Problem Set 5

Overview:

In this problem set you will be working on a shared repository with your homework group. Similar to the last problem set, we will not be asking for your git commands and output– you do not have to paste your git commands in R markdown and can simply use your command line interface. You will only need to submit (via pushing to your Github repository) your R script and any data files you create. We will be using the **rtweet** package in R to practice working with Twitter data, the **stringr** package (part of tidyverse) to work with strings, and the **lubridate** package for working with dates and times. *Note: Make sure you have a Twitter account (user name and password).* Lastly, we encourage you all to communicate with your group by creating issues on your shared repository.

Part I: Command Line & Organizating Project/Script

1. Have a new member of your group create a new private repository on GitHub here and name it `<team_name>_ps5`. Under the `Add .gitignore` option, select `R`, and then click `Create repository`.

   Invite the other group members as collaborators under `Settings` > `Manage access` > `Invite teams or people`. All members will clone this repository to their local machines.

2. Create a new RStudio project, setting this repository directory as the project working directory. Note that your R Console and RStudio Terminal will start up in this directory now.

   In your RStudio Terminal, check `git status`. You will see that an `.Rproj` file has been generated. For the purposes of this class, you can add `.Rproj` to `.gitignore`.

   Have a second member of your group add this to `.gitignore`, then add/commit this change and push to the remote. All other members will pull this change.

3. You can begin working individually now. You will be doing all work on a branch.

   Create a new branch called `dev_[initials]` and switch to it.

4. Create a directory called `data` where you will save your data file(s).

5. Create an R script and call it `<last_name>_script.R`

   Don't forget to follow the template provided here.

   In your R script, create a `data_dir` object that stores the file path to the `data` directory. Remember to write the path relative to the project directory.

6. Next, install (if necessary) and import the following packages in your R script: `tidyverse`, `rtweet`, and `lubridate`.

Part II: Working with strings

1. You will be using the rtweet package to fetch data from the twitter accounts of three news outlets– CNN, Fox News, and Univision.

2. Take a few minutes to skim the `rtweet` package documentation here.

3. In your R script, use the following to create an object called `news` that is a character vector of the twitter handles of the following three news outlets.

```r
news <- c("CNN", "FoxNews", "UniNoticias") #twitter handles

news
```

```
## [1] "CNN"         "FoxNews"     "UniNoticias"
```

4. Use the `search_tweets()` function to search for (1,000) tweets from the twitter handles of the character vector, `news`, we created above. Save the tweets dataframe in a variable called `news_df`.

   Hint: Do not forget to load your libraries (e.g. `tidyverse`, `rtweet`, `lubridate`)

5. Subset your dataframe `news` and create a new dataframe called `news_df2` keeping only the following variables: `user_id`, `status_id`, `created_at`, `screen_name`, `text`, `followers_count`, `friends_count`, `profile_expanded_url`.

6. Create a new column in `news_df2` called `text_len` that contains the length of the character variable `text`.

   What is the `class` and `type` of this new column?

7. Create another column in `news_df2` called `handle_followers` that stores the twitter handle and the number of followers associated with that twitter handle in a string. For example, the entries in the `handle_followers` column should look like this: "@[twitter_handle] has [number] followers."

   Now create another column called `handle_friends` that stores the twitter handle and the number of friends associated with that twitter handle in a string. Follow the steps you took for creating the `handle_followers` column.

   What is the `class` and `type` of these new columns?

8. Lastly, create a column in `news_df2` called `short_web` that contains a short version of the `profile_expanded_url` without the `http://www.` part of the url. The entries in that column should look something like this: "nytimes.com".

9. Use `saveRDS()` to save your `news_df2` in a file called `<last_name>_twitter_news.RDS` in the `data_dir`. Add this data file and commit with the message `add twitter news outlet data`.

Part III: Working with dates/times

We will be using twitter handles associated with the Cost of Living Adjustment (COLA) campaign across the 10 UC campuses.

Background:

In September 2019, organizers composed of graduate students, and organizations like the Graduate Student Association (GSA) and UAW2865 (Union representing tutors, readers, graduate student instructors and teaching assistants at the University of California) presented the following demand to the UCSC Administration:

> "A Cost of Living Adjustment for every graduate student, regardless of residence, visa, documentation, employment or funding status, to bring us:
>
> 1. Out of rent burden
> 2. Without raising tuition or campus fees
> 3. With a guarantee of non-retaliation"

After months of meetings and intimidation and threats from administration, COLA organizers declared a strike on February 10th, culminating in the support of hundreds of graduate and undergraduate students, faculty, and staff at the UCSC campus. Similar demonstrations and acts of solidarity followed across the UC campuses demanding for fair wages and living conditions. COLA organizers continue to organize admist the COVID-19 pandemic. For more information see https://payusmoreucsc.com/.

1. Create an object called `cola` that is a character vector of the twitter handles of the 10 UC's and affiliated COLA accounts.

   Your code should look similar to part II, question 3.

2. Use the `search_tweets()` function to search for (1,000) tweets from the twitter handles of the character vector, `cola`, we created above. Save the dataframe and call it `cola_df`.

   Subset your dataframe and call it `cola_df2` and keep only the following variables: `user_id`, `status_id`, `created_at`, `screen_name`, `text`, `followers_count`, `friends_count`.

3. Using the column `created_at`, create a new column in `cola_df2` called `dt_chr` that is a character version of `created_at`.

4. Create another column in `cola_df2` called `dt_len` that stores the length of `dt_chr`.

5. Next, create additional columns in `cola_df2` for each of the following date/time components:

   a. Create a new column `date_chr` for date (e.g. 2020-03-26) using the column `dt_chr` and the `str_sub()` function.
   b. Do the same for year `yr_chr` (e.g. 2020).
   c. Do the same for month `mth_chr` (e.g. 03).
   d. Do the same for day `day_chr` (e.g. 26).
   e. Do the same for time `time_chr` (e.g. 22:41:09).

6. Using the column we created in the previous question `time_chr`, create additional columns in `cola_df2` for the following time components:

   a. Create a new column `hr_chr` for hour (e.g. 22) using the column `time_chr` and the `str_sub()` function.
   b. Do the same for minutes `min_chr` (e.g. 41).
   c. Do the same for seconds `sec_chr` (e.g. 09).

7. Now let's get some practice with the `lubridate` package.

   a. Using the `year()` function from the `lubridate` package, create a new column in `cola_df2` called `yr_num` that contains the year (e.g. 2020) extracted from `date_chr`.
   b. Do the same for month `mth_num`.
   c. Do the same for day `day_num`.
   d. Do the same for hour `hr_num`, but extract from `created_at` column instead of `date_chr`.
   e. Do the same for minutes `min_num`.
   f. Do the same for seconds `sec_num`.

8. Using the new numeric columns you've created in the previous step, reconstruct the date and datetime columns. Namely, add the following columns to `cola_df2`:

   a. Use `make_date()` to create new column called `my_date` that contains the date.
   b. Use `make_datetime()` to create new column called `my_datetime` that contains the datetime.

   What is the `class` for your `my_date` and `my_datetime` columns?

9. Use `saveRDS()` to save your `cola_df2` in a file called `<last_name>_twitter_cola.RDS` in the `data_dir`. Add this data file and commit with the message `add twitter cola data`.

Part IV: I got issues

1. Navigate to the issues tab for the **rclass2** repository here.

   You can either:

   - Create a new issue posting a question you have about the class/problem set (assign instructors)

- Answer a question that another student posted
- Create a new issue posting about something new you learned or figured out from this class
  - If you choose this option, please mention the other members of your team and assign yourself

Paste the link to the issue you contributed to as a comment in `<last_name>_script.R`.

Please make sure to close the issue once your question has been resolved or within 1 week.

Part V: Wrapping up

1. How much time did you spend on this problem set? Write your response as a comment in `<last_name>_script.R`.
2. Finally, add your `<last_name>_script.R` file and make a commit. Push your branch `dev_[initials]` to the remote (hint: you will need to set upstream branch on initial push).

   Open a pull request for your branch to be merged into `master`. Assign one of your teammates to be responsible for merging in your pull request. Do not assign the same member who assigned you to merge their branch.