Homework-6

2022-12-02

```
library(tidyr)
library(ggplot2)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

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

```
## Loaded glmnet 4.1-4
```

```
library(dplyr)
```

```
##
## 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(stringr)
library(tokenizers)
library(lubridate)
```

```
## Loading required package: timechange
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v tibble  3.1.8     v purrr   0.3.5
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x Matrix::expand()         masks tidyr::expand()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x Matrix::pack()           masks tidyr::pack()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
## x Matrix::unpack()         masks tidyr::unpack()
```

```
library(readr)
library(tidytext)
```

Part-A

Problem-1

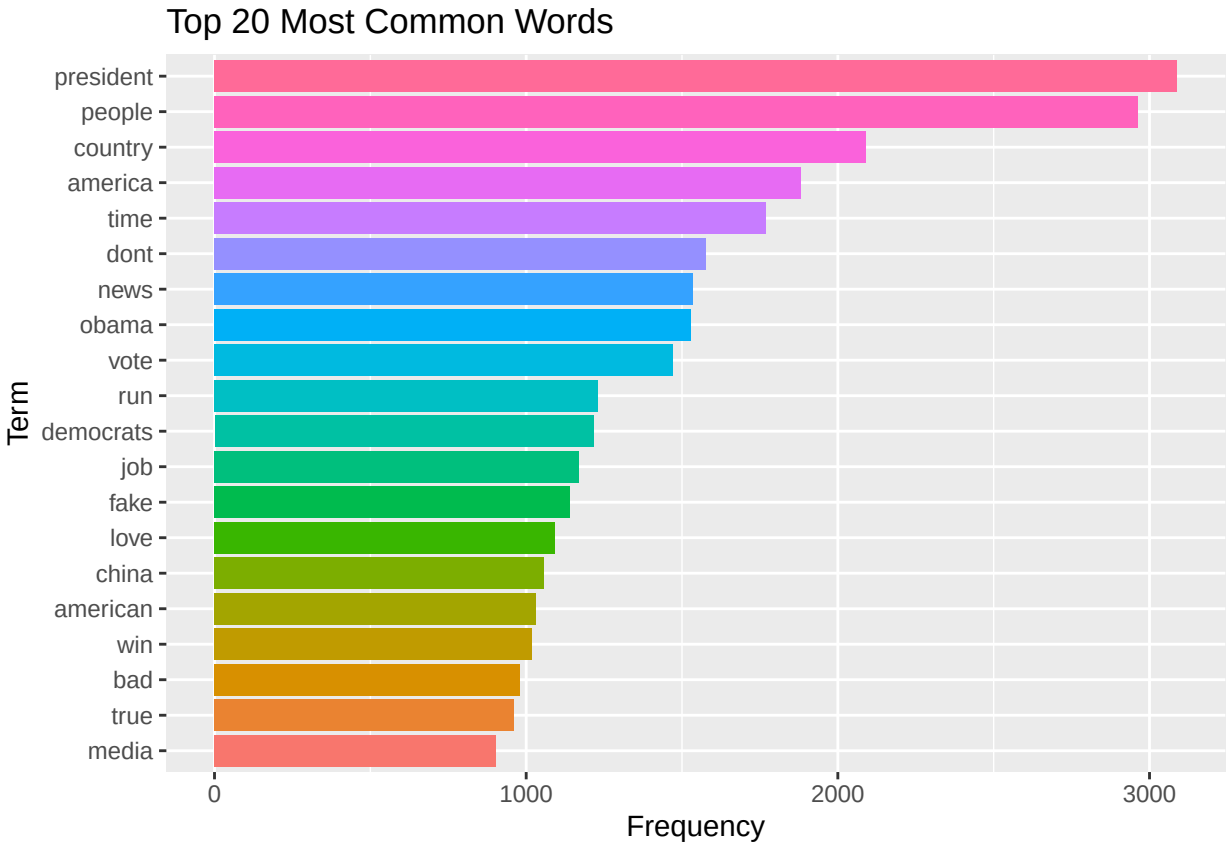
```
#read the data
data <- read_csv("S:/IDMP/Assignments/Assignment6/twitter/twitter/realDonaldTrump-20201106.csv", col_ty

# tidy the data
tidy_data <- data %>% filter(!isRetweet, str_detect(text,"[:space:]"))
# remove stop words and '&amp'. Although removing amp removes that too.
trump_names <- c("donaldtrump", "donald", "trump","realdonaldtrump", "amp","$")
tidy_data <- tidy_data %>%
  unnest_tokens("word", text, token="tweets", to_lower=TRUE) %>%
  anti_join(stop_words, by="word")
```

```
## Using ‘to_lower = TRUE‘ with ‘token = ’tweets’‘ may not preserve URLs.
```

```
tidy_data <- tidy_data %>%
  filter(!word %in% trump_names,
  !str_detect(word, "http"),
  !str_detect(word, "@"))

tidy_data$word <- sapply(tidy_data$word, function(x) gsub("[^\x01-\x7F]",NA_character_, x))
tidy_data <- na.omit(tidy_data)
tidy_data %>% count(word) %>%
  top_n(20) %>%
  ggplot(aes(x=reorder(word, n), y=n, fill=reorder(word, n))) +
```

```
  geom_col(show.legend=FALSE)+
  coord_flip() +
  labs(x="Term", y="Frequency", title="Top 20 Most Common Words")
```

```
## Selecting by n
```



From The above visualization we can see the Top 20 tweets by Donald Trump with President being the most common tweet and other tweet are more or less related to politics too.

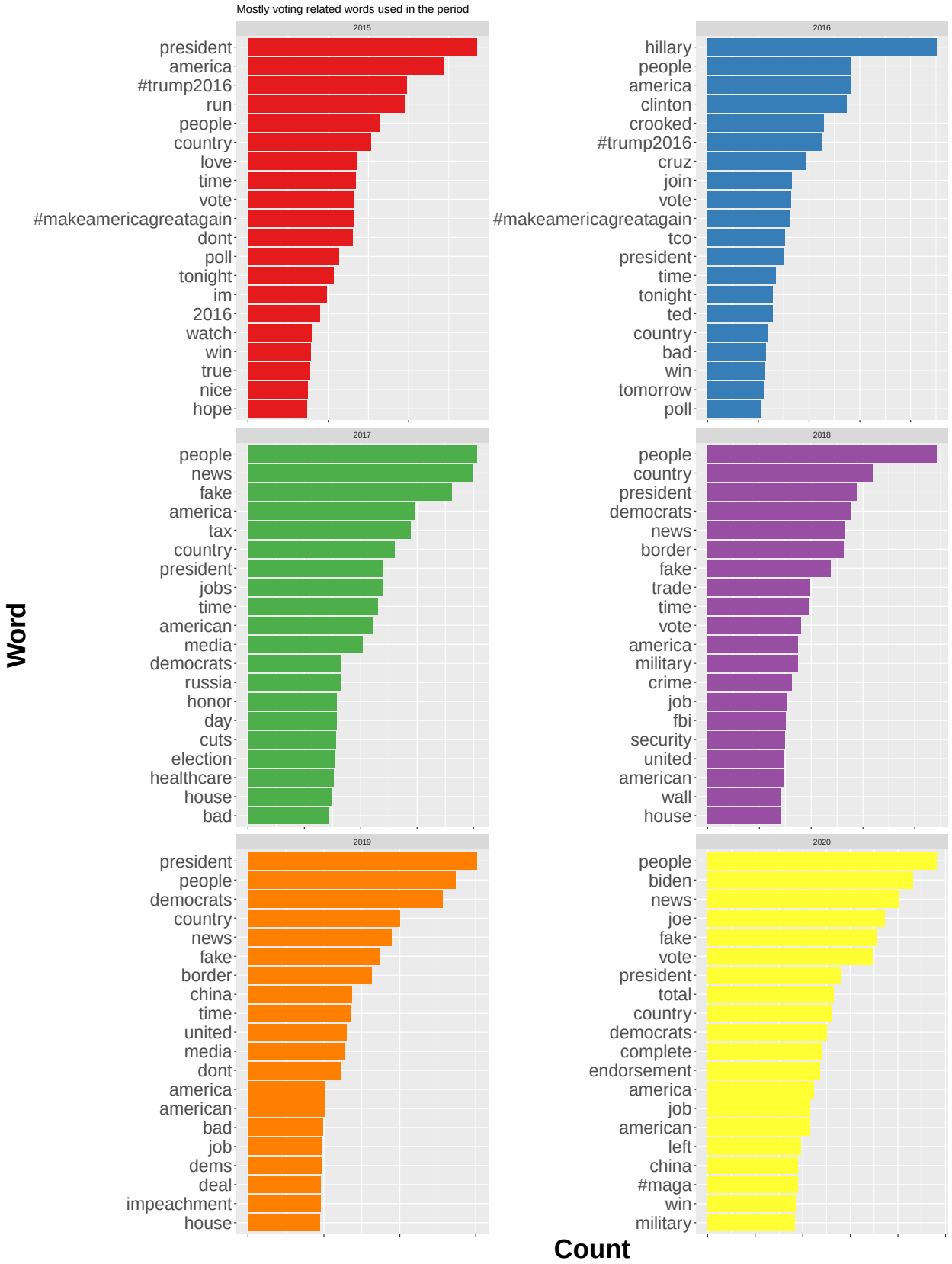
Problem-2

Lets visualize Donald Trump's tweets for each year from 2015-2020

```
tidy_data %>% mutate(year=year(date)) %>%
  filter(year %in% 2015:2020)%>%
  count(word, year) %>%
  group_by(year) %>%
  top_n(20) %>%
  ggplot(aes(x=reorder_within(word, n, year), y=n, fill=as.factor(year))) +
  geom_col(show.legend=FALSE) +
  coord_flip() +
  scale_x_reordered()+
  facet_wrap(~year, scales="free", ncol=2)+
  scale_fill_brewer(palette="Set1")+
```

```
  scale_y_continuous(labels=NULL)+
  labs(x="Word", y="Count", title="Mostly voting related words used in the period") +theme(axis.text=el
        axis.title=element_text(size=30,face="bold"))
```

```
## Selecting by n
```



As we can see from the above plot Donald Trump's tweet was mostly a about Make America Great Again pre election and Hillary Clinton. Post victory China, jobs and democrats were among the top topics Trump

tweeted about Finally pre election we can see MAGA popping up again along with Joe (President, Joe Biden)

Problem 3

```
years <- tidy_data %>%
  mutate(year=year(date)) %>%
  filter(year %in% 2015:2020) %>%
  count(word, year) %>%
  bind_tf_idf(word, year, n)

years %>%
  group_by(year) %>%
  top_n(20, tf_idf) %>%
  ggplot(aes(x=reorder_within(word, tf_idf, year),
  y=tf_idf, fill=as.factor(year))) +

  geom_col(show.legend = FALSE) +
  coord_flip() +
  facet_wrap(~year, scale="free", ncol=2)+
  scale_x_reordered() +
  scale_y_continuous(labels=NULL) +
  labs(title="",x="Word", y="tf_idf")+ theme(axis.text=element_text(size=20),
        axis.title=element_text(size=30,face="bold"))
```

make america great again trump2016 being most tweeted was not a surprise since 2015-16 was the election

rally time. Following years the policies which Trump brought were tweeted other than Fake news, Trump used to call out fake news a lot in his tweets. To no ones surprise Covid, Coronavirus were most talked about Trump in 2020.

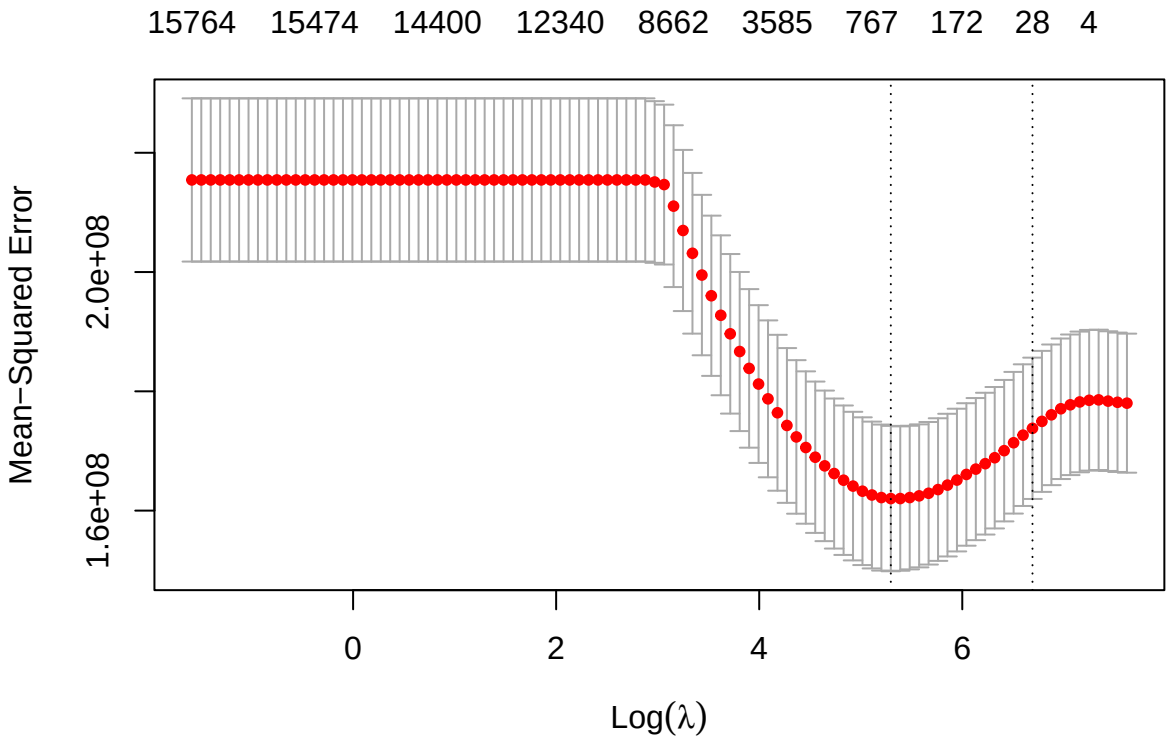
Part-B

Problem-4

```
set.seed(30)
#excluding 2015
tidy_data <- tidy_data %>%
  filter(year(date)>=2016) %>%
  count(id, word) %>%
  cast_sparse(id, word, n)

tweetIds <- tibble(id=rownames(tidy_data))

#join by id
tweetIds <- tweetIds %>% left_join(data)
```

```
## Joining, by = "id"
```

```
rt <- tweetIds$retweets

fit1 <- cv.glmnet(tidy_data, rt)
plot(fit1)
```




```
fit1
```

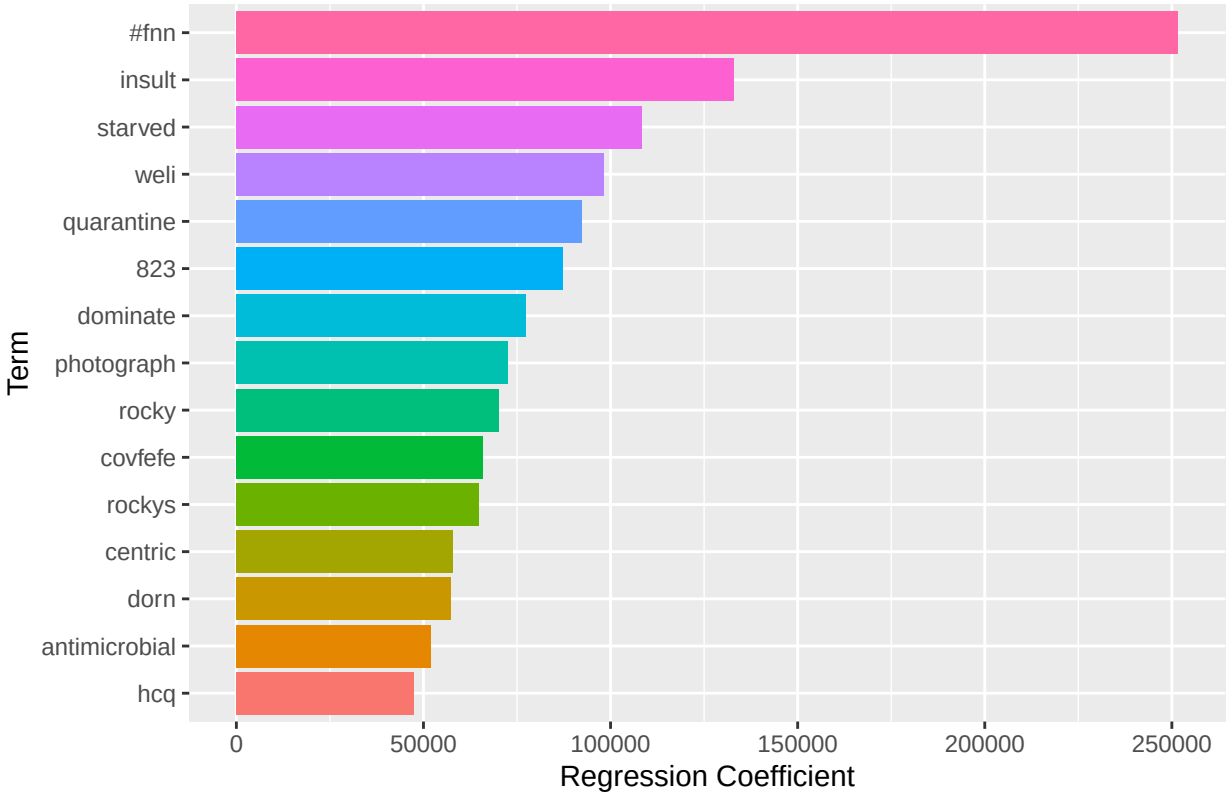
```
##
## Call:  cv.glmnet(x = tidy_data, y = rt)
##
## Measure: Mean-Squared Error
##
##     Lambda Index   Measure       SE Nonzero
## min  199.6    26 162016875 12185302     557
## 1se  805.7    11 173798335 11847642      28
```

As per Mean Squared Error, the best model has lambda 199.6 and uses 557 terms.

Problem 5

```
coef_ <- coef(fit1, s="lambda.min")
coef_ <- tibble(word=rownames(coef_), coef=as.numeric(coef_))
coef_ %>% top_n(15) %>%
ggplot(aes(x=reorder(word, coef), y=coef, fill=reorder(word, coef))) +
geom_col(show.legend=FALSE) +
coord_flip() +
labs(x="Term",y="Regression Coefficient", title="")
```

```
## Selecting by coef
```



The re-tweets with the highest regression coefficient have "#fnn" (Fox News Networks) in them