KIT306/606 Tutorial 4

Student ID	Name

The following tutorial work should be completed by tutorial 5 (week6).

Install the `rvest` package

Install the rvest package which includes wrappers around the XML and httr packages to make it easy to download, then manipulate, both html and xml.

Let's install the rvest package by using the following command:

```
> install.packages("rvest")
--- Please select a CRAN mirror for use in this session ---
```

Select one of the CRAN mirror (Recommend 'Melbourne')

Your R console will automatically download and install the rvest package.

```
trying URL 'http://cran.ms.unimelb.edu.au/bin/macosx/mavericks/contrib/3.2/rvest_0.2.0.tgz'
Content type 'application/x-gzip' length 2865764 bytes (2.7 MB)
==================================================
downloaded 2.7 MB


The downloaded binary packages are in
    /var/folders/3l/pnv29h0d5bn39t71b1j19blh0000gn/T//Rtmp6GyQcx/downloaded_packages
>
```

Once rvest package is downloaded, you are now ready to use rvest package by using the following command.

```
> library("rvest")
```

Web page scraping

Let's try to scrape one of the web pages in Wikipedia website. Before we get started, it is important to choose which section you would like to extract.

The following url allows to view the information of Barack Obama in Wikipedia.
https://en.wikipedia.org/wiki/Barack_Obama

Assume that we would like to extract the red-coloured section, which is surrounded by the table form, and contains the brief summary of 'Barack Obama'

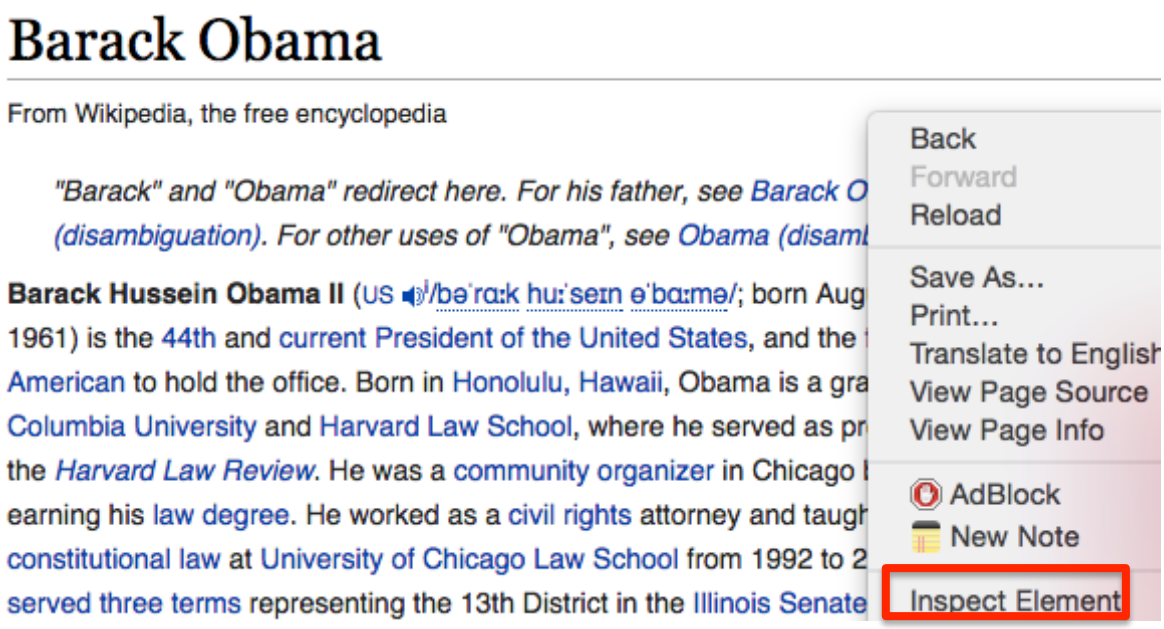
First, we need to scrap whole page of Barack Obama Wiki page.

1. Define the url that you would like to collect
2. Use `html` function that allows you to parse an HTML page.

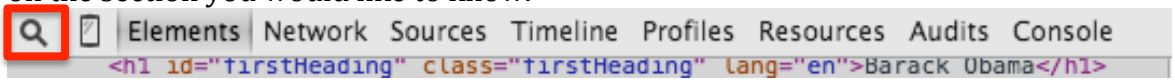
```
> wikiobama<-"https://en.wikipedia.org/wiki/Barack_Obama"
> obamapage<-html(wikiobama)
```

Then, it is crucial to define which section you would like to extract. The expected section should be checked with the source code.

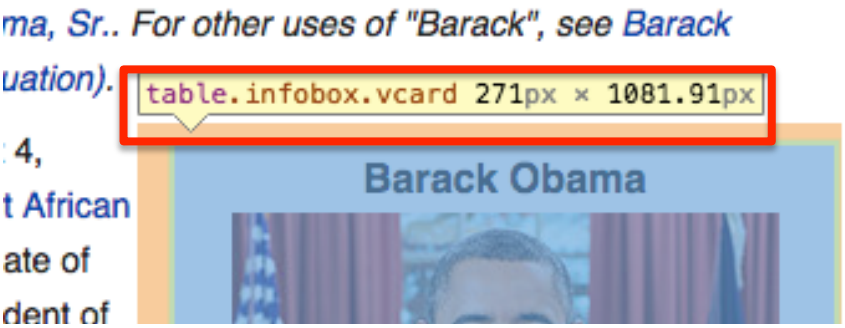
Go to the web page (https://en.wikipedia.org/wiki/Barack_Obama), do the right-click, and select 'Inspect Element' (* In case, you use Chrome as a browser. If you use Firefox, you can see the detailed source code with firebug)



Once you selected 'Inspect Element' option, you can select the following icon, and check the HTML tag details (including class, id, or any other attributes) by hovering on the section you would like to know.



The following image shows what will be happened when you hover on the table mentioned above.



As you can see the image (with red-coloured box), it shows .infobox.vcard.
In the CSS, a **class** selector is a name preceded by a **full stop**(".") and an **ID** selector is a name preceded by a **hash character** ("#").

The function html_nodes allows you to select nodes from an HTML document.
(NOTE: we made a 'obamapage' variable above by using html function)
You can find the detailed information of html_nodes function using help function.

```
> html<-html_nodes(obamapage,".infobox.vcard")
> text<-html_text(html)
```

html nodes select the nodes and extract the contents, and html_text allows you to extract attributes, text and tag name from html.

Let's print out and see what happened.

```
> print(text)
[1] "Barack Obama\n\n44th President of the United States\nIncumbent
\nAssumed office\nJanuary 20, 2009\nVice President\nJoe Biden
\nPreceded by\nGeorge W. Bush\nUnited States Senator\nfrom Illinois
\nIn office\nJanuary 3, 2005 – November 16, 2008\nPreceded by\nPeter
Fitzgerald\nSucceeded by\nRoland Burris\nMember of the Illinois
Senate\nfrom the 13th district\nIn office\nJanuary 8, 1997 –
November 4, 2004\nPreceded by\nAlice Palmer\nSucceeded by\nKwame
Raoul\nPersonal details\nBorn\nBarack Hussein Obama II(1961-08-04)
August 4, 1961 (age 54)Honolulu, Hawaii, U.S.\nNationality\nAmerican
\nPolitical party\nDemocratic\nSpouse(s)\nMichelle Robinson (1992-
present)\nChildren\n2\nResidence\nWhite House\nEducation\nPunahou
School\nAlma mater\n\n\nOccidental College\nColumbia University
(B.A.)\nHarvard Law School (J.D.)\n\n\nReligion
\nProtestantism[1]\nSignature\n\nWebsite\nbarackobama.com\n"
[2] "\nThis article is part of a series aboutBarack Obama\n\nEarly
life and career\nIllinois Senate\nU.S. Senate\n2008 Democratic
primaries\nPolitical positions\nPublic image\nFamily\nPresident of
the United StatesIncumbent\nFirst term\nCampaign for the Presidency
\n2008\n\nTransition\n1st inauguration\nPresidency\nFirst 100 days
\nNobel Peace Prize\nAffordable Care Act\nForeign trips\n\nTimeline:
'09\n'10\n'11\n'12\n\n\nPolicies \n\nClimate
changeEconomicEnergySocialForeign\n\n\nSecond term\nReelection
\n2012\nReactions\n\n2nd inauguration\nPresidency\nObama Doctrine
\nImmigration executive action\nIran Deal\nCuban Thaw\n\nTimeline:
'13\n'14\n'15\n\nElectoral history\nThe Audacity of Hope\nPlanned
Library\n\n\n\nv\nt\ne\n\n\n"
```

paste function allows you to concatenate vectors after converting to character.

```
> paste(text, collapse=" ")
[1] "Barack Obama\n\n44th President of the United States\nIncumbent
\nAssumed office\nJanuary 20, 2009\nVice President\nJoe Biden
\nPreceded by\nGeorge W. Bush\nUnited States Senator\nfrom Illinois
\nIn office\nJanuary 3, 2005 – November 16, 2008\nPreceded by\nPeter
Fitzgerald\nSucceeded by\nRoland Burris\nMember of the Illinois
Senate\nfrom the 13th district\nIn office\nJanuary 8, 1997 –
November 4, 2004\nPreceded by\nAlice Palmer\nSucceeded by\nKwame
Raoul\nPersonal details\nBorn\nBarack Hussein Obama II(1961-08-04)
August 4, 1961 (age 54)Honolulu, Hawaii, U.S.\nNationality\nAmerican
\nPolitical party\nDemocratic\nSpouse(s)\nMichelle Robinson (1992-
present)\nChildren\n2\nResidence\nWhite House\nEducation\nPunahou
School\nAlma mater\n\n\nOccidental College\nColumbia University
(B.A.)\nHarvard Law School (J.D.)\n\n\nReligion
\nProtestantism[1]\nSignature\n\nWebsite\nbarackobama.com\n \nThis
article is part of a series aboutBarack Obama\n\nEarly life and
career\nIllinois Senate\nU.S. Senate\n2008 Democratic primaries
\nPolitical positions\nPublic image\nFamily\nPresident of the United
StatesIncumbent\nFirst term\nCampaign for the Presidency\n2008\n
\nTransition\n1st inauguration\nPresidency\nFirst 100 days\nNobel
Peace Prize\nAffordable Care Act\nForeign trips\n\nTimeline:
'09\n'10\n'11\n'12\n\n\nPolicies \n\nClimate
changeEconomicEnergySocialForeign\n\n\nSecond term\nReelection
\n2012\nReactions\n\n2nd inauguration\nPresidency\nObama Doctrine
\nImmigration executive action\nIran Deal\nCuban Thaw\n\nTimeline:
'13\n'14\n'15\n\nElectoral history\nThe Audacity of Hope\nPlanned
Library\n\n\n\nv\nt\ne\n\n\n"
```

You can make a function (as below), and change the keyword to search different types of Wikipedia pages. ☺

```
> wiki<-function(keyword){
+
+ wikiurl<-"https://en.wikipedia.org/wiki/"
+ theurl<-paste0(wikiurl,keyword)
+
+ wikipage<-html(theurl)
+
+ infohtml<-html_nodes(wikipage,".infobox.vcard")
+ infotext<-html_text(infohtml)
+
+ paste(infotext, collapse=" ")
+ print(infotext)
+
+ }
```

After you made the above function, let's try to collect different types of web pages and check what will be happened.

```
> wiki("Barack Obama")

> wiki("Tony Abbott")

 > wiki("David Cameron")

> wiki("Beyonce")
```

- **HTML page scarping with xml_structure function**

First, we can collect the google website as well by using html function.

```
> google<-html("http://google.com")
```

The function xml_structure will allow you to show the structure of an html/xml document. %>%(a.k.a. %in%) is a binary operator called chain operator. The following codes shows that html_structure of the html page (google.com). After you extracted the html structure, you are now ready to extract all contents in p tag and a tag.

```
 > google %>% xml_structure()
> google %>% html_nodes("p")
 > google %>% html_nodes("a")
```

The following examples show that both hierarchical command or %>% chain command can derive the same result.

```
> ateam <- html("http://www.boxofficemojo.com/movies/?id=ateam.htm")
> html_nodes(ateam, "center")
> html_nodes(ateam, "center font")
> html_nodes(ateam, "center font b")
> ateam %>% html_nodes("center") %>% html_nodes("font")
> ateam %>% html_nodes("center") %>% html_nodes("font") %>% html_nodes("b")
```

Your page creation

You can also create the html page by using html function as follows:

```
> userpage <-html("<h1 id='title'>KIT306/606</h1><p class='a'><b>KIT306/606 is very
informative unit</b></p>")
> userpage
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-
html40/loose.dtd">
<html><body>
<h1 id="title">KIT306/606</h1>
<p class="a"><b>KIT306/606 is very informative unit</b></p>
</body></html>
```

Check whether the page is created properly using html_node function.

```
> h1<-html_node(userpage,"h1")
> h1
<h1 id="title">KIT306/606</h1>
> p<-html_node(userpage,"p")
> p
<p class="a">
  <b>KIT306/606 is very informative unit</b>
</p>
```

The above variable 'p' is saved in a vector format so you can check the child node of the p tag with the following command:

```
> p[[1]]
<b>KIT306/606 is very informative unit</b>
```

XML node

Xml function is the function that allows you to use all methods, which work the same as their HTML equivalents. Currently xml parses XML files as HTML because I can't find another way to ignore namespaces.

The following xml url contains the technology news in Sydney morning herald.
http://www.smh.com.au/rssheadlines/technology-news/article/rss.xml
Please check the above url and see the content of the page.

```
> technology<- xml("http://www.smh.com.au/rssheadlines/technology-news/article/rss.xml")
```

The news item variable includes the extracted item nodes in the technology xml page.

```
> newsitem <- technology %>% xml_nodes("item")
```

You can find the structure of the item nodes in technology xml page.

```
> newsitem[[1]] %>% xml_structure()
<item>
  <title> {text}
  <link>
  {text}
  <description> {text}
  <pubdate> {text}
  <guid [ispermalink]> {text}
```

The following command will extract the description content from each items.

```
> newsitem %>% xml_node("description") %>% xml_text()
```

Tutorial Questions

1. **Make a variable `userpage`, which contains an html page with the following code and `html` function.**

```
<h1>Caren's home page</h1>
<p>My first paragraph.</p>
<a>This is a link</a>
<table>
  <tr>
    <th>Student Name</th>
  </tr>
  <tr>
    <td>Jane Smith</td>
  </tr>
  <tr>
    <td>Michael Jones</td>
  </tr>
</table>
```

2. **Make a variable `userpage_xml` by using the following command:**

   ```
   userpage_xml= xml(userpage)
   ```

   , And make a variable `table_row` which contains the following result. (You should use a function `xml_nodes`.)

```
> table_row
[[1]]
<tr><th>Student Name</th>
  </tr>

[[2]]
<tr><td>Jane Smith</td>
  </tr>

[[3]]
<tr><td>Michael Jones</td>
  </tr>
```