Homework 9 – Due Monday November 16, 2015

Cheryl Calhoun

11/015/2015

```
## Setting up the work environment.
##setwd("C:/Users/07001412/OneDrive/Education/UF/2015/Fall/EDF6938/Week 11")
library("dplyr") ##, lib.loc="~/R/win-library/3.2")

## Setting up for Twitter API access.
## Install `twitterR` and supporting packages if not already installed.
##install.packages("twitteR")
##install.packages("base64enc")  ## This solved some of my problems, so you sho
uld install it too.
##download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pe
m") ##-- this was suggested if you are running a Windows machine.

## Loading twitterR and base64end libraries.
library("twitteR") ##, lib.loc="~/R/win-library/3.2")
library("base64enc") ##, lib.loc="~/R/win-library/3.2")

## 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="sourc
e")
library(sentiment)

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

##download.file(url="http://cran.r-project.org/bin/windows/Rtools/")"
##install.packages("RCurl")
library(RCurl)
```

```
## Loading variables.

## Setting search patterns for usertags and hashtags.
usertag <- "@[A-z_0-9]+"
hashtag <- "#[A-z0-9]+"

## Load the subjectivity and emotions files from (https://github.com/timjurka/s
entiment/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=F
ALSE)
```

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 hasgtag 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 apparant that this data set includes tweets from all Gator sports as well as many other miscellanous tweets. Futher 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 #	Date	Opponent	Location
Game 11	Nov 21	Fl. Atlantic	Gainesville
Game 12	Nov 28	Florida State	Gainesville

```r
## Collect weekly tweets beginning Thursday before game and ending on Sunday after the game. Store tweets in a .csv file.  This code will be executed on a weekly 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="2015-11-07", n=10000, retryOnRateLimit=120)
##Florida9.df <- rbind_all (lapply (Florida9, function(rr) rr$toDataFrame()))
##write.csv(Florida9.df, file = "Vanderbilt2.csv")

## Collect data for game 10 using both #GoGators and #UFvsSC. The #GoGators data will be stored to compare with results from #UFvsSC data.
##FloridaGG <- searchTwitter("#GoGators", since="2015-11-12", until="2015-11-14", n=10000, retryOnRateLimit=120)
##FloridaGG.df <- rbind_all (lapply (FloridaGG, function(rr) rr$toDataFrame()))
##write.csv(FloridaGG.df, file = "CarolinaGG.csv")

## Florida10 <- searchTwitter("#UFvsSC", since="2015-11-12", until="2015-11-16", n=10000, retryOnRateLimit=120)
## Florida10.df <- rbind_all (lapply (Florida10, function(rr) rr$toDataFrame()))
## write.csv(Florida10.df, file = "Carolina.csv")

## Collect data for game 11
##Florida11 <- searchTwitter("#GoGators", since="2015-11-19", until="2015-11-23", n=10000)
##Florida11.df <- rbind_all (lapply (Florida11, function(rr) rr$toDataFrame()))
##write.csv(Florida11.df, file = "FLAtl.csv")

## Collect data for game 12
##Florida12 <- searchTwitter("#GoGators", since="2015-11-26", until="2015-11-30", n=10000)
##Florida12.df <- rbind_all (lapply (Florida12, function(rr) rr$toDataFrame()))
##write.csv(Florida12.df, file = "FloridaSt.csv")

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

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

Gathering Data for Games 11-12 - Coming soon…

Collect a set of 10000 tweets for each of the last five ***Thursday-Saturday*** blocks on which a football game was played by the University of Florda. Repeat the hashtag collection and user tagging exercises for each of these samples. Do the same users appear to produce the same volume each week? Do the same secondary hashtags appear?

```
## Games 11-12 coming soon...
```

Game 9: Vanterbilt

Evaluating HashTags

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

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 user tags in the tweet.
##Game9 <- mutate (Game9, usertags=regmatches(Game9$text, gregexpr(users, Game9$text)))
##Game9 <- mutate (Game9, UQty=as.integer(lapply(Game9$users, function(x) length(x))))

Game9 <- mutate (Game9, usertags=regmatches(Game9$text, gregexpr(usertag, Game9$text)))
Game9 <- mutate (Game9, UQty=as.integer(lapply(Game9$usertags, function(x) length(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$text, gregexpr (usertag, Game9$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
## ..            ...  ...
```

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.
frequenttweeters9 <- summarise(group_by(Game9, screenName), count=n())
frequenttweeters9 <- frequenttweeters9[order(-frequenttweeters9$count),]
head(frequenttweeters9, 25)
```

```
## Source: local data frame [25 x 2]
##
##         screenName count
## 1       SECstagram    79
## 2     HotCorner_10    78
## 3    FloridaGators    41
## 4     gogators1974    30
## 5    MrMarques2727    26
## 6          Trippo7    25
## 7         Tebow815    23
## 8      CliffWilkin    22
## 9  _Whoa_itsPayge_    20
## 10       GatorsSRH    16
## ..             ...   ...
```

```
## ## Find the 25 users that tweeted most frequently during Game 9.
frequenttweeters10 <- summarise(group_by(Game10, screenName), count=n())
frequenttweeters10 <- frequenttweeters10[order(-frequenttweeters10$count),]
head(frequenttweeters10, 25)
```

```
## Source: local data frame [25 x 2]
##
##         screenName count
## 1   gator_fbreport   121
## 2      LFTorresIII    82
## 3   TylerWardFilms    60
## 4         GatorsFB    54
## 5        GatorsSRH    49
## 6   JayrockJenkins    49
## 7    FloridaGators    45
## 8  _Whoa_itsPayge_    44
## 9  ESPNGainesville    44
## 10     TampaBaySRH    43
## ..             ...   ...
```

```
## Remove tweets from FloridaGators, GatorsFB and UF
GameTweets <- filter(GameTweets, screenName!="FloridaGators")
GameTweets <- filter(GameTweets, screenName!="GatorsFB")
GameTweets <- filter(GameTweets, screenName!="UF")

## 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),]
head(frequenttweeters, 25)
```

```
## Source: local data frame [25 x 2]
##
##          screenName count
## 1   gator_fbreport   121
## 2      LFTorresIII    82
## 3       SECstagram    80
## 4     HotCorner_10    78
## 5        GatorsSRH    65
## 6  _Whoa_itsPayge_    64
## 7   TylerWardFilms    60
## 8   JayrockJenkins    58
## 9      TampaBaySRH    58
## 10 ESPNGainesville    44
## ..             ...   ...
```

The top 25 individual tweeters for each game are:

(Note: We’ve removed the following UF users from this list: FloridaGators, GatorsFB, UF)

Game 9	Game 10	Game 11	Game 12
SECstagram	gator_fbreport		
HotCorner_10	LFTorresIII		
gogators1974	TylerWardFilms		
MrMarques2727	GatorsSRH		
Trippo7	JayrockJenkins		
Tebow815	*Whoa_itsPayge*		
CliffWilkin	ESPNGainesville		
Whoa_itsPayge	TampaBaySRH		
GatorsSRH	gatorlane		
gatorzonenews	SWNY315		
jhern2498	brittanylaughsx		
TampaBaySRH	BrookinsOneil		
2103_amber	haleythecoolkid		
bmanning96	bmanning96		
major1029	Emmy_ArmadaFC		
ClaireabellGatr	ChaseRojas		

Game 9	Game 10	Game 11	Game 12
CVan_Huss	the_timbo_slice		
hunterlynch29	cdcoldplay1		
krizielyvonne	mikecombs386		
REGarrison	ChompingGators		
ktharris	NikHarper		
rebeccavelaz17	2_kontagious_		
TOTODUVALDIVA	DarlingReina		
BrookinsOneil	Ls_Up_850		
ChristophersZen	pbpsports		

Starting Homework 9 specific analysis

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

First we'll produce a sample of 10 tweets from Game 9. We're looking for 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.

Now 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)
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"
```

```
##       POS                NEG                 POS/NEG
##  [1,] "16.5333679773693" "8.78232285939751"  "1.88257346513717"
##  [2,] "25.670707456461"  "26.1492093145274"  "0.981701096491643"
##  [3,] "1.03127774142571" "0.445453222112551" "2.31512017476245"
##  [4,] "17.2265151579293" "0.445453222112551" "38.6718836070664"
##  [5,] "9.47547003995745" "8.78232285939751"  "1.07892526745566"
##  [6,] "8.78232285939751" "26.1492093145274"  "0.33585424147102"
##  [7,] "1.03127774142571" "26.8423564950873"  "0.0384197915564697"
##  [8,] "1.03127774142571" "9.47547003995745"  "0.108836578774127"
##  [9,] "25.670707456461"  "8.78232285939751"  "2.92299746518566"
## [10,] "17.2265151579293" "0.445453222112551" "38.6718836070664"
##       BEST_FIT
##  [1,] "neutral"
##  [2,] "negative"
##  [3,] "positive"
##  [4,] "positive"
##  [5,] "neutral"
##  [6,] "negative"
##  [7,] "negative"
##  [8,] "negative"
##  [9,] "positive"
## [10,] "positive"
```

```
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"
```

```
emotion <- classify_emotion (TenTweets$text)
```

Now we'll run the classifiers on the full data set `GameTweets` which currently contains tweets from Game 9 & Game 10. Add the classifiers (and only the classifiers, not the raw scores) as columns to your existing data frames. How many tweets are classified "positive" subjectivity? How many are classified as "joy" emotions?

```
## Create a dataframe to hold the polarity classifiers.
polarity.df <- as.data.frame(classify_polarity(GameTweets$text))

## Bind the polarity classifier back to the original GameTweets data set.
GameTweets <- cbind(GameTweets, BEST_FIT = polarity.df$BEST_FIT)
```

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 <- GameTweets %>%
  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, 25)
```

```
## Source: local data frame [25 x 9]
##
##         screenName Positive Neutral Negative Total   PosFrac   NeutFrac
## 1   gator_fbreport       89       9       23   121 0.7355372 0.07438017
## 2      LFTorresIII       47      11       24    82 0.5731707 0.13414634
## 3       SECstagram       56       7       17    80 0.7000000 0.08750000
## 4     HotCorner_10       40       9       29    78 0.5128205 0.11538462
## 5        GatorsSRH       51       2       12    65 0.7846154 0.03076923
## 6  _Whoa_itsPayge_       35       8       21    64 0.5468750 0.12500000
## 7   TylerWardFilms       25      10       25    60 0.4166667 0.16666667
## 8   JayrockJenkins       39       7       12    58 0.6724138 0.12068966
## 9      TampaBaySRH       46       1       11    58 0.7931034 0.01724138
## 10 ESPNGainesville       27       8        9    44 0.6136364 0.18181818
## ..             ...      ...     ...      ...   ...       ...        ...
## Variables not shown: NegFrac (dbl), TotFrac (dbl)
```

6. Which users have the greatest propensity for “joy”? Which for “sadness”? Is this consistent from day to day in the data set?

```
## Create a dataframe to hold the emotion classifiers.
emotion.df <- as.data.frame(classify_emotion(GameTweets$text))
emotion.df <- rename(emotion.df, EBEST_FIT = BEST_FIT)
## Bind the emotion classifiers back to the original GameTweets data set
GameTweets <- cbind(GameTweets, emotion.df)

## Create a new dataframe `polarity.df` to hold the summary table.
emotion.df <- GameTweets %>%
  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),]
head(emotion.df, 25)
```

```
## Source: local data frame [25 x 8]
##
##         screenName ANGER DISGUST FEAR JOY SADNESS SURPRISE Total
## 1        04Gator_2     1       1    2   5       1        2     1
## 2   0fficial_Ferny     1       1    2   5       1        2     1
## 3  1000Girlfriends     1       1    2   5       1        2     1
## 4        11Zette17     1       1    2   5       1        2     1
## 5      137saltlife     1       1    2   5       1        2     1
## 6           1971vw     1       1    2   5       1        2     1
## 7      1986rbgator     1       1    2   5       1        2     1
## 8    1stevehallett     1       1    2   5       1        2     1
## 9      1wesmoulden     1       1    2   5       1        2     1
## 10      411charlie     1       1    2   5       1        2     1
## ..             ...   ...     ...  ... ...     ...      ...   ...
```

```
emotion.df <- emotion.df[order(-emotion.df$SADNESS),]
head(emotion.df, 25)
```

```
## Source: local data frame [25 x 8]
##
##         screenName ANGER DISGUST FEAR JOY SADNESS SURPRISE Total
## 1   47CremeBrulees     1       1    2   5       3        2     1
## 2          AREaton     1       1    2   5       3        2     1
## 3       Armando526     1       1    2   5       3        2     1
## 4  BrettHamilton32     1       1    2   5       3        3     1
## 5     Brett_Burgin     1       1    2   5       3        2     1
## 6    Cgbaxley4Chad     1       1    2   5       3        3     1
## 7        Chelsieg1     1       1    2   5       3        2     1
## 8     CoferRichard     1       1    2   5       3        2     1
## 9     Daryl_Manuel     1       1    2   5       3        3     1
## 10 Figgz_Nation928     1       1    3   5       3        2     1
## ..             ...   ...     ...  ... ...     ...      ...   ...
```

```
```