Lab-11

your name here

April 15, 2022

Preface

The goal of this assignment is to help you gain more familiarity with processing text data. As always, please come to office hours and reach out to your teaching staff if you have any questions.

Data

We will work with data on data scientist job postings in the U.S. scraped from popular job boards by JobSpikr.

```
# FYI the code chunk option "message = FALSE" omits read_csv()'s feedback from the PDF.
# instead, the output is printed in the console so you can still see the message
job_posts <- read_csv("data_scientist_united_states_job_postings.csv") %>%
  select(-cursor, -contains("contact"), -uniq_id, -html_job_description) %>%
  relocate(crawl_timestamp, url, .after = last_col())
job_posts # the PDF will still include regular output, such as this
```

```
## # A tibble: 10,000 x 17
##    job_title        category  company_name    city  state  country inferred_city
##    <chr>            <chr>     <chr>           <chr> <chr>  <chr>   <chr>
##  1 Enterprise Data~ Accounti~ Farmers Insura~ Wood~ CA     Usa     Woodland hil~
##  2 Data Scientist   <NA>      Luxoft USA Inc  Midd~ NJ     Usa     Middletown
##  3 Data Scientist   <NA>      Cincinnati Bel~ New ~ NY     Usa     New york
##  4 Data Scientist,~ Accounti~ BlackRock       New ~ NY 10~ Usa     New york
##  5 Senior Data Sci~ biotech   CyberCoders     Char~ NC     Usa     Charlotte
##  6 CIB - Fixed Inc~ Accounti~ JP Morgan Chase New ~ NY 10~ Usa     New york
##  7 Data Scientist,~ Accounti~ Spotify         New ~ NY 10~ Usa     New york
##  8 Sr. Data Scient~ <NA>      Xoriant Corpor~ Sant~ CA     Usa     Santa clara
##  9 Data Scientist,~ Accounti~ BlackRock       New ~ NY 10~ Usa     New york
## 10 Data Scientist   <NA>      Adroit Resourc~ San ~ CA     Usa     San francisco
## # ... with 9,990 more rows, and 10 more variables: inferred_state <chr>,
## #   inferred_country <chr>, post_date <date>, job_description <chr>,
## #   job_type <chr>, salary_offered <chr>, job_board <chr>, geo <chr>,
## #   crawl_timestamp <chr>, url <chr>
```

1. Let's start by looking at the job title. We see from the first few entries that most job titles include "data scientist." Tokenize `job_title` to 2-grams, and use a bar chart to show the top ten 2-grams that appear in `job_title`. What are they? Do they make sense to you?

Note: To avoid case sensitivity, `unnest_tokens` converts text to all lower cases by default.

2. From question 1 we see that some of the job titles include words indicating the job level, such as "senior", "sr", "lead", "principal", etc. Use `str_detect()` to classify jobs into three different levels: "junior", "senior", and "principal", based on the description of the job title. Then use a bar chart to show the corresponding number of postings for each level.

Tip: You may use `str_detect()` to detect whether `job_title` contains specific strings that indicate the job level. Junior jobs may contain "junior", "jr", "i", "1"; senior jobs may contains "senior", "sr", "ii", "2"; principal jobs may contain "principal", "lead", "iii", "3", "4". For job postings that don't contain any of these words, treat them as junior jobs. You should also either start by using `str_to_lower()` to convert job titles to lowercase, or use `regex(., ignore_case = TRUE)` within `str_detect()` to avoid issues with cases.

3. Let's look at the category of the job. Tokenize `category` into individual words and use a bar chart to show the top 10 words.

Tip: You might want to remove stop words.

4. Try using a word cloud to visualize the `category` text. Use your tokenized text from question 3 to make a word cloud plot using `wordcloud()` function. Does the plot seem easy to digest?

Tip: Use `wordcloud()` from the package `wordcloud` to make a word cloud plot. Remember that you can type ?wordcloud to get the help file to understand how to use a new function.

5. Where are these jobs located? Use a bar chart to show the number of job postings of the top 10 cities.

6. What software skills are most commonly required for these jobs? To find out, create logical variables to indicate whether each `job_description` contains skill requirements, such as excel, python, R, tableau, java, sql, matlab, etc. Then calculate the share of postings that require each of these skills, and show them in a bar plot. Do your results make sense? If not, can you improve them?

Tip: You may use `str_detect()` to detect whether `job_descriptions` contain each skill. To compute the share of postings the require each skill, `summarize_all(mean)` may come in handy. You should also either start by using `str_to_lower()` to convert job titles to lowercase, or use `regex(., ignore_case = TRUE)` within `str_detect()` to avoid issues with cases.

7. Do something else interesting with the data. For example, you may explore a bit more about the job description, and see whether you might be able to find useful information such as minimun working experiences, salary ranges, etc. Another option would be to explore education requirements for the jobs. Get creative and have fun!