Analysis of people completing the Getting and Cleaning Data Project

Sara Wang

Questions to ask: In the Coursera platform, thousands of students are taking data analysis online classes. Each of these classes includes a final project. For the *Getting and Cleaning Data project*, we want to explore the main sources of variation in how people complete the project.

Data Source: Getting and Cleaning Data project

Data Descriptions: There are 31,920 repository results for the *Getting and Cleaning Data Project*. For each repository name, it composes of username and the course title, such as `dnolan15/GettingAndCleaningDataCourseProject`. Within each of the repository name, it should contain a file named `run_analysis.r`, which contains code to get and clean data. Majority of the repositories contain the `run_analysis.r` file, which is the required file which proves their completion of the course. However, there are a few cases that students do not have `run_analysis.r` file under their repositories. The reason is that some of them have a `rmd` file instead to contain their getting and cleaning data code, or some students simply do not finish their project. Since we have a large population, with majority of the repositories, we are able to conduct analysis on how students get and clean data using r code. As `R` becomes increasingly popular for analysts to analyze data, it is interesting to know the variation of methods that people write in R to conduct their first step in analyzing data: how to get and clean data in R.

How to pull repositories and save data: When pulling data from github with the autherized key set-up, we can only pull 1,000 repositories at a time. No matter how much sleep time we put in the code to slow down the repository pulling command using `Sys.sleep()`, github gives the same 1,000 repositories each time. In order to get more repositories other than the same 1,000 ones, we can sort the github repositories by their creating dates. By doing this, we can use the `lubridate` package from R. According to the Coursera class *Getting and Cleaning Data Project* instructor, Jeff Leek, created the course on December, 2015. However, when pulling repositories created after 2015-12-01, we can only pull around 5,000 repositories instead of the total 31,920 repositories. I concluded that one reason could be that students created a repository earlier than the course started and changed the repository's name to be the course name. Another reason could be that the course was created earlier than December, 2015. For either reason, we need to trace back the repository created date earlier than 2015 to get a complete data set. By doing some explorations, I have found that if we can trace back the created dates to 2009, we can pull majority of the repositories from github. In order to receive students' codes to summarize the variations in how people complete the *Getting and Cleaning Data Project*, I script github website to get every student's repository name saved as `repos.rda`. By going into each

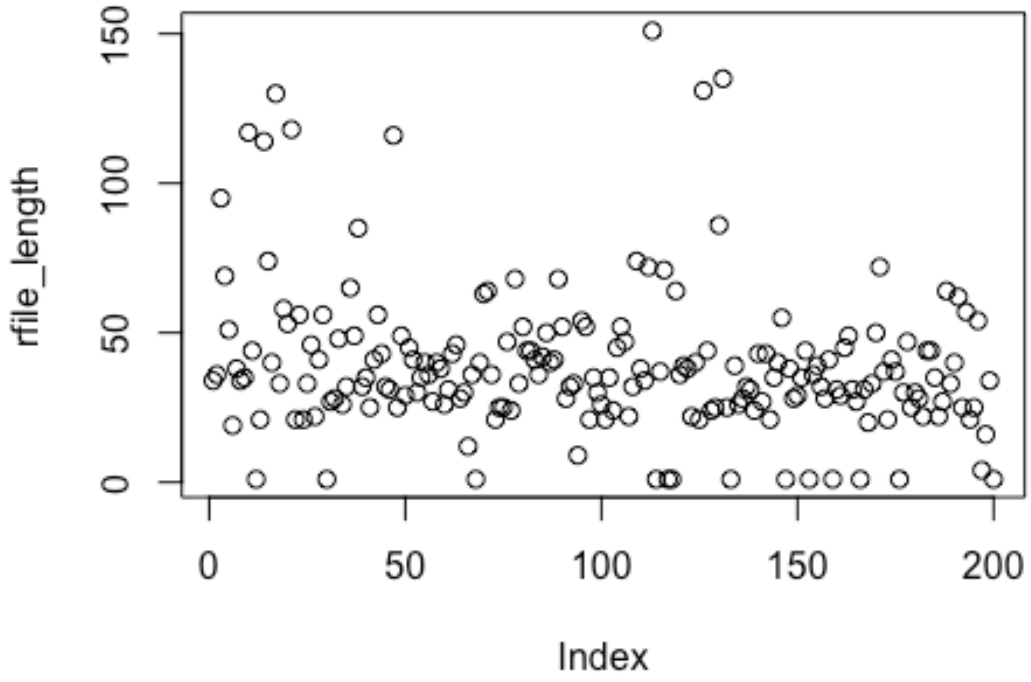
repository in `repos`, I find the `run_analysis.r` file and save each r file in a list named `rfile_list.rda`.

```
# sort based on dates. lubridate
library(dplyr)
library(gh)
library(lubridate)

# "Cite Andrew Leroux's code to get dates"
# start from 2008

date_start <- ymd("2015-12-01")        ## start date
day_inc  <- 14                        ## increment days by 14 at a time
dates <- c()
i <- 1
while(date_start < Sys.Date() - (day_inc+1)) {
        dates[[i]] <- c(rep(date_start,2) %m+% c(days(-1),days(day_inc+1)))
        date_start <- date_start + days(day_inc + 1)
        i <- i + 1
}


### NOTE: Need to create a personal access authentication token for using GET
/seach/code!!!
### Do this here: https://github.com/settings/tokens
token <- readLines("githubtoken.txt")
repos <- c()
for (dates_num in length(dates)){

### Only 100 results per page (the max). Change page=1 parameter to get all
the repositories.
gh_date <- paste("created:", paste(dates[[dates_num]], collapse=".."),
sep="")

  #1:10
  for (page_num in 1:10){

    repo_name <- paste0("GET
/search/repositories?q=getting+and+cleaning+data+",
            gh_date, "&per_page=100")
    x <- try(gh(repo_name, page = page_num, .token = token))

    if ("try-error" %in% class(x) == FALSE) {

      repos <- c(sapply(x[[3]], "[[", "full_name"),repos)
      print(page_num)
    }

  }
```

```
  Sys.sleep(60)
}

save(repos, file = "repos.rda")

library(dplyr)
library(gh)

load("repos.rda")
token <- readLines("githubtoken.txt")

rfile_list <- list()
for (i in 1:length(repos)){
  rfile_list[[i]] <- NA
}
# "Cite Stephen's code to use gh function getting repo names"


for (i in 1:length(repos)){
  string <- paste0("GET /search/code?q=repo:", repos[[i]],"+extension:r")
  res <- try(gh::gh(string, .token=token), silent = TRUE)

  if ("try-error"  %in% class(res) == FALSE) {
  # loop_path to get all .R files
    path <- try(res[[3]][[1]]$path, silent = TRUE)

    if ("try-error" %in% class(path) == FALSE) {

      code.url <- file.path("https://raw.githubusercontent.com",repos[[i]],
"master", path)
      code.url <- gsub(" ","%20",code.url)
      code <- code.url %>% readLines(warn = FALSE)
      # check comments later
      #execode <- code[!grepl("^#", code) & !grepl("\\ ^#", code) & code !=
"" & code != " "]

      print(i)
      rfile_list[[i]] <- code
      Sys.sleep(10)
    }
  }
}



save(rfile_list, file = "rfile_list.rda")
```

analysis for length of code: check how many lines of code each student writes to get and clean data and make a scatter plot. We observe that majority students write less than 50 lines of code to complete their project.

```
load("rfile_list.rda")
rfile_length <- unlist(lapply(rfile_list,length))
rfile_length_mean <- mean(rfile_length)
rfile_length_median <- median(rfile_length)
plot(rfile_length)
```



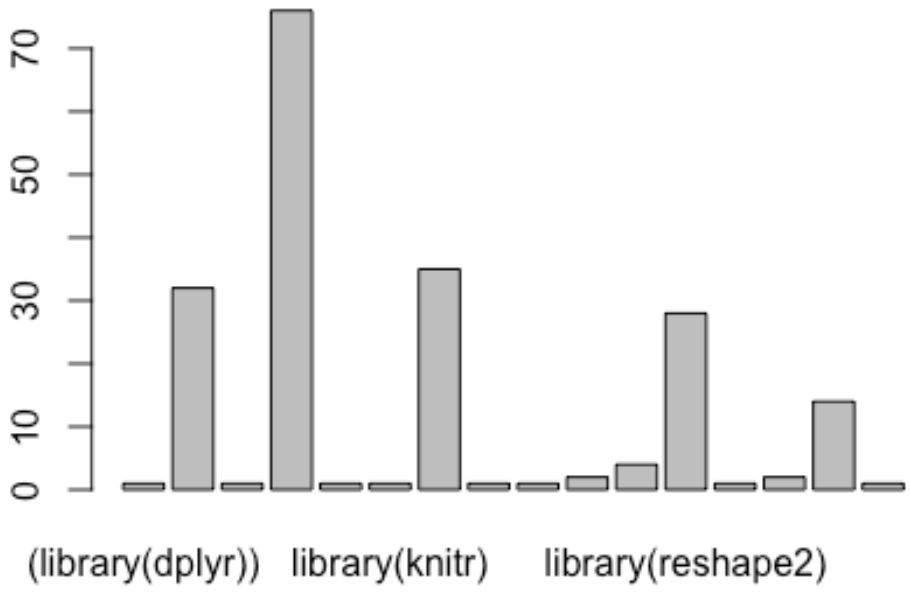
The mean of lines of code is `40`. The median of lines of code is `35`

analysis for library usage of code: check what libraries are used more often for students to finish their projects.

```
load("rfile_list.rda")
rfile_text <- unlist(rfile_list)
library_usage <- rfile_text[grep("library\\(", rfile_text)]
library_usage <- sapply(library_usage, function(x)
gsub("\'|\t|\"|suppressMessages| |;","",x))
names(library_usage) <- NULL
table(library_usage)
```

```
## library_usage
##    (library(dplyr)) library(data.table)      library(descr)
##                   1                  32                  1
##      library(dplyr)       library(httr)      library(knitr)
##                  76                   1                  1
##       library(plyr)      library(psych)       library(Rcpp)
##                  35                   1                  1
##      library(RCurl)    library(reshape)   library(reshape2)
##                   2                   4                 28
##      library(sqldf)    library(stringr)      library(tidyr)
##                   1                   2                 14
##      library(utils)
##                   1
```

```
barplot(table(library_usage))
```




```
function_usage <- rfile_text[grep("\\(", rfile_text)]


#table(function_usage)
```