Getting and Cleaning Data - Data Science - Quiz 2 - Coursera

Getting and Cleaning Data Quiz 2

This is Quiz 2 from the Getting and Cleaning Data course within the Data Science Specialization on Coursera. Topics include querying an API and SQL database, html scraping, and fixed width files.

Questions

---

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.

- **2013-11-07T13:25:07Z**

```
library(httr)


oauth_endpoints("github")
## <oauth_endpoint>
##  authorize: https://github.com/login/oauth/authorize
##  access:    https://github.com/login/oauth/access_token
gitapp <- oauth_app("github",
  key = "e84aefedbce3a0690faf",
  secret = "bf589b8260cb6a26719b9cc64fa205d5da1abf26")


github_token <- oauth2.0_token(oauth_endpoints("github"), gitapp)


gtoken <- config(token = github_token)
req <- GET("https://api.github.com/rate_limit", gtoken)
stop_for_status(req)
content(req)
```

```
## $resources
## $resources$core
## $resources$core$limit
## [1] 5000
##
## $resources$core$remaining
## [1] 5000
##
## $resources$core$reset
## [1] 1477141265
##
##
## $resources$search
## $resources$search$limit
## [1] 30
##
## $resources$search$remaining
## [1] 30
##
## $resources$search$reset
## [1] 1477137725
##
##
## $resources$graphql
## $resources$graphql$limit
## [1] 200
##
## $resources$graphql$remaining
## [1] 200
##
## $resources$graphql$reset
## [1] 1477141265
##
##
##
## $rate
## $rate$limit
## [1] 5000
```

```
##
## $rate$remaining
## [1] 5000
##
## $rate$reset
## [1] 1477141265
library(jsonlite)
json1 = content(req)
json2 = jsonlite::fromJSON(toJSON(json1))
repo <- json2[5]
names(repo)
## [1] NA
repo$created_at
## NULL
```

---

2. The 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?

- **sqldf("select pwgtp1 from acs where AGEP < 50")**

```
library(sqldf)
download.file("https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06pid.csv", destf
ile = "quiz2data.csv")


acs <- read.csv("quiz2data.csv")


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)

- **sqldf("select distinct AGEP from acs")**

```
sqldf("select distinct AGEP 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)

- **45 31 7 25**

```
require(httr);require(XML)
## Loading required package: XML
URL <- url("http://biostat.jhsph.edu/~jleek/contact.html")
lines <- readLines(URL)
close(URL)
c(nchar(lines[10]), nchar(lines[20]), nchar(lines[30]), nchar(lines[100]))
## [1] 45 31  7 25
```

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)

- **32426.7**

```r
url <- "https://d396qusza40orc.cloudfront.net/getdata%2Fwksst8110.for"
widths <- c(1, 9, 5, 4, 1, 3, 5, 4, 1, 3, 5, 4, 1, 3, 5, 4, 1, 3)
fixed <- read.fwf(url, widths, header = FALSE, skip = 4)
sum(fixed$V8)
```

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