Homework 10 – Due Monday November 21, 2015

Cheryl Calhoun

11/22/2015

```
## Setting up the work environment.
##setwd("C:/Users/07001412/OneDrive/Education/UF/2015/Fall/EDF6938/Week 11")

## Setting up for Twitter API access.
## Install `twitterR` and supporting packages if not already installed.
## download.file(url="http://cran.r-project.org/bin/windows/Rtools/")"
## download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pe
m") ##-- this was suggested if you are running a Windows machine.
## install.packages("twitteR", "RCurl", "RJSONIO", "stringr", "base64enc")

## Loading twitterR and other supporting libraries.

library("dplyr")
library(twitteR)
library(RCurl)
library(RJSONIO)
library(stringr)
library("base64enc")

## Loading twitter API and Access keys.
source ("twitter-access-keys.R")
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

```
## [1] "Using direct authentication"
## Setting up for sentiment analysis. Install packages if necessary.
## First, install an alternate repository.
## install.packages("Rstem", repos = "http://www.omegahat.org/R", type="source")
library(sentiment)

## Second, we can use packages hosted on GitHub.  Install packages if necessary.
## install.packages("devtools")
## install_github ("timjurka/sentiment/sentiment")
library(devtools)

## Loading variables.

## Setting search patterns for usertags and hashtags.
usertag <- "@[A-z_0-9]+"
hashtag <- "#[A-z0-9]+"
dates <- c("Oct 13", "Sep 01", "Nov 09", "Nov 13", "Dec 12")

## Load the subjectivity and emotions files from (https://github.com/timjurka/sentiment/tree/master/sentiment)
subjectivity <- read.csv ("http://www.acthomas.ca/FSSS/data/subjectivity.csv", header=FALSE)
emotions <- read.csv ("http://www.acthomas.ca/FSSS/data/emotions.csv", header=FALSE)
```

Gathering data on Football related tweets at the University of Florida

Context: Every member of the class was assigned a school in the Southeastern Conference (SEC) as specified in the spreadsheet located at: [https://docs.google.com/spreadsheets/d/1IKRXc0hN1C9e5S845LmgY-rlLdCQ1xOQrI3thxKIgVA/edit?usp=sharing (https://docs.google.com/spreadsheets/d/1IKRXc0hN1C9e5S845LmgY-rlLdCQ1xOQrI3thxKIgVA/edit?usp=sharing)]

For this exercise, data will be captured from twitter each week, saved to a file, and reload from that file as needed. This will ensure both reproducibility and ease of use on Twitter’s servers.

The original plan was to use geolocation to find our tweets. To do this, we need to find the latitude and longitude of our school on Google Maps. (This is a manual process of looking up the coordinates.) The latitude and longitude of Ben Hill Griffin stadium at the University of Florida is: 29.649898, -82.348429. Now, we can use these coordinates to extract a sample of tweets from near that location. For this exercise, we will use a 5 mile radius.

Unfortunately, geolocation does not seem to work, so we will use the most common hashtag associated with Florida Sports and Florida Football. After some initial data exploration, the #GoGators hashtag is selected.

11/14/15 Update: After review of tweets obtained using the #GoGators hashtag, it is apparent that this data set includes tweets from all Gator sports as well as many other miscellaneous tweets. Further research through twitter and watching the game confirms there is a game specific hashtag each week. The game specific hashtag for Game 10 is #UFvsSC. I will use this hashtag to gather the tweets for the S. Carolina game. I will also gather tweets using the #GoGators hashtag as a comparison.

There are four remaining games this season. They are:

Game #	**Date**	**Opponent**	**Location**
Game 9	Nov 7	Vanderbilt	Gainesville
Game 10	Nov 14	S. Carolina	Columbia
Game 11	Nov 21	Fl. Atlantic	Gainesville
Game 12	Nov 28	Florida State	Gainesville

```
## Collect weekly tweets beginning Thursday before game and ending on Sunday af
ter the game. Store tweets in a .csv file.  This code will be executed on a wee
kly basis until we have gathered data for the remaining 4 games of the season.

##The original geolocation code.
## Florida <- searchTwitter('', geocode='29.649898,-82.348429,5mi', since="2015
-11-05", until="2015-11-09", n=10000)

##Collect data for Game 9 using the updated #GoGators hashtag code.
##Florida9 <- searchTwitter("#GoGators", since="2015-11-05", until="2015-11-0
7", n=10000, retryOnRateLimit=120)
##Florida9.df <- rbind_all (lapply (Florida9, function(rr) rr$toDataFrame()))
## Remove tweets from FloridaGators, GatorsFB and UF
## Florida9.df <- filter(Florida9.df, screenName!="FloridaGators")
## Florida9.df <- filter(Florida9.df, screenName!="GatorsFB")
## Florida9.df <- filter(Florida9.df, screenName!="UF")
## Florida9.df <- filter(Florida9.df, text!="RT @GatorsFB: #Gators are back hom
e in The Swamp this week taking on FAU. #FAUvsUF https://t.co/w9C1YyK4AG")
##write.csv(Florida9.df, file = "Game9.csv")

## Collect data for game 10, using the games specific hashtage #UFvsSC.
## Florida10 <- searchTwitter("#UFvsSC", since="2015-11-12 00:00:00", until="20
15-11-16 00:00:00", n=10000, retryOnRateLimit=120)
## Florida10.df <- rbind_all (lapply (Florida10, function(rr) rr$toDataFrame
()))
## Remove tweets from FloridaGators, GatorsFB and UF
## Florida10.df <- filter(Florida10.df, screenName!="FloridaGators")
## Florida10.df <- filter(Florida10.df, screenName!="GatorsFB")
## Florida10.df <- filter(Florida10.df, screenName!="UF")
## Florida10.df <- filter(Florida10.df, text!="RT @GatorsFB: #Gators are back h
ome in The Swamp this week taking on FAU. #FAUvsUF https://t.co/w9C1YyK4AG")
## write.csv(Florida10.df, file = "Game10.csv")

## Collect data for game 11 using both the game specific hashtag #FAUvsUF and t
he username @GatorsFB
## Florida11 <- searchTwitter("#FAUvsUF OR @GatorsFB", since="2015-11-13", unti
l="2015-11-20", n=10000)
## Florida11.df <- rbind_all (lapply (Florida11, function(rr) rr$toDataFrame
()))
## Remove tweets from FloridaGators, GatorsFB and UF
## Florida11.df <- filter(Florida11.df, screenName!="FloridaGators")
## Florida11.df <- filter(Florida11.df, screenName!="GatorsFB")
## Florida11.df <- filter(Florida11.df, screenName!="UF")
## Florida11.df <- filter(Florida11.df, text!="RT @GatorsFB: #Gators are back h
ome in The Swamp this week taking on FAU. #FAUvsUF https://t.co/w9C1YyK4AG")
## write.csv(Florida11.df, file = "Game11.csv")

## Collect data for game 12 using both the game specific hashtag #FAUvsUF and t
```

```
he username @GatorsFB
##Florida12 <- searchTwitter("#UFvsFSU or @GatorsFB", since="2015-11-26", until
="2015-11-30", n=10000)
##Florida12.df <- rbind_all (lapply (Florida12, function(rr) rr$toDataFrame()))
## Florida12.df <- filter(Florida12.df, screenName!="FloridaGators")
## Florida12.df <- filter(Florida12.df, screenName!="GatorsFB")
## Florida12.df <- filter(Florida12.df, screenName!="UF")
## Florida12.df <- filter(Florida12.df, text!="RT @GatorsFB: #Gators are back h
ome in The Swamp this week taking on FAU. #FAUvsUF https://t.co/w9C1YyK4AG")
##write.csv(Florida12.df, file = "Game12.csv")

# Read previously stored data from data file.
Game9 <- read.csv("Game9.csv")
Game10 <- read.csv("Game10.csv")
Game11 <- read.csv("Game11.csv")
## Game12 <- read.csv("FloridaSt.csv")

## Combine all game tweets into one big file.
GameTweets <- bind_rows(Game9, Game10, Game11)
```

Homework #8 specific analysis begins here.

Evaluating HashTags for Game 9: Vanderbilt

Determining the number of hashtags used in each tweet and overall in the sample.

```
## Find hashtags for Game 9. Add a column for hashtags and a column for number
of hashtags.
Game9 <- mutate (Game9, hashtags=regmatches(Game9$text, gregexpr(hashtag, Game9
$text)))
Game9 <- mutate (Game9, HQty=as.integer(lapply(Game9$hashtags, function(x) leng
th(x))))
```

Determining the number of tweets at each hashtag level.

```
## Create a Hashtags per Tweet table.
HashtagsperTweet = table(Game9$HQty)
HashtagsperTweetTable = as.data.frame(HashtagsperTweet)
names(HashtagsperTweetTable)[1] = 'Number of Hashtags'
names(HashtagsperTweetTable)[2] = 'Number of Tweets'
HashtagsperTweetTable
```

```
##    Number of Hashtags Number of Tweets
## 1                    0               21
## 2                    1             4760
## 3                    2             3478
## 4                    3             1102
## 5                    4              365
## 6                    5              153
## 7                    6               58
## 8                    7               34
## 9                    8               19
## 10                   9                7
## 11                  10                1
## 12                  12                1
## 13                  14                1
```

Determining the number of hashtags in the data set.

```
## Determine total number of hashtags.
all.hashtags <- unlist(regmatches(Game9$text, gregexpr (hashtag, Game9$text)))

## Determine total number of unique hashtags.
unique.hashtags <- unique(all.hashtags <- unlist(regmatches(Game9$text, gregexp
r (hashtag, Game9$text))))
## all.hashtags
```

Using the `length()` function, there are a total of 18084 hashtags in the data set.

Using the `length()` function, there are 1109 unique hashtags in the data set.

Finding the most frequently used hashtags.

```
## Count the hashtag usage.
hashtags.df <- data.frame(cbind(all.hashtags))
hashcount <- count(hashtags.df, all.hashtags)
hashcount <- hashcount[order(-hashcount$n),]
hashcount
```

```
## Source: local data frame [1,109 x 2]
##
##         all.hashtags    n
## 1          #GoGators 9068
## 2         #VANDYvsUF 1477
## 3               #SEC  747
## 4          #gogators  645
## 5  #GatorsHeismanDay  486
## 6        #GatorNation  363
## 7            #Gators  330
## 8         #BeatVandy  273
## 9  #SECEastChampions  266
## 10           #TurnUp  240
## ..               ...  ...
```

Evaluating UserTags for Game 9: Vanderbilt

Determining the number of users tagged in each tweet.

```
## Find users for game 9.Add a column for usertags and a column for number of u
ser tags in the tweet.
Game9 <- mutate (Game9, usertags=regmatches(Game9$text, gregexpr(usertag, Game9
$text)))
Game9 <- mutate (Game9, UQty=as.integer(lapply(Game9$usertags, function(x) leng
th(x))))
```

The number of users tagged per tweet.

```
## Create a Hashtags per Tweet table.
UserTagsperTweet = table(Game9$UQty)
UserTagsperTweetTable = as.data.frame(UserTagsperTweet)
names(UserTagsperTweetTable)[1] = 'Number of Users'
names(UserTagsperTweetTable)[2] = 'Number of Tweets'
UserTagsperTweetTable
```

```
##   Number of Users Number of Tweets
## 1               0             2944
## 2               1             5893
## 3               2              865
## 4               3              212
## 5               4               26
## 6               5               59
## 7               6                1
```

The total number of usertags in the data set.

```
## Determine total number of hashtags.
all.usertags <- unlist(regmatches(Game9$text, gregexpr (usertag, Game9$text)))
## all.hashtags
length(all.usertags)
```

```
## [1] 8664
```

The total number of unique usertags in the data set.

```
## Determine total number of unique hashtags.
unique.usertags <- unique(all.usertags <- unlist(regmatches(Game9$text, gregexp
r (usertag, Game9$text))))
length(unique.usertags)
```

```
## [1] 788
```

Finding the most commonly used usertags.

```
## Count the hashtag usage.
usertags.df <- data.frame(cbind(all.usertags))
usercount <- count(usertags.df, all.usertags)
usercount <- usercount[order(-usercount$n),]
usercount
```

```
## Source: local data frame [788 x 2]
##
##      all.usertags    n
## 1       @GatorsFB 1923
## 2  @FloridaGators 1643
## 3  @CoachMcElwain  524
## 4    @AlbertGator  467
## 5  @GatorsGameday  419
## 6     @ImShmacked  263
## 7             @UF  251
## 8       @ufalumni  244
## 9   @GatorsSoccer  120
## 10  @Jakeallen_14   81
## ..            ...  ...
```

Finding the top 25 most frequent tweeters.

The following 25 users tweeted most frequently during each game period. The usertags were selected after eliminating users that were clearly accounts for the team proper or their PR department. That is, we're trying to find a fan community for your team that tweets about their team's games on a regular basis. This community will form the basis for the project looking ahead.

```
## Find the 25 users that tweeted most frequently during Game 9 & 10.
frequenttweeters <- summarise(group_by(GameTweets, screenName), count=n())
frequenttweeters <- frequenttweeters[order(-frequenttweeters$count),]
frequenttweeters$screenName[1:25]
```

```
##  [1] "_Whoa_itsPayge_" "gator_fbreport"  "JayrockJenkins"
##  [4] "FloridaGators"   "LFTorresIII"     "SECstagram"
##  [7] "HotCorner_10"    "GatorsFB"        "GatorsSRH"
## [10] "bmanning96"      "TampaBaySRH"     "TylerWardFilms"
## [13] "BrookinsOneil"   "GatorVisionTV"   "Emmy_ArmadaFC"
## [16] "ChaseRojas"      "cdcoldplay1"     "ESPNGainesville"
## [19] "OHareTyler22"    "DarlingReina"    "gogators1974"
## [22] "Tebow815"        "gatorlane"       "Ls_Up_850"
## [25] "2_kontagious_"
```

The top 25 individual tweeters for game9, 10 and 11 are: *Whoa_itsPayge*, gator_fbreport, JayrockJenkins, FloridaGators, LFTorresIII, SECstagram, HotCorner_10, GatorsFB, GatorsSRH, bmanning96, TampaBaySRH, TylerWardFilms, BrookinsOneil, GatorVisionTV, Emmy_ArmadaFC, ChaseRojas, cdcoldplay1, ESPNGainesville, OHareTyler22, DarlingReina, gogators1974, Tebow815, gatorlane, Ls_Up_850, 2_kontagious_

Homework #9 specific analysis begins here.

Looking at user opinions

Now that we have our top 25 users we want to look at their opinions about their team's upcoming performance in the next football game. For now we'll manipulate the data using the sentiment analysis tools we just acquired, and inspecting the capabilities of the R package `sentiment` to judge its usefulness

We'll begin by analyzing a small sample of 10 tweets from Game 9.

We've hand picked ten tweets that have a large number of words and express some form of emotion or "positive/negative" spin.

```
TenTweets <- slice(Game9, 1)
TenTweets <- bind_rows(TenTweets, slice(Game9, 125))
TenTweets <- bind_rows(TenTweets, slice(Game9, 4564))
TenTweets <- bind_rows(TenTweets, slice(Game9, 4550))
TenTweets <- bind_rows(TenTweets, slice(Game9, 4612))
TenTweets <- bind_rows(TenTweets, slice(Game9, 4632))
TenTweets <- bind_rows(TenTweets, slice(Game9, 5188))
TenTweets <- bind_rows(TenTweets, slice(Game9, 5254))
TenTweets <- bind_rows(TenTweets, slice(Game9, 5484))
TenTweets <- bind_rows(TenTweets, slice(Game9, 4526))
TenTweets$text
```

```
##  [1] Good defense,  Not a "Targeting' Penalty !\n#SECFootball #GatorNation #
ItsGreatUF #GoGators #SECEastChamps https://t.co/gG7WVuhb2
M

##  [2] RT @maxstarks78: Wow, what a great tackle by Poole! Gators walk away
w/ a hard fought Victory in the Swamp, and the SEC East Division Title<U+0085
>

##  [3] Somehow #Vanderbilt always almost ruins homecoming for us, key word bei
ng almost. #GoGators  https://t.co/719URH01m
J

##  [4] RT @jhargrove1: @LostLettermen The coach doesn't even know the kickers
name! Haha! #TheDentist #GoGators https://t.co/XNxbyyPoz
a

##  [5] Boy that was not pretty.   But a win is a win! #GoGators #SECChamp
s!


##  [6] Thanks for the heart palpitations...that game was way too close! Gator
s are headed to the SEC Championship! #GoGator
s

##  [7] SEC East CHAMPS Gators!!!!! Sloppy sloppy game, Refs screwed us badly,
but Gators still finished when it mattered ! #GatorNation #GoGator
s

##  [8] "Anything inside the 20 we give to the dentist" - coach Mac on the fiel
d goal kicker in dental school <ed><U+00A0><U+00BD><ed><U+00B8><U+0082><ed><U+0
0A0><U+00BD><ed><U+00B8><U+0082><ed><U+00A0><U+00BD><ed><U+00B8><U+0082> #gogat
ors #thatwasugly #gatorfb
##  [9] The SEC is similar to the NBA.  All the talent is in the west and the e
ast champion is just sheep waiting for slaughter.  #GoGator
s

## [10] RT @jhargrove1: @LostLettermen The coach doesn't even know the kickers
name! Haha! #TheDentist #GoGators https://t.co/XNxbyyPoz
a

## 4159 Levels: 'Bama... See you in Atlanta #GoGators ...
```

Results: In examining the words contained within these tweets, some of them, in my opinion, do express emotional or “positive/negative” spin. We see words like: “Good”, “Wow”, “great”, “ruins” “Haha”, “Sloppy”. Some of the tweets I picked actually use a negative qualifier with a positive word, such as “not pretty”. I choose these because I can see how they would create a challenge in a textual analysis.

Let’s see what the classifiers have to offer.

We’ll run the classifiers on the `TenTweets` extraction of text, using `verbose=TRUE` so we can see how each word is being scored by the classifier.

```
polarity.df <- as.data.frame(classify_polarity(TenTweets$text))
TenTweets <- cbind(TenTweets, polarity.df$BEST_FIT)
polarity <- classify_polarity (TenTweets$text, verbose = TRUE)
```

```
## [1] "DOCUMENT 1"
## [1] "WORD: defense CAT: positive POL: weaksubj SCORE: 7.7510451179718"
## [1] "WORD: good CAT: positive POL: weaksubj SCORE: 7.7510451179718"
## [1] "WORD: penalty CAT: negative POL: weaksubj SCORE: 8.33686963728496"
## [1] "POS: 16.5333679773693 NEG: 8.78232285939751 RATIO: 1.88257346513717"
##
## [1] "DOCUMENT 2"
## [1] "WORD: division CAT: negative POL: weaksubj SCORE: 8.33686963728496"
## [1] "WORD: great CAT: positive POL: strongsubj SCORE: 8.44419229853175"
## [1] "WORD: hard CAT: negative POL: weaksubj SCORE: 8.33686963728496"
## [1] "WORD: swamp CAT: negative POL: strongsubj SCORE: 9.0300168178449"
## [1] "WORD: victory CAT: positive POL: weaksubj SCORE: 7.7510451179718"
## [1] "WORD: wow CAT: positive POL: strongsubj SCORE: 8.44419229853175"
## [1] "POS: 25.670707456461 NEG: 26.1492093145274 RATIO: 0.981701096491643"
##
## [1] "DOCUMENT 3"
## [1] "POS: 1.03127774142571 NEG: 0.445453222112551 RATIO: 2.31512017476245"
##
## [1] "DOCUMENT 4"
## [1] "WORD: even CAT: positive POL: weaksubj SCORE: 7.7510451179718"
## [1] "WORD: know CAT: positive POL: strongsubj SCORE: 8.44419229853175"
## [1] "POS: 17.2265151579293 NEG: 0.445453222112551 RATIO: 38.6718836070664"
##
## [1] "DOCUMENT 5"
## [1] "WORD: boy CAT: negative POL: weaksubj SCORE: 8.33686963728496"
## [1] "WORD: pretty CAT: positive POL: strongsubj SCORE: 8.44419229853175"
## [1] "POS: 9.47547003995745 NEG: 8.78232285939751 RATIO: 1.07892526745566"
##
## [1] "DOCUMENT 6"
## [1] "WORD: close CAT: negative POL: weaksubj SCORE: 8.33686963728496"
## [1] "WORD: game CAT: negative POL: weaksubj SCORE: 8.33686963728496"
## [1] "WORD: heart CAT: positive POL: weaksubj SCORE: 7.7510451179718"
## [1] "WORD: way CAT: negative POL: strongsubj SCORE: 9.0300168178449"
## [1] "POS: 8.78232285939751 NEG: 26.1492093145274 RATIO: 0.33585424147102"
##
## [1] "DOCUMENT 7"
## [1] "WORD: badly CAT: negative POL: strongsubj SCORE: 9.0300168178449"
## [1] "WORD: game CAT: negative POL: weaksubj SCORE: 8.33686963728496"
## [1] "WORD: sloppy CAT: negative POL: strongsubj SCORE: 9.0300168178449"
## [1] "POS: 1.03127774142571 NEG: 26.8423564950873 RATIO: 0.0384197915564697"
##
## [1] "DOCUMENT 8"
## [1] "WORD: mac CAT: negative POL: strongsubj SCORE: 9.0300168178449"
## [1] "POS: 1.03127774142571 NEG: 9.47547003995745 RATIO: 0.108836578774127"
##
## [1] "DOCUMENT 9"
## [1] "WORD: champion CAT: positive POL: weaksubj SCORE: 7.7510451179718"
## [1] "WORD: just CAT: positive POL: strongsubj SCORE: 8.44419229853175"
## [1] "WORD: slaughter CAT: negative POL: weaksubj SCORE: 8.33686963728496"
## [1] "WORD: talent CAT: positive POL: strongsubj SCORE: 8.44419229853175"
## [1] "POS: 25.670707456461 NEG: 8.78232285939751 RATIO: 2.92299746518566"
##
## [1] "DOCUMENT 10"
## [1] "WORD: even CAT: positive POL: weaksubj SCORE: 7.7510451179718"
## [1] "WORD: know CAT: positive POL: strongsubj SCORE: 8.44419229853175"
## [1] "POS: 17.2265151579293 NEG: 0.445453222112551 RATIO: 38.6718836070664"
```

```
emotion <- classify_emotion (TenTweets$text, verbose = TRUE)
```

```
## [1] "DOCUMENT 1"
## [1] "WORD: good CAT: joy SCORE: 6.31535800152233"
## [1] "DOCUMENT 2"
## [1] "WORD: great CAT: joy SCORE: 6.31535800152233"
## [1] "WORD: walk CAT: joy SCORE: 6.31535800152233"
## [1] "DOCUMENT 3"
## [1] "DOCUMENT 4"
## [1] "DOCUMENT 5"
## [1] "DOCUMENT 6"
## [1] "WORD: close CAT: joy SCORE: 6.31535800152233"
## [1] "WORD: heart CAT: joy SCORE: 6.31535800152233"
## [1] "DOCUMENT 7"
## [1] "DOCUMENT 8"
## [1] "DOCUMENT 9"
## [1] "DOCUMENT 10"
```

Results: In evaluating the question, “Are there any definitions of polarity that surprise you given their context?”, I have to say, I can’t really see where the positive, neutral, and negative polarities are highly correlated to how I would view these tweets." I suspect, that sports analogies may be difficult to rate because the sentiments are sometimes inverse to how they might be evaluated. For example, how is “not pretty” evaluated?

Now let’s look at the full data set.

We’ll run the classifiers on the full data set for Game 9, then add the classifiers (and only the classifiers, not the raw scores) as columns to the existing Game 9 data frame. Next, we’ll look for the number of tweets that are classified “positive” subjectivity as well as the number classified as “joy” emotions?

```
## Create a dataframe to hold the polarity classifiers. Save this dataframe so
it can be reloaded in later iterations.  This will save time in processing.
## polarity9.df <- as.data.frame(classify_polarity(Game9$text))
## save (polarity9.df, file="polarity.RData")
load ("polarity.RData")

## Bind the polarity classifier back to the original GameTweets data set.
Game9 <- cbind(Game9, BEST_FIT = polarity9.df$BEST_FIT)
BestFitTable <- Game9 %>% count(BEST_FIT)
```

Results: Using the `count` function, the number of tweets that can be classified as “positive” are: 7124.

```
## Create a dataframe to hold the emotion classifiers. Save this dataframe so i
t can be reloaded in later iterations.  This will save time in processing.
## emotion.df <- as.data.frame(classify_emotion(Game9$text))
## emotion.df <- rename(emotion.df, EBEST_FIT = BEST_FIT)
## save (emotion.df, file="emotion.RData")
load ("emotion.RData")

## Bind the emotion classifiers back to the original GameTweets data set
Game9 <- cbind(Game9, emotion.df)
BestFitTable <- Game9 %>% count(EBEST_FIT)
```

Results: Using the `count` function, the number of tweets that can be classified as “joy” are: 1551.

——I STILL HAVE SOME WORK TO DO HERE ——————-

Now we will use `group_by` and `summarize` to find the fraction of tweets that are positive, neutral and negative for each user-day combination.

```
## Create a new dataframe `polarity.df` to hold the summary table.
polarity.df <- Game9 %>%
  group_by(screenName) %>%
  summarise(Positive = sum(BEST_FIT=="positive"),
            Neutral = sum(BEST_FIT=="neutral"),
            Negative = sum(BEST_FIT=="negative"),
            Total = n(),
            PosFrac = sum(BEST_FIT=="positive")/n(),
            NeutFrac = sum(BEST_FIT=="neutral")/n(),
            NegFrac = sum(BEST_FIT=="negative")/n(),
            TotFrac = n()/n())

## Sort the dataframe by total so that output will display the most active twee
ters at the top.
polarity.df <- polarity.df[order(-polarity.df$Total),]
head(polarity.df, 10)
```

```
## Source: local data frame [10 x 9]
##
##         screenName Positive Neutral Negative Total   PosFrac   NeutFrac
## 1       SECstagram       56       6       17    79 0.7088608 0.07594937
## 2     HotCorner_10       40       9       29    78 0.5128205 0.11538462
## 3    FloridaGators       21       8       12    41 0.5121951 0.19512195
## 4     gogators1974       22       1        7    30 0.7333333 0.03333333
## 5    MrMarques2727       11       3       12    26 0.4230769 0.11538462
## 6          Trippo7       16       5        4    25 0.6400000 0.20000000
## 7         Tebow815       13       5        5    23 0.5652174 0.21739130
## 8      CliffWilkin       14       5        3    22 0.6363636 0.22727273
## 9  _Whoa_itsPayge_       11       2        7    20 0.5500000 0.10000000
## 10       GatorsSRH       15       0        1    16 0.9375000 0.00000000
## Variables not shown: NegFrac (dbl), TotFrac (dbl)
```

Now let's find which users have the greatest propensity for "joy"? Which for "sadness"? Is this consistent from day to day in the data set?

```
## Create a new dataframe `emotion.df` to hold the summary table.
emotion.df <- Game9 %>%
  group_by(screenName) %>%
  summarise(ANGER = mean(ANGER),
            DISGUST = mean(DISGUST),
            FEAR = mean(FEAR),
            JOY = mean(JOY),
            SADNESS = mean(SADNESS),
            SURPRISE = mean(SURPRISE),
            Total = n())

## Sort the dataframe by total so that output will display the most active twee
ters at the top.
emotion.df <- emotion.df[order(-emotion.df$JOY),]
mostjoy <- emotion.df$screenName[1]
head(emotion.df, 1)
```

```
## Source: local data frame [1 x 8]
##
##   screenName ANGER DISGUST FEAR JOY SADNESS SURPRISE Total
## 1  __di3go__     1       1    2   4       1        2     1
```

```
emotion.df <- emotion.df[order(-emotion.df$SADNESS),]
mostsaddness <- emotion.df$screenName[1]
head(emotion.df, 1)
```

```
## Source: local data frame [1 x 8]
##
##    screenName ANGER DISGUST FEAR JOY SADNESS SURPRISE Total
## 1 _JohnGandy     1       1    2   4       3        2     1
```

```
BestFitTable <- Game9 %>% count(EBEST_FIT)
```

The user with the greatest propensity for joy is **di3go**, the user with the greatest propensity for saddness is _JohnGandy.

Homework 10 starts here:

At this point we have three weekends worth of data (Game9, Game10 & Game11). We'll continue to capture data Game12 on November 28. This will give us four games worth of data using the twitterR function. To gather more data, we'll look for a way to reach back to earlier games.

Getting tweets directly from the twitter website.

Trial 1: Scraping tweets directly from a user page.

For this trial, we will put together our own manual collection of tweets directly scraping from the user page. Except… it's queried to scrolling. Worst case, we have to do this ourselves, and manually. We'll start by looking at the three most frequent users in our tweets so far and use the "getURL" solution to grab their history of tweets on their visible page.

```
## Pulling tweets from _Whoa_itsPayge_
## This one does not work.  #2 & #3 below work just fine with same code.
## WhoaitsPayge <- getURL("https://twitter.com/_Whoa_itsPayge_") ##, ssl.verify
peer = FALSE)
## writeLines (WhoaitsPayge, "WhoaitsPayge.txt") ## check it out
## tweets <- regmatches(WhoaitsPayge, gregexpr ('(?<=<p class=\"TweetTextSize T
weetTextSize--16px js-tweet-text tweet-text\" lang=\"en\" data-aria-label-part=
\"0\">).*(?=</p>)', WhoaitsPayge, perl=TRUE))
## tweets <- unlist(tweets)
## tweets <- as.data.frame(tweets)
## Pulling tweets from gator_fbreport
gator_fbreport <- getURL("https://twitter.com/gator_fbreport") ##, ssl.verifype
er = FALSE)
writeLines (gator_fbreport, "gator_fbreport.txt") ## check it out
tweets2 <- regmatches(gator_fbreport, gregexpr ('(?<=<p class=\"TweetTextSize T
weetTextSize--16px js-tweet-text tweet-text\" lang=\"en\" data-aria-label-part=
\"0\").*(?=</p>)', gator_fbreport, perl=TRUE))
tweets2 <- unlist(tweets2)
tweets2 <- as.data.frame(tweets2)
dim(tweets2)[1]
```

```
## [1] 14
```

```
## Pulling tweets from JayrockJenkins
JayrockJenkins <- getURL("https://twitter.com/JayrockJenkins") ##, ssl.verifype
er = FALSE)
writeLines (JayrockJenkins, "JayrockJenkins.txt") ## check it out
tweets3 <- regmatches(JayrockJenkins, gregexpr ('(?<=<p class=\"TweetTextSize T
weetTextSize--16px js-tweet-text tweet-text\" lang=\"en\" data-aria-label-part=
\"0\">).*(?=</p>)', JayrockJenkins, perl=TRUE))
tweets3 <- unlist(tweets3)
tweets3 <- as.data.frame(tweets3)
dim(tweets3)[1]
```

```
## [1] 19
```

Results: There are 14 tweets in both of these data sets. Unfortunately, I was not able to get the #1 tweeter to work. I was able to get both #2 and #3 working with the existing example code. From this initial analysis, this does not seem to be an effective method. First, 14 or 19 tweets is not enough to give us any statistical significance. Second, an analysis of the tweets reveals we have very little about the actual game we are trying to capture.

Trial 2: Scraping tweets by copying from the source page.

In this step I'll use the "copy from source" solution to copy from the web page source file. I'm also going to try using a different search option. I'll look for tweets with either the game specific hashtag "#UFvsUGA", or that contain both UF and UGA in the text. To capture the tweets, I'll use the view source code option and copy and paste the source code into a text file. Next well develop a regexp that will extract the tweet content from the text file.

```
Game8.2 <- paste(readLines ("Georgia2.txt"), collapse = "")
Game8.2 <- regmatches(Game8.2, gregexpr ('(?<=<p class="TweetTextSize  js-tweet
-text tweet-text" lang="en" data-aria-label-part="0">).*?(?=</p>)', Game8.2, pe
rl=TRUE))
Game8.2 <- unlist(Game8.2)
Game8.2 <- as.data.frame(Game8.2)
dim(Game8.2)
```

```
## [1] 16  1
```

Results: This also did not seem to be an effective method. We again only captured 16 tweets as above. This is a bit better than in Trial 1, because these tweets are focused on the game we are trying to capture.

Trial 3: Scraping tweets by copying directly from the page.

For trial 3, we'll again search for tweets with with either the game specific hashtag "#UFvsUGA", or that contain both UF and UGA in the text. We are looking for tweets beginning three days before game day up to game day. Once the tweets are displayed on the screen, we will need to scroll down and reveal all of the tweets back to three days prior to game day, highlight and copy the entire page and paste into a text file. Did you need to alter the HTML from the existing example above in order to make this work? Next well develop a regexp that will extract the tweet content from the text file. Since we no longer have html code in this file, we cannot use the same extraction method as in Trial 2.

```
## The Georgia.txt file used below was created by using the advanced search uti
lity on twitter on the hash code #UFvsUGA, with the dates of 10/28/2015 - 10/3
0/2015. ## After executing the search, I scrolled down until all tweets were ex
posed, then copied and pasted the content into a text file.
copiedPage <- paste (readLines("Georgia.txt"), collapse = " ")
Game8 <- regmatches(copiedPage, gregexpr ('[@].*?(?=[0-9] retweets)', copiedPag
e, perl=TRUE))
Game8 <- unlist(Game8)
Game8 <- as.data.frame(Game8)
head(Game8,5)
```

```
#
#



     Game8
##
1
                                    @CFB_Update  Oct 30 Man.. re-watching the p
ast few games. LSU included.. and i'm really excited about tomorrow. #UFvsUGA #
BeatUGA
##
2
                                             @Commando074  Oct 30 Hey, where di
d all the red and black go? #UFvsUGA  #GoGators #UGAHateWeek  Embedded image pe
rmalink
## 3 @GatorsScott  Oct 30 Scott Carter Retweeted Seth Emerson #UFvsUGA Scott Ca
rter added, Seth Emerson @SethEmersonAJC For the record: Still not 100% convinc
ed Bauta is starting, based on my reporting. But safe to say the plan is for hi
m to play.
## 4                                      @CFB_Update  Oct 30 Looks like #UGA i
s going to be dressing up as #RockyTop for Halloween this year. #UFvsUGA htt
p://florida.247sports.com/Bolt/Three-questions-for-Florida-Georgia-40665405 â<U
+0080>¦ View summary
##
5
                                                                  @JalenParker2
0  Oct 30 Florida, USA Ugh can the #UFvsUGA start already. My patience is weari
ng out.
```

```
write.csv(Game8, file = "Game8.csv")
```

Results: The solutions that is the most productive is the copy from page solution. We were able to capture 180 tweets using this method. I'll use this to capture the remaining data for the project.

Extracting the data into a table.

For each selections of tweets gathered we'll create a data frame with three columns: the username, the date of the tweet, and the contents of the tweet itself.

```
Game8 <- mutate (Game8, username=regmatches(Game8$Game8, gregexpr("^[@].* " , G
ame8$Game8)))
Game8 <- mutate (Game8, date=regmatches(Game8$Game8, gregexpr("([JFMASOND][a-z]
{2}) ([0-9]{2})", Game8$Game8)))
## write.csv(Game8, file = "Game8.csv")

## Way 1: gsub and dplyr
## dates2 <- gsub ("([A-Z][a-z]{2}) ([0-9]{2})", "2015-\\1-\\2", dates)
## dates3 <- gsub ("Sep", "09", dates2) %>% gsub ("Oct", "10", .) %>% gsub ("No
v", "11", .) %>% gsub ("Dec", "12", .)

## Way 2: as.Date
## as.Date (dates, format = "%b %d")
## as.Date (dates, format = "%b %d") - 2
## head(Game8, 5)
```

Creating one big file of all our game tweets

Once all of the tweets have been extracted, and fit into a common table format, we can join all the game tweets together using `rbind_list` into one giant data frame.

Coming Soon…

Current Status Report

I still have work to do in building the model that will gather the appropriate game related tweets and then extract the username, date and tweet from that data. I“m getting close. I've probably put in more than 15 hours this week already on this project. Once I have an effective model, I should be able to replicate it fairly easily for all of the games. ***

Rest of Project Instructions

This assumes we have all the data we need to finish the project from the above.

1. Get the record for your team in all games from September 1 until November 14. This means a data frame with columns for date (use YYYY-MM-DD), opponent, "did they win", your team's score and the opponent's score. If at all possible, see if you can find the closing "spread" – the market estimate for the median margin of victory for your team – and add this to the list. You will have to construct this yourself in your favorite spreadsheet program; save this to a `.csv` file.

2. Repeat the analyses for the sentiment analysis from the previous homework for the data set you have put together on your 25 "sensors". That means each tweet will have a column indicating the likely positive/negative subjectivity and the relevant most-likely emotion.

3. Collect these using `dplyr` for the average subjectivity (positive/negative) and the total fraction of each emotion (joy, anger, etc) for the tweets for a user on a particular day. Collect these for the two days preceding a game. This means you should have 50 rows corresponding to each game, times as many games as you have in your collection.

4. Add the result for the game that followed (win/loss) as a column to this table. Can you do this automatically by joining the date in your table to the date in your game result? You should be able to manipulate your result in Question 1 to do this; you may need the `as.Date()` function in R.

5. Conduct a logistic regression on the win-loss outcome. Include the emotional and subjective variables in your analysis. Were any of them predictive of winning?

6. Repeat the previous two steps, but indicate whether your team instead won the game by more than 7 points. Do this again for whether your team lost by more than 7 points. Do this again for if they beat or tied the spread.

7. Given the content of your prior analyses, do you believe that social media censoring is a reasonable way to get predictive information about the game that might not otherwise be useful?

```
```