Homework 3

Due by 9PM, September 27th

Please submit R code, explanations, and / or plots for any section marked with the **[To submit]** heading by emailing jakeporway+itp@gmail.com.

READINGS

Background Reading for This Homework

- Go back and read through the notes in Lecture 3
- Subsetting in R: Sections 4.3.1 and 4.3.2 in http://cran.r-project.org/doc/contrib/Lam-IntroductionToR_LHL.pdf
- Text manipulation in R: Section 4.5 in http://cran.r-project.org/doc/contrib/Lam-IntroductionToR_LHL.pdf
- If you want to know a bit more about Python: Lists and dictionaries in Python: http://www.sthurlow.com/python/lesson06/
- Using the Twitter Streaming API: http://mike.teczno.com/notes/streaming-data-from-twitter.html. This guy takes a slightly different approach than we did, but it's pretty close.

Reading for Next Class

- I asked Jer to talk about mapping data (geospatial data) next class so I *hope* he'll be able to. If he doesn't, we'll talk about it the week I get back so I thought it would be good to use some maps of the Stop and Frisk data to illustrate some ideas about mapping data:
    - http://www.wnyc.org/articles/wnyc-news/2012/jul/16/wnyc-map-police-find-guns-where-they-stop-and-frisk-less/
    - http://www.dashboardinsight.com/news/news-articles/analysis-of-wnyc-stop-frisk-guns-graphic.aspx
    - http://spatialityblog.com/2012/07/27/nyc-stop-frisk-cartographic-observations/

In class last time we saw how to convert tweets into something we could analyze in R. We didn't get much of a chance to explore our data, however, so let's get into it here in the homework. This homework will walk you through the real questions one asks when faced with a new dataset. It'll have some subsetting examples to build on what we learned in the last two weeks as well as introduce you to some text processing.

Part One – Warm Up

Let's start with the dataset we had in class – a collection of tweets containing the word "libya". Download http://jakeporway.com/teaching/data/libya_tweets.json and run the code at http://jakeporway.com/teaching/code/tweets_to_csv.py on it to get a CSV file you can load into R (there was an error in class so I'd redownload this file). Change lines 9 and 123 to point to the appropriate files you want to read and write.

Each row in this dataset is a tweet with information about the user who tweeted, which is a pretty rich dataset if you think about it. When I first see a new dataset, the first thing I want to dig into (after the basic stuff like how many rows there are and what types of columns are in here) are properties of the variables in the dataset. This usually takes the form of looking at the distributions of variables or finding the top N "things". Let's look at some top N lists and distributions to familiarize ourself with what's in the data and see if we see anything interesting.

[To submit]

- I'm always curious about people who have high follower counts, because they might be interesting people tweeting about this topic. How many unique users have more than 100000 followers? What are their screen names?
- It'd be interesting to see what part of the world users are tweeting from. What are the top 3 locations people are from (not counting blanks)?
- Retweets can often indicate what's important, or at least influential. What is the text of the tweet that was retweeted the most times and who tweeted it?

Cool! We could dig into these details a lot more thoroughly, but this at least gives us some sense of who the most popular tweeters are and what they're tweeting. But what about the rest of the people in this dataset? Let's now look at some of the dynamics of our tweeters as a whole. In particular, let's look at the number of other users that people tend to follow.

[To submit] Plot the distribution of the number of people the users are following (don't worry about the fact that some people will be counted multiple times – just pretend each row is a different user). NOTE: We don't want to use table() here because we don't want to know how many people had *exactly* 4014 followers, for example, we just want to see the overall distribution, so use hist() to plot the distribution of "following". What do you see?

[To submit] Huh, it looks like there are a few people with LOTS of followers who are skewing our distribution, making it hard to look at the bulk of the data. It's probably those people from the first question. Let's reduce our set to just people with fewer than 5000 followers and look at the histogram again. What do you see now? Have you tried using different breaks? Does anything surprise you?

Part Two: Getting To Know You

OK, we've identified our top tweeters, we've seen something about the distribution of how many other users people follow, but I'd love to get a broader sense of what the people tweeting about this topic are *like*. Who are they? Where are they from? What are their hopes and dreams? Notice that each tweet has a description, which is a self-provided line of text that describes the person tweeting. Ah ha! Why don't we look at the words that commonly show up in the descriptions of our users? That way we can get a rough sense of what people are like. To do this we want to count up all the words used in people's descriptions and find the top N words used.

[To submit] Write code to find the 5 most popular words used in the descriptions of our users (again, just treat each row as if it's a unique user, even though that means we'll be counting users who tweeted more than once multiple times). Unless you're an R wizard, you're going to need to know some extra stuff to do this:

1. You should use strsplit(), which you just read about.
2. You may notice strsplit() gives you back a funky type of object in R that we haven't learned about yet called a "list". Lists are like hash maps in R. What happened here is that every description has been split into a vector of words, and each vector lives in its own bucket, one bucket per row. You don't need to worry about lists yet (unless you're curious and want to read up on 'em) – what you need to know is that you can turn the list into one big long vector using the unlist() function on your results.
3. It's going to be annoying to count "Libya" and "libya" as different words, so let's use the tolower() function to get all the words into lowercase.

Now you should be able to find the top 5 words using techniques you've seen before.

If you did this step right, you're going to find that the words we got back are, well, pretty uninteresting. That's because almost everyone uses the word "the". Let's clean out super common words, often referred to as stopwords, so we can just focus on the interesting words people are using. That means we need to remove any common words from our big ol' word vector we created in step 2 above. Hmm, how can we remove specific elements of a vector? Let's try this:

1. Create a vector of words that you want to omit from your final results (e.g. "the", "a", "and"). To do this, let's load up the list of stopwords listed at this URL: http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop You can use read.csv() to load all the stopwords from the link above as a data frame with one column (make sure you use the as.is=TRUE argument!)
2. Ugh, there will still be punctuation and blanks in our words as well. Add the following values to your stopwords vector: "" (a blank), "&", "-", "|"
3. OK, next we have to remove these stopwords from our big vector of all the words in the descriptions. There's a really great logical operator called "%in%" that does set membership. Here's an example:

   ```
   X <- c(1, 2, 3, 4)
   Y <- c(3, 4, 5, 6)
   # 3 and 4 are the only values in X that are in Y
   X[X %in% Y]
   # [1] 3 4
   ```

   One little annoyance with %in% is that to negate it you have to negate the *whole* expression, you can't just do !%in% as you might think:

   ```
   X[X !%in% Y] # ERROR!
   X[ !(X %in% Y)]  # Ah, better
   # [1] 1 2
   ```

[To submit] Using your skills with %in% and a vector of stopwords, remove the stopwords from the descriptions and recompute the top 5 words our Twitter users use to describe themselves. What do you think of the results? Do you have a sense of what types of users are most common in our dataset?

Part Three:

We saw we could use Twitter's streaming API to download tweets matching certain search terms and then convert them to CSV using our Python script. Maybe Libya's not your thing though, so teach me about a topic you're interested in. Use the Twitter streaming API to download tweets matching some terms you're interested in, then convert the results to CSV (don't forget to change lines 9 and 123 of the Python script tweets_to_csv.py, if you're using it).

[To submit] Tell me what search terms you used and then teach me something about your dataset, preferably something you find interesting. You can just repeat one of the exercises we did above on your new data but, if you have the time, I'd encourage you to dig into the data and try to find something that surprises you. It's only when you really start trying to answer questions for yourself and dig in Sherlock Holmes style that you really start to get this stuff :)

BONUS:
Now that you've been using R for a bit, what's one thing you wish you knew how to do? This thing could be something you've wanted to learn from the beginning of this course or something you've tried to do with the data in R and realized you couldn't do yet. I've got plans for a bunch of cool stuff to do together, but I'd love to hear what you guys want to do.