Week 2 - Quiz 1

Raphael Carvalho

18/07/2019

1. Register an application with the Github API here https://github.com/settings/applications. Access the API to get information on your instructors repositories (hint: this is the url you want "https://api.github.com/users/jtleek/repos"). Use this data to find the time that the datasharing repo was created. What time was it created? This tutorial may be useful (https://github.com/hadley/httr/blob/master/demo/oauth2-github.r). You may also need to run the code in the base R package and not R studio.

```
oauth_endpoints("github")
```

```
## <oauth_endpoint>
##  authorize: https://github.com/login/oauth/authorize
##  access:    https://github.com/login/oauth/access_token
```

```
myapp <- oauth_app("github",
                   key = Sys.getenv("GIT_KEY"),
                   secret = Sys.getenv("GIT_SECRET")
)
github_token <- oauth2.0_token(oauth_endpoints("github"), myapp)

gtoken <- config(token = github_token)
req <- GET("https://api.github.com/users/jtleek/repos", gtoken)
stop_for_status(req)
output <- content(req)

datashare <- which(sapply(output, FUN=function(X) "datasharing" %in% X))
list(output[[datashare]]$name, output[[datashare]]$created_at)
```

```
## [[1]]
## [1] "datasharing"
##
## [[2]]
## [1] "2013-11-07T13:25:07Z"
```

[ ] 2014-03-05T16:11:46Z

[x] 2013-11-07T13:25:07Z

[ ] 2012-06-20T18:39:06Z

[ ] 2014-02-06T16:13:11Z

2.he sqldf package allows for execution of SQL commands on R data frames. We will use the sqldf package to practice the queries we might send with the dbSendQuery command in RMySQL. Download the American Community Survey data and load it into an R object called `acs`(https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06pid.csv). Which of the following commands will select only the data for the probability weights pwgtp1 with ages less than 50?

```r
fileURL <- "https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06pid.csv"
download.file(fileURL, destfile = "acs.csv")
acs <- read.csv("acs.csv")
result <- sqldf("SELECT pwgtp1 FROM acs WHERE AGEP < 50")
str(result)
```

```
## 'data.frame':    10093 obs. of  1 variable:
##  $ pwgtp1: int  87 88 94 91 539 192 153 232 205 226 ...
```

[ ] sqldf(“select * from acs”)

[ ] sqldf(“select pwgtp1 from acs”)

[ ] sqldf(“select * from acs where AGEP < 50”)

[x] sqldf(“select pwgtp1 from acs where AGEP < 50”)

3. Using the same data frame you created in the previous problem, what is the equivalent function to unique(acs$AGEP)

```r
result <- sqldf("select distinct AGEP from acs")
str(result)
```

```
## 'data.frame':    91 obs. of  1 variable:
##  $ AGEP: int  43 42 16 14 29 40 15 28 30 4 ...
```

[x] sqldf(“select distinct AGEP from acs”)

[ ] sqldf(“select unique AGEP from acs”)

[ ] sqldf(“select distinct pwgtp1 from acs”)

[ ] sqldf(“select AGEP where unique from acs”)

4. How many characters are in the 10th, 20th, 30th and 100th lines of HTML from this page: http://biostat.jhsph.edu/~jleek/contact.html (Hint: the nchar() function in R may be helpful)

```r
htmlUrl <- url("http://biostat.jhsph.edu/~jleek/contact.html")
htmlCode <- readLines(htmlUrl)
close(htmlUrl)
c(nchar(htmlCode[10]), nchar(htmlCode[20]), nchar(htmlCode[30]), nchar(htmlCode[100]))
```

```
## [1] 45 31  7 25
```

[x] 45 31 7 25

[ ] 43 99 7 25

[ ] 45 0 2 2

[ ] 45 31 2 25

[ ] 43 99 8 6

[ ] 45 31 7 31

[ ] 45 92 7 2

5. Read this data set into R and report the sum of the numbers in the fourth of the nine columns (https://d396qusza40orc.cloudfront.net/getdata%2Fwksst8110.for). Original source of the data: http://www.cpc.ncep.noaa.gov/data/indices/wksst8110.for (Hint this is a fixed width file format)

```r
fileUrl <- "https://d396qusza40orc.cloudfront.net/getdata%2Fwksst8110.for"
dt <- read.fwf(fileUrl, skip=4, widths=c(12, 7, 4, 9, 4, 9, 4, 9, 4))
sum(dt[, 4])
```

```
## [1] 32426.7
```

[ ] 35824.9

[ ] 101.83

[x] 32426.7

[ ] 36.5

[ ] 28893.3

[ ] 222243.1