Appendix(A) - R script code for data cleanse up

November 17, 2016

```r
rm(list = ls())
getexpr = function(s,g)substring(s,g,g+attr(g,”match.length”)-1)
thepage <- readLines('https://en.wikipedia.org/wiki/Historical_polling_for_U.S._
    Presidential_elections')

tbregexp <- '<table class=”wikitable”>'
tbBlock <- grep(tbregexp, thepage)

regExp <- c('<caption><b><a href=.* title=.*>([0-9]*)</a>', #year
          '<th>([^<]*)<?.* \\(.*\\).*</th>',  #candidate name
          '<td.*>[^>]*>*([A-Z][a-z]*).*</td>', #month name
          '<td.*>[^0-9]*([0-9]+)%.*</td>')

totalYr <- c()
totalMn <- c()
totalCan <- c()
totalRt <- c()
for(i in 1:length(tbBlock)) {
  result <- list()
  for (j in 1:3) {
    ofrom <- tbBlock[i]
    if (i == length(tbBlock))
      oto = length(thepage)-1
    else oto = tbBlock[i+1]
    datalines <- grep(regExp[j],thepage[ofrom:oto],value=TRUE)
    gg <- gregexpr(regExp[j],datalines)
    matches <- c()
    for(k in 1:length(gg)) {
      matches <- c(matches,getexpr(datalines[k],gg[[k]]))
    }

    result[[j]] <- gsub(regExp[j],'\\1',matches)
  }
  #datalines at this time should store line# of different month
  datalines <- grep(regExp[3],thepage[ofrom:oto])
  repMon <- c()
  wmatches <- c()
  cand <- c()

  for(j in 1:(length(datalines))) {
    from = ofrom+datalines[j]-1
    if (j == length(datalines)) to = oto
    else to = ofrom+datalines[j+1]-1

    rate <- grep(regExp[4], thepage[from:to],value=TRUE)
    gg <- gregexpr(regExp[4],rate)

    for(k in 1:length(gg)) {
      matches <- getexpr(rate[k], gg[[k]])
      wmatches <- c(wmatches,matches)
      repMon <- c(repMon, rep(result[[3]][j], time=length(matches)))
    }
    wmatches <- gsub(regExp[4],'\\1',wmatches)
  }

  for(k in 1:(length(repMon)/length(result[[2]])))
    cand <- c(cand, result[[2]])


}
totalYr <- c(totalYr, rep(result[[1]], length(wmatches)))
totalMn <- c(totalMn, repMon)
totalCan <- c(totalCan, cand)
totalRt <- c(totalRt, wmatches)
```

```
rm(list = ls())

library(rvest)
library(stringr)

tb <- "https://www.archives.gov/federal-register/electoral-college/scores.html" %>%
      read_html() %>%
      html_nodes(xpath = '//tr/td/table') %>%
      .[[1]]

hPath <- c('//tr[1]/th[1]',
          '//tr[2]/th',
          '//tr[3]/th',
          '//tr[4]/th',
          '//tr[4]/th',
          '//tr[5]/th',
          '//tr[5]/th',
          '//tr[6]/th',
          '//tr[7]/th',
          '//tr[8]/th')
tPath <- c('//tr[1]/th[2]',
          '//tr[2]/td',
          '//tr[3]/td[1]',
          '//tr[4]/td[1]',
          '//tr[4]/td[2]',
          '//tr[5]/td[1]',
          '//tr[5]/td[2]',
          '//tr[6]/td',
          '//tr[7]/td',
          '//tr[8]/td')
dtset <- data.frame(index=1:53)

#Election Column
head <- tb  %>%
  html_nodes(xpath = hPath[1]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[1]) %>%
  html_text(trim = TRUE)
colname <- head[1]
```

```
dtset[colname] <- text

#President Column
head <- tb  %>%
  html_nodes(xpath = hPath[2]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[2]) %>%
  html_text(trim = TRUE)

remove <- c("") #Used to Eliminate the extra "" col
head <- head[!head %in% remove]
text <- text[!text %in% remove]
colname <- head[1]
dtset[colname] <- text

#Main Opponent Column
head <- tb  %>%
  html_nodes(xpath = hPath[3]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[3]) %>%
  html_text(trim = TRUE)

pat = '([^[:digit:]])*\\]$'
head <- head[!head %in% remove]
text <- text[grep(pat, text)]
colname <- head[1]
dtset[colname] <- text

#Winner Electoral Column
head <- tb  %>%
  html_nodes(xpath = hPath[4]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[4]) %>%
  html_text(trim = TRUE)

head <- head[!head %in% remove]
text <- text[!text %in% remove] #Used to Eliminate the extra "" col
colname <- paste(head[1],"-Winner")
dtset[colname] <- text

#Opponent Electoral Column
head <- tb  %>%
  html_nodes(xpath = hPath[5]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[5]) %>%
  html_text(trim = TRUE)

head <- head[!head %in% remove]
text <- text[!text %in% remove] #Used to Eliminate the extra "" col
colname <- paste(head[1],"-Opponent")
dtset[colname] <- text

#Popular Vote Winner Column
head <- tb  %>%
  html_nodes(xpath = hPath[6]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[6]) %>%
  html_text(trim = TRUE)

pat = '\r'
head <- head[!head %in% remove]
text <- text[!grepl(pat,text)]
colname <- paste(head[1],”-Winner”)
dtset[colname] <- text

#Popular Vote Opponent Column
head <- tb  %>%
  html_nodes(xpath = hPath[7]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[7]) %>%
  html_text(trim = TRUE)

pat = 'Return to Index'
head <- head[!head %in% remove]
text <- c(rep(”no record”, time=9), text[!grepl(pat,text)])
colname <- paste(head[1],”-Opponent”)
dtset[colname] <- text

#Vote for Others Column
head <- tb  %>%
  html_nodes(xpath = hPath[8]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[8]) %>%
  html_text(trim = TRUE)

pat = 'Votes for Others'

head <- head[!head %in% remove]
text <- text[!text %in% remove]
condition <- !grepl(pat,head)
copy <-text
text[condition] <- 'NA'

colname <- head[1]
dtset[colname] <- text

#Vice President Column
head <- tb  %>%
  html_nodes(xpath = hPath[9]) %>%
  html_text(trim = TRUE)
text <- tb  %>%
  html_nodes(xpath = tPath[9]) %>%
  html_text(trim = TRUE)

pat = 'Vice President'
rmpat = 'Notes|(Return to Index)'
head <- head[!head %in% remove]
text <- text[!text %in% remove]
text <- text[!grepl(rmpat,text)]
head[condition] <- pat
prev <- text
```

```r
text[condition] <- copy[condition]

colname <- head[1]
dtset[colname] <- text


#Further Cleanse up
dtset <- dtset[!names(dtset) %in% 'index']
partyPattern = '\\[([^[:digit:]])*\\]'
newCol1 = unlist(str_extract_all(dtset[2][,],partyPattern))
newCol2 = unlist(str_extract_all(dtset[3][,],partyPattern))

newCol1 <- gsub('(\\[)|(\\])','',newCol1)
newCol2 <- gsub('(\\[)|(\\])','',newCol2)

dtset[2][,] <-gsub(' \\[([^[:digit:]])*\\]','',dtset[2][,])
dtset[3][,] <-gsub(' \\[([^[:digit:]])*\\]','',dtset[3][,])

dtset$WinnerParty <- newCol1
dtset$OpponentParty <- newCol2

for (i in 4:7) {
  dtset[i][,] <- gsub('[^[:digit:]]*', '', dtset[i][,])
}

vpevpat <- ' (\\([0-9]*\\))'
vpev <- unlist(str_extract_all(dtset[9][,],vpevpat)) %>%
        gsub(pattern='(\\()|(\\))',replacement = '')
vpev <- c(rep('',time = 4),vpev)
dtset[9][,] <- gsub(' \\([0-9]*\\)', '', dtset[9][,])
dtset$VicePresidentEV <- vpev

#Candidate Pool Dataset
data.frame(dtset$Election ,dtset$`Votes for Others`)
ele <- dtset$Election
cpYr <- c()
cpLst <- c()
cpVote <- c()

for (i in 1:length(ele)) {
  sigYrVote <- dtset$`Votes for Others`[i]
  pat <- '[:alpha:]([:alpha:]| |\\.)*[:alpha:] \\([0-9]*\\)'
  if (grepl('^NA$', sigYrVote)) {
    cpYr <- c(cpYr, ele[i])
    cpLst <- c(cpLst, '')
    cpVote <- c(cpVote, '')
  } else {
    temp <- unlist(str_extract_all(sigYrVote, pat))
    sigYrVote <- temp %>% gsub(pattern=vpevpat, replacement='')
    vote <- temp %>% gsub(pattern='[^0-9]', replacement='')
    cpYr <- c(cpYr, rep(ele[i], length(sigYrVote)))
    cpLst <- c(cpLst, sigYrVote)
    cpVote <- c(cpVote, vote)
  }
}

dtset <- dtset[!names(dtset) %in% 'Votes for Others']
data.frame(Election=cpYr, 'Candidate List'=cpLst, 'Candidate Vote'=cpVote) %>% write.csv(row.names=FALSE, file='C:/Users/Yufan/Desktop/CMSC424/CandidatePool.csv')
```

```
dtset %>% write.csv(row.names=FALSE, file='C:/Users/Yufan/Desktop/CMSC424/
    ElectoralVoteDataSet.csv')
```

1