**WSMA Group Assignment**

**Objective: Brand perception analysis (Social media analytics - Text mining)**

**Group Members**

1. Akshay Surve
2. Amay Saxena
3. Aditya Godse

**Table of Contents**

[1] Topic Selection 3](#_Toc36410425)

[2] Exploratory Data Analysis 3](#_Toc36410426)

[ EDA through Tableau 4](#_Toc36410427)

[ EDA through R 6](#_Toc36410428)

[3] Data Cleaning 7](#_Toc36410429)

[1. Remove character string between < > 8](#_Toc36410430)

[2. Converting all text to Lowercase 8](#_Toc36410431)

[3. Remove the Links (URLs) 8](#_Toc36410432)

[4. Removing usernames from tweets 8](#_Toc36410433)

[5. Stemming words in the corpus 9](#_Toc36410434)

[6. Remove Single Letter Words 9](#_Toc36410435)

[7. Replace words with the proper ones 9](#_Toc36410436)

[8. Remove anything except the English language and space 10](#_Toc36410437)

[9. Removing Extra Whitespaces 10](#_Toc36410438)

[10. Removing Stop words 10](#_Toc36410439)

[4] Analyzing Text frequency 11](#_Toc36410440)

[1. Creating term document matrix for text frequency analysis 11](#_Toc36410441)

[2. Find the most frequently used terms in our corpus 11](#_Toc36410442)

[3. Word Frequency Analysis (Keeping word frequency low to high) 11](#_Toc36410443)

[4. Plotting word frequency for different frequency of word 13](#_Toc36410444)

[5] Word Cloud 15](#_Toc36410445)

[1. Word cloud for Overall corpus 15](#_Toc36410446)

[2. Creating Separate word cloud for Positive and Negative Sentiments. 15](#_Toc36410447)

[ Positive Words 16](#_Toc36410448)

[ Negative Words 18](#_Toc36410449)

[6] Sentiment Analysis, polarity - positive or negative 19](#_Toc36410450)

[ Sentiment Analysis: Plot by date - understanding cumulative sentiment score movement 21](#_Toc36410451)

[ Sentiment Analysis: Plot by date - Understanding average sentiment score movement 22](#_Toc36410452)

[7] Correlation chart of top keywords, including Word Association 23](#_Toc36410453)

[ Finding association of specific keyword with other words in tweets 23](#_Toc36410454)

[ Topic Modelling to identify latent/hidden topics using LDA technique 25](#_Toc36410455)

**Answer**

# **1] Topic Selection**

As Indian summer comes, India waits for IPL to start as it is a month full of excitement, cricket and fun. But as we know this year Coronavirus has hampered a lot of sports. So, to check what people are thinking about IPL, we are doing IPL tweets analysis.

# **2] Exploratory Data Analysis**

Loading required packages for text mining,

library(SnowballC) – For Stemming Operation

library(tm) – For Text Mining

library(ggplot2) – For Data Visualization

library(wordcloud) - To create word clouds, visualize differences and similarity between texts

library(topicmodels) - To fit topic models based on data structures from the text mining package tm

library(data.table) – For Fast aggregation of data

library(stringi) – Provide platform-independent functions

library(syuzhet) – To understand emotional valence in tweets

library(dplyr) – For data frame operations

library(plyr) – data splitting and combining

library(grid) – To produce graphical output directly

library(DT) - Provides an R interface to the JavaScript library DataTables

library(stringr) - Provide a cohesive set of functions

library(RSentiment) - Analyze Sentiment of English Sentences

library(readr) - to provide a fast and friendly way to read rectangular data

library(RColorBrewer) -

## **EDA through Tableau**

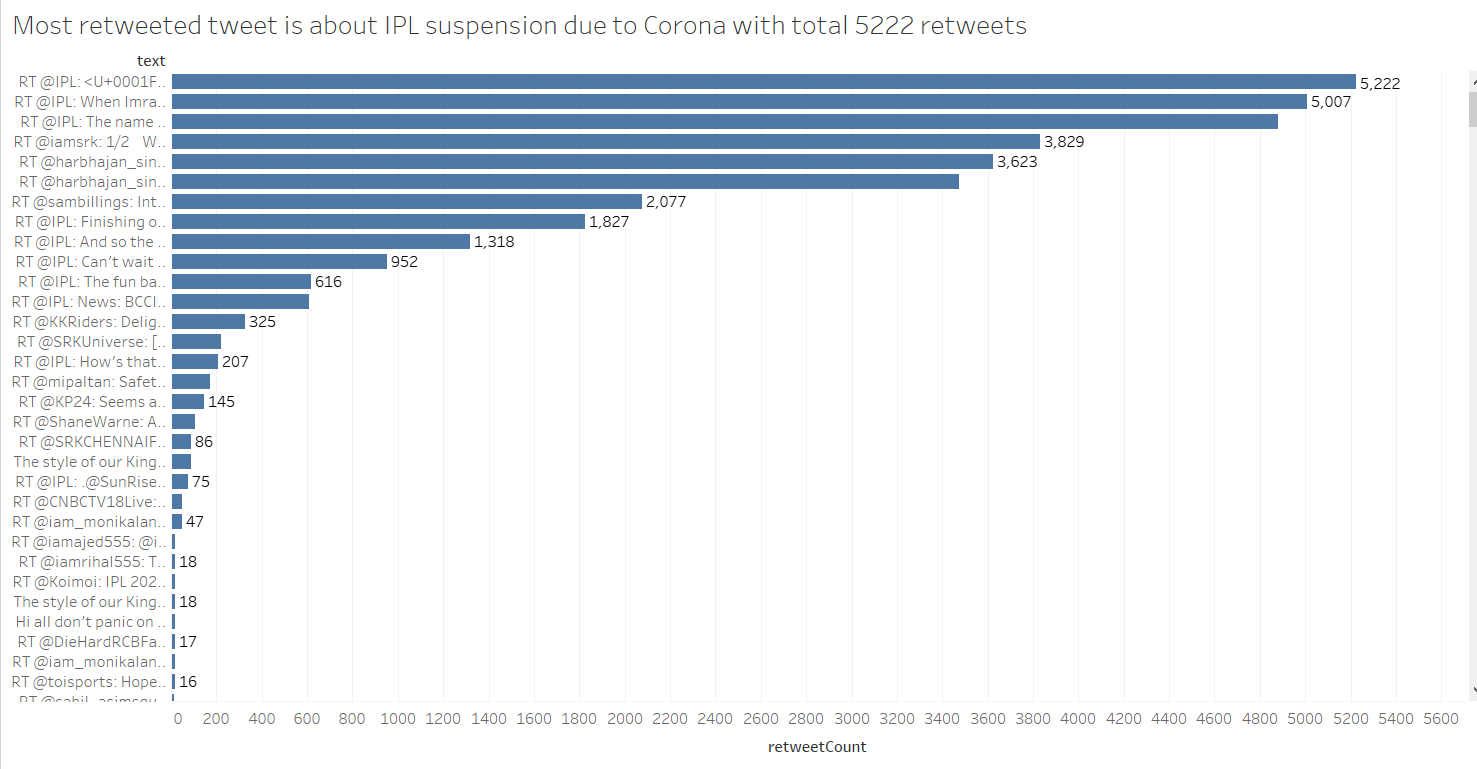


Fig1.Retweetcount vs. tweet text

Tweet: RT @IPL: <U+0001F6A8>Announcement<U+0001F6A8>: #VIVOIPL suspended till 15th April 2020 as a precautionary measure against the ongoing Novel Corona Virus (COVID-1…

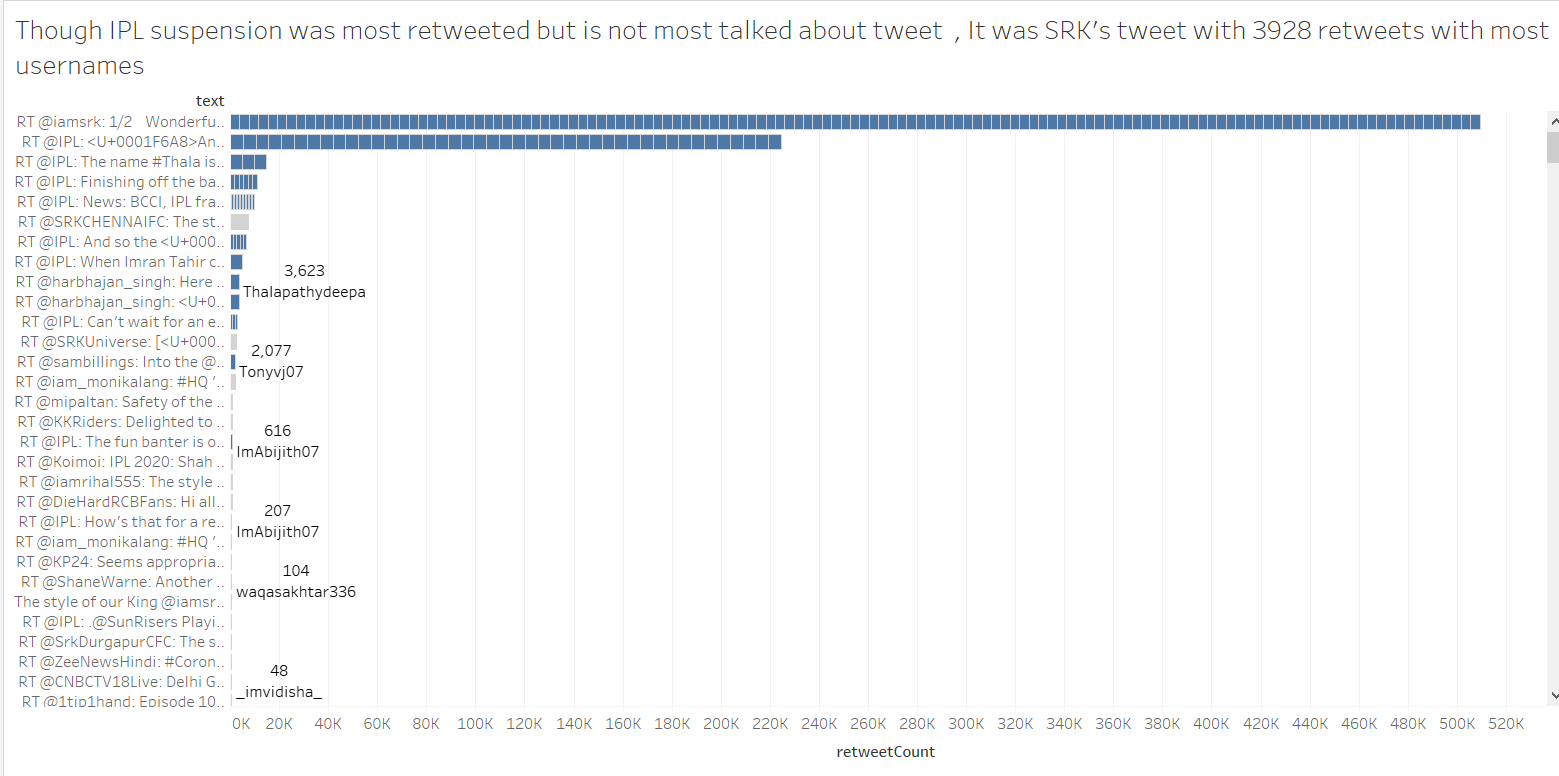


Fig2. Retweetcount vs. tweet text with usernames

Tweet: RT @iamsrk: 1/2 Wonderful to meet all the Franchise owners ‘off the field’ so to say. The meeting by @Bcci and @ipl was to reiterate wh…

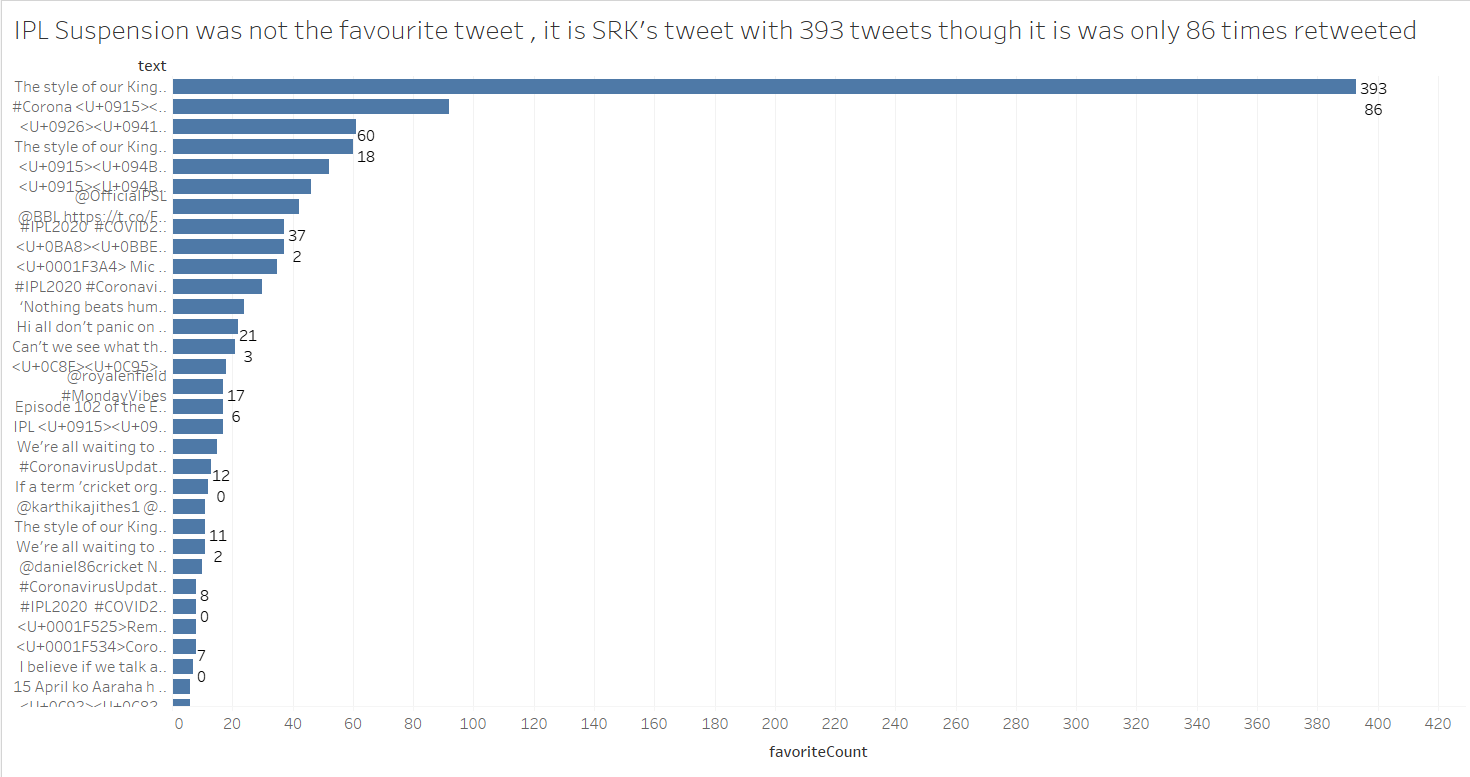


Fig3. favoritecount vs. tweet text

Tweet: RT @SRKCHENNAIFC: The style of our King @iamsrk is unmatchable <U+0001F60D><U+0001F60E> No one can play the game of style more efficiently than our Baadshah SRK…

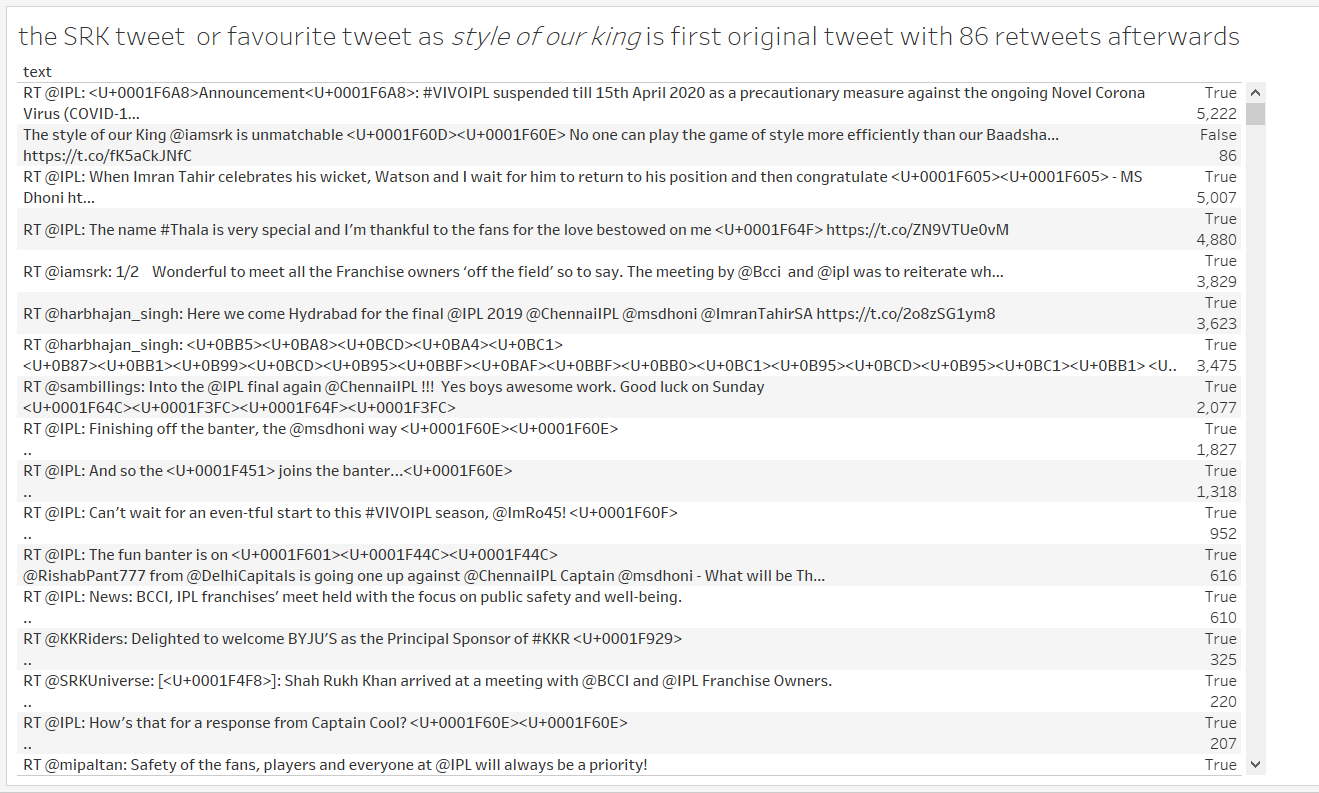


Fig4. Isretweet vs. tweet text

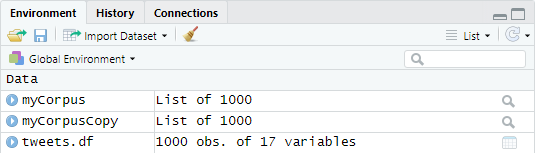
Tweet : RT @SRKCHENNAIFC: The style of our King @iamsrk is unmatchable <U+0001F60D><U+0001F60E> No one can play the game of style more efficiently than our Baadshah SRK…

## **EDA through R**

Read Tweeter data,

setwd("C:/Users/DELL/Desktop/Akshay/Group Assignments/Group Assignment WSMA")

tweets.df <- read.csv("WSMA.csv")



Further designate the correct format of date,

tweets.df$created <- as.Date(tweets.df$created, format= "%d-%m-%y")

tweets.df$text <- as.character(tweets.df$text)

structure of current data,

str(tweets.df)

'data.frame': 1000 obs. of 17 variables:

$ X : int 1 2 3 4 5 6 7 8 9 10 ...

$ text : chr "RT @iamsrk: 1/2 Wonderful to meet all the Franchise owners ‘off the field’ so to say. The meeting by @Bcci "| \_\_truncated\_\_ "@IPLCricket @IPL @RCBTweets" "RT @1711Pooja: <U+0001F525>Remember the moment<U+0001F525>\n@ABdeVilliers17 @DaleSteyn62 @RCBTweets @IPL @AbdeF"| \_\_truncated\_\_ "RT @SRKCHENNAIFC: The style of our King @iamsrk is unmatchable <U+0001F60D><U+0001F60E> No one can play the gam"| \_\_truncated\_\_ ...

$ favorited : logi FALSE FALSE FALSE FALSE FALSE FALSE ...

$ favoriteCount: int 0 0 0 0 0 0 2 0 11 0 ...

$ replyToSN : Factor w/ 108 levels "1tip1hand","AamAadmiParty",..: NA 46 NA NA 98 NA NA NA NA NA ...

$ created : Date, format: NA NA NA ...

$ truncated : logi FALSE FALSE FALSE FALSE FALSE FALSE ...

$ replyToSID : num NA 1.24e+18 NA NA 1.24e+18 ...

$ id : num 1.24e+18 1.24e+18 1.24e+18 1.24e+18 1.24e+18 ...

$ replyToUID : num NA 1.13e+18 NA NA 1.35e+08 ...

$ statusSource : Factor w/ 13 levels "<a href=\"http://kaptaankohli.com\" rel=\"nofollow\">The Gujarat Lions FC</a>",..: 4 4 4 4 4 4 4 11 7 10 ...

$ screenName : Factor w/ 754 levels "\_chiggsworld\_",..: 318 576 700 474 211 211 10 145 712 428 ...

$ retweetCount : int 3829 0 4 86 0 2 0 2 2 0 ...

$ isRetweet : logi TRUE FALSE TRUE TRUE FALSE TRUE ...

$ retweeted : logi FALSE FALSE FALSE FALSE FALSE FALSE ...

$ longitude : logi NA NA NA NA NA NA ...

$ latitude : logi NA NA NA NA NA NA ...

Creating word corpus of tweeter data, Corpus here indicates large set of text, i.e. in this case 1000 tweets.

myCorpus<- Corpus(VectorSource(tweets.df$text))

writeLines(strwrap(myCorpus[[792]]$content,60))

@IPL @IPL should b postponed til da #coronavirus issue

settle downs.Thr is no use in conducting such games wtout

cr… https://t.co/JUA0aPWP0R

Keeping a copy of “myCorpus” for stem completion later,

myCorpusCopy<- myCorpus

# **3] Data Cleaning**

Text in this case is just a sequence of words more precisely, a sequence of characters. But when dealing with natural language processing, we are more concerned about the words as a whole, instead of character-level depth of text data.

In order to deal with this, following techniques have been used.

## **Remove character string between < >**

Each Unicode in the tweet is different and doesn’t reflect any useful information. If left as it is, it will hamper the word frequency and corpus will have meaning less words left in it. Hence, it becomes necessary to remove these unicodes.

remove\_unicode <- function(x) gsub("\\<U[^\\>]\*\\>","", x)

myCorpus <- tm\_map(myCorpus, content\_transformer(remove\_unicode))

writeLines(strwrap(myCorpus[[792]]$content,60))

@IPL @IPL should b postponed til da #coronavirus issue

settle downs.Thr is no use in conducting such games wtout

cr… https://t.co/JUA0aPWP0R

## **Converting all text to Lowercase**

Converting every text to lowercase will limit the appearance of word twice if it has been used in variable cases i.e. uppercase.

myCorpus <- tm\_map(myCorpus, content\_transformer(stri\_trans\_tolower))

writeLines(strwrap(myCorpus[[792]]$content,60))

@ipl @ipl should b postponed til da #coronavirus issue

settle downs.thr is no use in conducting such games wtout

cr… https://t.co/jua0apwp0r

## **Remove the Links (URLs)**

Again, URL’s present in the data after proper data cleaning will remain in corpus and will form meaning less words. Hence, it becomes essential to remove these.

removeURL <- function(x) gsub("http[^[:space:]]\*", "", x)

myCorpus <- tm\_map(myCorpus, content\_transformer(removeURL))

writeLines(strwrap(myCorpus[[792]]$content,60))

@ipl @ipl should b postponed til da #coronavirus issue

settle downs.thr is no use in conducting such games wtout

cr…

## **Removing usernames from tweets**

Username does not reflect any useful data in this case. They will bunch up in word cloud and will affect real scenario in word cloud and sentiments.

removeUsername <- function(x) gsub("@[^[:space:]]\*", "", x)

myCorpus <- tm\_map(myCorpus, content\_transformer(removeUsername))

writeLines(strwrap(myCorpus[[792]]$content,60))

should b postponed til da #coronavirus issue settle

downs.thr is no use in conducting such games wtout cr…

## **Stemming words in the corpus**

Stemming basically refers to return word in a text to original stem. In this case return any possible verb to its original stem.

myCorpus<-tm\_map(myCorpus, stemDocument)

writeLines(strwrap(myCorpus[[792]]$content,60))

should b postpon til da #coronavirus issu settl downs.thr

is no use in conduct such game wtout cr

## **Remove Single Letter Words**

Individual characters don’t have a lot of “context”. These individual letter don’t hold any context individually, but when rearranged in the form of a word which might explain some activity in word frequency.

removeSingle <- function(x) gsub(" . ", " ", x)

myCorpus <- tm\_map(myCorpus, content\_transformer(removeSingle))

writeLines(strwrap(myCorpus[[792]]$content,60))

should postpon til da #coronavirus issu settl downs.thr is

no use in conduct such game wtout cr…

## **Replace words with the proper ones**

This is essential since over eliminating can lead to bad results and can hamper final result. Here we are replacing/readjusting several words with to their proper meaning.

removeSingle <- function(x) gsub(" . ", " ", x)

myCorpus <- tm\_map(myCorpus, content\_transformer(removeSingle))

writeLines(strwrap(myCorpus[[792]]$content,60))

Replace words with the proper ones,

replaceWord <- function(corpus, oldword, newword)

{ tm\_map(corpus, content\_transformer(gsub), pattern=oldword, replacement=newword) }

myCorpus<- replaceWord(myCorpus, "precautionari", "precautionary")

myCorpus<- replaceWord(myCorpus, "meeting|meets", "meet")

myCorpus<- replaceWord(myCorpus, "franchisee|franchis|owners|ownerse", "owners")

myCorpus<- replaceWord(myCorpus, "effici|efficientently|efficientent","efficient")

myCorpus<- replaceWord(myCorpus, "measuree|measur","measure")

myCorpus<- replaceWord(myCorpus, "srk|shah|khan|shahrukhkhan|baadsrk","srk")

myCorpus<- replaceWord(myCorpus, "games|game|match","game")

## **Remove anything except the English language and space**

It never happens, that all of the text written in the tweets is in same language. There are certain words which are have local meaning to it but on big scale it is meaning less in English language. Hence, in order to increase accuracy and to have good results these must be removed.

removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]\*", "", x)

myCorpus <- tm\_map(myCorpus, content\_transformer(removeNumPunct))

writeLines(strwrap(myCorpus[[792]]$content,60))

should postpon til da coronavirus issu settl downsthr is no

use in conduct such game wtout cr

## **Removing Extra Whitespaces**

This involves stripping spaces in between the words, which helps in further cleaning operation.

myCorpus<- tm\_map(myCorpus, stripWhitespace)

writeLines(strwrap(myCorpus[[792]]$content,60))

should postpon til da coronavirus issu settl downsthr is no

use in conduct such game wtout cr

## **Removing Stop words**

Listing stop words is essential operation in data cleaning. It basically involves words which are banal. It may happen that these words have more frequency of occurring and this may affect further results. Hence, adding these words to stop words list is essential.

myStopWords<- c((stopwords('english')),

c("rt","use", "used","say","can","wh","says","wh","amp","bhi","th","r","ho","hq", "dont","reiter","b","reiterate","ipl","style","go","will","mi","hi","htt","psl","ht", "khelbolega","sr","m","one","t","bbl","h","ms","wi","till","re","meet","u","efficient","field","wonderful","owners","srk","play","game","efficientently","ownerse","baadsrk","measuree"))

myCorpus<- tm\_map(myCorpus,removeWords , myStopWords)

writeLines(strwrap(myCorpus[[792]]$content,60))

postpon til da coronavirus issu settl downsthr conduct

wtout cr

# **4] Analyzing Text frequency**

## **Creating term document matrix for text frequency analysis**

tdm<- TermDocumentMatrix(myCorpus, control= list(wordLengths= c(1, Inf)))

tdm

<<TermDocumentMatrix (terms: 1140, documents: 1000)>>

Non-/sparse entries: 4377/1135623

Sparsity : 100%

Maximal term length: 28

Weighting : term frequency (tf)

## **Find the most frequently used terms in our corpus**

(freq.terms <- findFreqTerms(tdm, lowfreq = 25))

[1] "owner" "wonder" "king" "ungame" "coronavirus"

[6] "announcement" "april" "corona" "covid" "measure"

[11] "novel" "ongo" "precautionary" "suspend" "virus"

[16] "vivoipl" "rcb" "cricket" "rukh" "show"

[21] "event" "best" "bet" "curtail" "debat"

[26] "scenario"

term.freq <- rowSums(as.matrix(tdm))

term.freq <- subset(term.freq, term.freq > 25)

df <- data.frame(term = names(term.freq), freq= term.freq)

## **Word Frequency Analysis (Keeping word frequency low to high)**

Keeping word frequency more than or equal to 10:

Below is the list of words with word frequency keeping as low as 10. This basically means, all the words which appeared at least 10 times will appear in list.

(freq.terms <- findFreqTerms(tdm, lowfreq = 10))

[1] "owner" "wonder" "king" "ungame"

[5] "start" "hear" "kane" "read"

[9] "richardson" "wait" "wer" "coronavirus"

[13] "now" "announcement" "april" "corona"

[17] "covid" "measure" "novel" "ongo"

[21] "precautionary" "suspend" "virus" "vivoipl"

[25] "coronavirusupd" "zeejankarioncorona" "rcb" "stadium"

[29] "veri" "time" "march" "csk"

[33] "fan" "bcci" "detail" "focus"

[37] "held" "news" "ownerses" "public"

[41] "safeti" "wellbeing" "thank" "cricket"

[45] "season" "dhoni" "coronavirusoutbreak" "prevent"

[49] "spread" "take" "arriv" "rukh"

[53] "postpon" "sir" "show" "team"

[57] "come" "player" "year" "money"

[61] "countri" "hope" "subsid" "event"

[65] "best" "follow" "cautionyespanicno" "safe"

[69] "stay" "first" "franch" "later"

[73] "well" "bet" "curtail" "debat"

[77] "scenario" "clean" "disease" "hand"

[81] "panic" "precaut" "iplt" "see"

[85] "win" "world" "banter" "camp"

[89] "congratul"

term.freq <- rowSums(as.matrix(tdm))

term.freq <- subset(term.freq, term.freq > 10)

df1 <- data.frame(term = names(term.freq), freq= term.freq)

Keeping word frequency more than or equal to 25:

Below is the list of words with word frequency keeping as low as 25. This basically means, all the words which appeared at least 10 times will appear in list.

(freq.terms <- findFreqTerms(tdm, lowfreq = 25))

[1] "owner" "wonder" "king" "ungame" "coronavirus"

[6] "announcement" "april" "corona" "covid" "measure"

[11] "novel" "ongo" "precautionary" "suspend" "virus"

[16] "vivoipl" "rcb" "cricket" "rukh" "show"

[21] "event" "best" "bet" "curtail" "debat"

[26] "scenario"

term.freq <- rowSums(as.matrix(tdm))

term.freq <- subset(term.freq, term.freq > 25)

df <- data.frame(term = names(term.freq), freq= term.freq)

Keeping word frequency more than or equal to 55:

Below is the list of words with word frequency keeping as low as 55. This basically means, all the words which appeared at least 10 times will appear in list.

(freq.terms <- findFreqTerms(tdm, lowfreq = 55))

[1] "owner" "wonder" "king" "ungame" "coronavirus" "corona"

[7] "covid" "virus"

term.freq <- rowSums(as.matrix(tdm))

term.freq <- subset(term.freq, term.freq > 55)

df2 <- data.frame(term = names(term.freq), freq= term.freq)

Keeping word frequency more than or equal to 85:

Below is the list of words with word frequency keeping as low as 85. This basically means, all the words which appeared at least 10 times will appear in list.

(freq.terms <- findFreqTerms(tdm, lowfreq = 85))

[1] "owner" "wonder" "king" "ungame" "covid"

term.freq <- rowSums(as.matrix(tdm))

term.freq <- subset(term.freq, term.freq > 85)

df3 <- data.frame(term = names(term.freq), freq= term.freq)

## **Plotting word frequency for different frequency of word**

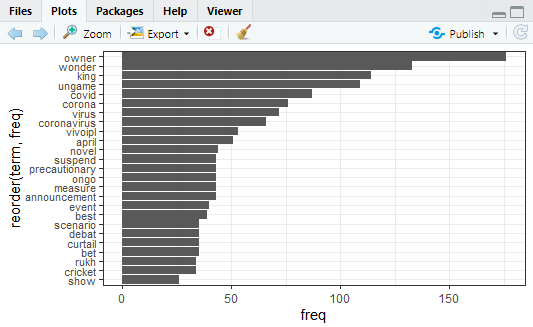
In continuation to the above word frequency lists,

p1=ggplot(df1, aes(reorder(term, freq),freq)) + theme\_bw() + geom\_bar(stat = "identity") + coord\_flip() +labs(list(title="@10", x="Terms", y="Term Counts")) + theme(axis.text.y = element\_text(size=7))

Though word frequency plot is not shown here but, most frequent word coming at top are **owner, wonder, king, ungame, covid, corona, virus, coronavirus, vivoipl, april, novel.**

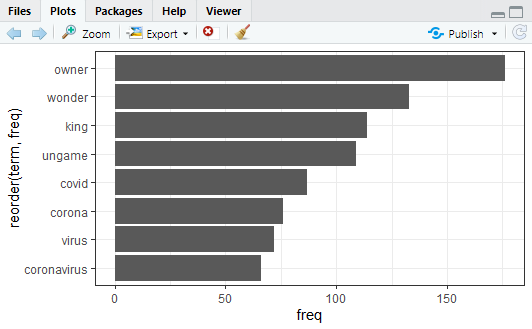
p2=ggplot(df, aes(reorder(term, freq),freq)) + theme\_bw() + geom\_bar(stat = "identity") + coord\_flip() +labs(list(title="@25", x="Terms", y="Term Counts"))+

theme(axis.text.y = element\_text(size=7))



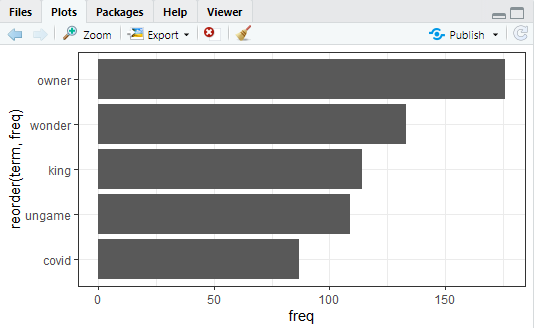
As seen in plot, above words have appeared in corpus for more than 25 times. And **owner, wonder, king, ungame, covid** are appearing more frequently. Words like debate, precautionary, April, suspend can also be seen which confirms as precautionary measure April IPL is suspended.

p3=ggplot(df2, aes(reorder(term, freq),freq)) + theme\_bw() + geom\_bar(stat = "identity") + coord\_flip() +labs(list(title="@55", x="Terms", y="Term Counts"))



As seen in plot, above words have appeared in corpus for more than 55 times. And **owner, wonder, king, ungame, covid** are appearing more frequently. It confirms the about covid-19 has created an unplayable condition.

p4=ggplot(df3, aes(reorder(term, freq),freq)) + theme\_bw() + geom\_bar(stat = "identity") + coord\_flip() +labs(list(title="@85", x="Terms", y="Term Counts"))



As seen in plot, above words have appeared in corpus for more than 85 times. And **owner, wonder, king** are appearing more frequently. This indicates people are mostly talking about these, Covid being reason for suspension of wonderful game of IPL and King or Shahrukh khan being most active among people.

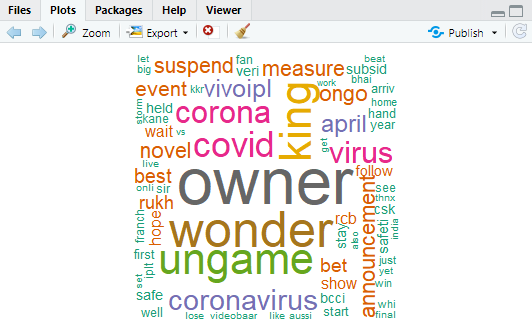
# **5] Word Cloud**

## **1. Word cloud for Overall corpus**

word.freq <-sort(rowSums(as.matrix(tdm)), decreasing= F)

pal<- brewer.pal(8, "Dark2")

wordcloud(words = names(word.freq), freq = word.freq, min.freq = 2, random.order = F, colors = pal, max.words = 170)



Word cloud tells us about owners of IPL teams being most active or talked about, as IPL is suspended and they are talking to their fan base and also talking about measures against corona. Also, King Khan and covid are the buzz words as told by frequency analysis. People are also talking about Kane Richardson, first cricketer to get covid-19.

## **2. Creating Separate word cloud for Positive and Negative Sentiments.**

Extract text data from tweet data,

setwd("C:/Users/DELL/Desktop/Akshay/Group Assignments/Group Assignment WSMA")

tweets.df <- read.csv("WSMA.csv")

r1 = as.character(tweets.df$text)

Pre - Processing data,

set.seed(100)

sample = sample(r1, (length(r1)))

corpus = Corpus(VectorSource(list(sample)))

corpus = tm\_map(corpus, removePunctuation)

corpus = tm\_map(corpus, content\_transformer(tolower))

corpus = tm\_map(corpus, removeNumbers)

corpus = tm\_map(corpus, stripWhitespace)

corpus = tm\_map(corpus, removeWords, stopwords('english'))

corpus = tm\_map(corpus, stemDocument)

dtm\_up = DocumentTermMatrix(VCorpus(VectorSource(corpus[[1]]$content)))

freq\_up <- colSums(as.matrix(dtm\_up))

Calculating positive and negative sentiments separately,

sentiments\_up = calculate\_sentiment(names(freq\_up))

sentiments\_up = cbind(sentiments\_up, as.data.frame(freq\_up))

sent\_pos\_up = sentiments\_up[sentiments\_up$sentiment == 'Positive',]

sent\_neg\_up = sentiments\_up[sentiments\_up$sentiment == 'Negative',]

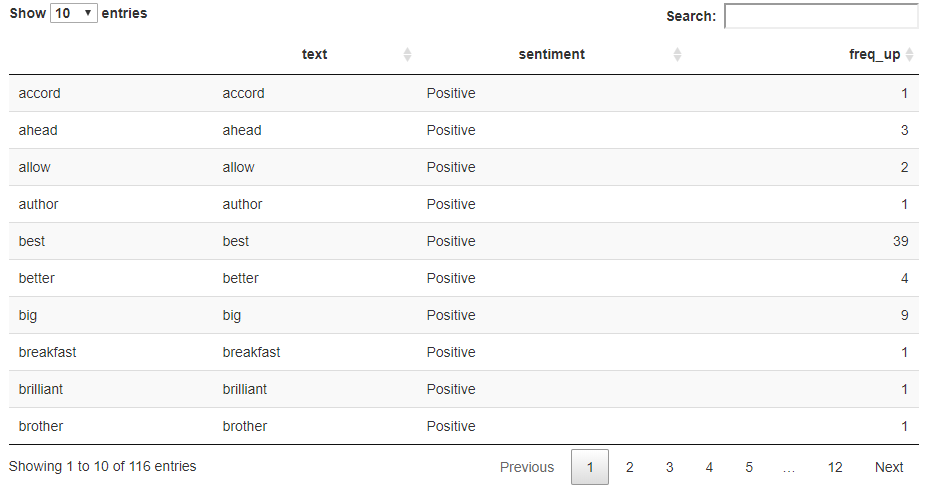
cat("We have far lower negative Sentiments: ",sum(sent\_neg\_up$freq\_up)," than positive: ",sum(sent\_pos\_up$freq\_up))

We have far lower negative Sentiments: 317 than positive: 697

### **Positive Words**

The table below shows the frequency of words in our text classified as positive. The same has been generated using the data table function.

DT::datatable(sent\_pos\_up)



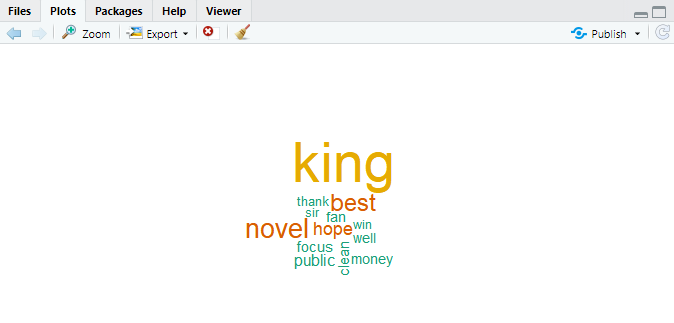
layout(matrix(c(1, 2), nrow=2), heights=c(1, 4))

par(mar=rep(0, 4))

plot.new()

set.seed(100)

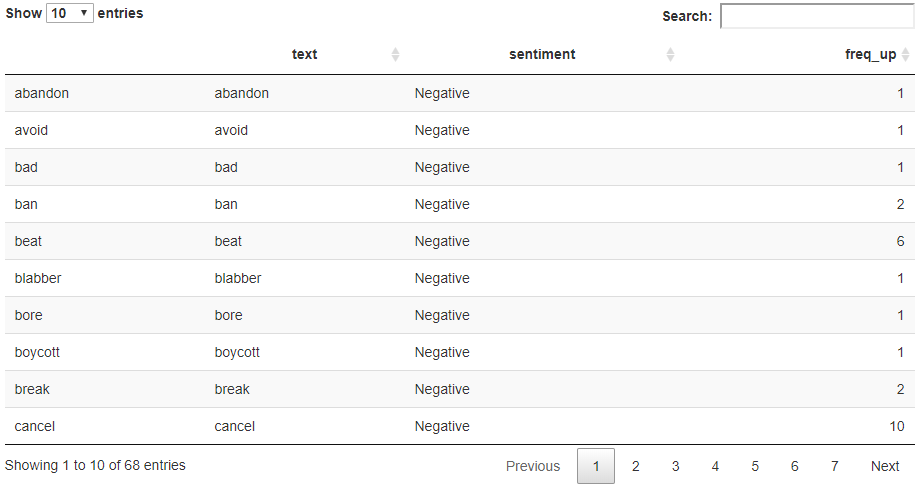
wordcloud(sent\_pos\_up$text,sent\_pos\_up$freq,min.freq=10,colors=brewer.pal(6,"Dark2"))



### **Negative Words**

The table below shows the frequency of words in our text classified as Negative. The same has been generated using the data table function.

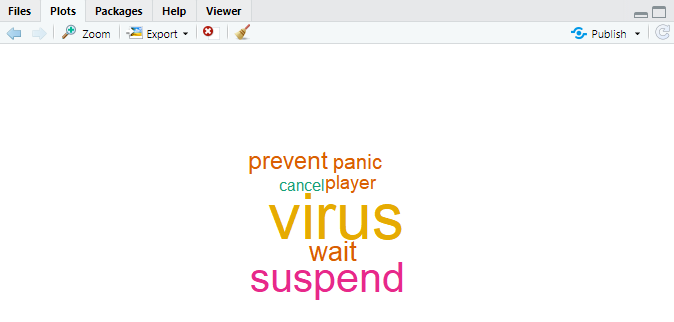
DT::datatable(sent\_neg\_up)



plot.new()

set.seed(100)

wordcloud(sent\_neg\_up$text,sent\_neg\_up$freq, min.freq=10,colors=brewer.pal(6,"Dark2"))



# **6] Sentiment Analysis, polarity - positive or negative**

Sentiment Analysis: understanding emotional valence in tweets using syuzhet

mysentiment<-get\_nrc\_sentiment((tweets.df$text))

# Get the sentiment score for each emotion

mysentiment.positive =sum(mysentiment$positive)

mysentiment.anger =sum(mysentiment$anger)

mysentiment.anticipation =sum(mysentiment$anticipation)

mysentiment.disgust =sum(mysentiment$disgust)

mysentiment.fear =sum(mysentiment$fear)

mysentiment.joy =sum(mysentiment$joy)

mysentiment.sadness =sum(mysentiment$sadness)

mysentiment.surprise =sum(mysentiment$surprise)

mysentiment.trust =sum(mysentiment$trust)

mysentiment.negative =sum(mysentiment$negative)

# Create the bar chart

yAxis <- c(mysentiment.positive,

+ mysentiment.anger,

+ mysentiment.anticipation,

+ mysentiment.disgust,

+ mysentiment.fear,

+ mysentiment.joy,

+ mysentiment.sadness,

+ mysentiment.surprise,

+ mysentiment.trust,

+ mysentiment.negative)

xAxis <- c("Positive","Anger","Anticipation","Disgust","Fear","Joy","Sadness",

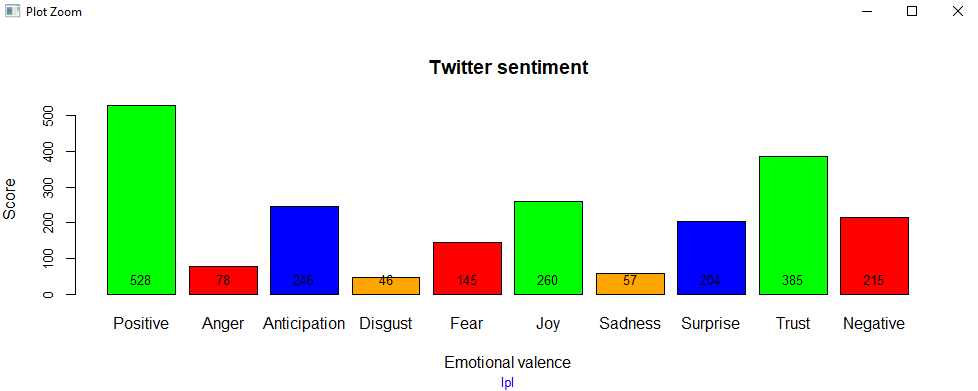
"Surprise","Trust","Negative")

colors <- c("green","red","blue","orange","red","green","orange","blue","green","red")

yRange <- range(0,yAxis)

bp <- barplot(yAxis, names.arg = xAxis, xlab = "Emotional valence", ylab = "Score", main = "Twitter sentiment", sub = "Ipl", col = colors, border = "black", xpd = F, ylim = yRange, axisnames = T, cex.axis = 0.8, cex.sub = 0.8, col.sub = "blue")

text(bp, 0, round(yAxis,1), cex = 0.8, pos = 3)



bp

[,1]

[1,] 0.7

[2,] 1.9

[3,] 3.1

[4,] 4.3

[5,] 5.5

[6,] 6.7

[7,] 7.9

[8,] 9.1

[9,] 10.3

[10,] 11.5

We can see that people are positive or welcoming IPL suspension as they are also looking to stay at home and don’t want to risk lives and control Corona. They have trust in BCCI that they will try to conduct IPL on some future date and have anticipation for same. But they are unhappy or sad for not able to enjoy the fun or watch it at home at least. We can see mix emotions from people on IPL suspension.

## **Sentiment Analysis: Plot by date - understanding cumulative sentiment score movement**

mysentimentvalues <- data.frame(get\_sentiment((tweets.df$text)))

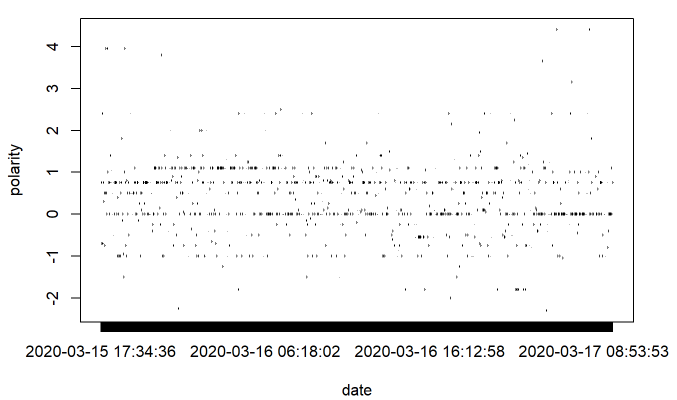
colnames(mysentimentvalues)<-"polarity"

mysentimentvalues$date <- tweets.df$created

result <- aggregate(polarity ~ date, data = mysentimentvalues, sum)

result

plot(result, type = "l")



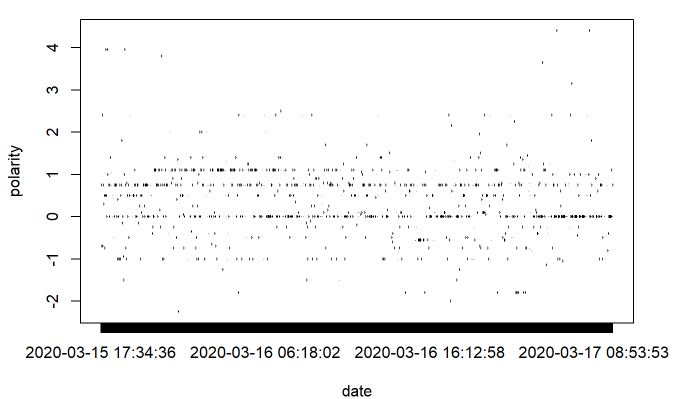
We can see many tweets have score of -1 to 1, means people are having mix emotions, and they are pretty much neutral about suspension. But also, people can be seen very confident about IPL suspension in respect to corona control and also very angry as they will miss money they bet, as it was seen in frequency analysis.

## **Sentiment Analysis: Plot by date - Understanding average sentiment score movement**

result1 <- aggregate(polarity ~ date, data = mysentimentvalues, mean)

result1

plot(result1, type = "l")



The average sentiment is matching with cumulative sentiment and share same thoughts about that.

# **7] Correlation chart of top keywords, including Word Association**

## **Finding association of specific keyword with other words in tweets**

Association of other words with keyword “vivoipl”:

list1<- findAssocs(tdm, "vivoipl", 0.25)

corrdf1 <- t(data.frame(t(sapply(list1,c))))

corrdf1

vivoipl

announcement 0.90

measure 0.90

ongo 0.90

precautionary 0.90

suspend 0.90

novel 0.89

april 0.82

virus 0.67

corona 0.63

covid 0.61

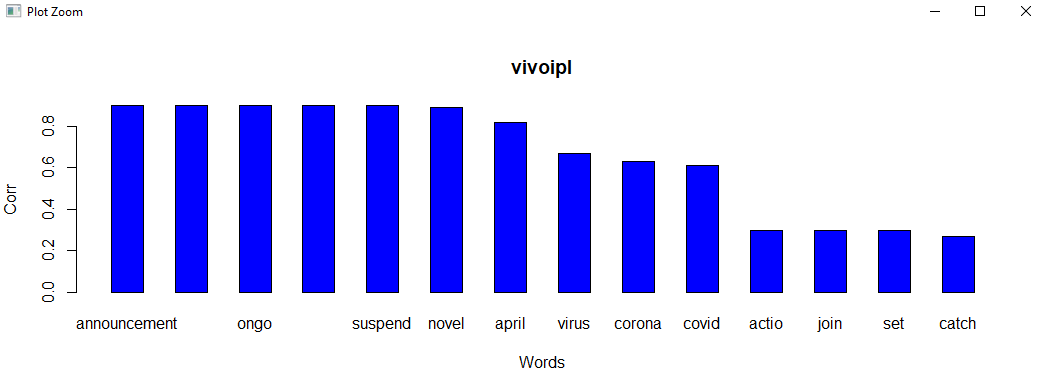
actio 0.30

join 0.30

set 0.30

catch 0.27

barplot(t(as.matrix(corrdf1)), beside=TRUE,xlab = "Words",ylab = "Corr",col = "blue",main = "vivoipl",border = "black")



Association of other words with keyword “covid”:

list1<- findAssocs(tdm, "covid", 0.20)

corrdf1 <- t(data.frame(t(sapply(list1,c))))

corrdf1

covid

announcement 0.69

measure 0.69

ongo 0.69

precautionary 0.69

suspend 0.69

novel 0.68

april 0.64

vivoipl 0.61

corona 0.52

virus 0.52

arriv 0.41

rukh 0.26

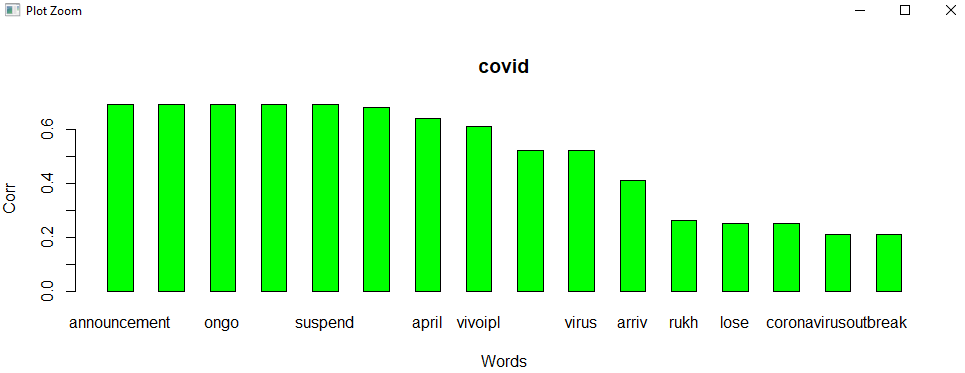
lose 0.25

moneyspin 0.25

coronavirusoutbreak 0.21

potent 0.21

barplot(t(as.matrix(corrdf1)), beside=TRUE,xlab = "Words",ylab = "Corr",col = "green",main = "covid",border = "black")



We can see from above two correlations, that people are mostly talking about Corona and IPL suspension because of it and they are also talking king khan and how owners are talking about corona.

## **Topic Modelling to identify latent/hidden topics using LDA technique**

dtm <- as.DocumentTermMatrix(tdm)

rowTotals <- apply(dtm , 1, sum)

NullDocs <- dtm[rowTotals==0, ]

dtm <- dtm[rowTotals> 0, ]

if (length(NullDocs$dimnames$Docs) > 0) {

tweets.df <- tweets.df[-as.numeric(NullDocs$dimnames$Docs),]

}

lda <- LDA(dtm, k = 5) # find 5 topic

term <- terms(lda, 7) # first 7 terms of every topic

(term <- apply(term, MARGIN = 2, paste, collapse = ", "))

Topic 1

"safeti, bcci, held, public, news, come, detail"

Topic 2

"event, best, owner, bet, curtail, debat, scenario"

Topic 3

"coronavirus, follow, prevent, safe, stay, clean, disease"

Topic 4

"owner, wonder, king, ungame, wait, hear, wer"

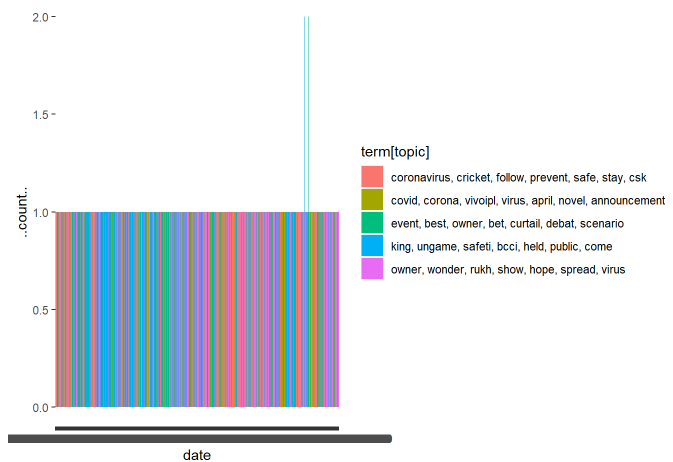
Topic 5

"covid, corona, virus, vivoipl, april, novel, announcement"

topics<- topics(lda)

topics<- data.frame(date=(tweets.df$created), topic = topics)

qplot (date, ..count.., data=topics, geom ="bar", fill= term[topic], position="stack")



Well LDA doesn’t give much information, but we can see topic 2 tell us about Corona and April IPL announcement.

Topic 4 tells about how BCCI and King Khan are talking about public safety when people will come to ground, so it is safety which is being talked about.

Topic 1 tells how CSK talking about coronavirus and asking people to stay safe, prevent themselves and still follow cricket.