README.md

Source code: quiz2.R and utils.R

Question 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.

```
# Find OAuth settings for github:
# http://developer.github.com/v3/oauth/
github <- oauth_endpoints("github")

# Replace your key and secret below.
myapp <- oauth_app("github",
                   key = "319cffb9580b74b3e3fc",
                   secret = "147be6cc25e9a526f22dc19df75a2c1d47340ea5")

# Get OAuth credentials
github_token <- oauth2.0_token(github, myapp)

# Use the API
gtoken <- config(token = github_token)
req <- with_config(gtoken, GET("https://api.github.com/users/jtleek/repos"))
stop_for_status(req)
repo_list <- content(req)

answer1 <- c()
for (i in 1:length(repo_list)) {
  repo <- repo_list[[i]]
  if (repo$name == "datasharing") {
    answer1 = repo
    break
  }
}

# Expected output: The repository 'datasharing' was created at 2013-11-07T13:25:07Z
if (length(answer1) == 0) {
```

```
  msg("No such repository found: 'datasharing'")
} else {
  msg("The repository 'datasharing' was created at", answer1$created_at)
}
```

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

Question 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?

1. sqldf("select pwgtp1 from acs where AGEP < 50")
2. sqldf("select * from acs")
3. sqldf("select pwgtp1 from acs")
4. sqldf("select * from acs where AGEP < 50 and pwgtp1")

```
fname <- "survey.csv"
download_if_not_exists(fname, "https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2Fss06pid.csv")
acs <- read.csv(fname, header = TRUE, sep = ",")

answer2 <- sqldf("select pwgtp1 from acs where AGEP < 50")
msg("Probability weights for people with ages less than 50:")
msg("sqldf('select pwgtp1 from acs where AGEP < 50'):", head(answer2))
```

Answer: sqldf("select pwgtp1 from acs where AGEP < 50")

Question 3

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

1. sqldf("select distinct pwgtp1 from acs")
2. sqldf("select unique * from acs")
3. sqldf("select distinct AGEP from acs")
4. sqldf("select AGEP where unique from acs")

```
expected_result <- unique(acs$AGEP)
queries <- list(
  q1 <- "select distinct pwgtp1 from acs",
  q2 <- "select unique * from acs",
  q3 <- "select distinct AGEP from acs",
  q4 <- "select AGEP where unique from acs"
)
answer3 <- c()
lapply(queries, function(q) {
  result <- try(sqldf(q), silent = TRUE)
  if (inherits(result, "try-error")) {
    msg("Invalid query:", q)
  } else if (identical(result$AGEP, expected_result)) {
    answer3 <<- c(answer3, q)
  }
})
```

Answer: sqldf("select distinct AGEP from acs")

Question 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

```
tryCatch({
  con <- url("http://biostat.jhsph.edu/~jleek/contact.html")
  html <- readLines(con)
}, finally = {
  close(con)
})

answer4 <- c()
sapply(c(10, 20, 30, 40), function(line) {
  answer4 <<- c(answer4, nchar(html[line]))
})
# Expected output:
msg("Characters in the 10th, 20th, 30th and 100th lines of HTML:", paste(as.character(answer4), collapse = ", "))
```

Answer: 45 31 7 2

Question 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

```
fname <- "wksst8110.for"
download_if_not_exists(fname, "https://d396qusza40orc.cloudfront.net/getdata%2Fwksst8110.for")

# column sequence: 5x empty space, SST column, SSTA column
col_seq <- c(-5, 4, 4)

# rows: skip first four lines
# cols (left to right):
#   empty space (-1)
#   nine characters (9)
#   etc.
df <- read.fwf(fname,
               widths = c(-1, 9, col_seq, col_seq, col_seq, col_seq),
               skip = 4)
answer5 <- sum(df[, 4])

# Expected output: "The sum of numbers in the fourth column is 32426.7"
msg("The sum of numbers in the fourth column is", answer5)
```

Answer: 32426.7