hw6-Deep-Bhalodia

Deep Bhalodia

4/3/2019

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Importing required packages

```
library(purrr)
library(tidyr)
library(tidyverse)
library(dplyr)
library(tidytext)
library(tokenizers)
library(stringr)
library(modelr)
library(mlbench)
```

PART A

Problem 1

Write a function that performs cross-validation for a linear model (fit using lm) and returns the average root-mean-square-error across all folds.

```
cvlm <- function(formula, data, nfold) {
  cvdata <- crossv_kfold(data, nfold)
  cvdata <- cvdata %>%
  mutate(fit = map(train, ~ lm(formula, data = .))) %>%
  mutate(rmse = map2_dbl(fit, test, rmse))
  c(cv_rmse=mean(cvdata$rmse))
}
```

Problem 2

Using 5-fold cross-validation, report the cross-validated RMSE of the model you fit in Homework 5, Problem 5, for predicting crim from the BostonHousing dataset in the mlbench package. Can you find a model with a better cross-valided RMSE?

```
set.seed(1)
```

```
data("BostonHousing")

cvlm(log(crim) ~ log(dis) + rad,data=BostonHousing, nfold=5)
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
##   cv_rmse
## 0.8916171
```

```
cvlm(log(crim) ~ log(dis) + rad + chas,data=BostonHousing, nfold=5)
```

```
##   cv_rmse
## 0.8894011
```

```
cvlm(log(crim) ~ log(dis) + rad + tax,data=BostonHousing, nfold=5)
```

```
##   cv_rmse
## 0.8960458
```

```
cvlm(log(crim) ~ log(dis) + rad + ptratio, data=BostonHousing, nfold=5)
```

```
## cv_rmse
## 0.89996
```

```
cvlm(log(crim) ~ log(dis) + rad + tax + ptratio, data=BostonHousing, nfold=5)
```

```
##   cv_rmse
## 0.8937894
```

Comment

Adding ptratio to our model from Homework 5, Problem 5 improves the cross-validated RMSE most. Adding additional variables after that shows no improvement.

PART B

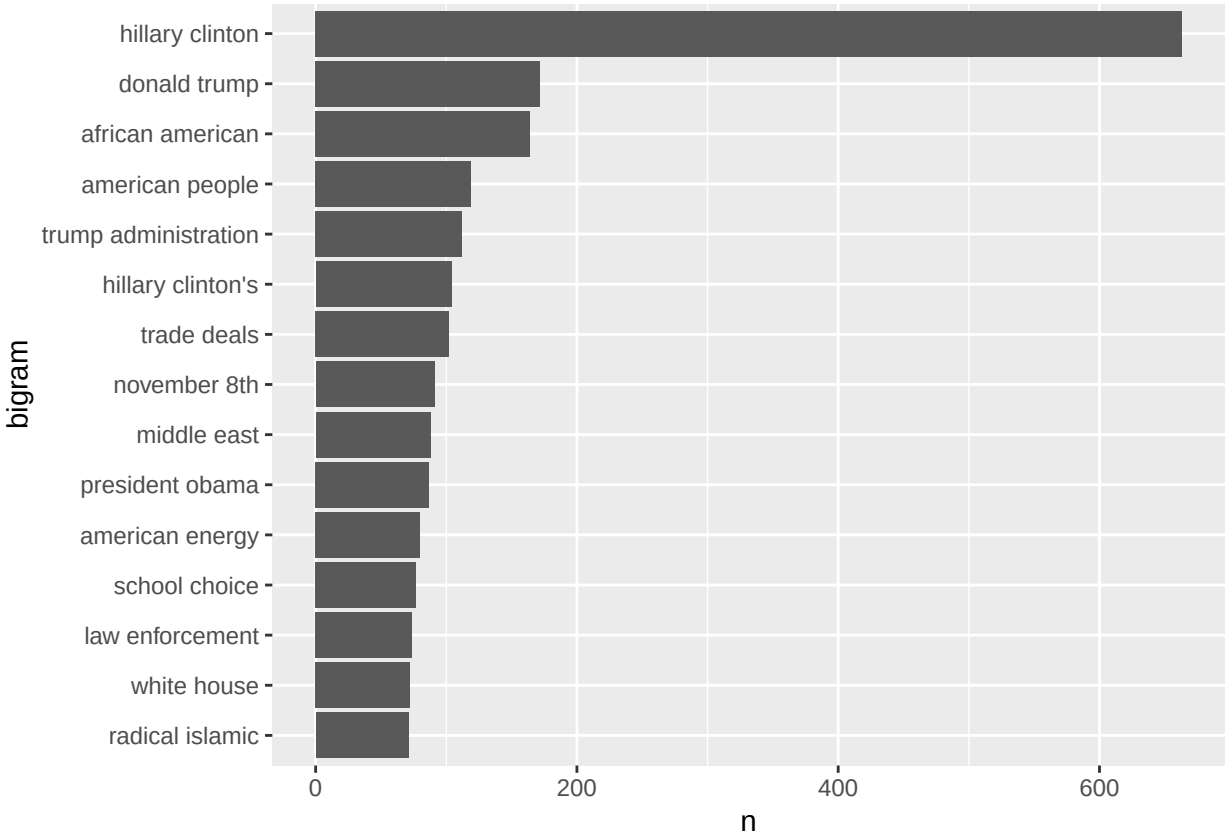
Problem 3

Import the text from all 56 Donald Trump speeches into R. Tokenize the data into a tidy text data frame, using bigrams as tokens.

```
text <- read_lines("full_speech.txt")


trump <- tibble(line=1:length(text), text=text)


trump_bigrams <- trump %>%
  unnest_tokens(bigram, text, token="ngrams", n=2) %>%
  separate(bigram, c("word1", "word2"), sep = " ")


trump_bigrams %>%
  filter(!word1 %in% c(stop_words$word, "applause")) %>%
  filter(!word1 %in% c("not", "no", "never", "without")) %>%
```

```
  filter(!word2 %in% c(stop_words$word, "applause")) %>%
  unite(bigram, word1, word2, sep = " ") %>%
  count(bigram, sort=TRUE) %>%
  mutate(bigram = reorder(bigram, n)) %>%
  top_n(15) %>%
  ggplot(aes(x=bigram, y=n)) +
  geom_col() +
  coord_flip()
```

```
## Selecting by n
```



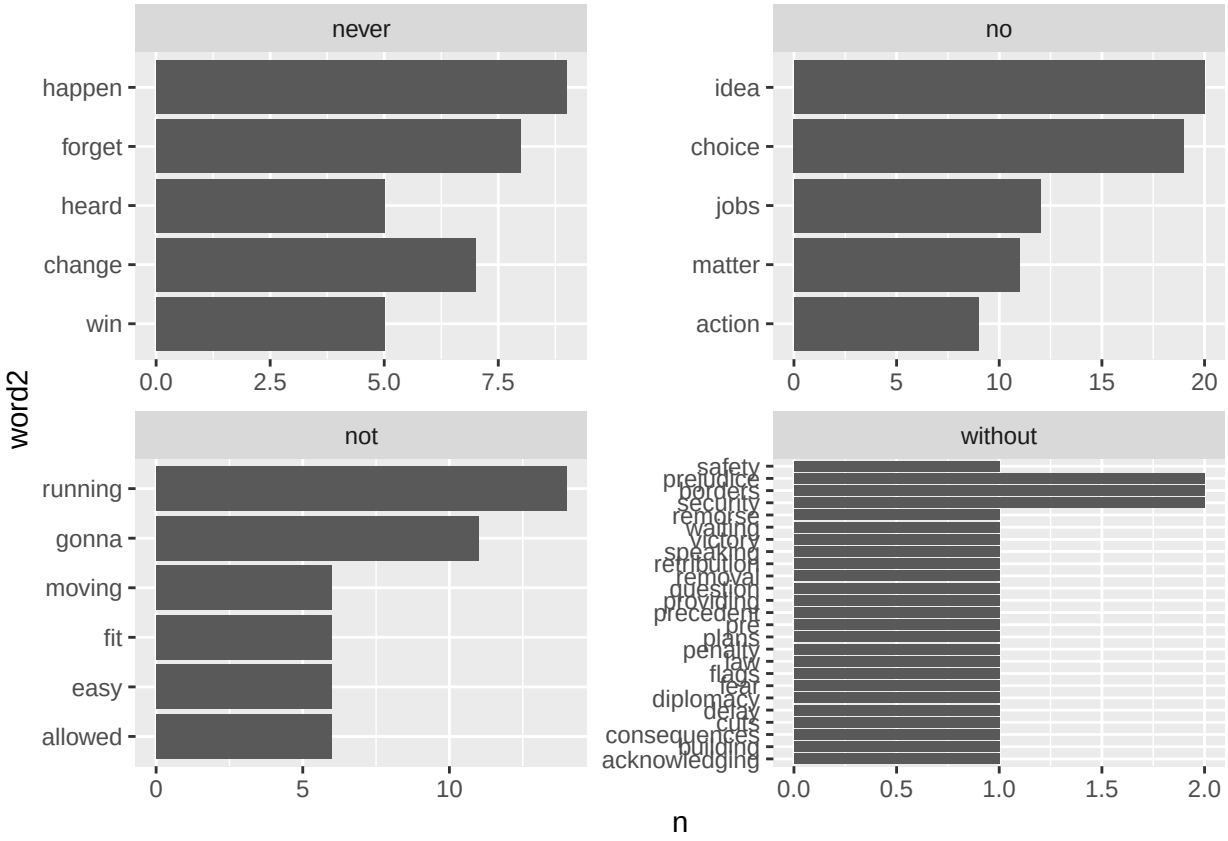
Problem 4

We would like to see the most commonly negated words in Donald Trump's speeches, and how they're negated.

```
trump_bigrams %>%
  filter(word1 %in% c("not", "no", "never", "without")) %>%
  filter(!word2 %in% c(stop_words$word, "applause")) %>%
  count(word1, word2, sort=TRUE) %>%
  mutate(word2 = reorder(word2, n)) %>%
  group_by(word1) %>%
  top_n(5) %>%
  ggplot(aes(x=word2, y=n)) +
  geom_col() +
```

```
  facet_wrap(~word1, scales="free") +
  coord_flip()
```

```
## Selecting by n
```

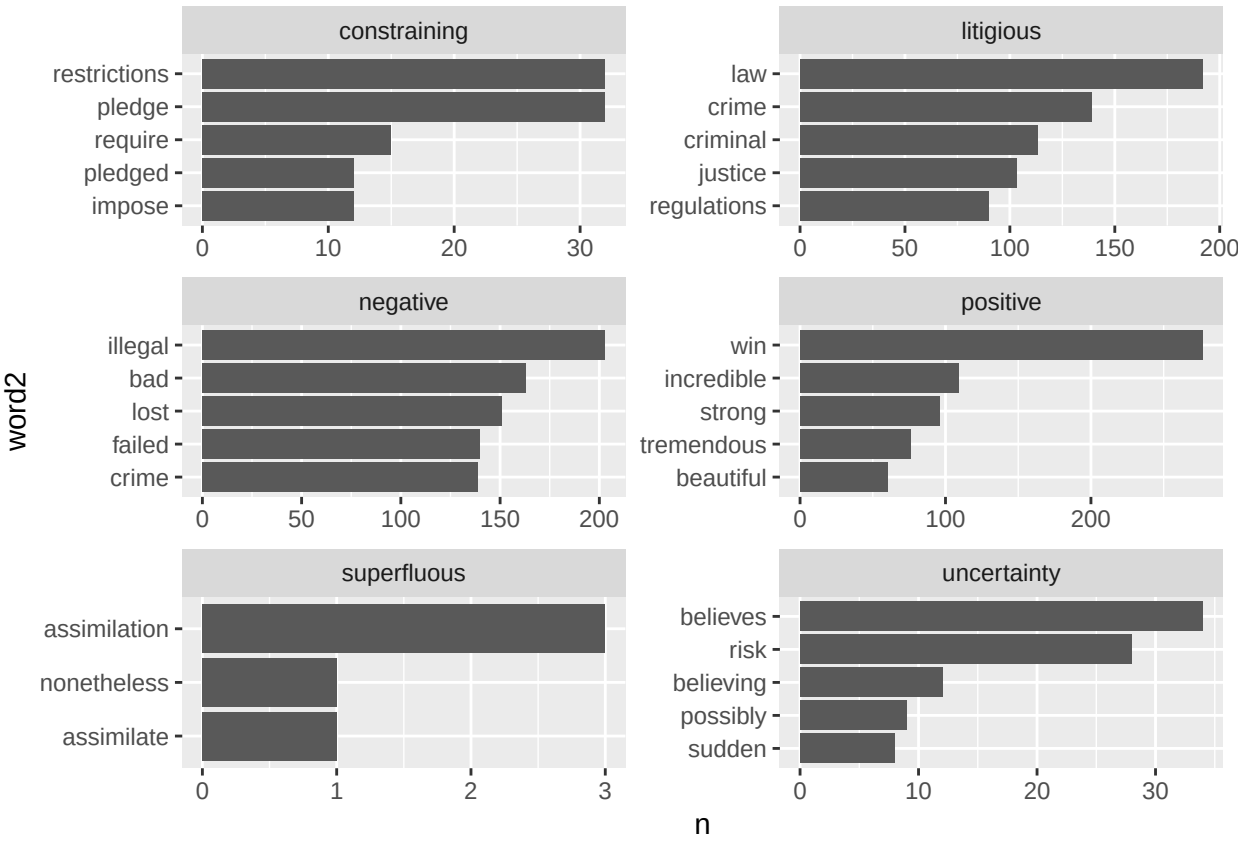


Problem 5

We would like to do 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 filter out all bigrams where the first word is any of the words "not", "no", "never", or "without".

```
trump_bigrams %>%
  filter(!word1 %in% c("not", "no", "never", "without")) %>%
  filter(!word2 %in% c(stop_words$word, "applause")) %>%
  inner_join(get_sentiments("loughran"), by=c("word2"="word")) %>%
  count(word2, sentiment, sort=TRUE) %>%
  mutate(word2 = reorder(word2, n)) %>%
  group_by(sentiment) %>%
  top_n(5) %>%
  ggplot(aes(x=word2, y=n)) +
  geom_col(show.legend=FALSE) +
  facet_wrap(~sentiment, ncol=2, scales="free") +
  coord_flip()
```

```
## Selecting by n
```


constraining
restrictions
pledge
require
pledged
impose
0
10
20
30
litigious
law
crime
criminal
justice
regulations
0
50
100
150
200
negative
illegal
bad
lost
failed
crime
0
50
100
150
200
positive
win
incredible
strong
tremendous
beautiful
0
100
200
superfluous
assimilation
nonetheless
assimilate
0
1
2
3
uncertainty
believes
risk
believing
possibly
sudden
0
10
20
30
word2
n