Donald Trump Speeches Sentiment Analysis- Rohan Chouthai

Code

How exactly did Donald Trump surprise the world and became the President of US? Let us take a look at what is it that he said to the people of US that appealed the most to them.

I will perform a sentiment analysis of all the 56 speeches Trump gave across US en route to the Oval office.

Let us load the libraries we are going to use.

Hide

```
suppressWarnings(library(readr))
suppressWarnings(library(tidyverse))
suppressWarnings(library(stringr))
suppressWarnings(library(tidytext))
suppressWarnings(library(tm))
```

Now, let us start by loading the text file of Trump's speeches.

Hide

```
Trump_speeches<-read_lines("C:/Users/rohan/Desktop/DMP/Assignment 3/full_speech.txt")
```

Create a corpus of all the speeches in the text file

Hide

```
Trump_corpus<-VCorpus(VectorSource(Trump_speeches))
print(Trump_corpus)
```

```
<<VCorpus>>
Metadata:  corpus specific: 0, document level (indexed): 0
Content:  documents: 74
```

We need to first clean the speeches of the common stop words, white spaces and punctuation marks.

Let us start of by converting all the upper case letters to lower case letters.

Hide

```
Trump_corpus_clean<-tm_map(Trump_corpus,content_transformer(tolower))
```

Let us now continue to remove all the numbers.

Hide

```
Trump_corpus_clean<-tm_map(Trump_corpus_clean,removeNumbers)
```

Next, we will remove all the stop words.

Hide

```
Trump_corpus_clean<-tm_map(Trump_corpus_clean,removeWords,stopwords())
```

We also want to remove the word “Applause”. Let us do that now.

Hide

```
Trump_corpus_clean<-tm_map(Trump_corpus_clean,removeWords,"applause")
```

Let us now move forward and remove the punctuation marks.

Hide

```
Trump_corpus_clean<-tm_map(Trump_corpus_clean,removePunctuation)
```

Let us now standardize the text by stemming.

Hide

```
library(SnowballC)
#Trump_corpus_clean<-tm_map(Trump_corpus_clean,stemDocument)
```

Finally, let us remove the white spaces in our corpus.

Hide

```
Trump_corpus_clean<-tm_map(Trump_corpus_clean,stripWhitespace)
```

Let us now make a Document Term Matrix of our corpus.

Hide

```
Trump_dcm<-DocumentTermMatrix(Trump_corpus_clean)
Trump_dcm
```

```
<<DocumentTermMatrix (documents: 74, terms: 8193)>>
Non-/sparse entries: 53339/552943
Sparsity           : 91%
Maximal term length: 22
Weighting          : term frequency (tf)
```

Now, let us remove the sparce terms from our DCM.

Hide

```
Trump_dcm_sparce<-removeSparseTerms(Trump_dcm,.5)
Trump_final<-as.data.frame(as.matrix(Trump_dcm_sparce))
dim(Trump_final)
```

```
[1]  74 300
```

Hide

```
Trump_final[,1:10]
```

	accomplish <dbl>	acro… <dbl>	administration <dbl>	africanamerican <dbl>	… <dbl>	al… <dbl>	alw… <dbl>	amazi… <dbl>	amend… <dbl>
1	0	2	2	0	4	1	4	1	2
2	0	0	1	0	0	2	0	0	0
3	0	0	6	0	1	11	4	0	0
4	0	1	2	0	1	7	0	0	1
5	0	5	1	2	1	8	1	2	0
6	0	5	3	0	0	7	1	0	0
7	0	0	3	0	1	5	1	5	4
8	1	3	3	2	3	8	2	3	2
9	0	0	0	0	13	6	2	1	1
10	1	2	0	0	9	5	4	0	3

1-10 of 74 rows | 1-10 of 10 columns Previous **1** 2 3 4 5 6 … 8 Next

Now let us plot the 15 most common words that featured in Trump's speeches.

Hide

```
Most_common<-colSums(Trump_final)
sort(Most_common,decreasing = TRUE)[1:15]
```

```
    will    going   people  country  hillary  clinton     jobs american      one     know    gre
at  america      new
    2522     1985     1421     1154      999      899      805      796      770      688      6
66      632      583
    said     just
     579      574
```

We will make a wordcloud of these words now.

Hide

```
wordcloud::wordcloud(names(Most_common),Most_common,random.order = FALSE,max.words = 15,colors =
 blues9)
```

Now, let us get the word frequencies in a dataframe so that we can plot them.

Hide

```
Word_freq<-as.data.frame(Most_common)
head(Word_freq)
```

	Most_common <dbl>
accomplish	51
across	102
administration	191
africanamerican	148
ago	120
also	435

6 rows

Hide

```
Word_freq$Word<-rownames(Word_freq)
rownames(Word_freq)<-c()
Word_freq<-Word_freq[c(2,1)]
colnames(Word_freq)<-c("Word","Frequency")
head(Word_freq)
```

	Word <chr>	Frequency <dbl>
1	accomplish	51
2	across	102
3	administration	191
4	africanamerican	148
5	ago	120
6	also	435

6 rows

Hide

```
Word_freq$Word<-as.factor(Word_freq$Word)
```

Let us now create a bar plot of the top 15 most commonly used words by Trump.

Hide

```
str(Word_freq)
```

```
'data.frame':   300 obs. of  2 variables:
 $ Word     : Factor w/ 300 levels "accomplish","across",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Frequency: num  51 102 191 148 120 435 72 91 89 632 ...
```

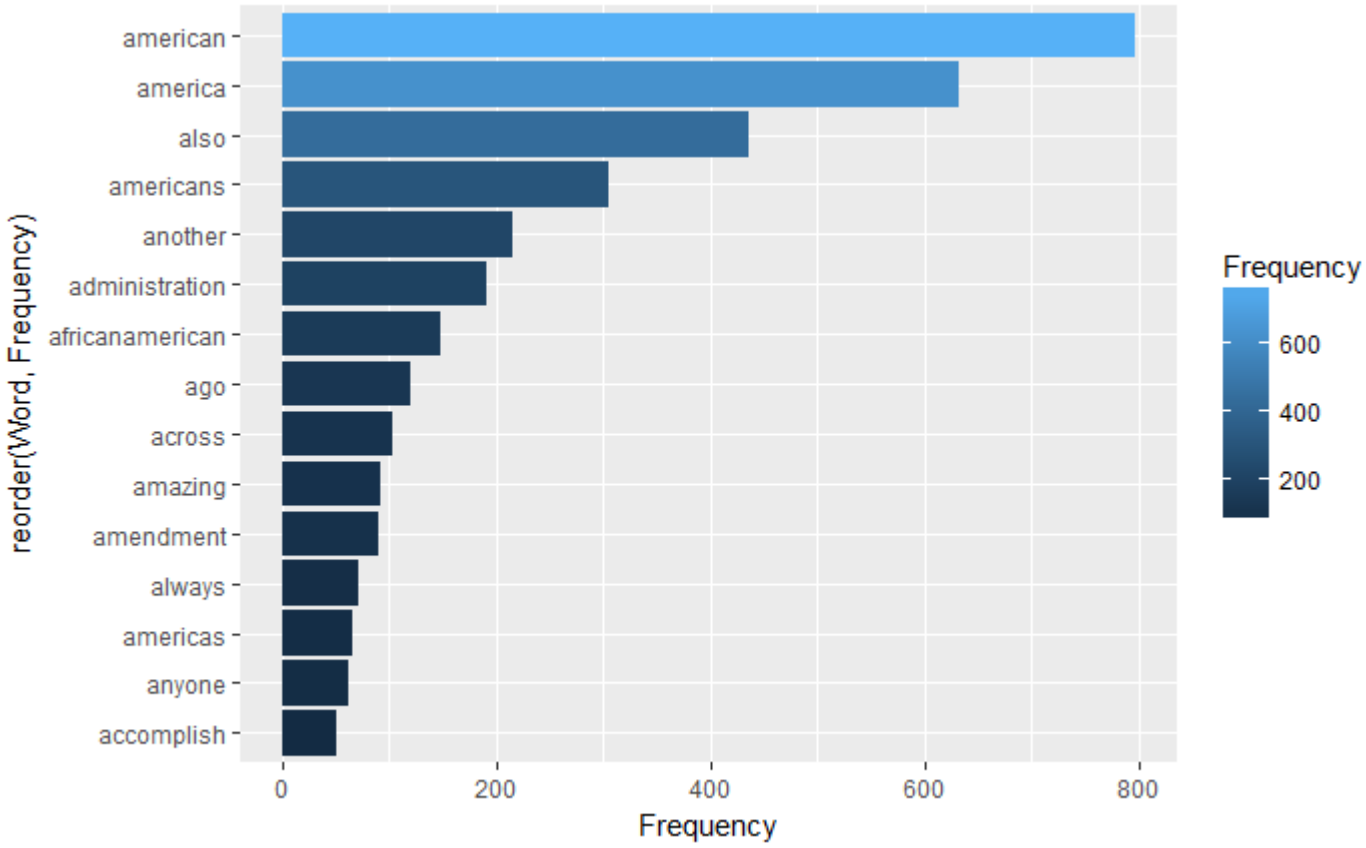
Hide

```
Top_15<-Word_freq[1:15,]
dim(Top_15)
```

```
[1] 15  2
```

Hide

```
ggplot(Top_15)+geom_bar(mapping = aes(reorder(Word,Frequency),Frequency,fill=Frequency),stat =
"identity")+coord_flip()
```



PART B

In this part, I will: Re-tokenize the text of all 56 Donald Trump Speeches into a new tidy text data frame, using bigrams as tokens. Remove each bigram where either word is a stop word or the word “applause”. Then plot the top 15 most common bigrams in Trump’s speeches.

For the part A, I have used the corpus format. Now, I will use the tidytext format.

Hide

```
Trump_s<-tibble(line=1:length(Trump_speeches),text=Trump_speeches)
Trump_s
```

line <int>
1
2
3
4
5
6
7
8

<int>
9
10

1-10 of 74 rows | 1-1 of 2 columns Previous **1** 2 3 4 5 6 … 8 Next

Hide

```
dim(Trump_s)
```

```
[1] 74  2
```

Now, let us tidy the data.

Hide

```
tidy_speeches<-Trump_s%>% unnest_tokens(word,text)
tidy_speeches
```

line <int>	word <chr>
1	trump
1	wow
1	whoa
1	that
1	is
1	some
1	group
1	of
1	people
1	thousands

1-10 of 235,237 rows Previous **1** 2 3 4 5 6 … 100 Next

Now, let us first see the most common words. This is essentially an easier way to solve the problem 6.

Before we proceed, let us first remove the common stop words.

Hide

```
new_list<-c("Applause")
new_list<-as.data.frame(new_list)
new_list$lexicon<-c("SMART")
colnames(new_list)<-c("word","lexicon")
new_list$word<-as.character(new_list$word)
stop_words<-rbind(stop_words,new_list)
tidy_speeches_imp<-tidy_speeches%>% anti_join(stop_words,by="word")%>% count(word,sort = TRUE)
```

```
package <U+393C><U+3E31>bindrcpp<U+393C><U+3E32> was built under R version 3.4.3
```

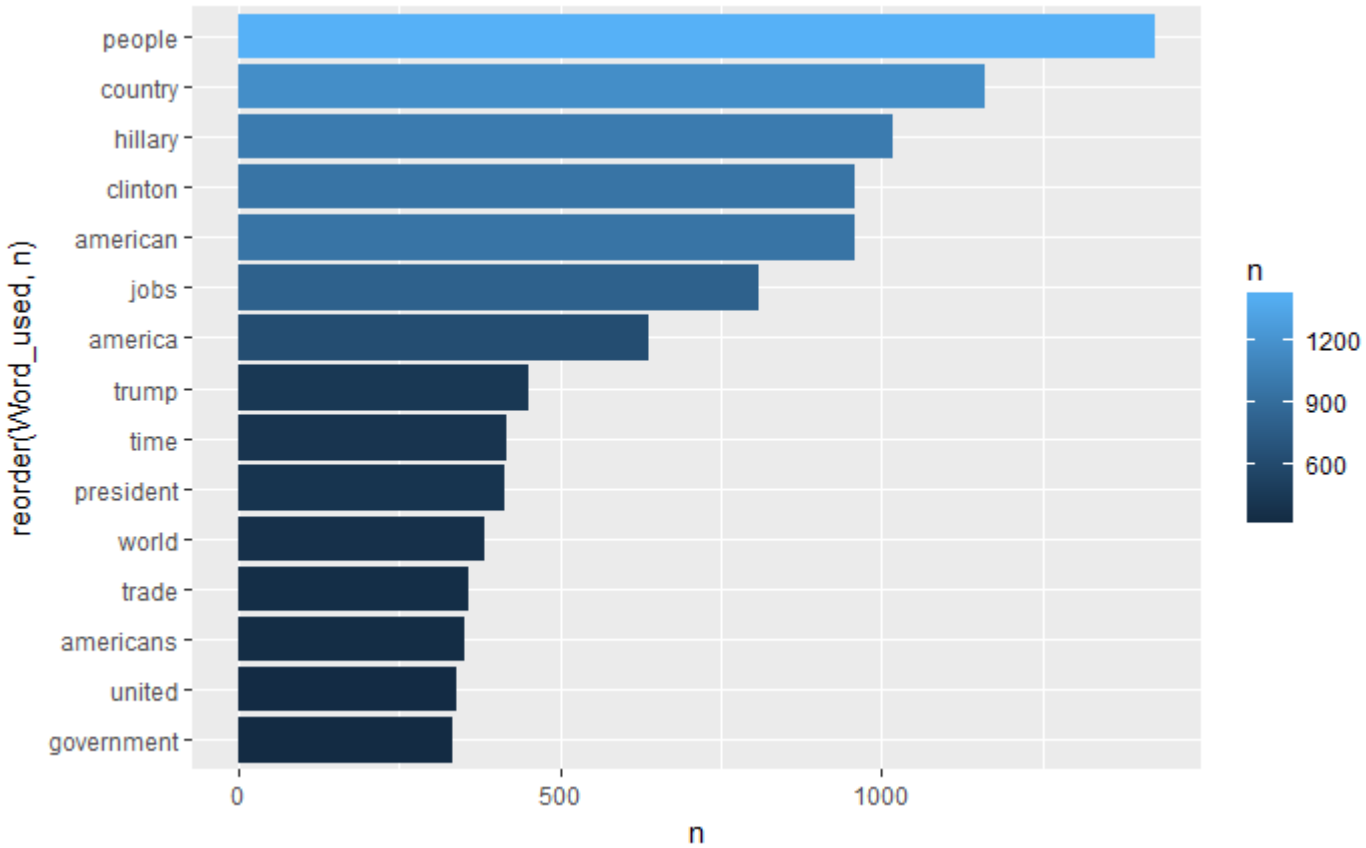
Hide

```
tidy_speeches_final<-tidy_speeches_imp[-3,] #removing the word applause
colnames(tidy_speeches_final)<-c("Word_used","n")
```

Now, let us plot the top 15 most used words.

Hide

```
Top15_words<-tidy_speeches_final[1:15,]
ggplot(Top15_words)+geom_bar(mapping = aes(reorder(Word_used,n),n,fill=n),stat = "identity")+coo
rd_flip()
```



Now, let us create a bigram of the Trump speeches. We will do so to get a deeper context in which the words were actually used.

Hide

```
Trump_bigrams<-Trump_s%>% unnest_tokens(bigram,text,token = "ngrams",n=2)
Trump_bigrams
```

line <int>	bigram <chr>
1	trump wow
1	wow whoa
1	whoa that
1	that is
1	is some
1	some group
1	group of
1	of people
1	people thousands
1	thousands so

1-10 of 235,163 rows Previous **1** 2 3 4 5 6 … 100 Next

Hide

```
tidy_speeches_big<-tidy_speeches%>% anti_join(stop_words,by="word")
str(tidy_speeches_big)
```

```
Classes □tbl_df□, □tbl□ and 'data.frame':   85814 obs. of  2 variables:
 $ line: int  1 1 1 1 1 1 1 1 1 1 ...
 $ word: chr  "trump" "wow" "whoa" "people" ...
```

Hide

```
str(Trump_s)
```

```
Classes □tbl_df□, □tbl□ and 'data.frame':   74 obs. of  2 variables:
 $ line: int  1 2 3 4 5 6 7 8 9 10 ...
 $ text: chr  "Trump: Wow. Whoa. That is some group of people. Thousands. So nice, thank you ver
y much. That's really nice. Th"| __truncated__ " Good evening. Thank you very much. I speak to y
ou today as a lifelong supporter and true friend of Israel. I'm"| __truncated__ "Thank you for t
he opportunity to speak to you, and thank you to the Center for the National Interest for honor
i"| __truncated__ "Thank you for joining me today. This was going to be a speech on Hillary Clin
ton and how bad a President, espec"| __truncated__ ...
```

Now let us check the most common bigrams.

Hide

```
Trump_bigrams<-Trump_bigrams%>%count(bigram,sort = TRUE)
Trump_bigrams
```

bigram <chr>	n <int>
going to	1821
of the	987
we will	819
we are	710
in the	669
hillary clinton	663
our country	575
are going	558
to be	538
we have	473

1-10 of 76,867 rows Previous **1** 2 3 4 5 6 … 100 Next

We can see that there are still stop words in the bigrams which we should eliminate.

For that, we will first split the bigrams into two words, eliminate the stop words and then reunite the words to form a bigram again.

Hide

```
library(tidyr)
Trump_bigrams_sep<-Trump_bigrams%>%separate(bigram,c("word1","word2"),sep = " ")
Trump_bigrams_sep
```

	word1 <chr>	word2 <chr>	n <int>
1	going	to	1821
2	of	the	987
3	we	will	819
4	we	are	710
5	in	the	669
6	hillary	clinton	663
7	our	country	575
8	are	going	558

			n
	<chr>	<chr>	<int>
9	to	be	538
10	we	have	473

1-10 of 76,867 rows Previous **1** 2 3 4 5 6 … 100 Next

Hide

```
Trump_bigrams_sep<-Trump_bigrams_sep%>% filter(!word1 %in% stop_words$word)%>%filter(!word2 %in%
 stop_words$word)
Trump_bigrams_sep
```

word1	word2	n
<chr>	<chr>	<int>
hillary	clinton	663
donald	trump	172
african	american	164
american	people	119
trump	administration	112
hillary	clinton's	104
trade	deals	102
november	8th	91
middle	east	88
president	obama	87

1-10 of 14,257 rows Previous **1** 2 3 4 5 6 … 100 Next

Now let us unite these bigrams again so that we can plot it.

Hide

```
Trump_bigrams_final<-Trump_bigrams_sep%>% unite(bigram,word1,word2,sep = " ")
Trump_bigrams_final
```

	bigram	n
	<chr>	<int>
1	hillary clinton	663
2	donald trump	172
3	african american	164
4	american people	119

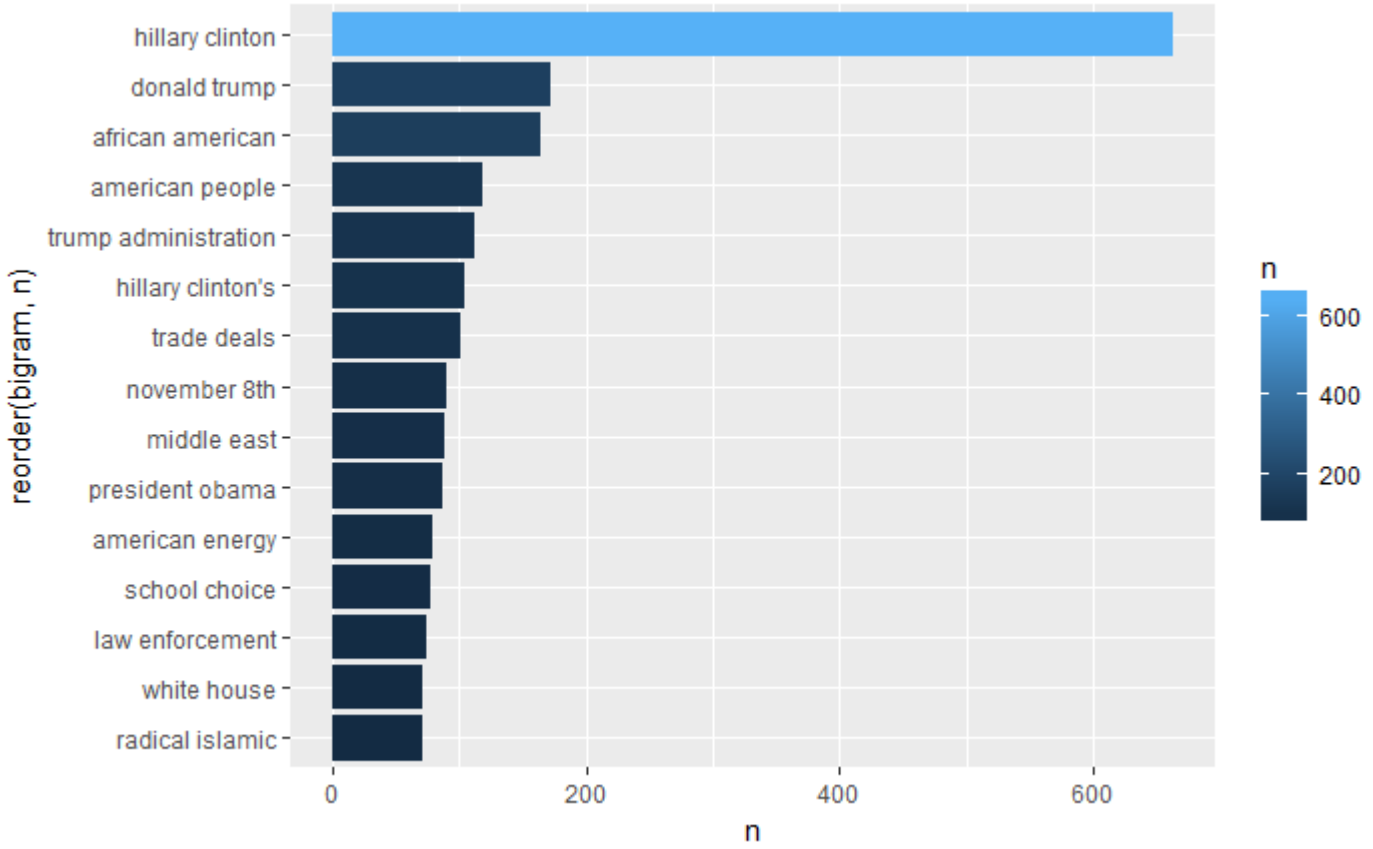
	bigram <chr>	n <int>
5	trump administration	112
6	hillary clinton's	104
7	trade deals	102
8	november 8th	91
9	middle east	88
10	president obama	87

1-10 of 14,257 rows Previous **1** 2 3 4 5 6 ... 100 Next

Now, let us plot the top 15 most commonly used bigrams in Trump's speeches.

Hide

```
Top15_bigram<-Trump_bigrams_final[1:15,]
ggplot(Top15_bigram)+geom_bar(mapping = aes(reorder(bigram,n),n,fill=n),stat = "identity")+coord
_flip()
```



PART C

For the part C, I will do the following:

A sentiment analysis of Donald Trump's speeches. In order to make sure sentiments are assigned to appropriate contexts, first tokenize the speeches into bigrams, and then filter out all bigrams where the first word is any of "not", "no", or "never".

Now, we have to remove the bigrams where the first words are "no","not","never".

For that, let us use the separated bigrams from the previous question.

Hide

```
Negative<-c("no","not","never")
Tr<-c("trump","applause")
Trump_bigram_senti<-Trump_bigrams_sep%>% filter(!word1 %in% Negative)%>% filter(!word2%in% Tr)
Trump_bigram_senti%>% filter(word2=="trump")# checking if the word elimination worked.
```

0 rows

Hide

```
Trump_bigram_senti%>% filter(word1=="no")
```

0 rows

Now let us get each of the 10 sentiments in the nrc into 10 separate dataframes.

We will need these to do the further analysis.

Hide

```
nrc<-get_sentiments("nrc")
unique(nrc$sentiment)
```

```
 [1] "trust"        "fear"         "negative"     "sadness"      "anger"        "surprise"
"positive"     "disgust"
 [9] "joy"          "anticipation"
```

Hide

```
nrc_trust<-nrc%>%filter(sentiment=="trust")
nrc_fear<-nrc%>%filter(sentiment=="fear")
nrc_negative<-nrc%>%filter(sentiment=="negative")
nrc_sadness<-nrc%>%filter(sentiment=="sadness")
nrc_anger<-nrc%>%filter(sentiment=="anger")
nrc_surprise<-nrc%>%filter(sentiment=="suprise")
nrc_positive<-nrc%>%filter(sentiment=="positive")
nrc_disgust<-nrc%>%filter(sentiment=="disgust")
nrc_joy<-nrc%>%filter(sentiment=="joy")
nrc_anticipation<-nrc%>%filter(sentiment=="anticipation")
```

Let us first create a new column in our dataframe so that we can later join the sentiment dataframes.

Hide

```
Trump_bigram_senti$word<-Trump_bigram_senti$word2
Trump_sentiment<-Trump_bigram_senti[,4]
class(Trump_sentiment)
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

Now, let us see the top 10 words in Trump's speeches associated with trust.

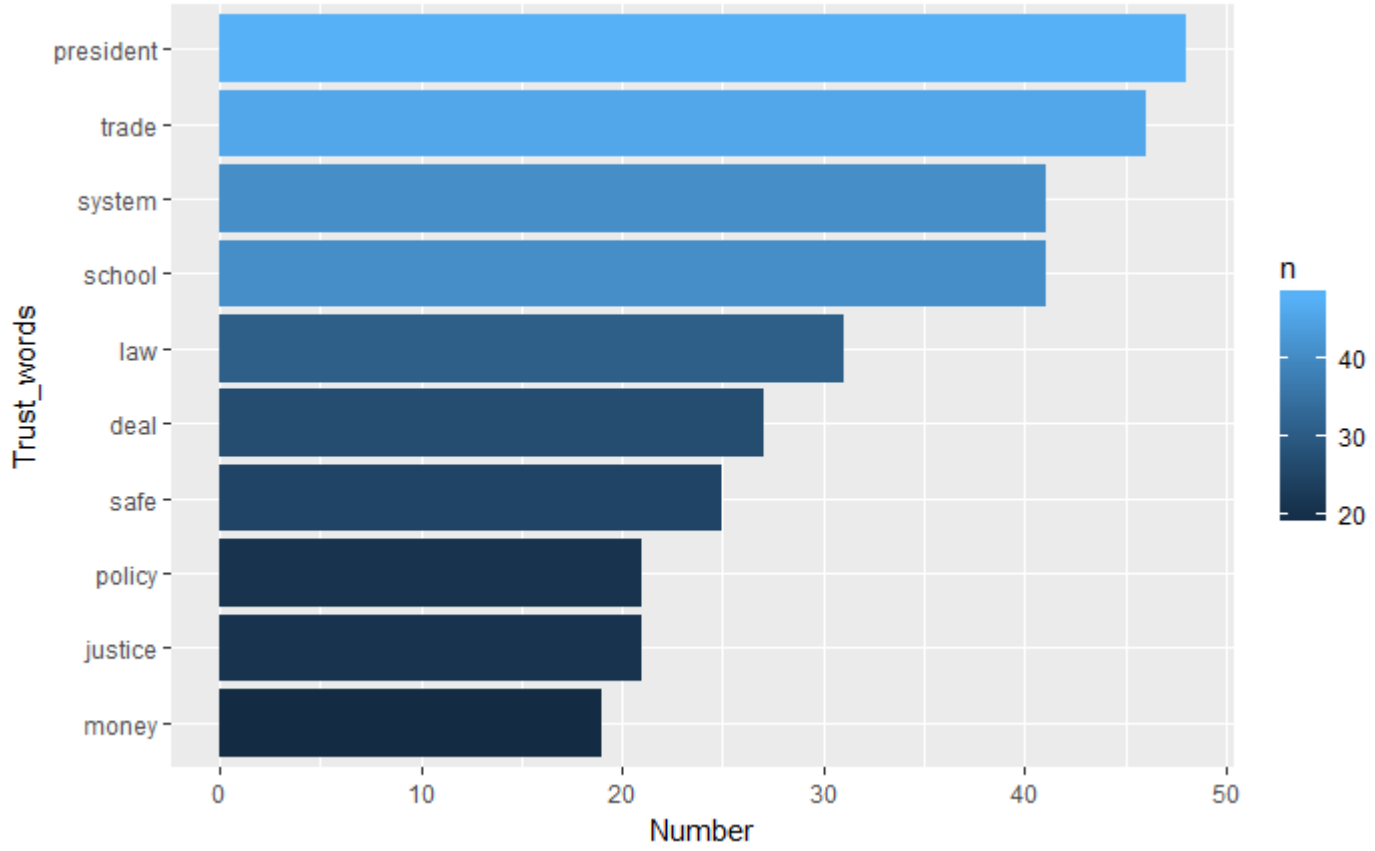
Hide

```
Trust_trump<-Trump_sentiment%>% inner_join(nrc_trust,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(Trust_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord_fl
ip()+labs(x="Trust_words",y="Number")
```



Now, let us see the top 10 words in Trump's speeches associated with Fear.

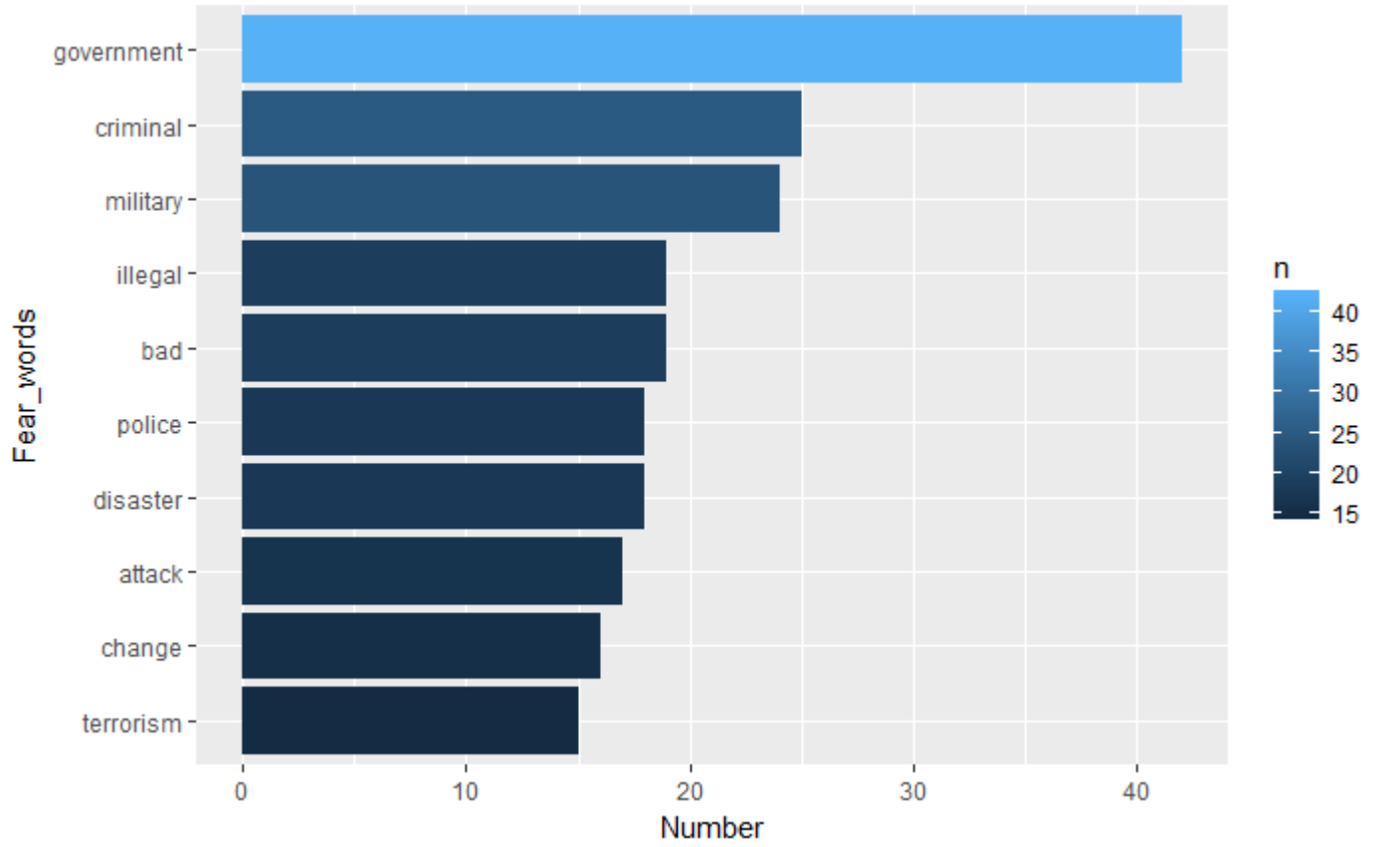
Hide

```
Fear_trump<-Trump_sentiment%>% inner_join(nrc_fear,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(Fear_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord_flip()+labs(x="Fear_words",y="Number")
```



Now, let us see the top 10 words in Trump's speeches associated with negative.

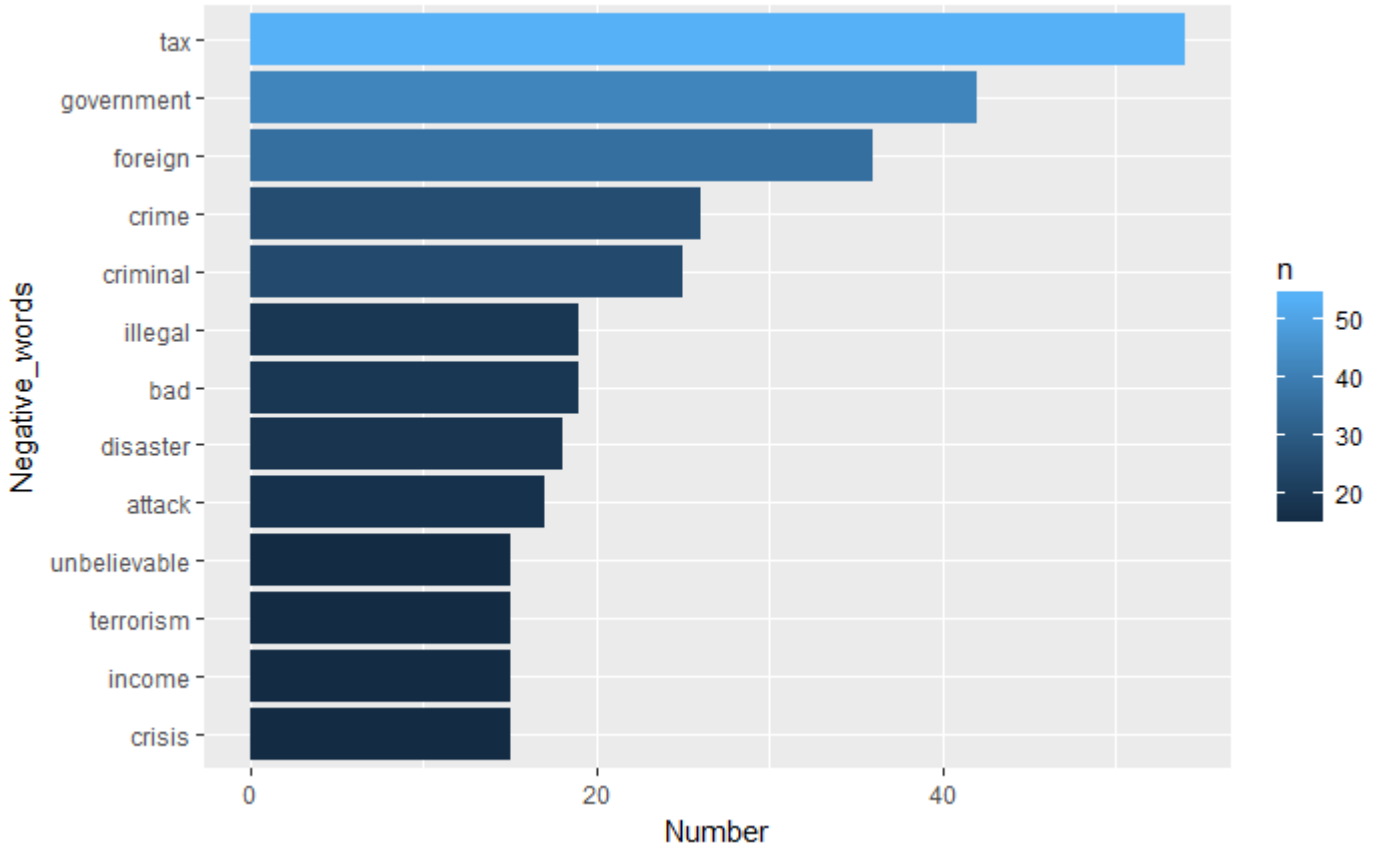
Hide

```
Negative_trump<-Trump_sentiment%>% inner_join(nrc_negative,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(Negative_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord_flip()+labs(x="Negative_words",y="Number")
```



Now, let us see the top 10 words in Trump's speeches associated with sadness.

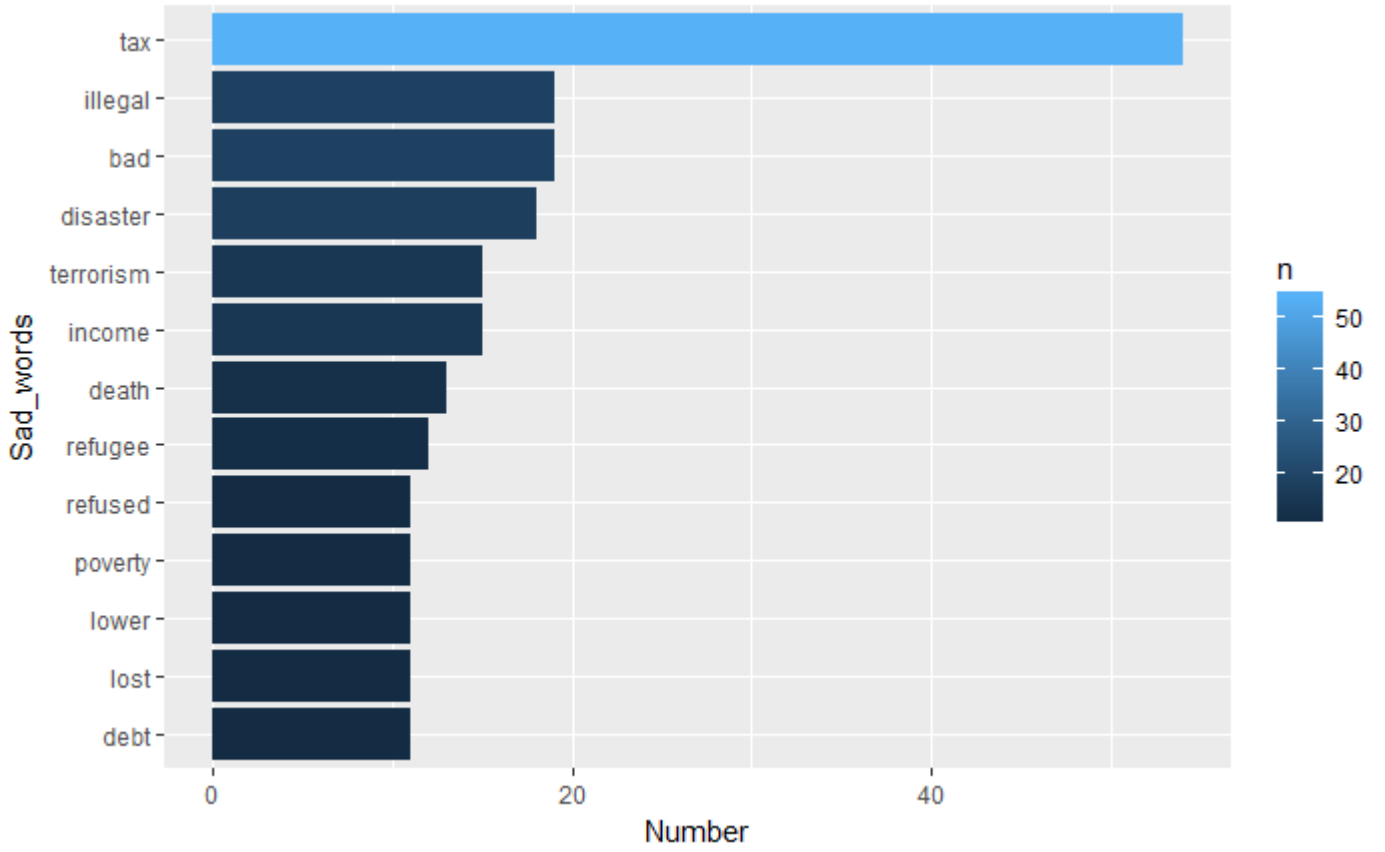
Hide

```
Sad_trump<-Trump_sentiment%>% inner_join(nrc_sadness,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(Sad_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord_flip
()+labs(x="Sad_words",y="Number")
```



Now, let us see the top 10 words in Trump's speeches associated with anger.

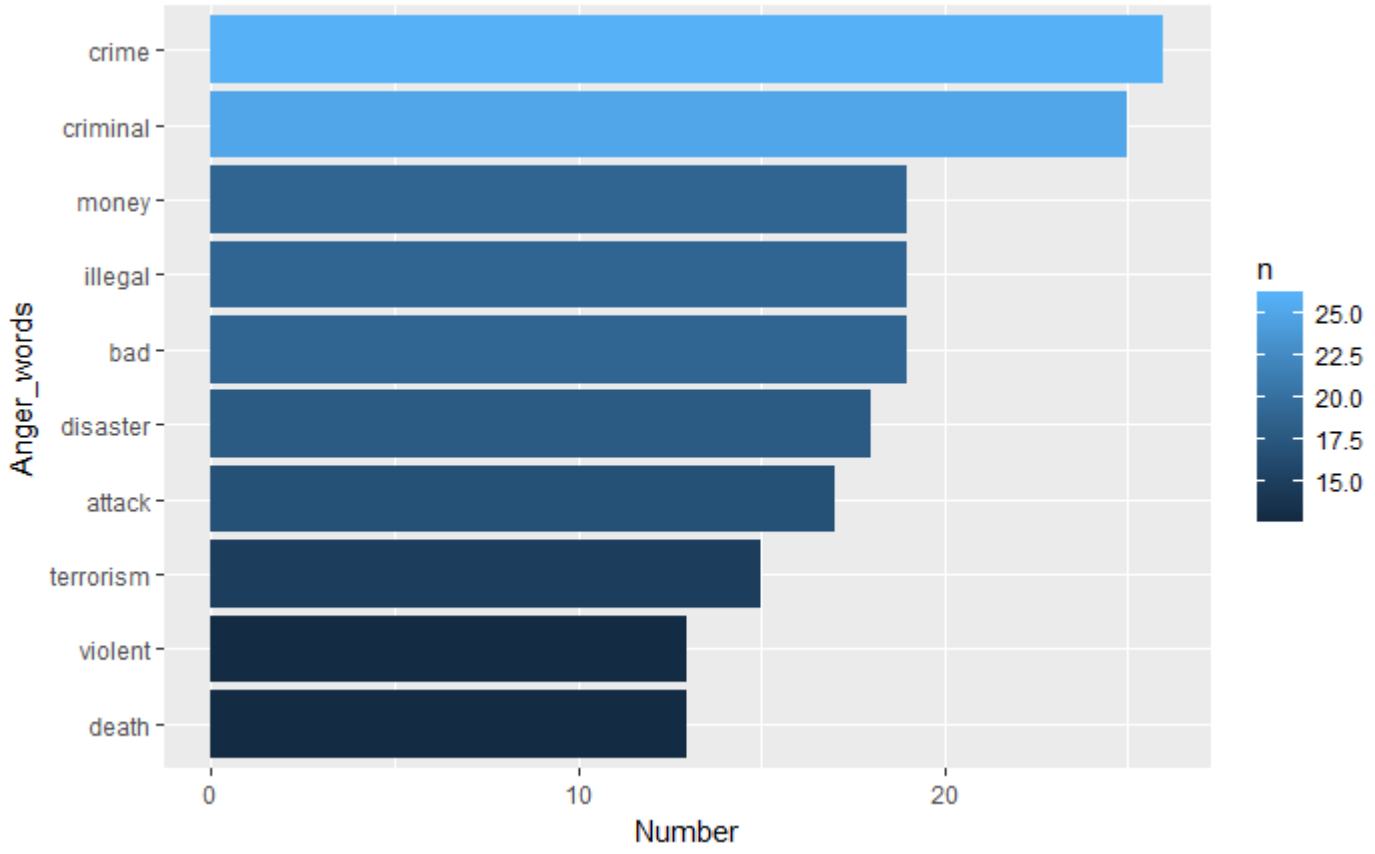
Hide

```
Anger_trump<-Trump_sentiment%>% inner_join(nrc_anger,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(Anger_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord_fl
ip()+labs(x="Anger_words",y="Number")
```



Now, let us see the top 10 words in Trump's speeches associated with surprise.

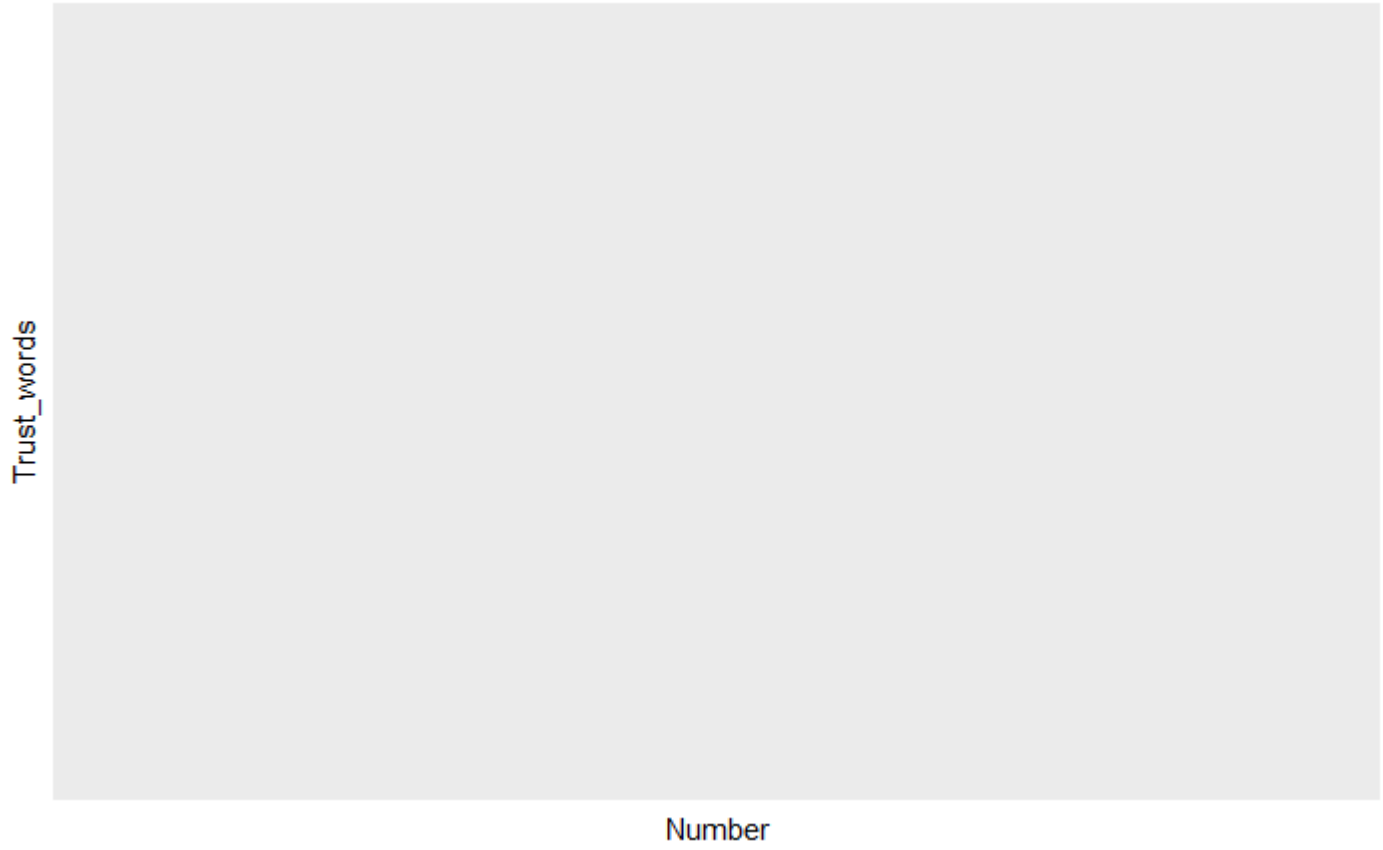
Hide

```
Surprise_trump<-Trump_sentiment%>% inner_join(nrc_surprise,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(Surprise_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord
_flip()+labs(x="Trust_words",y="Number")
```



There are no surprise words in trump's speeches.

Now, let us see the top 10 words in Trump's speeches associated with positive.

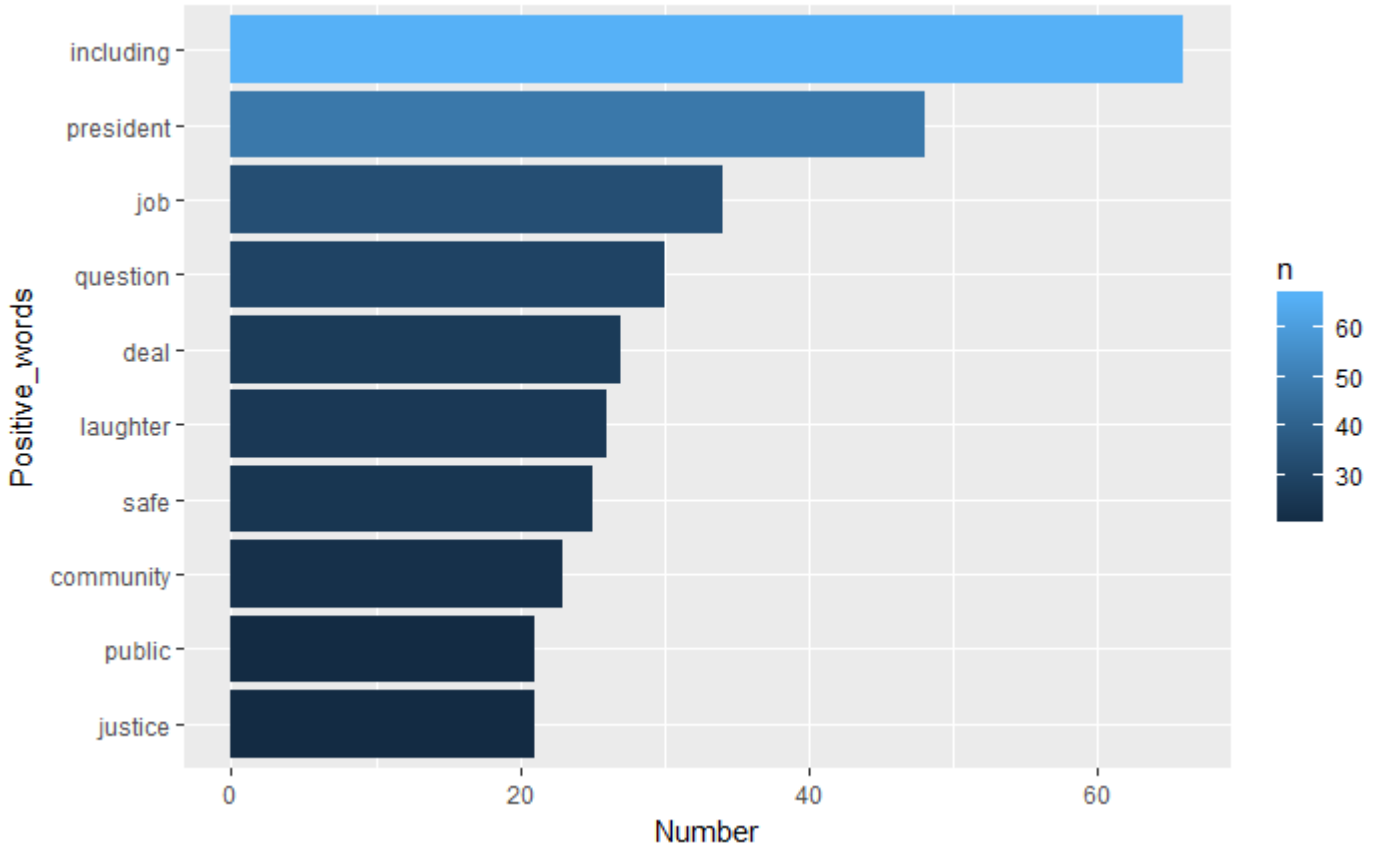
Hide

```
Positive_trump<-Trump_sentiment%>% inner_join(nrc_positive,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(Positive_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord
_flip()+labs(x="Positive_words",y="Number")
```



Now, let us see the top 10 words in Trump's speeches associated with disgust.

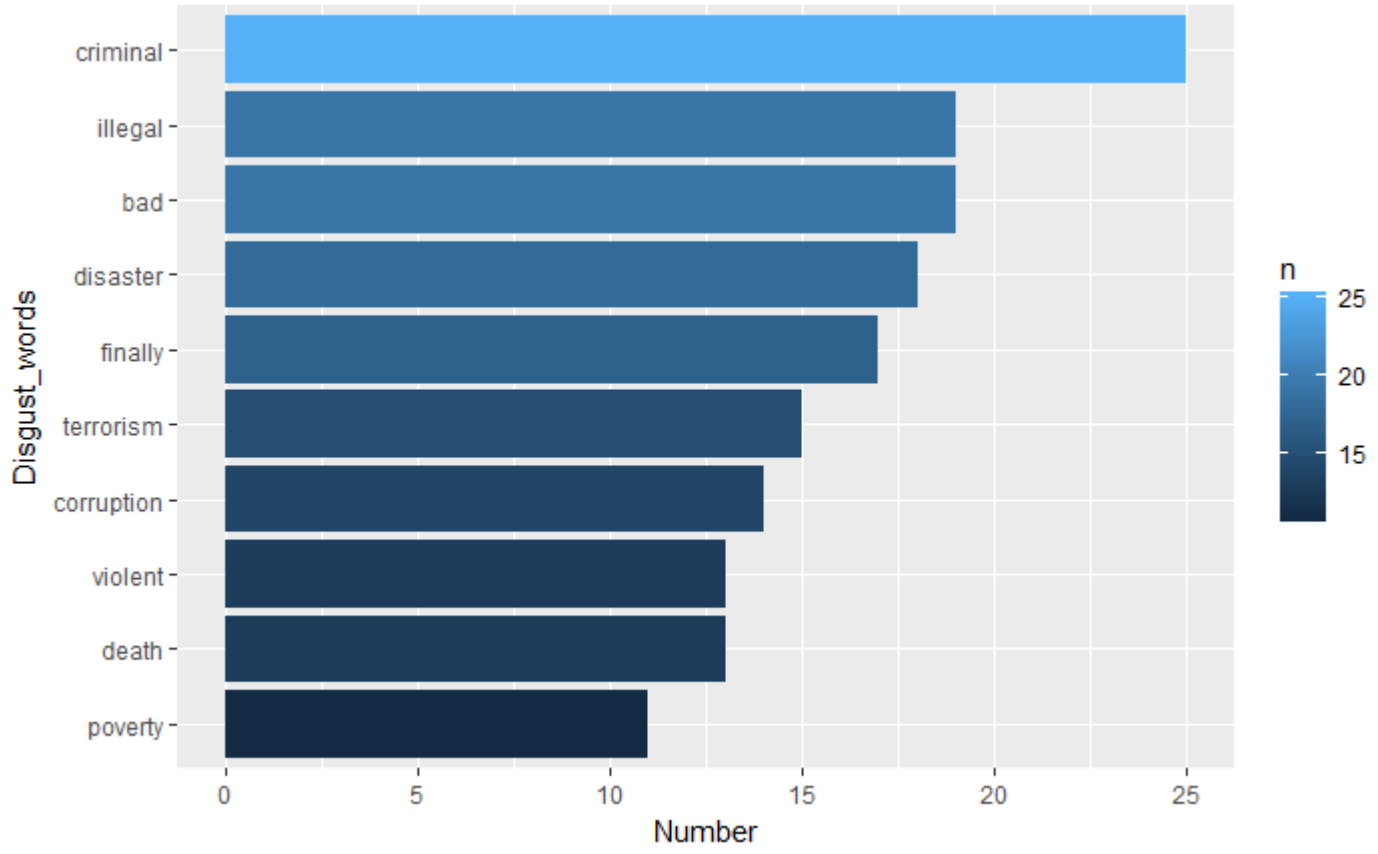
Hide

```
disgust_trump<-Trump_sentiment%>% inner_join(nrc_disgust,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(disgust_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord_
flip()+labs(x="Disgust_words",y="Number")
```



Now, let us see the top 10 words in Trump's speeches associated with joy.

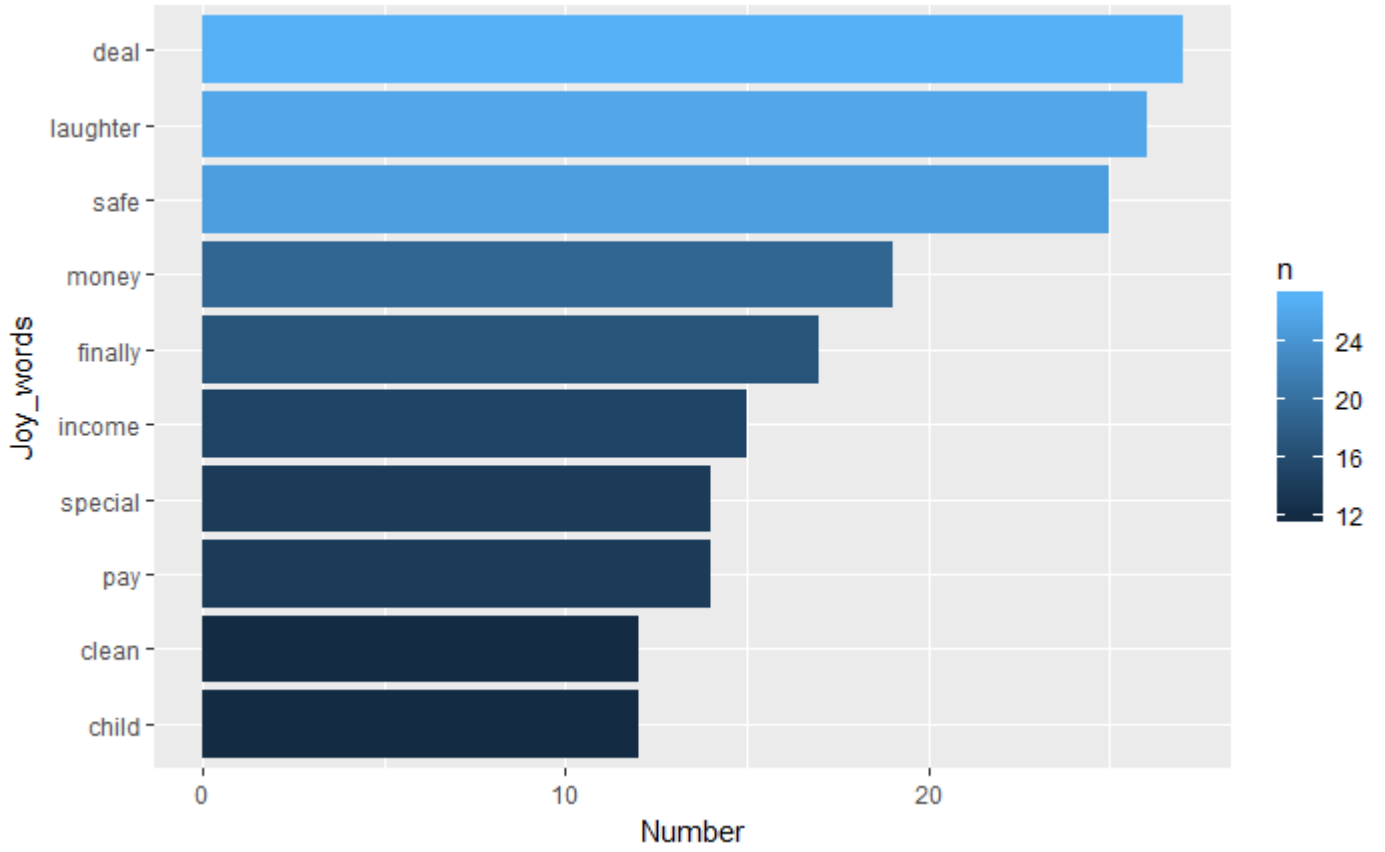
Hide

```
joy_trump<-Trump_sentiment%>% inner_join(nrc_joy,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(joy_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord_flip
()+labs(x="Joy_words",y="Number")
```



Now, let us see the top 10 words in Trump's speeches associated with anticipation.

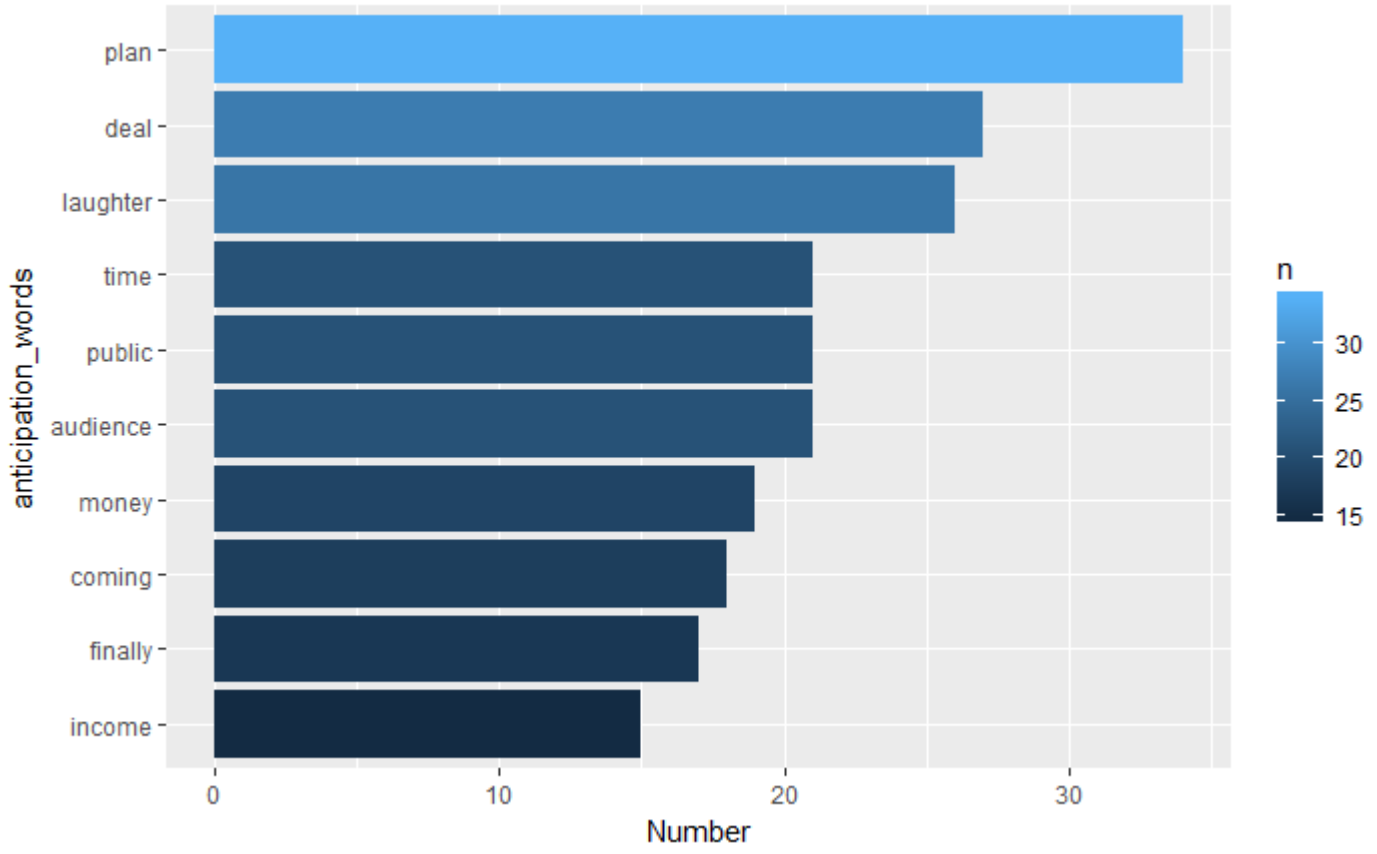
Hide

```
anticipation_trump<-Trump_sentiment%>% inner_join(nrc_anticipation,by="word")%>% count(word)%>%top_n(10)
```

```
Selecting by n
```

Hide

```
ggplot(anticipation_trump)+geom_bar(mapping = aes(reorder(word,n),n,fill=n),stat = "identity")+coord_flip()+labs(x="anticipation_words",y="Number")
```



PART D

In this part, I will write a function to tokenize an input corpus and spit out the most frequent words. The idea is to automate the process of data cleansing and get a glimpse into the data at the first go.

Hide

```
Text_analysis_corpus<-function(mysrc_clean,nwords){
  mysrc_clean<-tm_map(mysrc_clean,removeNumbers)
  mysrc_clean<-tm_map(mysrc_clean,removeWords,stopwords())
  mysrc_clean<-tm_map(mysrc_clean,removePunctuation)
  library(SnowballC)
  mysrc_clean<-tm_map(mysrc_clean,stripWhitespace)
 mysrc_dcm<-DocumentTermMatrix(mysrc_clean)

 mysrc_sparce<-removeSparseTerms(mysrc_dcm,.5)
mysrc_final<-as.data.frame(as.matrix(mysrc_sparce))
Most_common<-colSums(Trump_final)
Top_words<-sort(Most_common,decreasing = TRUE)[1:nwords]
Word_freq<-as.data.frame(Top_words)
Word_freq$Word<-rownames(Word_freq)
rownames(Word_freq)<-c()
Word_freq<-Word_freq[c(2,1)]
colnames(Word_freq)<-c("Word","Frequency")
Word_freq$Word<-as.factor(Word_freq$Word)
ggplot(Word_freq)+geom_bar(mapping = aes(reorder(Word,Frequency),Frequency,fill=Frequency),stat
 = "identity")+coord_flip()
 }
```

Let us test if the method works.

We will use the corpus from the question 6.

Hide

```
Text_analysis_corpus(Trump_corpus,5)
```