amd64
12:24:07.116 INFO - v2.52.0, with Core v2.52.0. Built from revision 4c2593c
12:24:07.446 INFO - Driver class not found: com.opera.core.systems.OperaDriver
12:24:07.447 INFO - Driver provider com.opera.core.systems.OperaDriver is not re
gistered
12:24:07.514 INFO - Driver provider org.openqa.selenium.safari.SafariDriver regi
stration is skipped:
registration capabilities Capabilities [{platform=MAC, browserName=safari, versi
on=}] does not match the current platform VISTA
12:24:08.628 INFO - RemoteWebDriver instances should connect to: http://127.0.0.
1:4444/wd/hub
12:24:08.628 INFO - Selenium Server is up and running
```

Using R to Access External Data

- Common web browser tasks using rvest and RSelenium
  - Once a page is loaded, Selenium can send keystrokes and mouse clicks

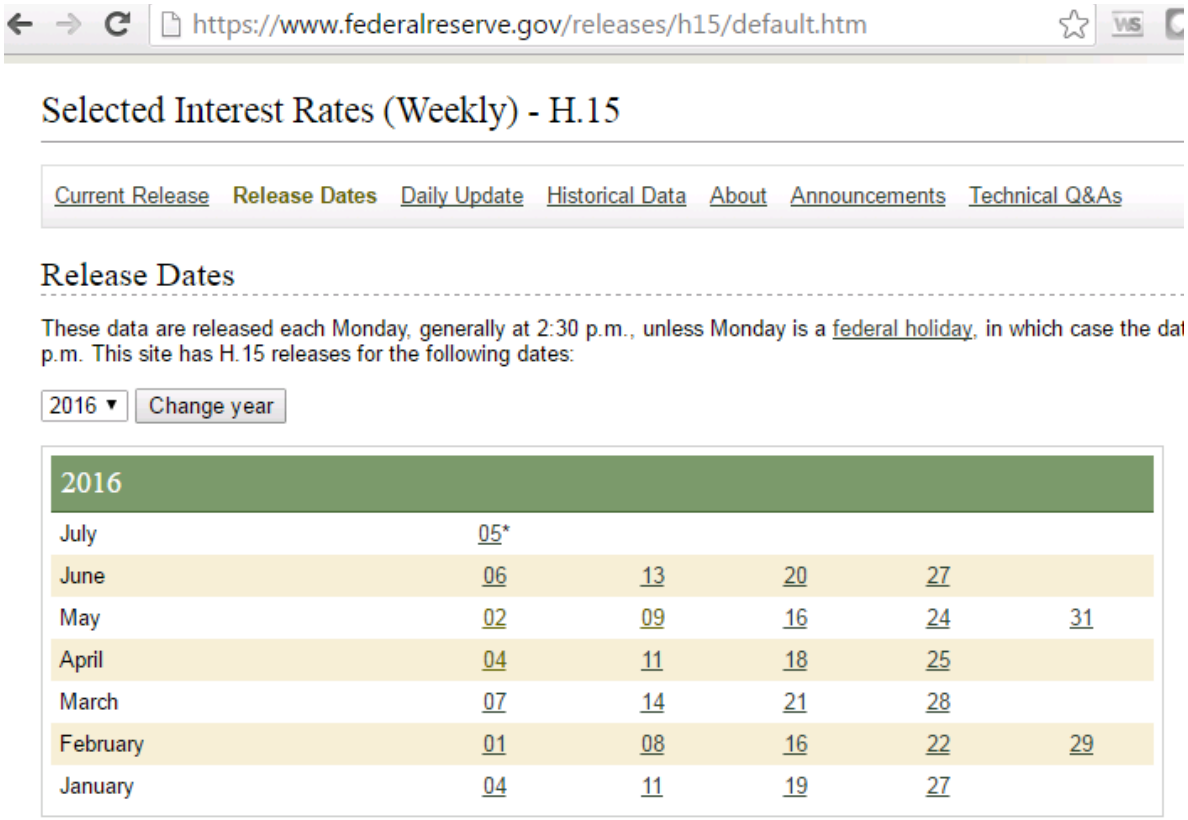
```
library(RSelenium)
remDr<-remoteDriver(remoteServerAddr = "localhost",
                    port=4444,
                    browserName = "firefox")
remDr$open()
remDr$navigate("http://www.xlcatlin.com/")
```



Illustrative Example

Illustrative Example

- Interest rate data



https://www.federalreserve.gov/releases/h15/default.htm

Selected Interest Rates (Weekly) - H.15

Current Release | **Release Dates** | Daily Update | Historical Data | About | Announcements | Technical Q&As

Release Dates

These data are released each Monday, generally at 2:30 p.m., unless Monday is a federal holiday, in which case the dat
p.m. This site has H.15 releases for the following dates:

2016 ▾ Change year

2016					
July	05*				
June	06	13	20	27	
May	02	09	16	24	31
April	04	11	18	25	
March	07	14	21	28	
February	01	08	16	22	29
January	04	11	19	27	

Illustrative Example

- Interest rate data

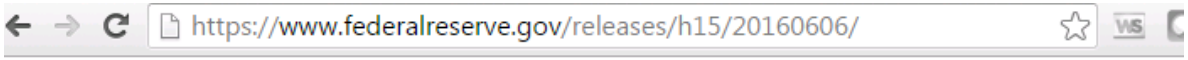
https://www.federalreserve.gov/releases/h15/current/

July 5, 2016
H.15 Selected Interest Rates
Yields in percent per annum

Instruments	2016 Jun 27	2016 Jun 28	2016 Jun 29	2016 Jun 30	2016 Jul 1	Week Ending		2016 Jun
						Jul 1	Jun 24	
Federal funds (effective) 1 2 3	0.41	0.41	0.41	0.30	0.41	0.40	0.38	0.38
Commercial Paper 3 4 5 6								
Nonfinancial								
1-month	0.39	0.37	0.40	0.36	0.37	0.38	0.36	0.38
2-month	0.45	0.41	0.42	0.41	0.40	0.42	0.40	0.43
3-month	0.52	0.45	0.48	0.45	0.45	0.47	0.48	0.49
Financial								
1-month	0.46	0.35	0.41	0.38	0.40	0.40	0.39	0.39
2-month	0.51	0.46	0.48	n.a.	n.a.	0.48	0.49	0.47
3-month	0.56	0.60	0.57	0.57	0.54	0.57	0.54	0.55
Eurodollar deposits (London) 3 7								
1-month	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48

Illustrative Example

- Interest rate data



June 6, 2016
H.15 Selected Interest Rates
Yields in percent per annum

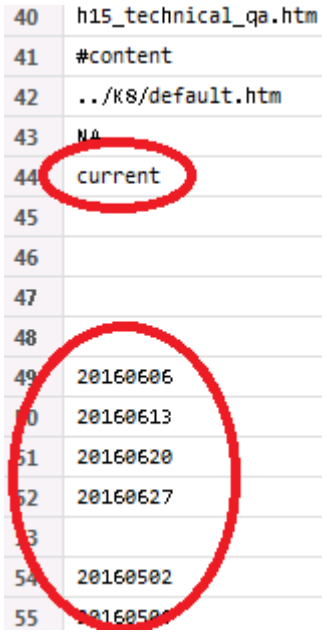
Instruments	2016 May 30*	2016 May 31	2016 Jun 1	2016 Jun 2	2016 Jun 3	Week Ending		2016 May
						Jun 3	May 27	
Federal funds (effective) 1 2 3	0.37	0.29	0.37	0.37	0.37	0.36	0.37	0.37
Commercial Paper 3 4 5 6								
Nonfinancial								
1-month		0.38	0.38	0.38	0.38	0.38	0.37	0.35
2-month		n.a.	0.44	n.a.	0.46	0.45	0.44	0.41
3-month		n.a.	n.a.	0.52	0.53	0.53	0.50	0.48
Financial								
1-month		0.36	0.43	0.34	0.35	0.37	0.40	0.38
2-month		0.48	0.52	0.44	0.46	0.48	0.52	0.47
3-month		0.59	0.60	0.56	0.57	0.58	0.61	0.57
Eurodollar deposits (London) 3 7								
1-month	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48

Illustrative Example

- Following the underlying links

2016					
July	05*				
June	06	13	20	27	
May	02	09	16	24	31
April	04	11	18	25	
March	07	14	21	28	
February	01	08	16	22	29
January	04	11	19	27	

```
webLink<-"https://www.federalreserve.gov/releases/h15/default.htm"

# Pull all the web site data from the link
webPage <- read_html(webLink)

# What are all the links on this page?
allLinks<-webPage %>%
  html_nodes("a") %>%
  html_attr("href")
allLinks<-data.frame(allLinks)
```



Illustrative Example

XL CATLIN

- Extract the relevant table

https://www.federalreserve.gov/releases/h15/20160606/

Instruments	2016 May 30*	2016 May 31	2016 Jun 1	2016 Jun 2	2016 Jun 3	Week Ending Jun 3	Week Ending May 27	2016 May
Federal funds (effective) 1 2 3	0.37	0.29	0.37	0.37	0.37	0.36	0.37	0.37
Commercial Paper 3 4 5 6								
Nonfinancial								
1-month		0.38	0.38	0.38	0.38	0.38	0.37	0.35
2-month		n.a.	0.44	n.a.	0.46	0.45	0.44	0.41
3-month		n.a.	n.a.	0.52	0.53	0.53	0.50	0.48

```
# Extract the table
dlTable<-webPage %>%
  html_nodes("table") %>%
  .[[3]] %>%
  html_table(fill=TRUE)
```

Instruments	2016May30*	2016May31	2016Jun1	2016Jun2	2016Jun3	Week Ending	NA	2016Ma
Jun3	May27	NA	NA	NA	NA	NA	NA	NA
Federal funds (effective) 1 2 3	0.37	0.29	0.37	0.37	0.37	0.36	0.37	0.37
Commercial Paper 3 4 5 6								
Nonfinancial								
1-month		0.38	0.38	0.38	0.38	0.38	0.37	0.35
2-month		n.a.	0.44	n.a.	0.46	0.45	0.44	0.41
3-month		n.a.	n.a.	0.52	0.53	0.53	0.50	0.48

Illustrative Example

- Problematic column header

https://www.federalreserve.gov/releases/h15/20160606/

	2016 May 30*	2016 May 31	2016 Jun 1	2016 Jun 2	2016 Jun 3	Week Ending Jun 3	Week Ending May 27	2016 May
3	0.37	0.29	0.37	0.37	0.37		0.37	0.37
		0.38	0.38	0.38	0.38	0.38	0.37	0.35
		n.a.	0.44	n.a.	0.46	0.45	0.44	0.41
		n.a.	n.a.	0.52	0.53	0.53	0.50	0.48

tr th

#col6 Clear (1)

Toggle Position XPath Help X

```
webPage %>%
html_node("#col4") %>%
html_text()
[1] "2016Jun2"
```

```
webPage %>%
html_node("#col5") %>%
html_text()
[1] "2016Jun3"
```

```
webPage %>%
html_node("#col6") %>%
html_text()
[1] "Jun3"
```

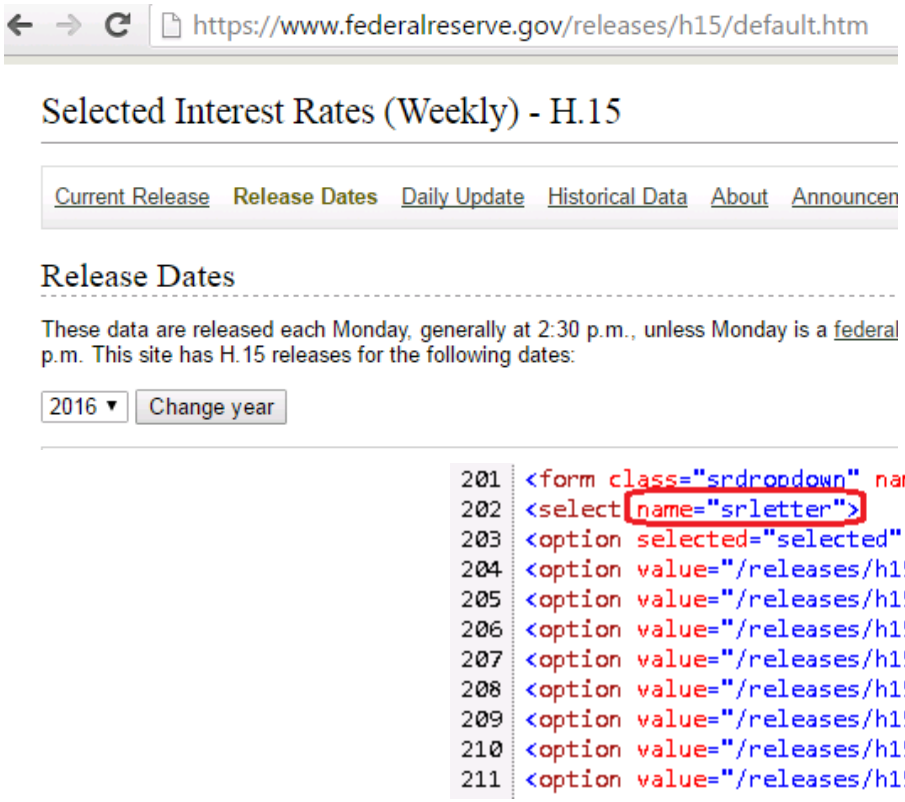
Illustrative Example

- Next steps
  - Perform data manipulation, then cycle through the remaining links and repeat, binding the tables from each iteration together
  - **NOTE!** The Federal Reserve provides much simpler ways to access data. This example was intended to show one way that R packages could be used to access data on a web page

Illustrative Example

XL CATLIN

- One way RSelenium could be used

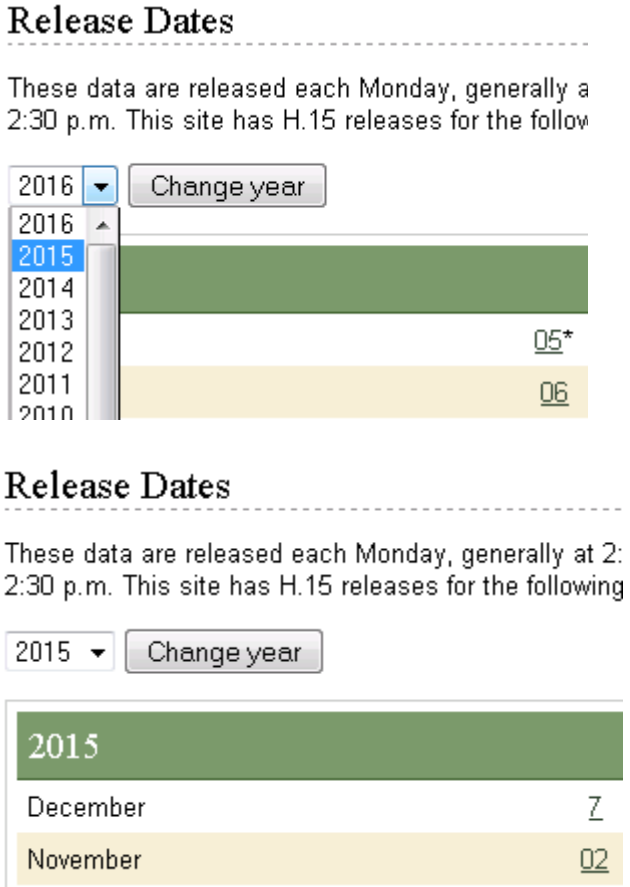



```
201 <form class="srdropdown" name="yearselect">
202 <select name="srletter">
203 <option selected="selected" value="default.htm">2016</option>
204 <option value="/releases/h15/2015.htm">2015</option>
205 <option value="/releases/h15/2014.htm">2014</option>
206 <option value="/releases/h15/2013.htm">2013</option>
207 <option value="/releases/h15/2012.htm">2012</option>
208 <option value="/releases/h15/2011.htm">2011</option>
209 <option value="/releases/h15/2010.htm">2010</option>
210 <option value="/releases/h15/2009.htm">2009</option>
211 <option value="/releases/h15/2008.htm">2008</option>
```

Illustrative Example

- One way RSelenium could be used

```
201 <form class="srdropdown" name="yearselect">
202 <select name="srletter">
203 <option selected="selected" value="default.htm">2016</option>
204 <option value="/releases/h15/2015.htm">2015</option>
webLink<-"https://www.federalreserve.gov/releases/h15/default.htm"
remDr$navigate(webLink)

# Go to the Year drop down
webElem<-remDr$findElement(using="name", value="srletter")

# Click on it
webElem$clickElement()

# Hit the home key
webElem$sendKeysToElement(list("\uE011"))

# Hit down key
webElem$sendKeysToElement(list("\uE015"))

# Go to the submit button
webElem<-remDr$findElement(using="name", value="button")

# Click on it
webElem$clickElement()
```



Conclusion

Conclusion

- Technical perspective
  - The functionality for navigating to web sites and extracting tables, links and free text using rvest provides useful building blocks
  - Identify items using Selector Gadget, which can then be used in rvest
  - Data on web sites which render interactively might be accessible with RSelenium
  - Find patterns in the way a web site arranges information
    - The illustrative example showed one approach, but there are other ways to extract data

Conclusion

- Final thoughts
  - External data can help the underwriters you work for make better pricing decisions
    - Even if that data can't be linked to policyholders/claims, it could be presented in a shiny app so that underwriting can consider it
  - Always follow a publisher's terms & conditions and be a considerate user

Questions?