Homework 1

Beau Harrison

http://www.trfetzer.com/wp-content/uploads/Homework1.pdf

Parameters:

pid (debate id), url, sequence within debate, debate, date, speaker, what was spoken (each fragment separately)

Libraries

```
library(xml2) # html_nodes
library(rvest) # %>% notation
library(data.table)
```

Get links within desired datespan

This block finds link nodes based on date

I'm using REGEX to find those debates from this election but I can remove this filter to return all elections

I'm not removing this filter because populating my data.table is extremely slow already

I find all rows with the class of 'docdate' and validate their text value against a regex date outputting only the rows I'm interested in

```
raw.html <- read_html('http://www.presidency.ucsb.edu/debates.php')
html.rows <-  raw.html %>%
              xml_nodes('tr')
html.docdates <-  html.rows %>%
                  xml_nodes('.docdate')
validDateRows <- grep('(January|February|March|April|May|June|July|August|September|October|November|De
linkRows <- sapply(html.docdates[validDateRows], function(x) max(grep(x, html.rows)))
```

Get attributes from each link

url

From the rows returned above find all anchor tags and return the value of the href property, url

```
urls <- sapply(html.rows[linkRows], function(x) x %>% xml_nodes('a') %>% xml_attr('href'))
```

pid

Use a string split to parse the pid from the url above

```
getPid <- function(url) return(unlist(strsplit(url, '='))[2])
```

date

Get the text from the docdates to use as the dates for each debate

```
getDate <- function(htmlPage) return(htmlPage %>% html_node('.docdate') %>% xml_text())
```

debate

Fortuantely there was a unique class name for the title of the debates

I use the paperstitle class to get the debate name for each debate

```
getDebate <- function(htmlPage) return(htmlPage %>% html_node('.paperstitle') %>% xml_text())
```

speaker

To find each speaker I find each paragraph and look for a bold tag inside the paragraph

If there is not a bold tag I ignore the line returned

I also do some clean up by parsing the colon out of each name

```
getSpeakers <- function(htmlPage) { #xml_child seems to be faster than html_node
  ps <- htmlPage %>% xml_nodes('p')
  bs <- unlist(lapply(ps[2:length(ps)], function(p) p %>% xml_child('b') %>% xml_text()))
  newSpeakerLines <- which(!is.na(bs))
  speakers <- unlist(lapply(newSpeakerLines, function(lineIndex) gsub(':','',bs[lineIndex])))
  return(speakers)
}
```

text

Here I use a similar strategy to the names but I'm parsing out the names

I also use the lines that I found the names on to combine p tags until I find another line with a name and create a new text block

```
getTexts <- function(htmlPage) { #xml_child seems to be faster than html_node
  ps <- htmlPage %>% xml_nodes('p')
  bs <- unlist(lapply(ps[2:length(ps)], function(p) p %>% xml_child('b') %>% xml_text()))
  newSpeakerLines <- which(!is.na(bs))
  psTextWNames <- ps %>% xml_text()
  psText <- gsub('^[A-z]+:', '', psTextWNames[2:length(psTextWNames)]) # parse out names
  newSpeakerLines <- c(newSpeakerLines, length(psText)) # append index for last line of text
  speakerText <- unlist(lapply(1:(length(newSpeakerLines)-1), function(lineNumber) paste(psText[newSpeal
  return(speakerText)
}
```

combined

Parameters:

pid (debate id), url, sequence within debate, debate, date, speaker, what was spoken (each fragment separately)

This was very slow with data.table even after unlisting all my lists

```
pidList <- integer()
urlList <- character()
seqList <- integer()
debateList <- character()
dateList <- character()
speakerList <- character()
textList <- character()

for(i in 1L:length(urls)) {
  htmlPage <- read_html(urls[i])
  pid <- getPid(urls[i])
  speakers <- getSpeakers(htmlPage)
  speakerList <- c(speakerList, speakers)
  texts <- getTexts(htmlPage)
  textList <- c(textList, texts)
  debate <- getDebate(htmlPage)
  date <- getDate(htmlPage)

  for(j in 1L:length(speakers)) {
    pidList <- c(pidList, pid)
    urlList <- c(urlList, urls[i])
    seqList <- c(seqList, j)
    debateList <- c(debateList, debate)
    dateList <- c(dateList, date)
  }
}

final <- data.table(pid=pidList,url=urlList,seq=seqList,debate=debateList,date=dateList,speaker=speakerl
```

Results

Number of Debates

```
sum(!duplicated(final$debate))
```

```
## [1] 31
```

Number of Speakers

```
sum(!duplicated(final$speaker))
```

```
## [1] 107
```

head() of data.frame or data.table objects

```
head(final)
```

```
##       pid                                                 url seq
## 1: 119039 http://www.presidency.ucsb.edu/ws/index.php?pid=119039   1
## 2: 119039 http://www.presidency.ucsb.edu/ws/index.php?pid=119039   2
## 3: 119039 http://www.presidency.ucsb.edu/ws/index.php?pid=119039   3
## 4: 119039 http://www.presidency.ucsb.edu/ws/index.php?pid=119039   4
## 5: 119039 http://www.presidency.ucsb.edu/ws/index.php?pid=119039   5
```

```
## 6: 119039 http://www.presidency.ucsb.edu/ws/index.php?pid=119039   6
##                                                              debate
## 1: Presidential Debate at the University of Nevada in Las Vegas
## 2: Presidential Debate at the University of Nevada in Las Vegas
## 3: Presidential Debate at the University of Nevada in Las Vegas
## 4: Presidential Debate at the University of Nevada in Las Vegas
## 5: Presidential Debate at the University of Nevada in Las Vegas
## 6: Presidential Debate at the University of Nevada in Las Vegas
##                date speaker
## 1: October 19, 2016 WALLACE
## 2: October 19, 2016 CLINTON
## 3: October 19, 2016 WALLACE
## 4: October 19, 2016   TRUMP
## 5: October 19, 2016 WALLACE
## 6: October 19, 2016 CLINTON
##
## 1:                 Good evening from the Thomas and Mack Center at the University of Nevada, Las Vega
## 2:  Thank you very much, Chris. And thanks to UNLV for hosting us.You know, I think when we talk abou
## 3:
## 4:
## 5:
## 6:
```