TEXT MINING AND NATURAL LANGUAGE PROCESSING

There are four rules that apply to this project:

a) Follow instructions *precisely*. If I do not tell you what to write on a particular line, leave it blank.
b) All code must be *scalable by sample size* unless specifically noted otherwise.
c) Always use standard headings unless otherwise specified: **R Studio API Code, Libraries, Data Import and Cleaning, Analysis, Visualization.** Delete any headings without code under them.
d) All code must be *refactored* to avoid future code rot (i.e., minimize unnecessary code complexity).

Before you start, here's a hint: As you work through this project, you may find you need to refactor code that you wrote earlier and was working fine before in order to make later steps work. Be sure to use **good refactoring technique**, or you could make your problems worse. To test your code, be sure to clear all of your environment variables, restart R, and re-run each chunk one-by-one to ensure it is fully reproducible the way you intend.

Part 1 – Data Import and Cleaning

1. In a file called **week13.Rmd, which must be annotated to explain all steps taken,** create a data frame called *imported_tbl* containing at least 5000 posts from a Twitter search of your choice and like/favorite counts using *twitteR*. Remove retweets from your dataset, but check that you have at least 1000 remaining cases. If not, try a different search or download more tweets.
2. Before finalizing imported_tbl, use the following code to modify the raw test in *imported_tbl*. We will discuss this during debrief:
   imported_tbl$text <- imported_tbl$text %>% iconv("UTF-8", "ASCII", sub="")
3. Save *imported_tbl* as **output/tweets_original.csv**
4. Create a corpus full of appropriately **preprocessed lemmas** called *twitter_cp*. Use good professional judgment when preprocessing. You may want to revise this step after creating visualizations. Remove all hashtags.
5. Convert your corpus into a unigram and bigram DTM called *twitter_dtm*. Eliminate sparse terms. Be careful not to remove so many terms that you can't complete the analyses in this project.
6. Created a new tibble called *dropped_tbl* using *imported_tbl* as input, but delete cases for tweets where no tokens were retained.

Part 2 – Visualization

7. Create a new tibble called *twitter_tbl* that contains all tokens from *twitter_dtm*. Remove your Twitter search term from this tibble if it's still there. Also remove all tokens that are or begin with web URLs.
8. Generate a word cloud of up to the top 50 most frequent words in *twitter_dtm*.
9. Generate a horizontal bar chart of the top 20 bigram lemmas in *twitter_dtm,* ordered by most common on the top and least common on the bottom.

Part 3 – Analysis: Topic Modeling

10. Create a subheading called **Topic Modeling**.
11. Use topic modeling to generate topic categorizations for each case. Create whatever tabular or graphical summaries you need, but at least two, to interpret the topics you've identified. Add a column with topic identifiers (e.g., 1, 2, 3) to *twitter_tbl* called *topic.* Briefly, try to interpret what your topics mean. But don't spend too much time on interpretation.
12. Create a tabular summary of most likely topic per tweet in your dataset.

Part 4 – Analysis: Machine Learning

13. Create a subheading called **Machine Learning**
14. Add tweet popularity from *dropped_tbl* into *twitter_tbl*.
15. Use 10-fold cross-validated **support vector regression** on *twitter_tbl* to predict tweet popularity from the tokens alone. Ensure you parallelize to the capabilities of your machine.
16. Use 10-fold cross-validated **support vector regression** on *twitter_tbl* to predict tweet popularity from the tokens plus topic assignments. Ensure you parallelize to the capabilities of your machine.
17. Conduct appropriate textual and graphical model comparisons between these two models.

Part 5 – Analysis: Final Interpretation

18. Create a subheading called **Final Interpretation**
19. In markdown under this subheading, respond to the following question:
    Given all the NLP you've done and analyses you've conducted, would you conclude that topics (and emotion) are important in predicting the popularity of tweets? Why or why not? Cite specific evidence from your earlier work on this project and also consider the research design and the impact of how topic modeling works carefully.
20. Export your work to a PDF. Before you do this, you may want to comment out your Twitter search and replace it with an import of **tweets_original.csv**, but this is up to you.