p2\_g8\_covid

g8

03/07/2020

## Loading libraries

library(rtweet)

## Warning: package 'rtweet' was built under R version 3.6.3

library(lubridate) #ymd\_hms

## Warning: package 'lubridate' was built under R version 3.6.3

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

#install.packages("magrittr") # package installations are only needed the first time you use it  
#detach("package:dplyr")   
#install.packages("dplyr") # alternative installation of the %>%  
library(magrittr) # needs to be run every time you start R and want to use %>%  
library(dplyr) # alternatively, this also loads %>% And glimpse

## Warning: package 'dplyr' was built under R version 3.6.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(kableExtra) # kable

##   
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':  
##   
## group\_rows

library(knitr)

## Warning: package 'knitr' was built under R version 3.6.3

library(dplyr) # %>% and unnest\_tokens  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.6.3

#install.packages("stringr") # Install stringr R package For Str\_c  
library("stringr") # Load stringr R package

## Warning: package 'stringr' was built under R version 3.6.3

#install.packages("tibble")  
library(tibble)

## Warning: package 'tibble' was built under R version 3.6.3

#install.packages("tidyverse")  
  
library(tidyr)

## Warning: package 'tidyr' was built under R version 3.6.3

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:magrittr':  
##   
## extract

library("tm") # For DataframeSource

## Warning: package 'tm' was built under R version 3.6.3

## Loading required package: NLP

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

library(tidytext) # For tidy

## Warning: package 'tidytext' was built under R version 3.6.3

#install.packages("wordcloud")  
library(wordcloud) # For wordcloud

## Warning: package 'wordcloud' was built under R version 3.6.3

## Loading required package: RColorBrewer

#install.packages("wordcloud2") # For wordcloud2  
library(gridExtra) # For grid.arrange

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(recommenderlab)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loading required package: arules

##   
## Attaching package: 'arules'

## The following object is masked from 'package:tm':  
##   
## inspect

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

## Loading required package: proxy

## Warning: package 'proxy' was built under R version 3.6.3

##   
## Attaching package: 'proxy'

## The following object is masked from 'package:Matrix':  
##   
## as.matrix

## The following objects are masked from 'package:stats':  
##   
## as.dist, dist

## The following object is masked from 'package:base':  
##   
## as.matrix

## Loading required package: registry

## Registered S3 methods overwritten by 'registry':  
## method from   
## print.registry\_field proxy  
## print.registry\_entry proxy

library(dplyr) # for spread  
  
  
#install.packages("tidyverse") #Try replacing library(dplyr) with library(tidyverse). The spread function now lives in the tidyr package which is part of the tidyverse along with dplyr.  
library(tidyverse) # for spread

## Warning: package 'tidyverse' was built under R version 3.6.3

## -- Attaching packages ----------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v readr 1.3.1 v forcats 0.5.0  
## v purrr 0.3.3

## Warning: package 'forcats' was built under R version 3.6.3

## -- Conflicts -------------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x NLP::annotate() masks ggplot2::annotate()  
## x lubridate::as.difftime() masks base::as.difftime()  
## x gridExtra::combine() masks dplyr::combine()  
## x lubridate::date() masks base::date()  
## x Matrix::expand() masks tidyr::expand()  
## x tidyr::extract() masks magrittr::extract()  
## x dplyr::filter() masks stats::filter()  
## x purrr::flatten() masks rtweet::flatten()  
## x kableExtra::group\_rows() masks dplyr::group\_rows()  
## x arules::intersect() masks lubridate::intersect(), base::intersect()  
## x dplyr::lag() masks stats::lag()  
## x Matrix::pack() masks tidyr::pack()  
## x arules::recode() masks dplyr::recode()  
## x purrr::set\_names() masks magrittr::set\_names()  
## x arules::setdiff() masks lubridate::setdiff(), base::setdiff()  
## x arules::union() masks lubridate::union(), base::union()  
## x Matrix::unpack() masks tidyr::unpack()

library(textdata) # For afinn

## Warning: package 'textdata' was built under R version 3.6.3

library(bit64) # For Integer64

## Loading required package: bit

## Attaching package bit

## package:bit (c) 2008-2012 Jens Oehlschlaegel (GPL-2)

## creators: bit bitwhich

## coercion: as.logical as.integer as.bit as.bitwhich which

## operator: ! & | xor != ==

## querying: print length any all min max range sum summary

## bit access: length<- [ [<- [[ [[<-

## for more help type ?bit

##   
## Attaching package: 'bit'

## The following object is masked from 'package:base':  
##   
## xor

## Attaching package bit64

## package:bit64 (c) 2011-2012 Jens Oehlschlaegel

## creators: integer64 seq :

## coercion: as.integer64 as.vector as.logical as.integer as.double as.character as.bin

## logical operator: ! & | xor != == < <= >= >

## arithmetic operator: + - \* / %/% %% ^

## math: sign abs sqrt log log2 log10

## math: floor ceiling trunc round

## querying: is.integer64 is.vector [is.atomic} [length] format print str

## values: is.na is.nan is.finite is.infinite

## aggregation: any all min max range sum prod

## cumulation: diff cummin cummax cumsum cumprod

## access: length<- [ [<- [[ [[<-

## combine: c rep cbind rbind as.data.frame

## WARNING don't use as subscripts

## WARNING semantics differ from integer

## for more help type ?bit64

##   
## Attaching package: 'bit64'

## The following object is masked from 'package:bit':  
##   
## still.identical

## The following objects are masked from 'package:arules':  
##   
## %in%, match

## The following objects are masked from 'package:base':  
##   
## %in%, :, is.double, match, order, rank

#install.packages("knitr")  
#install.packages("rmarkdown")  
library(rmarkdown)

## Warning: package 'rmarkdown' was built under R version 3.6.3

#-----------------  
#remove.packages('lubridate')  
#install.packages('Rcpp', dependencies = TRUE)  
#install.packages('lubridate', dependencies = TRUE)  
library(lubridate)  
  
library(caTools) # For sample.split  
library(rpart) # For rpart and tree plot  
library(rpart.plot)  
library(caret) # For trainControl

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

## The following objects are masked from 'package:recommenderlab':  
##   
## MAE, RMSE

## Loading the data

covid <- as\_tibble(data.table::fread(str\_c("d:/", "COVID.csv"), encoding= "UTF-8"))  
  
str(covid)

## tibble [60,160 x 22] (S3: tbl\_df/tbl/data.frame)  
## $ Tweet Id : chr [1:60160] "\"\"1233417783175778304\"\"" "\"\"1233417742520332290\"\"" "\"\"1233417741027225602\"\"" "\"\"1233417699264356357\"\"" ...  
## $ Tweet URL : chr [1:60160] "https://twitter.com/Giussi92/status/1233417783175778304" "https://twitter.com/LAMofficial/status/1233417742520332290" "https://twitter.com/mitchellvii/status/1233417741027225602" "https://twitter.com/HelenKennedy/status/1233417699264356357" ...  
## $ Tweet Posted Time (UTC) : chr [1:60160] "28 Feb 2020 15:44:49" "28 Feb 2020 15:44:40" "28 Feb 2020 15:44:39" "28 Feb 2020 15:44:29" ...  
## $ Tweet Content : chr [1:60160] "Also the entire Swiss Football League is on hold. Postponing games from the professional and amateur level... #"| \_\_truncated\_\_ "World Health Org Official: Trump’s press conference on #coronavirus ‘incoherent’\nWorld Health Organization Spe"| \_\_truncated\_\_ "I mean, Liberals are cheer-leading this #Coronavirus like it's their high school football team.\n\n#TDS" "Under repeated questioning, Pompeo refuses to say the #COVID19 coronavirus is not a hoax. https://t.co/PFESx4ePpD" ...  
## $ Tweet Type : chr [1:60160] "Tweet" "Tweet" "Tweet" "Tweet" ...  
## $ Client : chr [1:60160] "Twitter for iPhone" "Twitter Web App" "Twitter Web App" "Twitter for iPhone" ...  
## $ Retweets Received : int [1:60160] 0 0 23 4 0 14 3 11 1 2 ...  
## $ Likes Received : int [1:60160] 0 0 64 11 0 11 9 13 4 8 ...  
## $ Tweet Location : chr [1:60160] "" "Los Angeles CA USA" "Miami, FL" "NYC and the North Fork" ...  
## $ Lat : num [1:60160] NA NA NA NA NA NA NA NA NA NA ...  
## $ Long : num [1:60160] NA NA NA NA NA NA NA NA NA NA ...  
## $ Tweet Language : chr [1:60160] "English" "English" "English" "English" ...  
## $ User Id : chr [1:60160] "\"\"1556856595\"\"" "\"\"113738369\"\"" "\"\"17980523\"\"" "\"\"2199541\"\"" ...  
## $ Name : chr [1:60160] "Giuseppe Gentile" "London After Midnight" "Bill Mitchell" "Helen Kennedy" ...  
## $ Screen Name : chr [1:60160] "Giussi92" "LAMofficial" "mitchellvii" "HelenKennedy" ...  
## $ User Bio : chr [1:60160] "" "London After Midnight is a music project by writer, #vegan, political &amp; animal rights activist Sean Brennan"| \_\_truncated\_\_ "Host of YourVoice\231 America at http://yourvoiceamerica.tv, Mon-Frid 10am ET #TrustTrump #MAGA #Trump2020 #2A"| \_\_truncated\_\_ "Newspaperman (ex NY Daily News &amp; Boston Herald). I had a Trump joke here but it suddenly stopped being funn"| \_\_truncated\_\_ ...  
## $ Verified or Non-Verified : chr [1:60160] "Verified" "Verified" "Verified" "Verified" ...  
## $ Profile URL : chr [1:60160] "https://twitter.com/Giussi92" "https://twitter.com/LAMofficial" "https://twitter.com/mitchellvii" "https://twitter.com/HelenKennedy" ...  
## $ Protected or Non-protected: chr [1:60160] "Non-Protected" "Non-Protected" "Non-Protected" "Non-Protected" ...  
## $ User Followers : int [1:60160] 3071 4189 534045 27704 94269 954641 27990 954641 286384 577298 ...  
## $ User Following : int [1:60160] 100 84 10037 1060 1999 1320 2562 1320 362 131 ...  
## $ User Account Creation Date: chr [1:60160] "30 Jun 2013 00:27:50" "12 Feb 2010 21:43:17" "09 Dec 2008 01:54:21" "25 Mar 2007 19:14:46" ...  
## - attr(\*, ".internal.selfref")=<externalptr>

glimpse(covid)

## Rows: 60,160  
## Columns: 22  
## $ `Tweet Id` <chr> "\"\"1233417783175778304\"\"", "\"\"12...  
## $ `Tweet URL` <chr> "https://twitter.com/Giussi92/status/1...  
## $ `Tweet Posted Time (UTC)` <chr> "28 Feb 2020 15:44:49", "28 Feb 2020 1...  
## $ `Tweet Content` <chr> "Also the entire Swiss Football League...  
## $ `Tweet Type` <chr> "Tweet", "Tweet", "Tweet", "Tweet", "T...  
## $ Client <chr> "Twitter for iPhone", "Twitter Web App...  
## $ `Retweets Received` <int> 0, 0, 23, 4, 0, 14, 3, 11, 1, 2, 17, 0...  
## $ `Likes Received` <int> 0, 0, 64, 11, 0, 11, 9, 13, 4, 8, 20, ...  
## $ `Tweet Location` <chr> "", "Los Angeles CA USA", "Miami, FL",...  
## $ Lat <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ Long <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...  
## $ `Tweet Language` <chr> "English", "English", "English", "Engl...  
## $ `User Id` <chr> "\"\"1556856595\"\"", "\"\"113738369\"...  
## $ Name <chr> "Giuseppe Gentile", "London After Midn...  
## $ `Screen Name` <chr> "Giussi92", "LAMofficial", "mitchellvi...  
## $ `User Bio` <chr> "", "London After Midnight is a music ...  
## $ `Verified or Non-Verified` <chr> "Verified", "Verified", "Verified", "V...  
## $ `Profile URL` <chr> "https://twitter.com/Giussi92", "https...  
## $ `Protected or Non-protected` <chr> "Non-Protected", "Non-Protected", "Non...  
## $ `User Followers` <int> 3071, 4189, 534045, 27704, 94269, 9546...  
## $ `User Following` <int> 100, 84, 10037, 1060, 1999, 1320, 2562...  
## $ `User Account Creation Date` <chr> "30 Jun 2013 00:27:50", "12 Feb 2010 2...

names(covid)

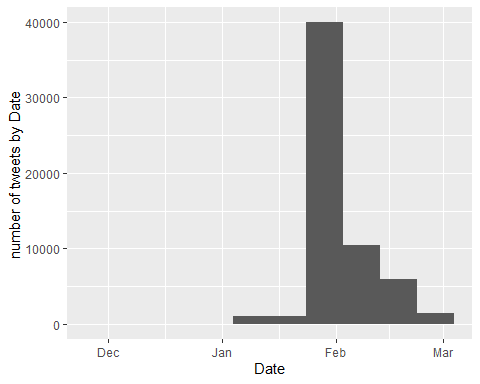
## [1] "Tweet Id" "Tweet URL"   
## [3] "Tweet Posted Time (UTC)" "Tweet Content"   
## [5] "Tweet Type" "Client"   
## [7] "Retweets Received" "Likes Received"   
## [9] "Tweet Location" "Lat"   
## [11] "Long" "Tweet Language"   
## [13] "User Id" "Name"   
## [15] "Screen Name" "User Bio"   
## [17] "Verified or Non-Verified" "Profile URL"   
## [19] "Protected or Non-protected" "User Followers"   
## [21] "User Following" "User Account Creation Date"

## preparation Date

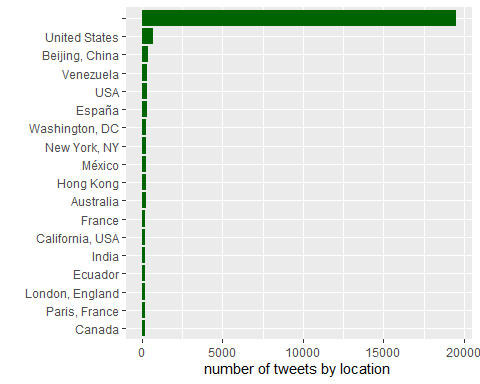
names(covid)[names(covid) == "Screen Name"] <- "Screen\_Name"  
names(covid)[names(covid) == "Tweet Language"] <- "Tweet\_Language"  
  
# We need this name for create VCorpus object  
names(covid)[names(covid) == "Tweet Content"] <- "text"   
names(covid)[names(covid) == "Tweet Id"] <- "doc\_id"   
covid$doc\_id <- gsub('"', "", covid$doc\_id, fixed = TRUE)  
covid$doc\_id <- as.integer64(as.character(covid$doc\_id))  
  
  
  
names(covid)[names(covid) == "Tweet Type"] <- "Tweet\_Type"  
  
names(covid)[names(covid) == "Tweet Posted Time (UTC)"] <- "time"  
  
names(covid)[names(covid) == "Tweet Location"] <- "location"  
  
covid$date\_time= parse\_date\_time(covid$time, '%d/%b/%Y %H:%M:%S')  
covid$date <- as.Date(covid$date\_time, format = "%Y.%m.%d")

## Exploratory Data Analysis (EDA)

ggplot(covid,aes(x=date)) + geom\_histogram(aes(y = (..count..)),binwidth=10) +   
labs(x="Date", y="number of tweets by Date ")



dt <- covid %>% group\_by(location) %>% count() %>% filter(n>=200) %>% arrange(desc(n)) %>% ungroup()  
ggplot(dt, aes(x=reorder(location, n), y=n)) +  
 geom\_bar(stat="identity", fill="darkgreen") + coord\_flip() +  
 labs(x="", y="number of tweets by location ") +  
 theme(legend.position = "none")



## Do covid tweet in Languages other than English?

Some tweets are in languages other than English. The “Undefined” ones are generally very short tweets

dt <- covid %>% group\_by(Tweet\_Language) %>% count() # %>% rename("Tweet\_Language"= "lang", n= "Number of Tweets")  
names(dt)

## [1] "Tweet\_Language" "n"

kable(dt) %>%  
 kable\_styling(bootstrap\_options = "striped", full\_width = F, position = "float\_left")

Tweet\_Language

n

1148

Arabic

79

Basque

1

Catalan

104

Chinese

283

Czech

9

Danish

19

Divehi

1

Dutch

93

English

33174

Estonian

12

Farsi

18

Finnish

14

French

2932

German

434

Greek

24

Hindi

15

ht

9

in

388

Italian

2958

iw

1

Japanese

204

km

1

Korean

18

Latvian

2

Lithuanian

2

ml

3

or

1

Pashto

1

Polish

69

Portuguese

912

Romanian

8

Russian

29

si

3

Slovenian

3

Spanish

15814

Swedish

17

Tagalog

131

Tamil

13

Thai

271

Turkish

884

Urdu

53

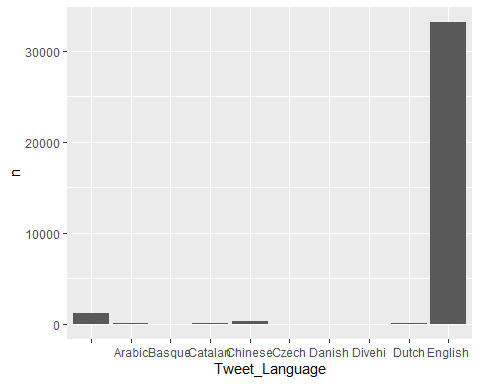
Vietnamese

3

Welsh

2

ggplot(dt[1:10,], aes(x = Tweet\_Language, y = n)) +  
 geom\_bar(stat = "identity") # visualization Language



## Remove all Twitter than English Language

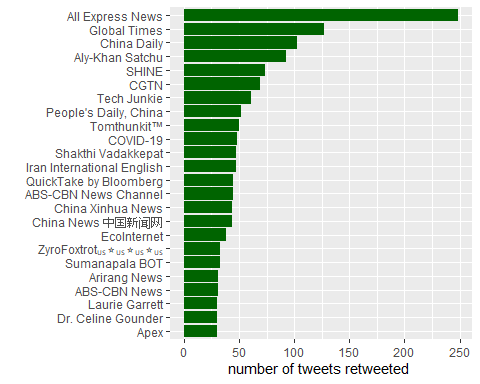
covid <- covid %>% filter(Tweet\_Language=="English")

## Now, let’s see if which people were interesting enough for Covid

p1 <- covid[c("Name")]  
  
p1 <- p1 %>% filter( Name != "" ) %>% group\_by (Name) %>% count() %>% filter(n>=30) %>% arrange(desc(n)) %>% ungroup()  
p1

## # A tibble: 24 x 2  
## Name n  
## <chr> <int>  
## 1 All Express News 249  
## 2 Global Times 127  
## 3 China Daily 103  
## 4 Aly-Khan Satchu 93  
## 5 SHINE 74  
## 6 CGTN 69  
## 7 Tech Junkie 61  
## 8 People's Daily, China 52  
## 9 Tomthunkit™ 50  
## 10 COVID-19 48  
## # ... with 14 more rows

ggplot(p1, aes(x=reorder(Name, n), y=n)) +  
 geom\_bar(stat="identity", fill="darkgreen") + coord\_flip() +  
 labs(x="", y="number of tweets retweeted ") +  
 theme(legend.position = "none")



## Text mining

##1 - Prepare The tweet texts

kable(head(covid %>% select(Name, Tweet\_Language, text), 20), format = "html") %>%  
 kable\_styling() %>%  
 column\_spec(1, bold = T, width = "2cm", border\_right = T) %>%  
 column\_spec(2, bold = T, width = "2cm", border\_right = T) %>%  
 column\_spec(3, width = "19cm")

Name

Tweet\_Language

text

Giuseppe Gentile

English

Also the entire Swiss Football League is on hold. Postponing games from the professional and amateur level… #coronavirus <https://t.co/UShMuqnAVC>

London After Midnight

English

World Health Org Official: Trump’s press conference on #coronavirus ‘incoherent’ World Health Organization Special Adviser to the Director Dr. Ezekiel Emanuel says "“I found most of what [Trump] said incoherent.”" <https://t.co/v4WIBW9Fld>

Bill Mitchell

English

I mean, Liberals are cheer-leading this #Coronavirus like it’s their high school football team.

#TDS

Helen Kennedy

English

Under repeated questioning, Pompeo refuses to say the #COVID19 coronavirus is not a hoax. <https://t.co/PFESx4ePpD>

Steve Herman

English

#coronavirus comments now from @larry\_kudlow here. <https://t.co/l45e8bUmBK>

QuickTake by Bloomberg

English

"“Are you going to see some schools shut down? Probably.”"

White House Chief of Staff Mick Mulvaney warned that the #coronavirus could disrupt U.S. schools as districts prepare their emergency plans <https://t.co/qADzyHNG4L>

Global Health Strategies

English

“Disease can emerge everywhere. Coronaviruses exist on a global basis. It’s important that we don’t ascribe blame to a geographic origin…the languge of stigma &amp; blame has become an unfortunate part of the global narrative”" @DrMikeRyan at @WHO briefing on #COVID19. #coronavirus <https://t.co/4IIvbTdN9j>

QuickTake by Bloomberg

English

“When the virus hit, I felt that I should stay to help.”

@selinawangtv connected with Li Wang. She’s part of a special force under China’s Uber @DidiGlobal that’s helping those stranded by the #coronavirus lockdown in #Wuhan #COVID2019 #<U+65B0><U+578B><U+80BA><U+708E> <https://t.co/CE5INLRC95>

NBC News World

English

NEW: WHO says it has increased its assessment of “the risk of spread and the risk of impact” of the coronavirus “to very high at a global level.” #COVID19

<https://t.co/dzmpKLK9mZ>

Radio Pakistan

English

President’s public awareness message about #Coronavirus <https://t.co/Qt2WFGqv5R> <https://t.co/pbafF9qVyy>

Mia Farrow

English

Harvard epidemiology professor Marc Lipsitch: “I think the likely outcome is that it will ultimately not be containable.” #Coronavirus - The Atlantic <https://t.co/hJVGqdsihO>

Summit County Health - Utah

English

The Summit County Health Department has issued the following statement about the #Coronavirus #COVID19 Outbreak: <https://t.co/maWQ4Adb4u>

There are NO confirmed cases in Summit County or Utah. #SummitCountyUT #ParkCity #Coalville #Kamas <https://t.co/Cg3VtMjO2R>

SIGNAL

English

Italian-Nigerian Footballer King Udoh, Contracts #Coronavirus <https://t.co/j67qRSIX13> <https://t.co/ofN8g8QrSN>

News18

English

#CoronavirusOutbreak | We are at a stage now that every country needs to be prepared. San Francisco has declared an emergency without a single case being reported: Dr. K.K. Aggarwal (Sr member of the Indian Medical Association) tells @AnchorAnandN on #TheRightStand

#COVID19 <https://t.co/wA4KUn8AuF>

Catherine Gasté

English

#Coronavirus outbreak: World Health Organization gives latest update on virus . Conférence de l’OMS | LIVE <https://t.co/C6SunFRpyS>

DT Next

English

Apart from that, several notifications were issued on how one could be affected by coronavirus, what the symptoms were and which preventive measures could be adopted. #India #Coronavirus

<https://t.co/AkUIJlCZNt>

newslaundry

English

Your #DailyDose of news has arrived!

@Memeghnad brings you the latest updates about #Delhi smoulders, #Coronavirus scare, new mini-moon, and more.

Listen up!

<https://t.co/htwPR8dUm2>

Texas Juvenile Justice

English

To keep up with the latest news about #coronavirus #COVID2019, visit @TexasDSHS where you’ll find everything you need to know.

<https://t.co/yqNjHAYE3C>

ANI

English

World Health Organisation (WHO): Our epidemiologists have been monitoring the developments continuously, and we have now increased our assessment of the risk of spread and the risk of impact of #COVID19 to very high at a global level. #coronavirus <https://t.co/yQrUvXIRpL>

GMA News

English

JUST IN: World Health Organization (WHO) upgrades global risk of #coronavirus spread to ‘very high’ | via @AFP <https://t.co/IjKMPEie9k>

# So text variable contains some Regex, such as \n for a new line and also \".  
covid$text[c(1,1)]

## [1] "Also the entire Swiss Football League is on hold. Postponing games from the professional and amateur level... #coronavirus https://t.co/UShMuqnAVC"  
## [2] "Also the entire Swiss Football League is on hold. Postponing games from the professional and amateur level... #coronavirus https://t.co/UShMuqnAVC"

#removing the \n as removePunctuation , URLs ,the & sign + converting the tweet text into ascii to remove emoji’s  
  
  
covid$text <- gsub('\"', " ", covid$text, fixed = TRUE)  
covid$text <- gsub('#', " ", covid$text, fixed = TRUE)  
  
covid$text <- str\_replace\_all(covid$text, "[\n]" , "") #remove new lines  
covid$text <- str\_replace\_all(covid$text, "&amp", "") # rm ampersand  
  
#URLs are always at the end and did not counts towards the 140 characters limit  
covid$text <- str\_replace\_all(covid$text, "http.\*" , "")  
  
covid$text <- iconv(covid$text, "latin1", "ASCII", sub="")  
  
covid$text[c(1,1)]

## [1] "Also the entire Swiss Football League is on hold. Postponing games from the professional and amateur level... coronavirus "  
## [2] "Also the entire Swiss Football League is on hold. Postponing games from the professional and amateur level... coronavirus "

##2- Creating a VCorpus object Corpora are collections of documents containing (natural language) text Corpus metadata contains corpus specific metadata in form of tag-value pairs

#names(covid)  
#names(all\_Tweets)  
#One column needs to have a unique document id (and must be named doc\_id), one column must be named ‘text’  
# we make that in advance  
  
all\_Tweets <- covid # %>% filter(Tweet\_Type=="Tweet" )  
  
all\_Corpus <- DataframeSource(all\_Tweets)  
all\_Corpus <- VCorpus(all\_Corpus)  
all\_Corpus

## <<VCorpus>>  
## Metadata: corpus specific: 0, document level (indexed): 22  
## Content: documents: 33174

content(all\_Corpus[[3]])

## [1] "I mean, Liberals are cheer-leading this Coronavirus like it's their high school football team. TDS"

##3 - # remove the English stopwords #convert all characters into lower characters (no more capitals) #remove numbers #remove all English stopwords. #remove punctuation #strip whitespaces # TermDocumentMatrix, which has all (remaining) terms as rows and all Documents (tweets) as columns #This is a processing step which involves creating a data frame where each term(i.e. each word from our reviews) is a column, and each review is a row, with

print(sort(stopwords("en")))

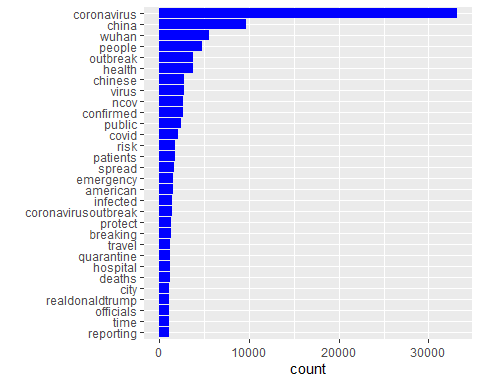
## [1] "a" "about" "above" "after" "again"   
## [6] "against" "all" "am" "an" "and"   
## [11] "any" "are" "aren't" "as" "at"   
## [16] "be" "because" "been" "before" "being"   
## [21] "below" "between" "both" "but" "by"   
## [26] "can't" "cannot" "could" "couldn't" "did"   
## [31] "didn't" "do" "does" "doesn't" "doing"   
## [36] "don't" "down" "during" "each" "few"   
## [41] "for" "from" "further" "had" "hadn't"   
## [46] "has" "hasn't" "have" "haven't" "having"   
## [51] "he" "he'd" "he'll" "he's" "her"   
## [56] "here" "here's" "hers" "herself" "him"   
## [61] "himself" "his" "how" "how's" "i"   
## [66] "i'd" "i'll" "i'm" "i've" "if"   
## [71] "in" "into" "is" "isn't" "it"   
## [76] "it's" "its" "itself" "let's" "me"   
## [81] "more" "most" "mustn't" "my" "myself"   
## [86] "no" "nor" "not" "of" "off"   
## [91] "on" "once" "only" "or" "other"   
## [96] "ought" "our" "ours" "ourselves" "out"   
## [101] "over" "own" "same" "shan't" "she"   
## [106] "she'd" "she'll" "she's" "should" "shouldn't"   
## [111] "so" "some" "such" "than" "that"   
## [116] "that's" "the" "their" "theirs" "them"   
## [121] "themselves" "then" "there" "there's" "these"   
## [126] "they" "they'd" "they'll" "they're" "they've"   
## [131] "this" "those" "through" "to" "too"   
## [136] "under" "until" "up" "very" "was"   
## [141] "wasn't" "we" "we'd" "we'll" "we're"   
## [146] "we've" "were" "weren't" "what" "what's"   
## [151] "when" "when's" "where" "where's" "which"   
## [156] "while" "who" "who's" "whom" "why"   
## [161] "why's" "with" "won't" "would" "wouldn't"   
## [166] "you" "you'd" "you'll" "you're" "you've"   
## [171] "your" "yours" "yourself" "yourselves"

CleanCorpus <- function(x){  
 x <- tm\_map(x, content\_transformer(tolower))  
 x <- tm\_map(x, removeNumbers) #remove numbers before removing words. Otherwise "trump2016" leaves "trump"  
 x <- tm\_map(x, removeWords, tidytext::stop\_words$word)  
 x <- tm\_map(x, removePunctuation)  
 x <- tm\_map(x, stripWhitespace)  
 return(x)  
}  
  
  
# Extra words removed   
RemoveNames <- function(x) {  
# x <- str\_replace\_all(x, "coronavirus" , "")  
 x <- tm\_map(x, removeWords, c("coronavirus"))  
 return(x)  
}  
  
  
# TermDocumentMatrix, which has all (remaining) terms as rows and all Documents (tweets) as columns  
  
CreateTermsMatrix <- function(x) {  
 x <- TermDocumentMatrix(x)  
 x <- as.matrix(x)  
 y <- rowSums(x)  
 y <- sort(y, decreasing=TRUE)  
 return(y)  
}  
  
all\_Corpus <- CleanCorpus(all\_Corpus)  
  
  
All\_tweeter\_Freq <- CreateTermsMatrix(all\_Corpus)  
  
  
  
content(all\_Corpus[[1]])

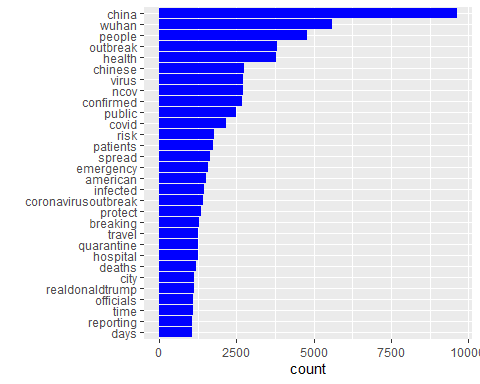
## [1] " entire swiss football league hold postponing games professional amateur level coronavirus "

##4- The words that Twitter used the most , make a Top20 of most used terms

all\_Word\_used <- data.frame(word=names(All\_tweeter\_Freq), count=All\_tweeter\_Freq)  
  
  
  
all\_Word\_used[1:30,] %>%  
 ggplot(aes(x=(reorder(word, count)), y=count)) +  
 geom\_bar(stat='identity', fill="blue") + coord\_flip() + theme(legend.position = "none") +  
 labs(x="")

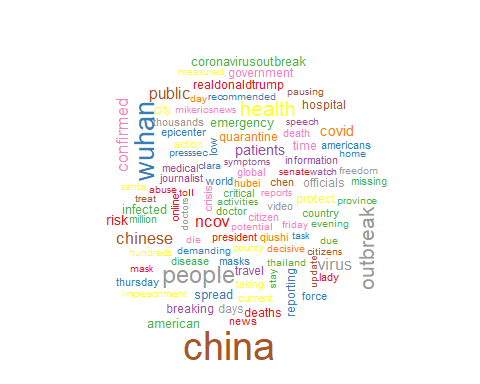


#As the names (especially ‘Coronavirus’ ) became very overwhelming in the wordcloud,   
# I removed the names (with the RemoveName function) from the corpus and created the Term Frequency Matrix again first.  
  
all\_Corpus <- RemoveNames(all\_Corpus)  
All\_tweeter\_Freq <- CreateTermsMatrix(all\_Corpus)  
all\_Word\_used <- data.frame(word=names(All\_tweeter\_Freq), count=All\_tweeter\_Freq)  
  
all\_Word\_used[1:30,] %>%  
 ggplot(aes(x=(reorder(word, count)), y=count)) +  
 geom\_bar(stat='identity', fill="blue") + coord\_flip() + theme(legend.position = "none") +  
 labs(x="")



##5 - standard” wordcloud

set.seed(2018)  
wordcloud(all\_Word\_used$word, all\_Word\_used$count, max.words = 100, scale=c(2.5,.5), random.color = TRUE, colors=brewer.pal(9,"Set1"))

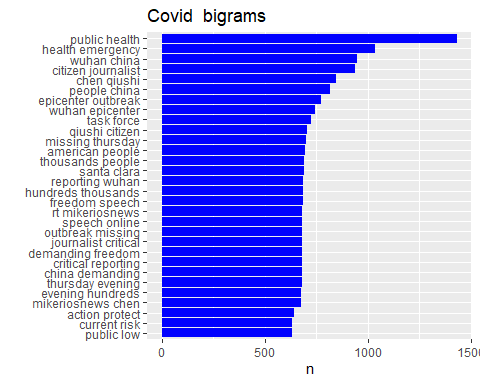


#This wordcloud is interactive and I really started to actually like wordclouds  
  
wordcloud2::wordcloud2(all\_Word\_used[1:100,], color = "random-light", backgroundColor = "grey", shuffle=FALSE, size=0.4)

## PhantomJS not found. You can install it with webshot::install\_phantomjs(). If it is installed, please make sure the phantomjs executable can be found via the PATH variable.

``` { r #Comparison cloud}

##6 - Bigrams : we use the tidy() method to construct a table with one row per document, including the metadata (such as id and datetimestamp) as columns alongside the text. becuase a corpus is a flexible storage method for documents, but doesn’t lend itself to processing with tidy tools  
  
  
```r  
# 6.1 Covid bigrams Top 30  
  
bigram\_all\_Tidy <- tidy(all\_Corpus)  
  
plotBigrams <- function(tibble, topN=30, title="", color="#FF1493"){  
 x <- tibble %>% select(text) %>%  
 unnest\_tokens(bigram, text, token = "ngrams", n = 2)  
 y <- x %>% count(bigram, sort = TRUE) %>% top\_n(topN, wt=n) %>%  
 ggplot(aes(x=reorder(bigram, n), y=n)) +  
 geom\_bar(stat='identity', fill=color) + coord\_flip() +  
 theme(legend.position="none") + labs(x="", title=title)  
}  
  
bi\_all <- plotBigrams(bigram\_all\_Tidy, title="Covid bigrams", color="blue")  
bi\_all



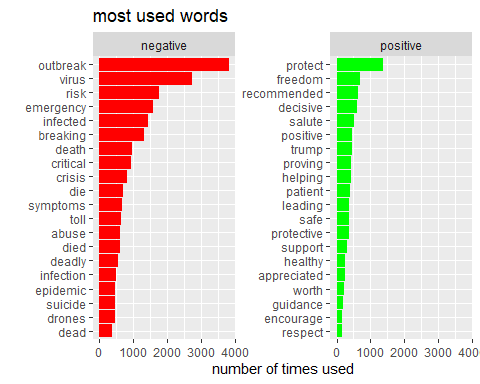
## Sentiment analysis

# ————– 1 The Bing lexicon (positive/negative, binary)

get\_sentiments("bing")

## # A tibble: 6,786 x 2  
## word sentiment  
## <chr> <chr>   
## 1 2-faces negative   
## 2 abnormal negative   
## 3 abolish negative   
## 4 abominable negative   
## 5 abominably negative   
## 6 abominate negative   
## 7 abomination negative   
## 8 abort negative   
## 9 aborted negative   
## 10 aborts negative   
## # ... with 6,776 more rows

# Positive and negative words used most frequently  
  
DocMeta\_all\_twittes <- meta(all\_Corpus)   
  
DocMeta\_all\_twittes$date <- date(DocMeta\_all\_twittes$date)  
  
bigram\_all\_Tidy$date <- DocMeta\_all\_twittes$date  
  
  
Words\_covid <- bigram\_all\_Tidy %>% unnest\_tokens(word, text)  
Bing\_covid <- Words\_covid %>% inner\_join(get\_sentiments("bing"), by="word")  
  
  
b1 <- Bing\_covid %>% count(word, sentiment, sort=TRUE) %>%  
 group\_by(sentiment) %>% arrange(desc(n)) %>% slice(1:20) %>%  
 ggplot(aes(x=reorder(word, n), y=n)) +  
 geom\_col(aes(fill=sentiment), show.legend=FALSE) +  
 coord\_flip() +  
 facet\_wrap(~sentiment, scales="free\_y") +  
 labs(x="", y="number of times used", title="most used words") +  
 scale\_fill\_manual(values = c("positive"="green", "negative"="red"))  
b1



#names(Bing\_covid)

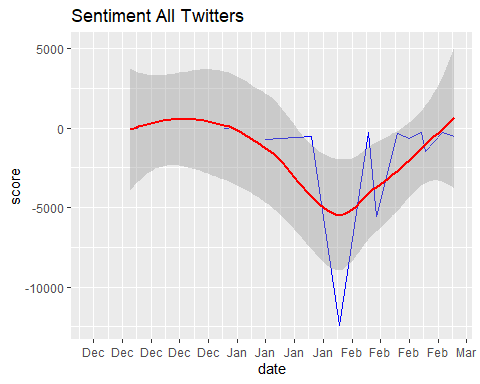
# ————– 2- Time series of sentiment

time\_all <- Bing\_covid %>% group\_by(date) %>% count(sentiment) %>%  
 spread(sentiment, n) %>% mutate(score=positive-negative) %>%  
 ggplot(aes(x=date, y=score)) +  
 scale\_x\_date(limits=c(as.Date("2019-12-01"), as.Date("2020-02-28")), date\_breaks = "1 week", date\_labels = "%b") +  
 geom\_line(stat="identity", col="blue") + geom\_smooth(col="red") + labs(title="Sentiment All Twitters")  
  
#+ geom\_smooth(col="red")   
  
Bing\_covid1 <- Bing\_covid %>% group\_by(date) %>% count(sentiment) %>%  
 spread(sentiment, n) %>% mutate(score=positive-negative)   
  
time\_all

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 9 rows containing non-finite values (stat\_smooth).

## Warning: Removed 2 row(s) containing missing values (geom\_path).

 # ————– 3- The AFFIN lexicon (positive/negative, with scores)

get\_sentiments("afinn")

## # A tibble: 2,477 x 2  
## word value  
## <chr> <dbl>  
## 1 abandon -2  
## 2 abandoned -2  
## 3 abandons -2  
## 4 abducted -2  
## 5 abduction -2  
## 6 abductions -2  
## 7 abhor -3  
## 8 abhorred -3  
## 9 abhorrent -3  
## 10 abhors -3  
## # ... with 2,467 more rows

Afinn\_covid <- Words\_covid %>% inner\_join(get\_sentiments("afinn"), by="word")  
t1 <- Afinn\_covid %>% select(id, date, word, value) %>% filter(date=="2020-02-28")  
 head(t1)

## # A tibble: 6 x 4  
## id date word value  
## <chr> <date> <chr> <dbl>  
## 1 3.16386707615403e-226 2020-02-28 postponing -1  
## 2 3.163835615731e-226 2020-02-28 questioning -1  
## 3 3.163835615731e-226 2020-02-28 hoax -2  
## 4 3.16378059713018e-226 2020-02-28 warned -2  
## 5 3.16378059713018e-226 2020-02-28 emergency -2  
## 6 3.16372945949632e-226 2020-02-28 blame -2

names(Afinn\_covid)[names(Afinn\_covid) == "value"] <- "score"  
a11 <- Afinn\_covid %>% group\_by(date) %>% summarise(score=sum(score))

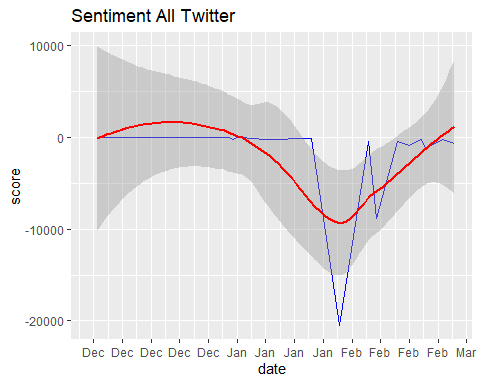
## `summarise()` ungrouping output (override with `.groups` argument)

a1 <- Afinn\_covid %>% group\_by(date) %>% summarise(score=sum(score)) %>%  
 ggplot(aes(x=date, y=score)) +  
 scale\_x\_date(limits=c(as.Date("2019-12-01"), as.Date("2020-02-287")), date\_breaks = "1 week", date\_labels = "%b") +  
 geom\_line(stat="identity", col="blue") + geom\_smooth(col="red") + labs(title="Sentiment All Twitter")

## `summarise()` ungrouping output (override with `.groups` argument)

a1

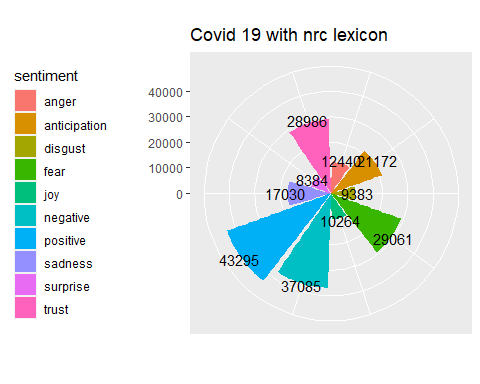
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

 # ————– 4 - The nrc lexicon (2 sentiment categories, and 8 basic emotions)

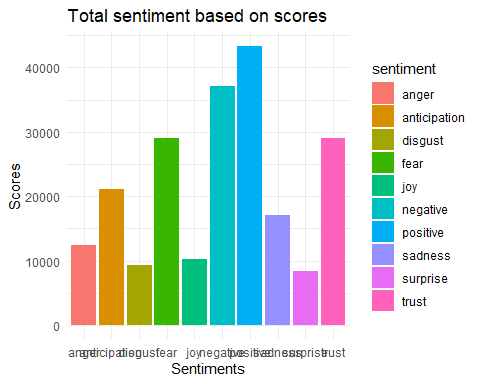
get\_sentiments("nrc")

## # A tibble: 13,901 x 2  
## word sentiment  
## <chr> <chr>   
## 1 abacus trust   
## 2 abandon fear   
## 3 abandon negative   
## 4 abandon sadness   
## 5 abandoned anger   
## 6 abandoned fear   
## 7 abandoned negative   
## 8 abandoned sadness   
## 9 abandonment anger   
## 10 abandonment fear   
## # ... with 13,891 more rows

Nrc\_covid <- Words\_covid %>% inner\_join(get\_sentiments("nrc"), by="word")  
  
n1 <- Nrc\_covid %>% count(sentiment) %>%  
 ggplot(aes(x=sentiment, y=n, fill=sentiment)) +  
 geom\_bar(stat="identity") + coord\_polar() +  
 theme(legend.position = "left", axis.text.x = element\_blank()) +  
 geom\_text(aes(label=n, y=n + 1000)) +  
 labs(x="", y="", title="Covid 19 with nrc lexicon")  
n1



# another plot for The nrc lexicon  
sentimentscores <- Nrc\_covid %>% count(sentiment)  
ggplot(data=sentimentscores,aes(x=sentiment,y=n))+  
 geom\_bar(aes(fill=sentiment),stat = "identity")+  
 theme(legend.position="none")+  
 xlab("Sentiments")+ylab("Scores")+  
 ggtitle("Total sentiment based on scores")+  
 theme\_minimal()



# Classification Tree

## Preperation the dataset

We are going to splitting the table into one-token-per-row by unnest\_tokens than join with The Bing lexicon sentament than make grouping with ID and Spread a key-value pair by spread function and callculating the score between positive and nigative and covert the score >0 positive sentiment. to True and <0 to negative sentiment. False the NA is neutral, but for simplicity purposes, we will only attempt to predict the positive and negative sentiment, and we will revisit neutral later This is because our goal is to train a model to recognize positive or negative language. we Have 16566 False and 4597 True

Words\_for\_reg <- covid %>% unnest\_tokens(word, text)  
#Bing\_covid\_for\_reg <- Words\_covid %>% inner\_join(get\_sentiments("bing"), by="word")  
  
#c  
Bing\_covid\_for\_reg <- Words\_for\_reg %>% inner\_join(get\_sentiments("bing"), by="word")  
covid\_with\_sent\_location <- Bing\_covid\_for\_reg  
  
names(Bing\_covid\_for\_reg)[names(Bing\_covid\_for\_reg) == "id"] <- "doc\_id"  
Bing\_covid\_for\_reg <- Bing\_covid\_for\_reg %>% group\_by(doc\_id) %>% count(sentiment)  
  
Bing\_covid\_for\_reg <- Bing\_covid\_for\_reg %>% spread(sentiment, n)  
Bing\_covid\_for\_reg[is.na(Bing\_covid\_for\_reg)] <- 0  
Bing\_covid\_for\_reg <- Bing\_covid\_for\_reg %>% mutate(score=positive-negative)  
Bing\_covid\_for\_reg <- Bing\_covid\_for\_reg %>% mutate(sent\_fin = ifelse(score > 0 , "TRUE",ifelse(score <0 , "FALSE" , "NA")))  
  
all\_twittes\_with\_sent <- merge(x = covid, y = Bing\_covid\_for\_reg, by = "doc\_id", all.x = TRUE)  
sum(is.na(all\_twittes\_with\_sent$sent\_fin)) # neutral

## [1] 7878

#all\_twittes\_with\_sent <- all\_twittes\_with\_sent %>% drop\_na()  
all\_twittes\_with\_sent <- all\_twittes\_with\_sent %>% filter(!sent\_fin=="NA")  
all\_twittes\_with\_sent$sent\_fin = as.factor(all\_twittes\_with\_sent$sent\_fin)  
table(all\_twittes\_with\_sent$sent\_fin)

##   
## FALSE TRUE   
## 15242 6468

# 2— Creating input variables

Clean Corpus by transforming it into lowercase, removing the puntuation and the common English words. build our Document Term Matrix. This is a processing step which involves creating a data frame where each term(i.e. each word from our reviews) is a column, and each review is a row, with corresponding values for the number of times each term appears in each review. We then remove infrequent terms and focus on the ones that appear in multiple reviews

corpus\_for\_reg = VCorpus(VectorSource(all\_twittes\_with\_sent$text))  
corpus\_for\_reg = CleanCorpus(corpus\_for\_reg)  
frequencies = DocumentTermMatrix(corpus\_for\_reg)  
sparse = removeSparseTerms(frequencies, 0.99)  
all\_twitter\_Sparse = as.data.frame(as.matrix(sparse))  
colnames(all\_twitter\_Sparse) = make.names(colnames(all\_twitter\_Sparse))

# now add our dependent variable back and let’s take a look at the result

all\_twitter\_Sparse$sent\_fin = all\_twittes\_with\_sent$sent\_fin  
  
names (all\_twitter\_Sparse)

## [1] "abuse" "action" "activities"   
## [4] "afp" "american" "americans"   
## [7] "amid" "announced" "appreciated"   
## [10] "authorities" "bay" "bioweapon"   
## [13] "bravetheworld" "breaking" "california"   
## [16] "call" "care" "cdc"   
## [19] "chen" "china" "chinese"   
## [22] "cities" "citizen" "citizens"   
## [25] "city" "clara" "confirmed"   
## [28] "congress" "constitution" "continue"   
## [31] "control" "corona" "coronavirus"   
## [34] "coronaviruschina" "coronavirusoutbreak" "countries"   
## [37] "country" "county" "covid"   
## [40] "crisis" "critical" "cruise"   
## [43] "current" "date" "day"   
## [46] "days" "dead" "deadly"   
## [49] "deal" "death" "deaths"   
## [52] "decisive" "declared" "declares"   
## [55] "demanding" "die" "died"   
## [58] "director" "disease" "doctor"   
## [61] "dont" "drericding" "due"   
## [64] "emergency" "engineered" "engineering"   
## [67] "entry" "epicenter" "epidemic"   
## [70] "epochtimeschina" "evening" "family"   
## [73] "fears" "feel" "fight"   
## [76] "fighting" "flights" "flu"   
## [79] "force" "foreign" "found"   
## [82] "freedom" "friday" "global"   
## [85] "government" "hands" "health"   
## [88] "healthy" "helping" "hiv"   
## [91] "home" "hong" "hospital"   
## [94] "hospitals" "howroute" "hubei"   
## [97] "humanity" "hundreds" "immediately"   
## [100] "impeachment" "india" "infected"   
## [103] "infection" "information" "inserts"   
## [106] "international" "iran" "jan"   
## [109] "january" "japan" "jenniferatntd"   
## [112] "journalist" "kong" "leading"   
## [115] "leave" "link" "loudobbs"   
## [118] "low" "major" "market"   
## [121] "mask" "masks" "measured"   
## [124] "media" "medical" "mikeriosnews"   
## [127] "million" "missing" "national"   
## [130] "ncov" "news" "official"   
## [133] "officials" "online" "original"   
## [136] "outbreak" "panic" "patient"   
## [139] "patients" "pausing" "people"   
## [142] "person" "placing" "positive"   
## [145] "posted" "president" "presssec"   
## [148] "protect" "protein" "province"   
## [151] "public" "pulled" "qiushi"   
## [154] "quarantine" "quarantined" "racism"   
## [157] "real" "realdonaldtrump" "recommended"   
## [160] "report" "reported" "reporting"   
## [163] "reports" "response" "rest"   
## [166] "reuters" "risk" "safe"   
## [169] "santa" "sars" "scientists"   
## [172] "senate" "ship" "shut"   
## [175] "sick" "sign" "social"   
## [178] "source" "special" "speech"   
## [181] "spread" "spreading" "stay"   
## [184] "steps" "stop" "stopthecoup"   
## [187] "support" "symptoms" "taking"   
## [190] "task" "test" "tested"   
## [193] "thousands" "thursday" "time"   
## [196] "toll" "tomfitton" "total"   
## [199] "travel" "trump" "understand"   
## [202] "update" "video" "virus"   
## [205] "watch" "week" "woman"   
## [208] "world" "wuhan" "wuhancoronavirus"   
## [211] "yesterday" "sent\_fin"

# Sentiment Analysis

## will start wil splitting the data into a training and testing set.

split = sample.split(all\_twitter\_Sparse$sent\_fin, SplitRatio = 0.7)  
all\_twitter\_Sparse$split = split  
train = subset(all\_twitter\_Sparse, split==TRUE)  
test = subset(all\_twitter\_Sparse, split==FALSE)  
table(train$sent\_fin)

##   
## FALSE TRUE   
## 10669 4528

10669 /nrow(train)

## [1] 0.7020465

# We then calculate our baseline accuracy that the model will have to surpass 70%

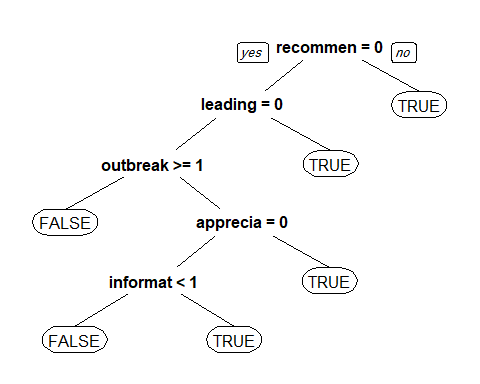
## Therefore the baseline accuracy is around 78%. This means that more than 3 quarters of all twitters are nigative Hence the dataset is biased towards nigative sentiment and the machine learning algorithm will also be more likely to predict nigative sentiment of tweets.

# Classification Tree

## build a CART model (this stands for Classification and Regression Trees, in this case it will be classification).

## it’s much more interpretable and can be visualized. In this way we will be able to have a look at which words were treated as predictors.

cartModel = rpart(sent\_fin ~ ., data=train, method="class")  
prp(cartModel)

 # if the twitter contains patient , it will be immediately labelled as negative. Words such as protects lead to positive twitter

# evaluate the performance of CART.

predictCART = predict(cartModel, newdata=test, type="class")  
table(test$sent\_fin, predictCART)

## predictCART  
## FALSE TRUE  
## FALSE 4536 37  
## TRUE 1443 497

(4535 + 482)/nrow(test)

## [1] 0.7703055

# The CART model has an 7% improvement over the baseline model which is a highly nigative sign

##Model Enhancement - Cross Validation

numFolds=trainControl(method = "cv", number = 10)  
cpGrid = expand.grid(.cp=seq(0.001, 0.01, 0.001))  
train(sent\_fin ~ ., data = train, method = "rpart", trControl = numFolds, tuneGrid = cpGrid)

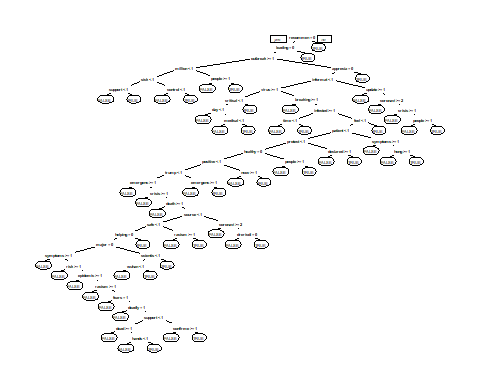
## CART   
##   
## 15197 samples  
## 212 predictor  
## 2 classes: 'FALSE', 'TRUE'   
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold)   
## Summary of sample sizes: 13678, 13677, 13677, 13677, 13677, 13677, ...   
## Resampling results across tuning parameters:  
##   
## cp Accuracy Kappa   
## 0.001 0.8505627 0.5949264  
## 0.002 0.8451669 0.5806165  
## 0.003 0.8360205 0.5530046  
## 0.004 0.8283872 0.5285356  
## 0.005 0.8108868 0.4644049  
## 0.006 0.8090446 0.4579741  
## 0.007 0.8018731 0.4302857  
## 0.008 0.7945656 0.4007269  
## 0.009 0.7943023 0.3998714  
## 0.010 0.7785785 0.3397954  
##   
## Accuracy was used to select the optimal model using the largest value.  
## The final value used for the model was cp = 0.001.

# cp Accuracy Kappa

0.001 0.9017154 0.6739517 0.002 0.8980705 0.6586304 0.003 0.8941561 0.6445860

# The cross validation gave me the optimal parameter cp = 0.001 so I will now re-build the tree with this parameter

cartModelImproved = rpart(sent\_fin ~ ., data=train, method="class", cp= 0.001)  
prp(cartModelImproved)

 # gives us a good overview of which words are used to make the split decisions and hence which words contribute most to the positive/negative sentiment.

# Let’s now obtain predictions using the new tree

predictCARTImproved = predict(cartModelImproved, newdata=test, type="class")  
table(test$sent\_fin, predictCARTImproved)

## predictCARTImproved  
## FALSE TRUE  
## FALSE 4485 88  
## TRUE 882 1058

(4426+1132)/nrow(test)

## [1] 0.8533702

# 85%

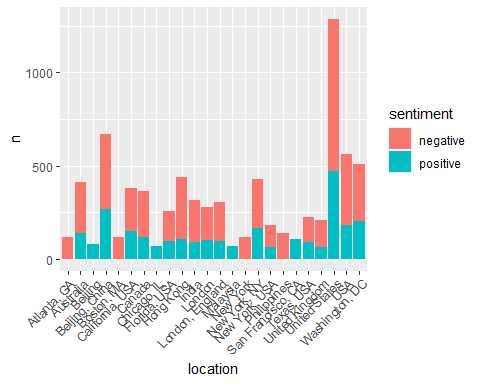
# 

Words\_for\_reg <- covid %>% unnest\_tokens(word, text)  
covid\_with\_sent\_location <- Words\_for\_reg %>% inner\_join(get\_sentiments("bing"), by="word")  
  
covid\_with\_sent\_location <- covid\_with\_sent\_location %>% group\_by(location,sentiment ) %>%   
 count() %>% arrange(desc(n))   
covid\_with\_sent\_location <- covid\_with\_sent\_location %>% filter(!location=="")  
  
  
a11 <- covid\_with\_sent\_location %>%  
 arrange\_(~ desc(n)) %>%  
 group\_by\_(~ sentiment) %>%  
 slice(1:20)

## Warning: `arrange\_()` is deprecated as of dplyr 0.7.0.  
## Please use `arrange()` instead.  
## See vignette('programming') for more help  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

## Warning: `group\_by\_()` is deprecated as of dplyr 0.7.0.  
## Please use `group\_by()` instead.  
## See vignette('programming') for more help  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

ggplot(a11, aes(x=location, y=n, fill=sentiment)) + geom\_bar(stat='identity') + theme(axis.text.x=element\_text(angle=45,hjust=1))



#textplot\_network(a11, min\_freq = 0.5, edge\_color = "orange", edge\_alpha = 0.8, edge\_size = 5)