data_scraping.R

michaeltang

2020-06-25

```
# created on June 24, 2020
```

```
# load packages
library(XML)
library(RCurl)
library(magrittr)
```

```
# 1. Data scraping exercise
# a) import data

# enter URL given
url <- "https://en.wikipedia.org/wiki/Opinion_polling_in_the_Canadian_federal_election,_2015"

# get data from url
urldata <- getURL(url)

# read data from campaign period polls, i.e. the first table on the website
# then do the same for pre-campiagn period polls
# set header = TRUE when table has the appropriate column names
data_camp <- readHTMLTable(urldata, header = TRUE, which = 1)
data_precamp <- readHTMLTable(urldata, header = TRUE, which = 2)

# since the two data frames have the same column names, use rbind to
# concatenate them vertically
data_table <- rbind(data_precamp, data_camp)

# b) cleaning data

# (i) remove rows on elections 2011 and 2015
# (ii) remove empty rows
# direct resulted data frame to output table named optable
optable <- data_table %>%
            subset(., .$"Polling firm" != "Election") %>%
            subset(., .$"Polling firm" != "")

# (iii) transform the last date of polling data to a numeric date format
# of year-month-day, as.Date default format is yyyy-mm-dd
optable$"Last dateof polling\n" <-
  as.Date(optable$"Last dateof polling\n", format = "%B %d, %Y")

# (iv) keep only polls from 2014 and after
```

```
optable <- subset(optable, optable$"Last dateof polling\n" >= "2014-01-01")

# (v) transform the margin of error data format to be simply numeric characters
# grepl("^[^0-9]+$", .) searches for strings containing numbers only,
# ifelse(grepl("^[^0-9]+$", .), "", .), if not found, replace with ""
optable$"Marginof error[1]" <-
  gsub(" pp", "", optable$"Marginof error[1]") %>%   # replace all " pp" strings with ""
  gsub("±", "", .) %>%   # replace all "±" with ""
  ifelse(grepl("^[^0-9]+$", .), "", .) %>%   # replace "n.a", "NA", or any non-numeric entries, with ""
  gsub("NA", "", .) %>% as.numeric(.)   # convert char strings to numeric type

# (vi) transform the 5 major federal parties data to be on a scale of 0-1,
# meaning a 38.1% results should become a 0.381
optable[, c("Cons.\n", "NDP\n", "Liberal\n", "BQ\n", "Green\n")] <-
  lapply(optable[ , c("Cons.\n", "NDP\n", "Liberal\n", "BQ\n", "Green\n")],
         function(x) {as.numeric(x)/100})

# (vii) remove the columns ‘Lead’ and ‘Link’
optable <- optable[, !(names(optable) %in% c("Lead\n", "Link\n"))]

# (viii) rename your columns to firm, lastDateOfPolling, lpc, cpc, ndp, bq, green,
# MOE, sampleSize and method.
colnames(optable)[colnames(optable) == "Polling firm\n"] = "firm"
colnames(optable)[colnames(optable) == "Last dateof polling\n"] = "lastDateOfPolling"
colnames(optable)[colnames(optable) == "Liberal\n"] = "lpc"
colnames(optable)[colnames(optable) == "Cons.\n"] = "cpc"
colnames(optable)[colnames(optable) == "NDP\n"] = "ndp"
colnames(optable)[colnames(optable) == "BQ\n"] = "bq"
colnames(optable)[colnames(optable) == "Green\n"] = "green"
colnames(optable)[colnames(optable) == "Marginof error[1]"] = "MOE"
colnames(optable)[colnames(optable) == "Samplesize[2]"] = "sampleSize"
colnames(optable)[colnames(optable) == "Polling method[3]"] = "method"

# (ix) reorder your data by last date of polling, from the oldest to most recent polls.
# default option of order() is ascending, i.e. from the oldest to the most recent dates
optable <- optable[order(optable$"lastDateOfPolling"), ]

# (v) write a csv named ‘e2015polls’
filedir = ’/Users/michaeltang/Downloads/ds_project’
filename = ’e2015polls.csv’
filepath = file.path(filedir, filename)

write.csv(optable, filepath, row.names=FALSE)
```

```
# ref:
# using readHTMLTable
# https://www.rdocumentation.org/packages/XML/versions/3.99-0.3/topics/readHTMLTable

# using rbind
# https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/cbind
```

data_reshaping.R

michaeltang

2020-06-25

```
# created on June 24, 2020
```

```
# load package
library(magrittr)
```

```
#2. Data re-shaping exercise

# enter directory and name of input file
fdir = ’/Users/michaeltang/Downloads/ds_project’
inputfname = ’Question2sample.csv’
inputfpath = file.path(fdir, inputfname)

# assign data frame with data from input file, skip the first two rows
inputdata <- read.csv(inputfpath, skip = 2)

# as we are interested in total population by age groups in different Canadian
# provinces and territories, do the following to achieve the data frame with relevant
# entries of data
# (i) select "Age characteristics" under the "topic" column,
# (ii) select the columns named "Prov_name", "Charcter.", "Total" only
# (iii) remove rows of 15, 16, 17, 18, and 19 years under the "Charac." column
#       because we already have age group of 15 - 19 years old
# (iv) assign resultant data frame to ltable
ltable <- inputdata %>% .[.$"Topic" == "Age characteristics", ] %>%
              .[, names(.) %in% c("Prov_Name", "Characteristic", "Total")] %>%
                  subset(., !(.$"Characteristic" %in% c("      15 years", "      16 years",
                                              "      17 years", "      18 years",
                                              "      19 years")))

# we have obtained a table listing all necessary info. in long format,
# use reshape() function to transform the table to wide format
# assign result to wtable
wtable <- reshape(ltable, idvar = "Characteristic", timevar = "Prov_Name",
                  v.names = "Total", direction = "wide")

# by inspecting column names of wtable, "Total." is found on the majority of
# the columns
# use grepl to search for the pattern ("Total."), if found,
# use gsub to replace it with "", which gives us the proper names of the columns
for (i in 1:length(colnames(wtable))){
  if (grepl("Total.", colnames(wtable)[i])){
    colnames(wtable)[i] = gsub("Total.", "", colnames(wtable)[i])
  }
}

# change name of "Characteristic" column to "Age groups"
colnames(wtable)[colnames(wtable) == "Characteristic"] = "Age groups"


# write a csv named ‘census2011ageByProv’
filedir = ’/Users/michaeltang/Downloads/ds_project’
filename = ’census2011ageByProv.csv’
filepath = file.path(filedir, filename)

write.csv(wtable, filepath)
```

```
#ref:
# more on reshape()
# http://www.datasciencemadesimple.com/reshape-in-r-from-wide-to-long-from-long-to-wide/
# https://stats.idre.ucla.edu/r/faq/how-can-i-reshape-my-data-in-r/
# https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/reshape
```

data_analysis.R

michaeltang

2020-06-25

```
# created on June 24, 2020
```

```
# load package
library(magrittr)
```

```
#3. Data analysis exercise

# enter directory and name of input file
fdir = ’/Users/michaeltang/Downloads/ds_project’
inputfname = ’Question2sample.csv’
inputfpath = file.path(fdir, inputfname)

# assign data frame with data from input file, skip the first two rows
inputdata <- read.csv(inputfpath, skip = 2)

# select data under topic of "family characteristics",
# then use subset to get data in provinces in the Atlantic region,
# get columns interested (i.e. prov., charac., and total)
# assign trimmed data frame to lfamtable
lfamtable <- inputdata %>% .[.$"Topic" == "Family characteristics", ] %>%
                subset(., .$"Prov_Name" %in% c("New Brunswick", "Nova Scotia",
                        "Prince Edward Island", "Newfoundland and Labrador")) %>%
                          .[, names(.) %in% c("Prov_Name", "Characteristic", "Total")]

# before transforming lfamtable to wide format,
# get the number of rows for each province, call it len_ent
len_ent = length(lfamtable[lfamtable$"Prov_Name" == "Newfoundland and Labrador",
                           "Prov_Name"])

# create a list that goes from 1 to the end of list len_ent
list_ent = 1:len_ent

# as there are 4 provinces interested, replicate the list 4 times
# to generate a set of Ids for rows in each province
id = rep(list_ent, times = 4)

# assign id to lfamtable (long format family table)
# with unique id assigned to each row in each prov.,
# specific data in each row can be obtained conveniently,
# and duplicate rows under column "Characteristic" would not be omitted when
# transforming data frame from long to wide format
```

```
lfamtable$"Id" <- id

# use reshape() to transform lfamtable to wide format
wfamtable <- reshape(lfamtable, idvar = c("Id", "Characteristic"), timevar = "Prov_Name",
                     v.names = "Total", direction = "wide")

# task 1: get percentage of couple families (married + common-law)
# having children at home in the Atlantic region,

#and in diff. prov., with which percent of common-law couple

# using the unique id for each row, and the columns interested,
# get sum of number of married couples with children at home (id = 10)
# get sum of number of common-law couples with children at home (id = 16)
# get number of couple families having children at home
numCouFamChildHome = sum(wfamtable[10, 3:6]) + sum(wfamtable[16, 3:6])

# get total number of couple families (id = 6)
totalNumFCouFam = sum(wfamtable[7, 3:6])

# get percentage of couple families having children living at home
perCouFamChildHome = (numCouFamChildHome/totalNumFCouFam)*100

# print messages to console
msg1str = sprintf("Percentage of coup. fam. having child. at home. = %.2f", perCouFamChildHome)
print("Task 1:")
```

```
## [1] "Task 1:"
```

```
print(msg1str)
```

```
## [1] "Percentage of coup. fam. having child. at home. = 46.43"
```

```
# task 2: dist. of size of census families in the Atlantic region
# get total number of 2-person census family (id = 2)
# do the same for 3-,4-, and 5 person census family (id = 3, 4, 5)
# get total nunber of census families (id =1)
cenFam2p = sum(wfamtable[2, 3:6])
cenFam3p = sum(wfamtable[3, 3:6])
cenFam4p = sum(wfamtable[4, 3:6])
cenFam5p = sum(wfamtable[5, 3:6])
totalNumCenFam = sum(wfamtable[1, 3:6])

# get percentage of families having 2 persons in their household,
# do the same for families of other sizes
perCenFam2p = cenFam2p/totalNumCenFam*100
perCenFam3p = cenFam3p/totalNumCenFam*100
perCenFam4p = cenFam4p/totalNumCenFam*100
perCenFam5p = cenFam5p/totalNumCenFam*100

# print messages to console
msg2str1 = sprintf("Percentage of 2-person families. = %.2f", perCenFam2p)
```

```r
msg2str2 = sprintf("Percentage of 3-person families. = %.2f", perCenFam3p)
msg2str3 = sprintf("Percentage of 4-person families. = %.2f", perCenFam4p)
msg2str4 = sprintf("Percentage of 5-person families. = %.2f", perCenFam5p)

print("Task 2:")
```

```
## [1] "Task 2:"
```

```r
print(msg2str1)
```

```
## [1] "Percentage of 2-person families. = 55.24"
```

```r
print(msg2str2)
```

```
## [1] "Percentage of 3-person families. = 22.27"
```

```r
print(msg2str3)
```

```
## [1] "Percentage of 4-person families. = 16.82"
```

```r
print(msg2str4)
```

```
## [1] "Percentage of 5-person families. = 5.68"
```

```r
# task 3: get percentage of lone-parent families in total and
# then with female parent (mother-child family)
# get sum of total lone-parent families (id = 20)
# get sum of total number of census families (id = 6)
# get percentage by using the two mentioned sums
numLoneParFam = sum(wfamtable[20, 3:6])
totalNumCenFam = sum(wfamtable[6, 3:6])
perLoneParFam = numLoneParFam/totalNumCenFam*100

# get sum of number of lone-mother families (id = 21),
# then use it to get percentage by lone-mother families among lone-parent families
numLoneMomFam = sum(wfamtable[21, 3:6])
perLoneMomFam = numLoneMomFam/numLoneParFam*100

# print messages to console
msg3str1 = sprintf("Percentage of lone-parent fam. = %.2f", perLoneParFam)
msg3str2 = sprintf("Percentage of mother-parent fam. among them = %.2f", perLoneMomFam)

print("Task 3:")
```

```
## [1] "Task 3:"
```

```r
print(msg3str1)
```

```
## [1] "Percentage of lone-parent fam. = 16.39"
```

```r
print(msg3str2)
```

```
## [1] "Percentage of mother-parent fam. among them = 80.18"
```

```r
# task 4: get percent of common-law couples in total census families, how many of
# have children at home

# get total number of common-law couples (id = 14)
# get percentage of common-law couple among total num. of couple families
totalNumComLawCou = sum(wfamtable[14, 3:6])
perComLawCou = totalNumComLawCou/totalNumFCouFam*100

# get sum of num. of common-law couple having children at home (id = 16)
# get percentage of common-law couple having children at home
totalNumConLawCouChildHome = sum(wfamtable[16, 3:6])
perComLawCouChildHome = totalNumConLawCouChildHome/totalNumComLawCou*100

# print messages to console
msg4str1 = sprintf("Percentage of common-law couple families. = %.2f", perComLawCou)
msg4str2 = sprintf("Percentage of common-law couple fam. having children at home = %.2f", perComLawCouC

print("Task 4:")
```

```
## [1] "Task 4:"
```

```r
print(msg4str1)
```

```
## [1] "Percentage of common-law couple families. = 17.15"
```

```r
print(msg4str2)
```

```
## [1] "Percentage of common-law couple fam. having children at home = 42.18"
```

```r
# task 5: get percent of children under 18 in census families
# get total number of children under 18 (id = 30, 31, 32)
# get total number of children (id = 29)
# get the required percentage
totalNumChildUnder18 = sum(wfamtable[30:32, 3:6])
totalNumChild = sum(wfamtable[29, 3:6])
perNumChildUnder18 = totalNumChildUnder18/totalNumChild*100

# print messages to console
msg5str = sprintf("Percentage of children under 18. = %.2f", perNumChildUnder18)

print("Task 5:")
```

```
## [1] "Task 5:"
```

```r
print(msg5str)
```

```
## [1] "Percentage of children under 18. = 67.98"
```

```r
# task 6: get avg. num. of persons per census family and private household
# in the Atlantic region

# repeat the above steps to get a table showing household characteristics
lhousetable <- inputdata %>% .[.$"Topic" == "Household and dwelling characteristics", ] %>%
  subset(., .$"Prov_Name" %in% c("New Brunswick", "Nova Scotia",
                                 "Prince Edward Island", "Newfoundland and Labrador")) %>%
  .[, names(.) %in% c("Prov_Name", "Characteristic", "Total")]


# get number of rows for each province under house categ.,  call it len_ent_house
# create a list that goes from 1 to the end of list len_ent_house
# replicate the id 4 times for 4 different provinces
# assign unique id to lhousetable
len_ent_house = length(lhousetable[lhousetable$"Prov_Name" == "Newfoundland and Labrador",
                           "Prov_Name"])
list_ent_house = 1:len_ent_house
id_house = rep(list_ent_house, times = 4)
lhousetable$"Id" <- id_house

# use reshape() to transform lhousetable to wide format
whousetable <- reshape(lhousetable, idvar = c("Id", "Characteristic"), timevar = "Prov_Name",
                       v.names = "Total", direction = "wide")

# get total number of persons in households (id = 48)
# get total number of households (id = 41)
totalNumPerHouse = sum(whousetable[48, 3:6])
totalNumHouse = sum(whousetable[41, 3:6])

# get average number of persons per household
avgNumPerHouse = totalNumPerHouse/totalNumHouse

# print messages to console
msg6str = sprintf("Avg. num. of persons per household = %.2f", avgNumPerHouse)

print("Task 6:")
```

```
## [1] "Task 6:"
```

```r
print(msg6str)
```

```
## [1] "Avg. num. of persons per household = 2.36"
```

Analysis report exercise

Michael Tang

25/06/2020

The Atlantic region might not have a significant population size (around 2.3 million measured in 2011), yet it remains an integral part of Canada that we should not overlook. Using the 2011 Census data, we looked at the characteristics and composition of families and households in that region. Below are some major findings that could be important to the client,

- 46.4% of couple (married + common-law) families had children living at home
- 55.2% of census families had a household size of 2 persons
- 16.4% of the total census families were lone-parent families, of which 80.1% were mother-parent families
- 17.2% of the total census couple families were common-law couples, of which 42.2% had children living at home
- Children under 18 accounted for 68.0% of the total children in census families
- Average number of persons per household was 2.36

Evidently, it would be more likely to find families living in 2-person households with less 50% chance of finding couples who had children living with them at home, Also, a significant portion of lone-parent families were maintained by mothers. With only data from 2011, it was not possible to deduce how the percentage of common-law couples within the total census couple population would evolve with time. Nonetheless, close to half of the common-law couples had children living with them at home in 2011. In addition, 60% of the children population were under 18, suggesting that there could be some demand for children's products, school supplies, or day care services. Depending on what industry the client is working on, the data above could offer some insights on the way that business strategies could be drafted.

The statistics were obtained by first trimming the input data down to the relevant columns interested, then assigning an unique id to each row of the data frame, which was then transformed from long to wide format.

```r
# select data under topic of "family characteristics",
# then use subset to get data in provinces in the Atlantic region,
# get columns interested (i.e. prov., charac., and total)
# assign trimmed data frame to lfamtable
lfamtable <- inputdata %>% .[.$"Topic" == "Family characteristics", ] %>%
                subset(., .$"Prov_Name" %in% c("New Brunswick", "Nova Scotia",
                        "Prince Edward Island", "Newfoundland and Labrador")) %>%
                          .[, names(.) %in% c("Prov_Name", "Characteristic", "Total")]

# get the number of rows for each province, call it len_ent
len_ent = length(lfamtable[lfamtable$"Prov_Name" == "Newfoundland and Labrador",
                           "Prov_Name"])

# create a list that goes from 1 to the end of list len_ent
list_ent = 1:len_ent

# as there are 4 provinces interested, replicate the list 4 times
# to generate a set of Ids for rows in each province
id = rep(list_ent, times = 4)

lfamtable$"Id" <- id

# use reshape() to transform lfamtable to wide format
wfamtable <- reshape(lfamtable, idvar = c("Id", "Characteristic"), timevar = "Prov_Name",
                     v.names = "Total", direction = "wide")
```

Then, useful information embedded in the data was conveniently extracted by using the id assigned to the specific rows. For example, to obtain the percentage of couple families who have children at home, we could perform the following calculations to arrive at the results

```r
# get sum of number of married couples with children at home (id = 10)
# get sum of number of common-law couples with children at home (id = 16)
# get number of couple families having children at home
numCouFamChildHome = sum(wfamtable[10, 3:6]) + sum(wfamtable[16, 3:6])

# get total number of couple families (id = 6)
totalNumFCouFam = sum(wfamtable[7, 3:6])

# get percentage of couple families having children living at home
perCouFamChildHome = (numCouFamChildHome/totalNumFCouFam)*100
```

Similarly, other calculations were conducted using ids on the relevant rows.