NBA Topic Mining

Shanshan Bradford

Section 1

```
options(tinytex.verbose = TRUE)

#clear up memory, set working directory and seeds
rm(list = ls())
setwd("/Users/syu/Library/CloudStorage/OneDrive-
St.JudeChildren'sResearchHospital/UDrive/Documents_syu_Backup/Github_deposit/
TextMining")

#Load twitter package
library(twitteR)
library(bitops)
library(RCurl)
library(ROAuth)

knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
```

#1.Retrieve tweets from Twitter with the hashtag #nba #

#Assign Twitter consumer key, secret and access token and secret

consumer_key <- “QLmWyDb3OmiMi7kkWta68F5rd”

consumer_secret <- “7YmvWUwlj6VwqA1B5P1TDetEelfJJnaqOyqGjIKisgvKFvXeib”

access.token <- “913200731829698560-N8rWlg3JqjK473rMWpqpEcvI8nHWC1B”

access.secret <- “Ka7zH3edvoOULrJC4CoudLZmqJiTxoI9JlkGOxBNJbGw2”

#connect to twitter and search tweets #nba

setup_twitter_oauth(consumer_key, consumer_secret, access.token, access.secret)

nba.tweets <- searchTwitter(“#nba”, n = 320, lang = “en”)

#strip retweets and check the number of tweets afterward

nba.nort <- strip_retweets(nba.tweets, strip_manual = T, strip_mt = T)

length(nba.nort)

#convert the tweets to dataframe and check out associated attributes

nba.df <- twListToDF(nba.nort)

colnames(nba.df)

#save tweets to csv

write.csv(nba.df, file = “NBA_tweets.csv”, na = “NA”)

Section 2

```r
options(tinytex.verbose = TRUE)

# 2. Clean up tweets
tweet.tb <- read.csv("NBA_tweets.csv", header = T, sep = ",", as.is = T)
nba.text <- gettext(tweet.tb$text)

# Replace @UserName with one space
# One space replacement is to avoid words being glued together
nba.modify <- gsub("@\\w{1,20}", " ", nba.text)

# Replace control character "\n" and "\n\n" with one space
nba.modify <- gsub("[[:cntrl:]]{1,10}", " ", nba.modify)

# Replace https links with one space
nba.modify <- gsub("(https)(://)(.*)[/]\\w+", " ", nba.modify)

# Replace punctuation with one space
nba.modify <- gsub("[[:punct:]]{1,20}", " ", nba.modify)

# Replace non graphical character with space
nba.modify <- gsub("[^[:graph:]]", " ", nba.modify)

# Replace tab and extra space introduced early with one space
nba.modify <- gsub("[ |\t]{2,}", " ", nba.modify)
nba.modify <- gsub("\\s+", " ", nba.modify)

# Remove extra blank space at the beginning and the end
nba.modify <- gsub("^ +", "", nba.modify)
nba.modify <- gsub(" $+", "", nba.modify)


# 3. Preprocess tweets further for analysis
library(NLP)
library("tm")
library(RColorBrewer)
library(wordcloud)
library("SnowballC")
library("lsa")

# generate corpus for the cleaned nba tweets and check out the corpus length
nba.corpus <- VCorpus(VectorSource(nba.modify))
length(nba.corpus)
```

```
## [1] 207

# transform the corpus to lower case, remove punctuation and numbers and
randomly check sample
trans.nbacorp <- tm_map(nba.corpus, content_transformer(tolower)) #convert to
lower cases
trans.nbacorp <- tm_map(trans.nbacorp, removePunctuation) # remove
punctuation
trans.nbacorp <- tm_map(trans.nbacorp, removeNumbers) # remove numbers
trans.nbacorp <- tm_map(trans.nbacorp, stripWhitespace)

# remove stop words
input.nbacorp <- tm_map(trans.nbacorp, removeWords,
                 c(stopwords("english"), "can", "don", "just", "nba", "via",
                   "e", "s", "y"))
#find an empty entry
inspect(input.nbacorp[[4]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 0

# 4. Generate a word cloud with the NBA tweets
set.seed(1234)

#generate document-term matrix in order to remove empty documents
nbacorp.dtm <- DocumentTermMatrix(input.nbacorp)
row.total <- apply(nbacorp.dtm, 1, sum)

#Correspondingly, remove the same empty entries from the corpus and document-
term matrix
input.nbacorp.noemp <- input.nbacorp[which(row.total > 0)]
nbacorp.docterm <- DocumentTermMatrix(input.nbacorp.noemp)
# or the following code will generate the same doc-term matrix with empty
entries removed
nbacorp.docterm <- nbacorp.dtm[which(row.total > 0),]

#The index of the matrix shifts accordingly, but the doc entry index remains
the same
inspect(nbacorp.docterm[15:16,])

## <<DocumentTermMatrix (documents: 2, terms: 772)>>
## Non-/sparse entries: 6/1538
## Sparsity           : 100%
## Maximal term length: 18
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs aaron absolutely accident account bizarre girlfriend scandals sex
wags
```

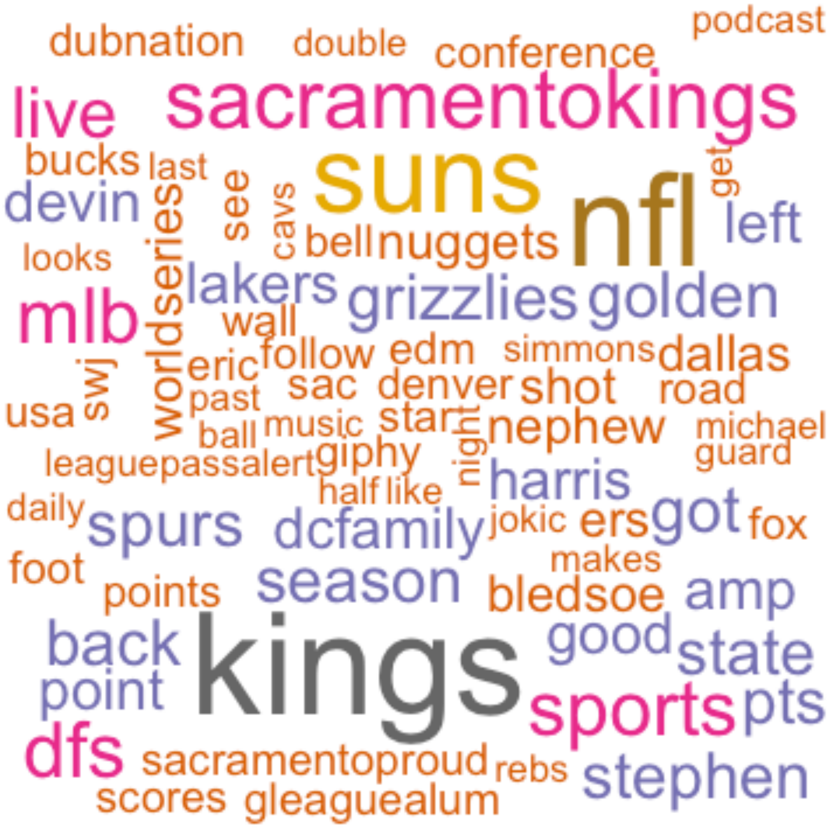
```
##   16     0          0        0       0       0          1         0   0
1
##   17     0          0        0       0       1          0         1   1
0
##     Terms
## Docs watson
##   16      1
##   17      0
```

```
# generate the term-document Matrix from the cleaned document-term matrix
nbacorp.terdoc <- t(nbacorp.docterm)
inspect(nbacorp.terdoc)
```

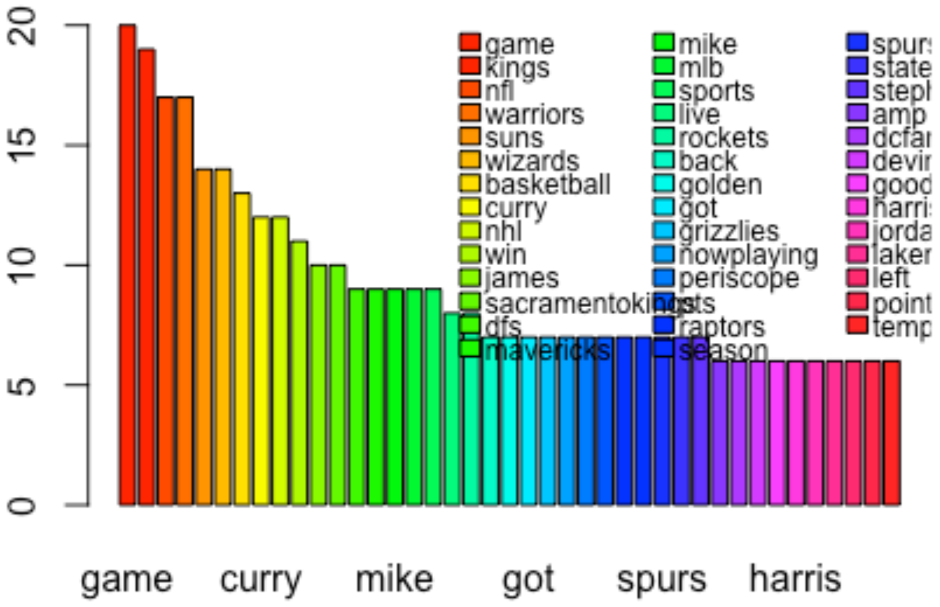
```
## <<TermDocumentMatrix (terms: 772, documents: 201)>>
## Non-/sparse entries: 1404/153768
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency (tf)
## Sample             :
##             Docs
## Terms         1 10 121 181 201 56 63 70 73 88
##   basketball 0  0   0   0   0  0  0  0  0  0
##   curry      0  0   0   1   1  0  1  0  0  0
##   game       0  1   0   0   0  1  0  0  0  0
##   kings      0  0   0   0   0  0  1  1  0  1
##   nfl        0  1   0   0   0  0  0  0  1  0
##   nhl        0  0   0   0   0  0  0  0  1  0
##   suns       0  0   0   0   0  0  0  2  0  0
##   warriors   0  0   0   1   1  0  0  0  0  0
##   win        0  0   0   0   0  0  0  1  0  0
##   wizards    0  0   0   0   0  0  0  0  0  0
```

```
# Find frequency of terms in term-doc matrix with frequency over 3
findFreqTerms(nbacorp.terdoc, lowfreq = 3)
```

```
##   [1] "actually"         "aldridge"         "amp"              "assists"
##   [5] "back"             "ball"             "basketball"       "bell"
##   [9] "bledsoe"          "buckle"           "bucks"            "cavs"
##  [13] "chriss"           "collegefootball"  "conference"       "consoles"
##  [17] "curry"            "daily"            "dallas"           "dcfamily"
##  [21] "denver"           "detroit"          "devin"            "dfs"
##  [25] "double"           "dubnation"        "edm"              "eric"
##  [29] "ers"              "first"            "follow"           "foot"
##  [33] "fox"              "game"             "games"            "garrett"
##  [37] "get"              "giphy"            "gleaguealum"      "going"
##  [41] "golden"           "good"             "got"              "grizzlies"
##  [45] "guard"            "half"             "harris"           "hornets"
##  [49] "james"            "jokic"            "jordan"           "kings"
##  [53] "lakers"           "last"             "latest"           "league"
##  [57] "leaguepassalert" "left"             "let"              "like"
##  [61] "live"             "looks"            "love"             "makes"
##  [65] "mavericks"        "memphis"          "michael"          "mike"
##  [69] "mlb"              "music"            "mvp"              "nephew"
##  [73] "nfl"              "nhl"              "night"            "now"
##  [77] "nowplaying"       "nuggets"          "past"             "periscope"
##  [81] "phx"              "play"             "podcast"          "point"
##  [85] "points"           "preview"          "pts"              "raptors"
##  [89] "rebs"             "recap"            "return"           "road"
##  [93] "rockets"          "sac"              "sacramentokings"
"sacramentoproud"
##  [97] "scores"           "season"           "see"              "shot"
## [101] "simmons"          "sports"           "spurs"            "star"
## [105] "state"            "steph"            "stephen"          "suns"
## [109] "swj"              "team"             "temple"           "thanks"
## [113] "tonight"          "two"              "usa"              "wall"
## [117] "warriors"         "washington"       "watson"           "week"
## [121] "win"              "wire"             "wizards"          "worldseries"

#sort the term by frequency and plot terms of frequency over 3 in a word
cloud
nbacorp.terdoc.matrix <- as.matrix(nbacorp.terdoc)
nbaterm.freqbydoc <- sort(rowSums(nbacorp.terdoc.matrix), decreasing = T,
na.last = NA)

wordcloud(names(nbaterm.freqbydoc), nbaterm.freqbydoc, min.freq = 3,
max.words = 100,
          textStemming = FALSE, colors=brewer.pal(8, "Dark2"))
```



```
# subset the terms with a frequency over 5 and generate a barplot
V.minfreq6 <- rowSums(nbacorp.terdoc.matrix) > 5
nbaterm.minfreq6 <- sort(rowSums(nbacorp.terdoc.matrix)[V.minfreq6],
decreasing = T)
term.barplot <- barplot(nbaterm.minfreq6, horiz = F, col =
rainbow(length(nbaterm.minfreq6)))

legend(20, 20.5, legend = names(nbaterm.minfreq6),fill =
rainbow(length(nbaterm.minfreq6)),
      cex = 0.75, ncol = 3, x.intersp = 0.2, y.intersp = 0.7, text.width = 9,
bty = "n")


# 5. Identify the top three pairs of tweets and the most frequently used
terms among these pairs
library(reshape2)
library(Matrix)
```



```
library("lsa")

#create consine similarity matrix and check result
nbacorp.cosi <- as.matrix(cosine(nbacorp.terdoc.matrix))
nbacorp.cosi[1:9, 1:9]

##    1          2 3         5 6         7         8         9         10
## 1  1 0.00000000 0 0.0000000 0 0.0000000 0.0000000 0.0000000 0.00000000
## 2  0 1.00000000 0 0.0000000 0 0.0000000 0.0000000 0.0000000 0.09805807
## 3  0 0.00000000 1 0.0000000 0 0.0000000 0.0000000 0.0000000 0.00000000
## 5  0 0.00000000 0 1.0000000 0 0.1005038 0.1348400 0.0000000 0.08362420
## 6  0 0.00000000 0 0.0000000 1 0.0000000 0.0000000 0.0000000 0.00000000
## 7  0 0.00000000 0 0.1005038 0 1.0000000 0.0000000 0.0000000 0.00000000
## 8  0 0.00000000 0 0.1348400 0 0.0000000 1.0000000 0.1581139 0.12403473
## 9  0 0.00000000 0 0.0000000 0 0.0000000 0.1581139 1.0000000 0.00000000
## 10 0 0.09805807 0 0.0836242 0 0.0000000 0.1240347 0.0000000 1.00000000

#replace all the diagonal value from 1 to NA
diag.replace <- function(x){
  for (i in 1: nrow(x)){
    if (x[i, i] == 1 | x[i, i] == 0)
    { x[i,i] <- NA }
  }
  return(x)
}

nbacorp.cosmod <- diag.replace(nbacorp.cosi)
nbacorp.cosmod[1:6, 1:6]
```

```
##    1  2  3         5  6         7
## 1 NA  0  0 0.0000000  0 0.0000000
## 2  0 NA  0 0.0000000  0 0.0000000
## 3  0  0 NA 0.0000000  0 0.0000000
## 5  0  0  0        NA  0 0.1005038
## 6  0  0  0 0.0000000 NA 0.0000000
## 7  0  0  0 0.1005038  0        NA
```

```
#convert the sparse matrix into a molten data frame and sort it based on the
cosine value
nbacorp.cosmolten <- melt(nbacorp.cosmod, na.rm = T, c("m.row.doc",
"m.col.doc"))
nbacorp.cosmolten <- nbacorp.cosmolten[order(nbacorp.cosmolten$value,
decreasing = T),]
nbacorp.cosmolten[1:25,]
```

```
##       m.row.doc m.col.doc value
## 2628         16        15     1
## 2828         15        16     1
## 8098         61        42     1
## 8496         57        44     1
## 10696        44        57     1
## 10933        82        58     1
## 10942        92        58     1
## 11498        42        61     1
## 15645       172        81     1
## 15733        58        82     1
## 15766        92        82     1
## 16364        87        86     1
## 16564        86        87     1
## 17542        58        92     1
## 17566        82        92     1
## 18990       100        99     1
## 19190        99       100     1
## 21617       114       112     1
## 22017       112       114     1
## 24040       125       124     1
## 24240       124       125     1
## 28304       168       145     1
## 32904       145       168     1
## 33645        81       172     1
## 36768       191       188     1
```

```
#inspect tweet pairs with cosine similarity of 1 --> These tweets seems to be
repost
inspect(input.nbacorp[[15]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 30
##
##   c j watson  girlfriend  wags

inspect(input.nbacorp[[16]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 30
##
##   c j watson  girlfriend  wags

inspect(input.nbacorp[[42]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 42
##
##  basket toops rs rock star michael jordan

inspect(input.nbacorp[[61]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 42
##
##  basket toops rs rock star michael jordan

inspect(input.nbacorp[[44]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 31
##
##   shot  mike james gleaguealum

inspect(input.nbacorp[[57]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 31
##
##    shot  mike james gleaguealum

#After empty entries removed, the doc index numbers in the matrix shift from
doc entry numbers
```

```
typeof(row.names(nbacorp.docterm)) # doc entry number in the doc-term matrix
are characters

## [1] "character"

# Therefore, the character value instead of numeric values can correctly
index doc entries
inspect(nbacorp.docterm[c("15","16","61","42","57","44"),])

## <<DocumentTermMatrix (documents: 6, terms: 772)>>
## Non-/sparse entries: 26/4606
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs basket girlfriend gleaguealum james jordan michael mike rock shot
star
##   15      0          1           0     0      0       0    0    0    0
0
##   16      0          1           0     0      0       0    0    0    0
0
##   42      1          0           0     0      1       1    0    1    0
1
##   44      0          0           1     1      0       0    1    0    1
0
##   57      0          0           1     1      0       0    1    0    1
0
##   61      1          0           0     0      1       1    0    1    0
1

#coerce the doc-term matrix to R matrix
nbacorp.docterm.matrix <- as.matrix(nbacorp.docterm)

#subset the matrix with reposted tweets
nbacorp.repost.matrix <-
nbacorp.docterm.matrix[c("15","16","61","42","57","44"),]
nbacorp.repost.matrix[, 1:20] #although subsetted, matrix inherited every
term from all the tweets

##     Terms
## Docs aaron absolutely accident account across action actions actually
addiction
##   15     0          0        0       0      0      0       0        0
0
##   16     0          0        0       0      0      0       0        0
0
##   61     0          0        0       0      0      0       0        0
0
##   42     0          0        0       0      0      0       0        0
0
```

```
##   57     0          0        0       0      0      0       0        0
0
##   44     0          0        0       0      0      0       0        0
0
##      Terms
## Docs addition airmax aldridge alex algorithm alive alley ally already
##   15        0      0        0    0         0     0     0    0       0
##   16        0      0        0    0         0     0     0    0       0
##   61        0      0        0    0         0     0     0    0       0
##   42        0      0        0    0         0     0     0    0       0
##   57        0      0        0    0         0     0     0    0       0
##   44        0      0        0    0         0     0     0    0       0
##      Terms
## Docs amicohoops amp
##   15          0   0
##   16          0   0
##   61          0   0
##   42          0   0
##   57          0   0
##   44          0   0

#which() & apply() index the terms that are only in the repost docs/tweets
term.inrepost <- names(which(apply(nbacorp.repost.matrix, 2, sum) > 0))

#The top 10 most used terms from all the tweets
top10.term <- names(nbaterm.freqbydoc[1:10])

#write a function to check whether any of the top 10 terms included in
subsetted similar tweets
identical.term <- function(x, y){
  for (i in 1: length(x)){
    if(length(grep(x[i], y)) > 0)
    {print(c(x[i],grep(x[i], y, value = T)))}
  }
}

top10.term

##  [1] "game"       "kings"      "nfl"        "warriors"   "suns"
##  [6] "wizards"    "basketball" "curry"      "nhl"        "win"

term.inrepost

##  [1] "basket"      "girlfriend"  "gleaguealum" "james"       "jordan"
##  [6] "michael"     "mike"        "rock"        "shot"        "star"
## [11] "toops"       "wags"        "watson"

identical.term(term.inrepost, top10.term)

## [1] "basket"     "basketball"
```

```
##inpect tweet pairs with cosine similarity less than 1
inspect(input.nbacorp[[182]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 66
##
## john wall guides  washwizards  road win  points  assists dcfamily

inspect(input.nbacorp[[200]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 77
##
## john wall guides  washwizards  road win  points  assists dcfamily
basketball

inspect(input.nbacorp[[50]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 42
##
##   shot  mike james gleaguealum  basketball

inspect(input.nbacorp[[44]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 31
##
##   shot  mike james gleaguealum

inspect(input.nbacorp[[138]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 95
##
## nowplaying live  periscope nfl  mlb amp indie music nfl  worldseries
collegefootball edm hiphop

inspect(input.nbacorp[[131]])

## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 99
##
## nowplaying live  periscope sports amp music unite nfl  worldseries
collegefootball edm hiphop indie
```

```
#subset the matrix with docs of cosine similarity
nbacorp.similar.matrix <- nbacorp.docterm.matrix[c("182", "200","50", "44",
"138", "131"), ]
nbacorp.similar.matrix[, 1:20] #although subsetted, matrix inherited every
term from all the tweets
```

```
##      Terms
## Docs  aaron absolutely accident account across action actions actually
##   182     0          0        0       0      0      0       0        0
##   200     0          0        0       0      0      0       0        0
##   50      0          0        0       0      0      0       0        0
##   44      0          0        0       0      0      0       0        0
##   138     0          0        0       0      0      0       0        0
##   131     0          0        0       0      0      0       0        0
##      Terms
## Docs  addiction addition airmax aldridge alex algorithm alive alley ally
##   182         0        0      0        0    0         0     0     0    0
##   200         0        0      0        0    0         0     0     0    0
##   50          0        0      0        0    0         0     0     0    0
##   44          0        0      0        0    0         0     0     0    0
##   138         0        0      0        0    0         0     0     0    0
##   131         0        0      0        0    0         0     0     0    0
##      Terms
## Docs  already amicohoops amp
##   182       0          0   0
##   200       0          0   0
##   50        0          0   0
##   44        0          0   0
##   138       0          0   1
##   131       0          0   1
```

```
#which() & apply() index the terms that are only in the similar docs/tweets
term.insimilar <- names(which(apply(nbacorp.similar.matrix, 2, sum) > 0))

#Check whether any of the top 10 terms are included in the similar tweets
top10.term
```

```
##  [1] "game"       "kings"      "nfl"        "warriors"   "suns"      
##  [6] "wizards"    "basketball" "curry"      "nhl"        "win"
```

```
term.insimilar
```

```
##  [1] "amp"             "assists"         "basketball"     
"collegefootball"
##  [5] "dcfamily"        "edm"             "gleaguealum"     "guides"         
##  [9] "hiphop"          "indie"           "james"           "john"           
## [13] "live"            "mike"            "mlb"             "music"          
## [17] "nfl"             "nowplaying"      "periscope"       "points"         
## [21] "road"            "shot"            "sports"          "unite"          
## [25] "wall"            "washwizards"     "win"             "worldseries"
```

```
identical.term(term.insimilar, top10.term)

## [1] "basketball" "basketball"
## [1] "nfl" "nfl"
## [1] "win" "win"

# 6. Identify terms with the highest weighted tf-idf among the top three
pairs of tweets
#calculate the tfidf of the document-term matrix created during # 4
nbacorp.dttfidf <- weightTfIdf(nbacorp.docterm)
inspect(nbacorp.dttfidf[1:6,])

## <<DocumentTermMatrix (documents: 6, terms: 772)>>
## Non-/sparse entries: 52/4580
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency - inverse document frequency
(normalized) (tf-idf)
## Sample             :
##     Terms
## Docs   brooks   dillon     fail   fanuel   nobody   paying      pts
survived
##    1 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.059572
0.000000
##    2 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000
##    3 0.000000 0.000000 0.000000 0.000000 1.912763 1.912763 0.000000
0.000000
##    5 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000
##    6 1.275175 1.275175 1.275175 1.275175 0.000000 0.000000 0.000000
1.275175
##    7 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000
##     Terms
## Docs      tix      won
##    1 0.000000 0.000000
##    2 0.000000 0.000000
##    3 1.912763 1.912763
##    5 0.000000 0.000000
##    6 0.000000 0.000000
##    7 0.000000 0.000000

#convert the document-term matrix to numeric matrix and calculate a total
tfidf of each document
nbacorp.dttfidf.matrix <- as.matrix(nbacorp.dttfidf)
nbadoc.countfidf <- sort(rowSums(nbacorp.dttfidf.matrix), decreasing = T)
nbadoc.countfidf[1:20]

##      195        3       39       49       97      141       17      146
## 7.651052 7.651052 7.651052 7.651052 7.651052 7.651052 7.651052 7.651052
##      166      207      179      164       29       38       41       46
## 7.651052 7.451052 7.365337 7.317718 7.251052 7.251052 7.251052 7.239892
##       23      117      123      143
## 7.222480 7.220225 7.208368 7.175132
```

```
#write a function to find the identical sum of tfidf of each document/tweets
same.tweets <- function(x) {
  temp.x <- x
  names(temp.x) <- NULL
  for(i in 1:length(temp.x))
  {
    if(identical(temp.x[i], temp.x[i+1]) == T)
    {print(x[c(i,i+1)])}
  }
}

same.tweets(nbadoc.countfidf)
```

```
##        3       39
## 7.651052 7.651052
##       39       49
## 7.651052 7.651052
##       49       97
## 7.651052 7.651052
##       97      141
## 7.651052 7.651052
##       17      146
## 7.651052 7.651052
##      146      166
## 7.651052 7.651052
##       29       38
## 7.251052 7.251052
##       38       41
## 7.251052 7.251052
##      143      148
## 7.175132 7.175132
##        2      133
## 7.129811 7.129811
##      133      161
## 7.129811 7.129811
##      109      163
## 7.110811 7.110811
##      110      185
## 7.051052 7.051052
##       32       33
## 6.901052 6.901052
##      59      75
## 6.85857 6.85857
##       24       60
## 6.772588 6.772588
##       25      134
## 6.727012 6.727012
##      145      168
## 6.651052 6.651052
##      168      190
## 6.651052 6.651052
##       52      120
## 6.525237 6.525237
##       80      189
## 6.504811 6.504811
##       15       16
## 6.456064 6.456064
##       16       51
## 6.456064 6.456064
##      86      87
## 6.43073 6.43073
##     137     188
## 6.35857 6.35857
##     188     191
## 6.35857 6.35857
##       74       98
## 6.264064 6.264064
##       99      100
## 6.151052 6.151052
##      100      154
## 6.151052 6.151052
##       42       61
## 6.122731 6.122731
##      81     172
## 6.04053 6.04053
##      112      114
## 5.929735 5.929735
##      152      170
## 5.572588 5.572588
##      197      198
## 5.447529 5.447529
##      124      125
## 5.370881 5.370881
##       58       82
## 4.996012 4.996012
##       82       92
## 4.996012 4.996012
##       44       57
## 4.947606 4.947606
```

```
# Inspect the content of the highest score of tweets
inspect(input.nbacorp[[3]])
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 25
##
## nobody   paying   won tix
```

```r
inspect(input.nbacorp[[39]])
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 24
##
##    overweight people
```

```r
inspect(input.nbacorp[[49]])
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 19
##
## meanwhile  phoenix
```

```r
inspect(input.nbacorp[[29]])
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 39
##
## remember   players spoke  mind  twitter
```

```r
inspect(input.nbacorp[[38]])
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 42
##
## wtf    gatorade tonight everybody wilding
```

```r
inspect(input.nbacorp[[143]])
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 68
##
##    say  much  hate   season already injuries irvingdiva coachesfired
```

```r
inspect(input.nbacorp[[148]])
```

```
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 57
##
## process servers  back  ready  hand  child support orders
```

```
#calculate  tfidf of all the terms and convert results to R matrix
nbacorp.tertfidf.matrix <- as.matrix(weightTfIdf(nbacorp.terdoc, normalize =
T))
#subset the matrix with 3 pairs of tweets having the highest tfidf sum
top3tweet.tfidf.matrix <- nbacorp.tertfidf.matrix[, c("3", "39", "29", "38",
"143", "148")]
top3tweet.tfidf.matrix[1:10,]#terms used in other tweets were inherited in
the subsetted matrix

##             Docs
## Terms        3 39 29 38 143 148
##   aaron      0  0  0  0   0   0
##   absolutely 0  0  0  0   0   0
##   accident   0  0  0  0   0   0
##   account    0  0  0  0   0   0
##   across     0  0  0  0   0   0
##   action     0  0  0  0   0   0
##   actions    0  0  0  0   0   0
##   actually   0  0  0  0   0   0
##   addiction  0  0  0  0   0   0
##   addition   0  0  0  0   0   0

term.top3tweet <- names(which(apply(top3tweet.tfidf.matrix, 1, sum) > 0))

#Harvest the top 10 terms of highest tfidf values
top10.tfidfterm <- sort(rowSums(nbacorp.tertfidf.matrix), decreasing =
T)[1:10]
top10.tfidfterm <- names(top10.tfidfterm)

#check the overlapped term with identical.term function
term.top3tweet

##  [1] "already"      "back"         "child"        "coachesfired"
"everybody"
##  [6] "gatorade"     "hand"         "hate"         "injuries"
"irvingdiva"
## [11] "mind"         "much"         "nobody"       "orders"
"overweight"
## [16] "paying"       "people"       "players"      "process"      "ready"
## [21] "remember"     "say"          "season"       "servers"      "spoke"
## [26] "support"      "tix"          "tonight"      "twitter"      "wilding"
## [31] "won"          "wtf"

top10.tfidfterm

##  [1] "suns"       "game"       "basketball" "kings"      "win"
##  [6] "chriss"     "nfl"        "warriors"   "dfs"        "wizards"

identical.term(term.top3tweet, top10.tfidfterm)
```

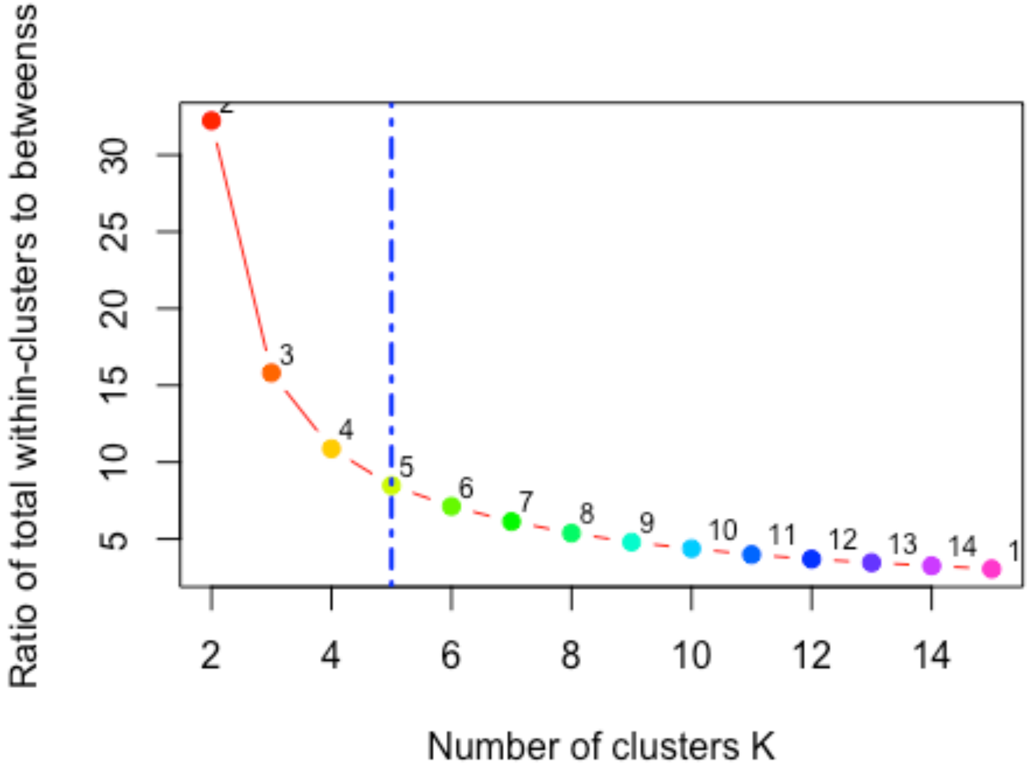
```
# 7. Determine the optimal numbers of clusters for the tweets
# Compute kmean and plot wss from k = 1 to k = 20.
set.seed(2345)
k.max <- 15
tot.wss <- sapply(2:k.max, simplify = T,
      function(k){kmeans(nbacorp.docterm.matrix, k, nstart = 50, iter.max =
100)$tot.withinss})

bet.ss <- sapply(2:k.max, simplify = T,
      function(k){kmeans(nbacorp.docterm.matrix, k, nstart = 50, iter.max =
100)$betweenss})
tot.wss
```

```
##  [1] 1466.000 1421.599 1384.195 1351.455 1324.348 1298.576 1271.605
1252.922
##  [9] 1229.817 1209.329 1189.088 1170.138 1155.059 1135.621
```

```
bet.ss
```

```
##  [1]  45.48259  89.88310 127.28802 159.40431 185.85846 211.55301 236.31668
##  [8] 261.37400 281.27125 303.25596 322.36029 338.26111 354.67941 372.48345
```

```
plot(2:k.max, tot.wss/bet.ss,
     type = "b", pch = 19, frame = T, lwd = 1, col= rainbow(k.max),
     xlab = "Number of clusters K", ylab = "Ratio of total within-clusters to
betweenss")
text(2:k.max, tot.wss/bet.ss, labels = 2:k.max, adj = c(-0.5, -0.5), cex =
0.75)
abline(v = 5, lwd = 2, lty = 4, col = "blue")
```



```
# 8. Identify the groups of tweets having similar characteristics
#pick up k-custer at 6
set.seed(2345)
nbacorp.cluster <- kmeans(nbacorp.docterm.matrix, 5, nstart = 30, iter.max =
50)
nbacorp.cluster$cluster[1:25]

##  1  2  3  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  1  1  1  1  1  3  1  1  5  1  1  5  5  1  1  1  1  4  1  1  1  1  1  1  3

#use sapply to extract the text from corpus
inputcorp.text <- t(data.frame(sapply(input.nbacorp.noemp, "[", "content")))

#index out empty entries of corpus from original tweets and extracted text,
#and combine the extract text, original tweet text and cluster vector
row.total.dataframe <- names(which(row.total > 0))
tweet.txtclust <- data.frame(tweet.tb[c(row.total.dataframe),]$text,
                        as.character(inputcorp.text),
                        nbacorp.cluster$cluster)

#change the column names and organize the table by clusters
names(tweet.txtclust) <- c("orginal tweets", "cleaned tweets", "K-clusters")
tweet.txtclust <- tweet.txtclust[order(tweet.txtclust$`K-clusters`,
decreasing = F),]

#subset cleaned tweet text data by clusters.
tweet.txtK1 <- tweet.txtclust[tweet.txtclust$`K-clusters` == 1, ]$`cleaned
tweets`
tweet.txtK2 <- tweet.txtclust[tweet.txtclust$`K-clusters` == 2, ]$`cleaned
tweets`
tweet.txtK3 <- tweet.txtclust[tweet.txtclust$`K-clusters` == 3, ]$`cleaned
tweets`
tweet.txtK4 <- tweet.txtclust[tweet.txtclust$`K-clusters` == 4, ]$`cleaned
tweets`
tweet.txtK5 <- tweet.txtclust[tweet.txtclust$`K-clusters` == 5, ]$`cleaned
tweets`

as.character(tweet.txtK1)[1:40]

##  [1] "houston rockets memphis grizzlies  eric gordon pts harden pts asts
marc gasol pts taps ennis rebs"
##  [2] " re playing name  ny player     happened gold  mlb yankees knicks"
##  [3] "nobody   paying   won tix"
##  [4] " trailblazers game worn portland trail blazers  summer league jersey
terrel harris xl"
##  [5] "survived  dillon brooks fail  dfs fanuel"
##  [6] "torantoraptors      game  follow basketball raptors"
##  [7] "watching  now   brutal bc    guys basketball iq  absolutely
horrendous smh"
##  [8] "   best back court    wall beal"
##  [9] "well   looking like   win season nbakings "
## [10] "  c j watson  girlfriend  wags"
## [11] "  c j watson  girlfriend  wags"
## [12] "  bizarre  sex scandals"
## [13] "  love   follow basketball"
## [14] " trade rumors la lakers  like  trade luol deng dnp cd treatment
continue  lakers"
## [15] "giannis antetokounmpo vs hornets pts ₒreb ₒast average pts reb ast
stl  fearthedeer"
## [16] "   t wanna  disrespected  t turn   amp get back  defense warriors"
## [17] " sorry  mavs   beat  pts  blame  rookie   dun"
## [18] "g anteto yes gon  mvp  season "
## [19] " welcome back nikola jokic  amp nikola jokic den"
## [20] "michael jordan   graphicdesign  basketball posterdesign"
## [21] " wizards   better record    teams including warriors  cavaliers
time   alive dcfamily nbatwitter"
## [22] "remember   players spoke  mind  twitter"
## [23] "     game  follow basketball bostonceltics"
## [24] "steals  finishes dcfamily milehighbasketball  nbapanel"
## [25] "hurry   blow  lead go swj "
## [26] "   changed   swj yokedjokic  unihistory"
## [27] "good half   fast   open court   nuts  see live swj wasvsden "
## [28] "  sick outlet pass swj yokedjokic wasvsden "
## [29] "nasty fam  nbaisback jordanbell warriors dubnation goldenstate
bayarea gswin gswvsdal mavericks"
## [30] "wtf    gatorade tonight everybody wilding "
## [31] "   overweight people    "
## [32] "trust  process  ers simmons roty joel markelle"
## [33] "shit  players tweeted  twitter blew "
## [34] " basket toops rs rock star michael jordan "
## [35] "    undefeated team   east   great team wizards "
## [36] "  shot  mike james gleaguealum "
## [37] "new promo hidden hours directed  dari arrington ے ے ے basketball
lakeshow"
## [38] "go   win "
## [39] "  shot  mike james gleaguealum couponsgod  sports news trending
fanclub "
## [40] "meanwhile  phoenix "
```

```
as.character(tweet.txtK2)[1:5]
```

```
## [1] "garrett temple makes foot pointer garrett temple makes foot point
jumper garrett temple makes foot point jumper  kings"
## [2] NA
## [3] NA
## [4] NA
## [5] NA
```

```
as.character(tweet.txtK3)[1:15]
```

```
##  [1] "steph curry shared  heartwarming moment  devin harris nephew
warriors "
##  [2] "repost  stephen curry consoles devin harris nephew  lost  father
car accident l"
##  [3] "video stephen curry consoles grieving nephew  dallas mavericks guard
devin harris sacramentokings kings "
##  [4] "stephen curry golden state warriors guard fined  throwing mouthpiece
sacramentokings kings "
##  [5] "golden state warriors blow  dallas mavericks might   worst team ever
golden state"
##  [6] "warriors stephen curry andre iguodala fined  actions  memphis "
##  [7] "usa  dallas mavericks golden state warriors "
##  [8] " golden state warriors used  second half surge behind stephen curry
kevin durant  rout  mavericks"
##  [9] "golden state warriors star stephen curry consoles grieving nephew
dallas mavericks guard devin harris "
## [10] NA
## [11] NA
## [12] NA
## [13] NA
## [14] NA
## [15] NA
```

```
as.character(tweet.txtK4)[1:20]
```

```
##  [1] " heart  hustle  inspiring   point shot  looking nice  appreciate
sac vet kings "
##  [2] " buckle   ve got  two point game suns sacramentokings left  play
leaguepassalert"
##  [3] "  sacramento kings go   win  tie   road  kings phoenixsuns"
##  [4] "gasol leads grizzlies  win  rockets sacramentokings kings "
##  [5] "aldridge murray power spurs past raptors sacramentokings kings "
##  [6] "game recap spurs raptors sacramentokings kings "
##  [7] "monday   suns fire watson banish bledsoe sacramentokings kings "
##  [8] "mike james hits  clutch   suns kings   chance  tie  win  scores suns
kingsupdate  sacramentoproud"
##  [9] "balling right now vs  suns  secs left   th snglv  kings"
## [10] "buckle   ve got  two point game suns sacramentokings left  play
leaguepassalert "
## [11] " sacramento kings game  driving  insane   comeback   kings let
finish   "
## [12] " freakin game man kings sacramentoproud "
## [13] "ok    game js  js  kings suns sunsvskings"
## [14] "buckle   ve got  two point game suns sacramentokings left  play
leaguepassalert "
## [15] "sac kings vs suns game  going    wire "
## [16] "  love  kings team fox  bogdanovic lt kings "
## [17] NA
## [18] NA
## [19] NA
## [20] NA
```

```
as.character(tweet.txtK5)[1:20]
```

```
##  [1] "sure nfl  boring game except  cowboys  sports analytics hardly
statistics bring  mlb  data"
##  [2] "nowplaying live  periscope nfl  worldseries dtongradio newmusic"
##  [3] "nhl  collegefootball nfl algorithm units yet documented"
##  [4] "nowplaying live  periscope sports amp music unite"
##  [5] "gymrant  myth needs  end conjugate conjugatemethod bjj jiujitsu nogi
mma judo wrestling nhl nfl"
##  [6] " rules errors  wolves last second win  æ sports nfl  mlb ncaaf nhl"
##  [7] "great breaks  tickets prices nfl  nhl"
##  [8] "nowplaying live  periscope nfl  worldseries musicmonday np rt"
##  [9] "nowplaying live  periscope sports amp music unite nfl  worldseries
collegefootball edm hiphop indie"
## [10] "tuesday   vip mlb nhl rc plays   nhl nhl incl  best bet run tgtbfc"
## [11] "nowplaying live  periscope nfl  mlb amp indie music nfl  worldseries
collegefootball edm hiphop"
## [12] "nowplaying live  periscope nfl  edm musicnmonday np rt"
## [13] "nowplaying live  periscope nfl  worldseries edm trance"
## [14] "  dominant dodgers  actually world series underdogs  æ sports nfl
mlb ncaaf nhl"
## [15] "every day  gameday  fantasydraft dailyfantasy nfl mlb  nhl pga"
## [16] "sporgy itunes podcast sports humor mlb nfl  nhl detroit"
## [17] "sporgy itunes podcast sports humor mlb nfl  nhl detroit"
## [18] NA
## [19] NA
## [20] NA
```

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
```