Teaching Machines to Cry, a.k.a
Sentiment Analysis, a.k.a
Mapping Feelings, a.k.a
Understanding the Human Network, a.k.a
Lab 6. "The Final Lab"

Lab in Brief:
This lab will be bringing together skills you've learned across the entire semester to conduct one "real world" analysis using real twitter data. You'll not only be downloading real tweets, but you'll be mapping them, building social networks out of them, creating word clouds, and conducting one new step: sentiment analysis. To prepare you for this final step, you'll be conducting a brief new analysis using the feedback you gave in class during lecture last Friday. Finally, you'll be putting your twitter analysis online using Rmarkdown.

Data and Materials:
We'll be using live tweets, which will require you to sign up for a special twitter "Developer" account at https://dev.twitter.com/

Objectives:
In this lab, you will be applying all of the skills you've learned in previous labs to:
(A) Download real, live data from Twitter
(B) Condense a very large dataset into meaningful, interactive visualizations.
(C) Analyze the data by sentiment, network effects, and geography.
(D) Create an interactive website using this data and analysis.

Deliverables:
At 12 Noon on December 10th, you will turn in the deliverables outlined on the final page of this lab.

Grading:
This lab will be graded by rubric, following the same grading strategy we've used for all of the labs to date. The rubric can be viewed on blackboard.

Part 1: An Introduction to Sentiment Analysis

(Note: You do not have to include this portion on your website, so echo=FALSE for the below section).

In this part of the lab, you'll be using the feedback you gave on this course as a baseline for something called sentiment analysis – the technical process of using a computer to understand if a collection of words are generally positive, negative, or neutral.

1-1: Let's get our R environment setup. Create a new .Rmd by going to file → new file → R Markdown. Just like in the last lab, you'll need to add ```{r, echo=TRUE} (or FALSE) before each code chunk, and "```" after each one.

1-2: Let's load in the old text mining library ("tm"), and install a new plugin for this library that allows for sentiment analysis. As this plugin is located on a different website than the tm library, you'll need to install it in a slightly different way than most packages:

```
library(devtools)
install_github("mannau/tm.plugin.sentiment")
library(tm.plugin.sentiment)
```

1-3: Let's give our new library a test drive. First, let's create a dummy corpus (you'll remember the text corpus creation from the social networks lab):

```
example = c("I hate, HATE apples." , "I love oranges." , "I have a love
hate relationship with plumbs")
exampleCorpus = Corpus(VectorSource(example))
```

1-4: Let's run our first sentiment analysis on these three statements. To do that, all you need to do is type in:

```
meta(score(exampleCorpus))
```

1-5: You just ran your first sentiment analysis! You'll get an output that has a number of different columns, and likely looks just like this:

	polarity	subjectivity	pos_refs_per_ref	neg_refs_per_ref	senti_diffs_per_ref
1	-1	0.6666667	0.0000000	0.6666667	-0.6666667
2	1	0.5000000	0.5000000	0.0000000	0.5000000
3	0	0.3333333	0.1666667	0.1666667	0.0000000

Dan recommends you read this part, even though it's wordy!: Let's briefly interpret each column – focusing on the first line. In this case, the first line is associated with the first example you typed in ("I hate, HATE apples.").

The first column - **polarity** is a -1 (or, a negative polarity). This means the computer thinks that the sentence is overall a very negative sentence. Polarity will always range between -1 and +1.

The second column – **subjectivity** – is the fraction of negative words + the fraction of positive words divided by all words (we ignore the word "I", because it's a stop word). A higher subjectivity means that the computer had to interpret more words as being positive or negative, relative to the length of the sentence. In the first row, 2/3 of the words (hate, HATE) were interpreted – only "apples" was not.

The third column – **pos_refs_per_ref** – is the percentage of the words the computer associated with being positive. In the first row, no words were positive.

The fourth column – **neg_refs_per_ref –** is the percentage of the words the computer associated with being negative. In the first row, 2/3 words were found to be negative.

The fifth column – **senti_diffs_per_ref –** is the sum of the positive and negative reference percentages. In the first row, this is -0.667. To calculate polarity, this fifth column is divided by the subjectivity score (i.e., the overall sentiment divided by the percent of words assessed for sentiment).

1-6: Go back to 1-3 and change the example data so you only have two rows. Make one of these rows have a subjectivity of +0.5, and one of these a subjectivity of -0.5 . Copy these examples to answer question 1-6.

1-7: Alright, let's apply the same sentiment analysis to the feedback you gave in class – we'll take a look at the general sentiment across every respondent, average sentiments, and sentiment by both sex and feedback on the overall difficulty of the course. To get started, load the CSV that was provided with this lab (studentfeedback.csv) into R. I named my object "studentFeedback". After you load the CSV, I recommend typing in View(studentFeedback) to take a quick look at the data.

1-8: Just like we did in the twitter lab, we're going to take the feedback and turn it into a text corpus. The corpus is necessary for sentiment analysis to work. In this case, we only want the "Feedback" column of data, and we're going to take all of the feedback:

```
allFeedbackCorpus = Corpus(VectorSource(studentFeedback$Feedback))
```

1-9: Let's run a sentiment analysis on every row – use the same code you used in 1-4, but with the feedback corpus instead of the example corpus.

1-10: The overall analysis is interesting, but what we're really after is summaries. Thankfully, R makes this very easy – just add the term "summary" before your sentiment analysis (your code will look something like this):

```
summary(meta(score(allFeedbackCorpus)))
```

Record the mean polarity and subjectivity, and write 1-2 sentences interpreting these results.

1-11: Let's look at the results by subgroup, now. First, let's assess only the students that are female, and contrast those to the students that are male. This requires us to build two different corpus – below is example code for female students; you'll have to figure out male students.

```
femaleFeedback = subset(studentFeedback, Sex == "F")
femaleFeedbackCorpus = Corpus(VectorSource(femaleFeedback$Feedback))
summary(meta(score(femaleFeedbackCorpus)))
```

Contrast your results – what group had a higher polarity? Lower polarity? Do you see any other important differences?

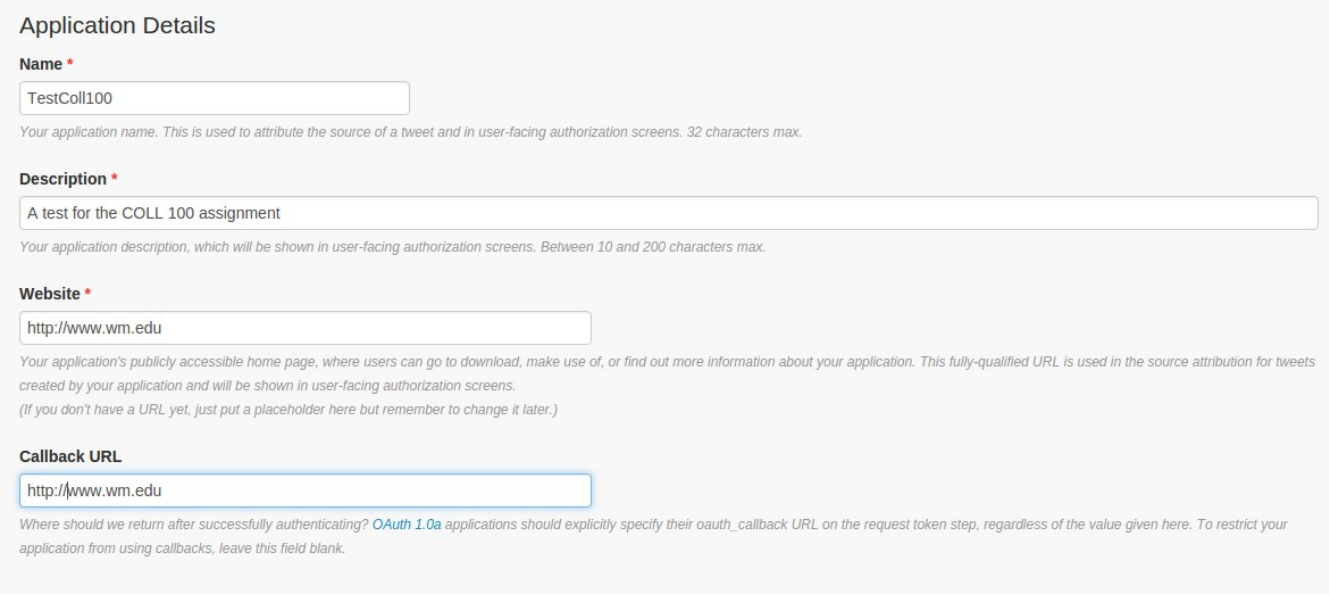
1-12: Now, do the same thing with the three groups of students that thought the class is too easy, too hard, and the correct difficulty. Contrast these results in the same way you contrasted for 1-11 – how do they differ? Use any of the tools you've learned in this class to make a visualization contrasting the polarities of these three groups; consider further splitting the groups by sex.

Part 2: Getting Real Data from Twitter
In this part of the lab, you're going to learn how to download real, live data from Twitter and run similar styles of sentiment analysis. For now, continue to keep echo=FALSE.

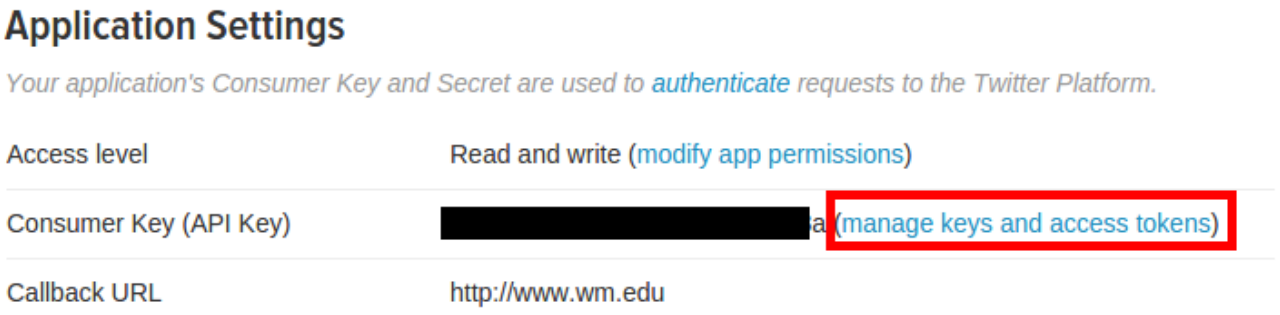
2-13: Go to https://dev.twitter.com/ and sign up for an account – you can use your existing Twitter user ID, or create a new one. Note you will need to confirm your account using a cellphone to access the API. You can add it here if you already have an account: https://twitter.com/settings/devices .

2-14: Once you're logged in, go to https://apps.twitter.com/ .

2-15: Click "create new application". You can name your app and describe it however you would like; put www.wm.edu as the website and callback URL, then scroll down and click Create your App.



2-16: On the next page, click on the "manage keys and access tokens" link:

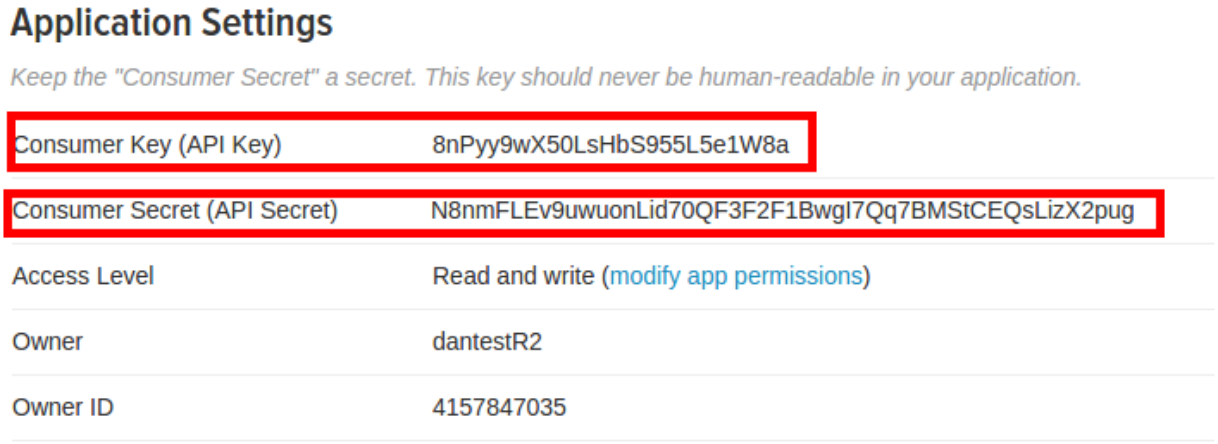


2-17: Now, we're going to use the data on that page to add new code into R – first, type the below into R:

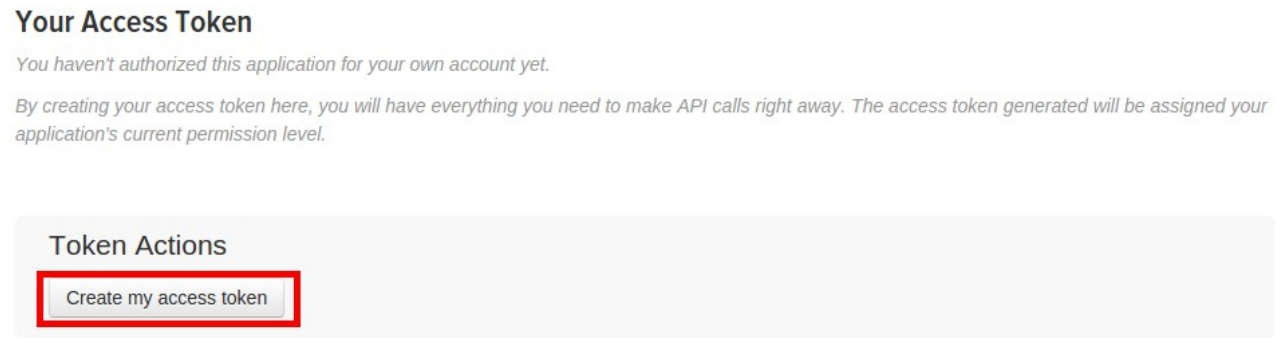
```
library(twitteR)
api_key <- "YOUR API KEY"
api_secret <- "YOUR API SECRET"

access_token <- "YOUR ACCESS TOKEN"
access_token_secret <- "YOUR ACCESS TOKEN SECRET"
```

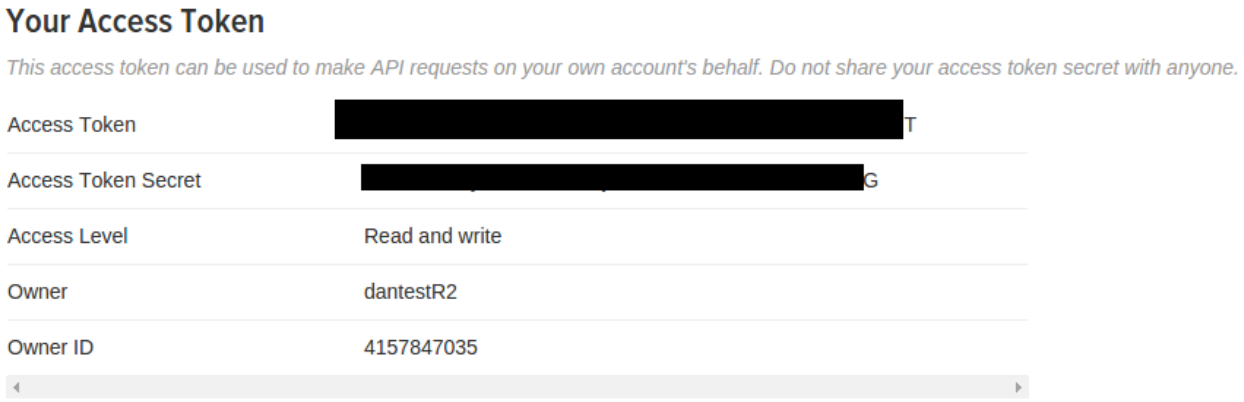
Now, replace where it says “YOUR API KEY” and “YOUR API SECRET” with the long strings you see on that website:



Now, scroll down to the bottom and click on “Create my Access Token”:



Now, replace your “YOUR ACCESS TOKEN” and “YOUR ACCESS TOKEN SECRET” with the two codes you find when you scroll down:



Finally, add this to your code and run everything – if prompted, push “1” on your keyboard:

```
library(base64enc)
setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
```

2-18: Let's make sure everything worked by doing a test search of twitter – the first line actually queries Twitter, then the second line translates what you get into a normal R data frame:

```
testTweetSearch = searchTwitter("William and Mary", n=100, lang="en")
tweetDF <- do.call("rbind", lapply(testTweetSearch, as.data.frame))
```

Type in “View(tweetDF)” and make sure you actually have a set of tweets!

2-19: Now, we're ready to rock and roll – first, let's run a sentiment analysis on the text of our tweets we just extracted:

```
tweetCorpus = Corpus(VectorSource(tweetDF$text))
summary(meta(score(tweetCorpus)))
```

Pro tip: If tweets included emojis, the sentiment analysis won't work because emojis will come out looking like this:  See if you can figure out how to fix this problem by reading the help on this page: http://stackoverflow.com/questions/15748190/emoticons-in-twitter-sentiment-analysis-in-r

What is the mean polarity you get for these tweets? How would you interpret this?

2-20: Let's look at individual tweets polarities now, and see if we can pick out any patterns. Add the polarity of every tweet to your dataframe:

```
tweetDF$polarity <- meta(score(tweetCorpus))[,1]
```

If you type in “View(tweetDF) now, you should see a new column called “polarity” at the end.

2-21: Now, let's create three dataframes – one for large positive polarities, one for large negative polarities, and one for neutral polarities. We're going to integrate all there of these into one big word-cloud in the next step:

```
lowPolarityTweets <- subset(tweetDF, polarity < -0.2)
neutralPolarityTweets <- subset(tweetDF, polarity < 0.2 & polarity > -0.2)
highPolarityTweets <- subset(tweetDF,  polarity > 0.2)
```

2-22: Integrate them all into one list for the word cloud:

```
allPolarityTweets = rep("", 3)

allPolarityTweets[1] = highPolarityTweets
allPolarityTweets[2] = lowPolarityTweets
allPolarityTweets[3] = neutralPolarityTweets
```

2-23: Build a word cloud, very similarly to how we've done it before:

```
library(tm)
library(wordcloud)

corpus = Corpus(VectorSource(allPolarityTweets))
tdm = TermDocumentMatrix(corpus)
tdm = as.matrix(tdm)
colnames(tdm) = c("pos", "neg", "neutral")

comparison.cloud(tdm, colors = brewer.pal(3, "Dark2"), scale = c(3,.5),
random.order = FALSE)
```

Copy this word cloud to answer question 2-23. The colors go along with the polarity of the tweets the words themselves were found in – orange words were commonly found in negative tweets, green positive tweets, and purple neutral tweets.

Part 3: Mapping the Twitters
In this part of the lab, you're going to map out the intensity of different tweets. Keep your echo=FALSE – the website is coming soon!

3-24: First, let's pull down tweets that are geotagged – that is, someone likely used their phone and the latitude / longitude they tweeted from was recorded. We'll take Washington, DC as our starting example, searching for all tweets within a 100 mile radius around the city center that mention “Obama”:

```
GeoTweetSearch = searchTwitter("Obama", n=1000, geocode='38.9047,-
77.0164,100mi')
```

Just like before, transform this search into a dataframe so we can use it with other tools:

```
geoDF <- do.call("rbind", lapply(GeoTweetSearch, as.data.frame))
```

3-25: We're going to do a throw-back all the way to the very first lab in this class, and use the data we just retrieved to make a map of all tweets mentioning Obama.

```
#load the library, note you may need to install this:
library(ggmap)

#remove cases where we don't have complete latitude and longitude informa-
tion,
#then convert the data into numeric values so our package can read it.
geoDF_noNA <- geoDF[complete.cases(geoDF["latitude"]),]
geoDF_noNA$latitude <- as.numeric(geoDF_noNA$latitude)
geoDF_noNA$longitude <- as.numeric(geoDF_noNA$longitude)

#Finally, create a map
qmplot(longitude, latitude, data = geoDF_noNA, colour = I('red'),
source="google")
```

Turn in the map the qmplot command creates to answer **3-25**.

Part 4: The real social network
In this part of the lab, you'll be building a social network from a subset of tweets you choose. Keep echo=FALSE – last time!

4-26: Let's do a new twitter search – this time, we're going to look at a controversial topic. Try:

```
NetworkTweetSearch = searchTwitter("Hillary Clinton", n=1000, lang="en")
```

Note this may take a little while – we're asking for 1000 tweets!

4-27: Just like before, let's convert this over to a dataframe:

```
NT_DF <- do.call("rbind", lapply(NetworkTweetSearch, as.data.frame))
```

4-28: Now, we're going to do the same steps we did on the twitter lab to build what the network of tweets look like. First, we only take the columns that tell us who the tweet was from and to. Then, we build a graph out of it and visualize that graph:

```
NT_DF_IDS <- NT_DF[c("replyToSN", "screenName")]
#Remove anyone who didn't tweet from or to someone:
NT_DF_IDS <- NT_DF_IDS[complete.cases(NT_DF_IDS),]

#Build our graph:
library(igraph)
Tweets_adj <- get.adjacency(graph.edgelist(as.matrix(NT_DF_IDS)))
MyGraph <- graph_from_adjacency_matrix(Tweets_adj)

#Plot the graph:
plot(MyGraph,                          #the graph to be plotted
     layout=layout.fruchterman.reingold,
     main='Twitter Network',           #specifies the title
     vertex.color = 'green',           #The color of the nodes
     vertex.label.dist=0.25,           #puts the name labels slightly off the dots
     vertex.frame.color='blue',        #the color of the border of the dots
     vertex.label.color='black',       #the color of the name labels
     vertex.label.font=0.1,            #the font of the name labels
     vertex.label.cex=0.5,             #specifies the size of the font of the labels
     vertex.size = 0.5,
     vertex.shape = "circle",          #Try out some circles
     edge.arrow.size = 0,              #More reasonable size arrows
     edge.lty = 6,                     #Different edge types; 1-6 are options.
     edge.curved=0.5                   #Make it look neat. Different values make it more or less curvy.
)
```

Turn this graph in to answer question 4-28.

Part 5: “The Final”

Using all of the skills you've learned in this class, coupled with the steps above, you will produce an interactive website that provides evidence for or against one of the following statement (you get to choose!). It is expected this website will have relevant graphs, maps, networks, and other figures as are necessary, as well as accompanying short descriptions of those figures.

Make sure to indicate which of these topics you chose. Select one of the following questions, or talk to me by no later than Wednesday of this week to get approval for a topic of your design:

- At this point, which republican or democratic candidate do you think will win the election? Who has broader geographic support and denser networks? How do the current sentiments of different candidates compare?
- Amongst a group of competing companies (no less than 4; for example, google, apple, twitter and yahoo), which do you believe will have a higher stock valuation in one month? What are the current trends of the stocks? Which have better sentiments on twitter?
- Across a group of academic teams (i.e., football; baseball – choose no less than 4), which has the strongest alumni network? This can be measured based on the density of social networks, sentiment, and geographic extent / density of tweets across the nation.
- What city is the most popular? Choose at least four cities, and run a sentiment analysis on tweets mentioning those cities. Identify if the support for those cities is country-wide or specific to the individual cities themselves, and consider the social network that supports those sentiments. Can you identify, for example, amongst what type of person a city is popular, or what characteristics of the city make it popular?

Deliverables:
Create a word document with the answers to the following questions:

1-6: Go back to 1-3 and change the example data so you only have two rows. Make one of these rows have a subjectivity of +0.5, and one of these a subjectivity of -0.5 . Copy these examples to answer question 1-6.

1-10: Record the mean polarity and subjectivity, and write 1-2 sentences interpreting these results.

1-11: Contrast your results – what group had a higher polarity? Lower polarity? Do you see any other significant differences?

1-12: Contrast these results in the same way you contrasted for 1-11 – how do they differ? Use any of the tools you've learned in this class to make a visualization contrasting the polarities of these three groups; consider further splitting the groups by sex .

2-19: What is the mean polarity you get for your test set of tweets? How would you interpret this?

2-23: Copy the word cloud to answer question 2-23.

3-25: Copy the map you create to answer this question.

4-28: Copy your twitter graph to answer this question.

5: Submit your website as an answer to part 5 of this assignment.