Analysis

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

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

# "Cite Andrew Leroux's code to get dates"
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()
#length(dates)
for (dates_num in length(dates[1:2])){

### 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 1:2){

    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)) break

    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)
      execode <- code[!grepl("^#", code) & !grepl("\\ ^#", code) & code != ""
& code != " "]

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

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

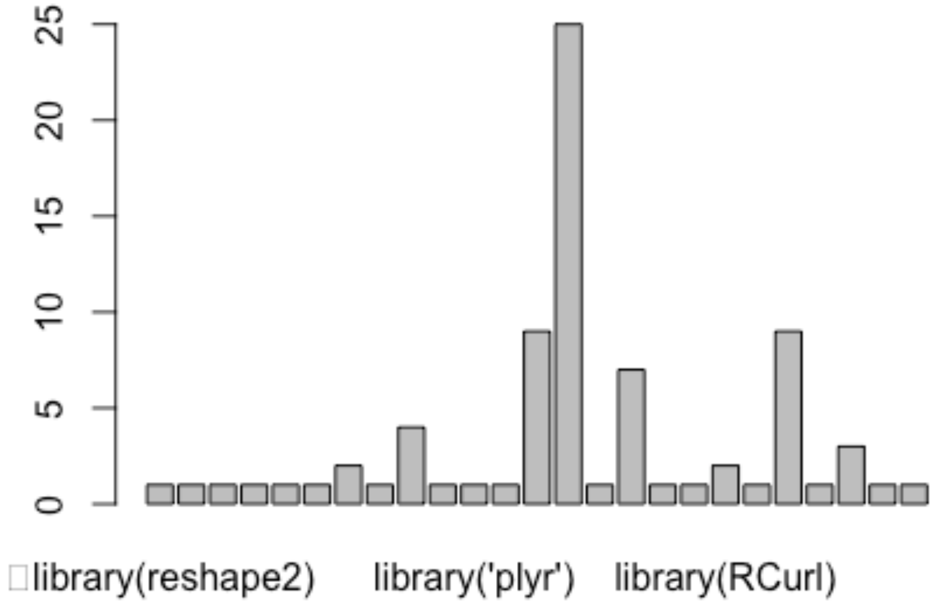
explotary analysis: check how many times libraries, functions, base R are used in the repos.

```r
load("rfile_list.rda")
rfile_text <- unlist(rfile_list)
library_usage <- rfile_text[grep("library\\(", rfile_text)]
function_usage <- rfile_text[grep("\\(", rfile_text)]

table(library_usage)

## library_usage
##               \tlibrary(reshape2)              library(data.table) 
##                                1                                1 
##                   library(dplyr)                library(reshape2) 
##                                1                                1 
##                   library(tidyr)              library(data.table) 
##                                1                                1 
##                   library(dplyr)              library(data.table) 
##                                2                                1 
##                   library(dplyr)                    library(plyr) 
##                                4                                1 
##                  library('plyr')                  library("plyr") 
##                                1                                1 
##              library(data.table)                   library(dplyr) 
##                                9                               25 
##                   library(knitr)                    library(plyr) 
##                                1                                7 
##                   library(plyr);                   library(psych) 
##                                1                                1 
##                   library(RCurl)                 library(reshape) 
##                                2                                1 
##                library(reshape2)                 library(stringr) 
##                                9                                1 
##                   library(tidyr)                   library(utils) 
##                                3                                1 
## suppressMessages(library(dplyr)) 
##                                1

barplot(table(library_usage))
```



```
#table(function_usage)
```