Problem Set 8

Clustering Part 1

[YOUR NAME]

Due Date: 2023-04-07

Getting Set Up

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...` . Accept defaults and save this file as `[LAST NAME]_ps8.Rmd` to your `code` folder.

Copy and paste the contents of this file into your `[LAST NAME]_ps8.Rmd` file. Then change the `author: [YOUR NAME]` (line 4) to your name.

All of the following questions should be answered in this `.Rmd` file. There are code chunks with incomplete code that need to be filled in.

This problem set is worth 10 total points, plus two extra credit points. The point values for each question are indicated in brackets below. To receive full credit, you must both have the correct code **and include a comment describing what each line does**. In addition, some questions ask you to provide a written response in addition to the code. Furthermore, some of the code chunks are totally empty, requiring you to try writing the code from scratch. Make sure to comment each line, explaining what it is doing!

You are free to rely on whatever resources you need to complete this problem set, including lecture notes, lecture presentations, Google, your classmates…you name it. However, the final submission must be complete by you. There are no group assignments. To submit, compiled the completed problem set and upload the PDF file to Brightspace by midnight on 2023/04/07.

Good luck!

Question 0

Require `tidyverse` and `tidytext` (for calculating AUC), and load the `Trump_tweet_words.Rds` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/8_Clustering_NLP/data/Trump_tweet_words.Rds?raw=true) data to an object called `tweet_words` . (Tip: use the `read_rds()` function with the link to the raw data.)

Also, load the `Trumptweets.Rds` (https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/8_Clustering_NLP/data/Trumptweets.Rds?raw=true) data to an object called `tweets` .

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.4
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.2      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
require(tidytext)
```

```
## Loading required package: tidytext
```

```
tweet_words <- read_rds(file="https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/8_Clustering_NLP/data/Trump_tweet_words.Rds?raw=true")

tweets <- read_rds(file="https://github.com/jbisbee1/DS1000_S2023/blob/main/Lectures/8_Clustering_NLP/data/Trumptweets.Rds?raw=true")
```

Question 1 [1 point + 2 EC]

Using the `tweet_words` object, calculate the most frequently used word by year.

a. Which is Trump's most commonly used word in 2011 and how often did he use it? [1 point]

b. EXTRA CREDIT: can you determine what this word means, using the `tweets` object to see it in context? [1 point] Based on this analysis, do you think we should drop this word? Why? [1 point] **HINT** get the list of tweet IDs (`document`) from the `tweet_words` object, then filter the `tweets` object based on the `id` column. (NB: `document` in the `tweet_words` object is the same as the `id` column from the `tweets` object.)

```
tweet_words %>%
  count(word,Tweeting.year) %>%
  group_by(Tweeting.year) %>%
  arrange(-n) %>%
  slice(1)
```

```
## # A tibble: 12 × 3
## # Groups:   Tweeting.year [12]
##    word      Tweeting.year     n
##    <chr>     <fct>         <int>
##  1 donald    2009             42
##  2 pm        2010             47
##  3 cont      2011             87
##  4 cont      2012            435
##  5 trump     2013            789
##  6 trump     2014           1107
##  7 trump     2015           1658
##  8 hillary   2016            444
##  9 people    2017            200
## 10 people    2018            425
## 11 president 2019            559
## 12 iran      2020             23
```

```
# EC:
docs <- tweet_words %>%
  filter(Tweeting.year == 2011,word == 'cont') %>%
  select(document)

tweets %>%
  filter(Tweeting.year == 2011,
         id %in% docs$document) %>%
  select(content)
```

```
## # A tibble: 87 × 1
##    content
##    <chr>
##  1 I want to personally congratulate President Obama and the men women of the A…
##  2 http://bit.ly/pwgGsQ http://bit.ly/pwgGsQ My interview yesterday on CNBC's S…
##  3 China demanded that we raise our debt ceiling and then their rating agency d…
##  4 The new President of OPEC is Mahmoud Ahmadinejad’s confidant Rostam Ghasemi,…
##  5 The S&P are losers. They did this for personal publicity in order to straigh…
##  6 Republicans and Democrats should get back to work immediately to work on res…
##  7 http://goo.gl/AMNEE Countdown to @ AmericaNowRadio as my former # Apprentice…
##  8 @ BarackObama took another swipe at the State of Israel for building in thei…
##  9 My interview on @ AmericaNowRadio with Andy Dean. Andy was a contestant on t…
## 10 I was interviewed by Greta Van Susteren today here at Trump Tower. Tune in t…
## # … with 77 more rows
```

- Trump’s most common word in 2011 was “cont”, which appeared 87 times.

- EC: this appears to be due to him writing longer tweets in this period, and indicating that the tweet would be “continued” by ending it with “(cont)”.

Question 2 [3 points]

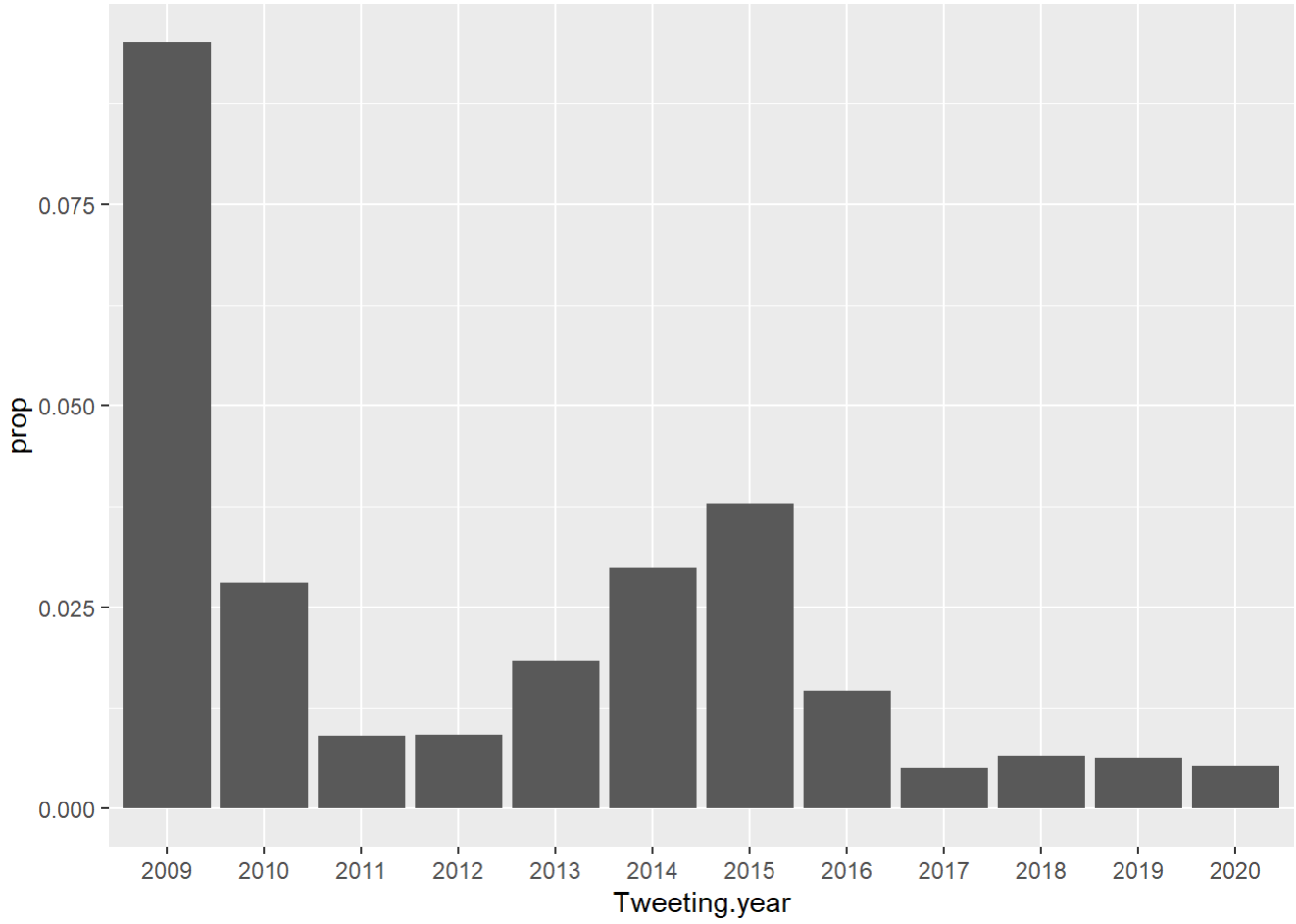
a. Calculate the proportion of times each word is used per year. [1 point]

b. Plot the proportion of times the word “trump” is used each year. Make sure to justify your choice of `geom_...()` ! [1 point]

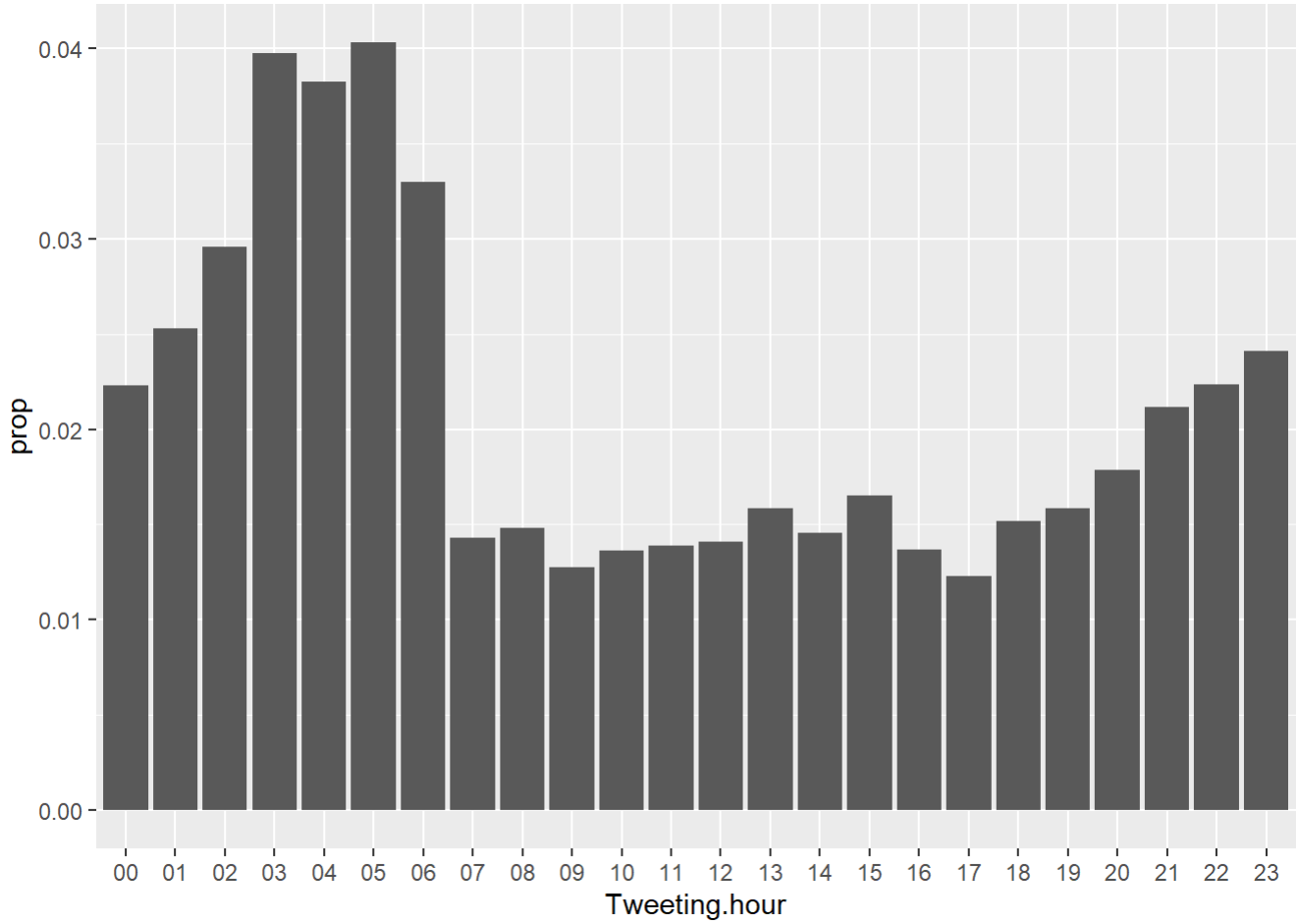
c. Do the same analysis, except calculate by hour of the day. [1 point]

```
#a.
toplot <- tweet_words %>%
  group_by(Tweeting.year) %>%
  mutate(totWords = n()) %>%
  count(Tweeting.year,word,totWords) %>%
  mutate(prop = n / totWords)

#b.
toplot %>%
  filter(word == 'trump') %>%
  ggplot(aes(x = Tweeting.year,y = prop)) +
  geom_bar(stat = 'identity')
```



```
#c.
tweet_words %>%
  group_by(Tweeting.hour) %>%
  mutate(totWords = n()) %>%
  count(Tweeting.hour,word,totWords) %>%
  mutate(prop = n / totWords) %>%
  filter(word == 'trump') %>%
  ggplot(aes(x = Tweeting.hour,y = prop)) +
  geom_bar(stat = 'identity')
```



- I chose `geom_bar()` for the year and the hour, since these are similar enough to categorical variables to make the visualization intuition.

Question 3 [3 points]

We want to only look at tweets written during Trump's campaign for president (June 16th, 2015 through November 8th, 2016), and are interested if there are patterns in what he talks about *by day*.

Prepare the data for topic modeling via k-means clustering, filtering to the campaign period and using `document` as the document.

a. Create a document-term matrix (`dtm`), dropping any words that appear fewer than 20 times total. [1 point]

b. Calculate the TF-IDF using the appropriate function from the `tidytext` package. [1 point]

c. Cast the DTM to wide format using the `cast_dtm()` function. [1 point]

```
#a.
dtm <- tweet_words %>%
  filter(Tweeting.date > as.Date('2015-06-16') & Tweeting.date < as.Date('2016-11-08')) %>%
  count(document,word) %>%
  group_by(word) %>%
  mutate(tot_n = sum(n)) %>%
  ungroup() %>% filter(tot_n > 20)

#b.
dtm.tfidf <- bind_tf_idf(tbl = dtm, term = word, document = document, n = n)

#c.
castdtm <- cast_dtm(data = dtm.tfidf, document = document, term = word, value = tf_idf)
```

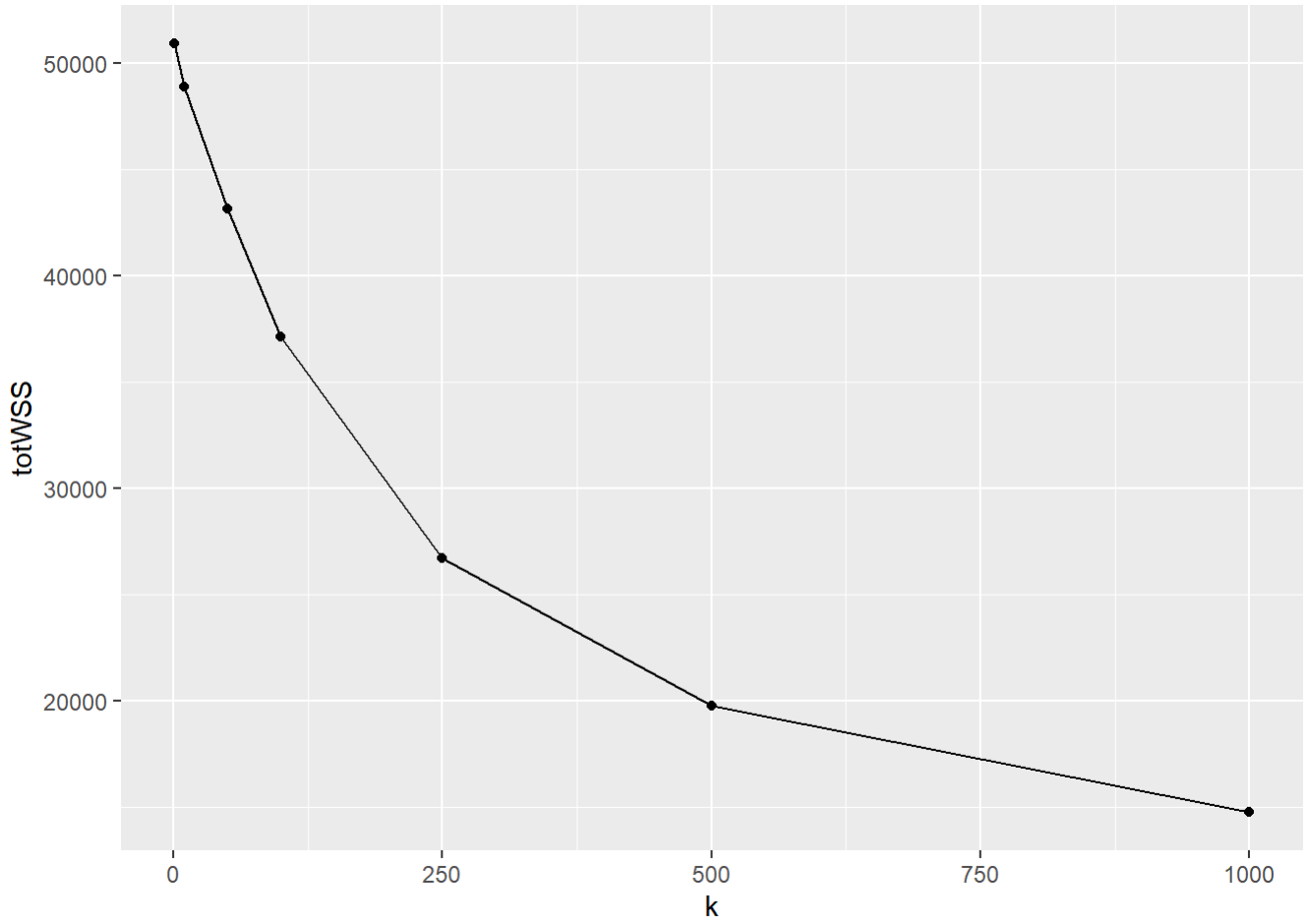
Question 4 [1 point]

Determine the optimal number of clusters / centers / topics / k by creating and manually inspecting an elbow plot. To save time, only examine the following sizes: `c(1,10,50,100,250,500,1000)` (this will still take a little while to run so be patient!). What value would you choose? [1 point]

```
set.seed(42)
totWSS <- NULL
for(k in c(1,10,50,100,250,500,1000)) {
  km_out <- kmeans(castdtm,
                centers = k,
                nstart = 5)

  totWSS <- data.frame(totWSS = km_out$tot.withinss,
                       k = k) %>%
    bind_rows(totWSS)
}

totWSS %>%
  ggplot(aes(x = k,y = totWSS)) +
  geom_point() +
  geom_line()
```



- I would choose about 250 clusters, based on this analysis.

Question 5 [2 points]

Re-run the k-means analysis using the number of clustered identified above and then `tidy()` the output.

a. Which are the top 5 most popular topics for Donald Trump in this period? [1 point]

b. Plot the top 10 highest scoring words for each of the top 5 most popular topics. What is each “about”? [1 point]

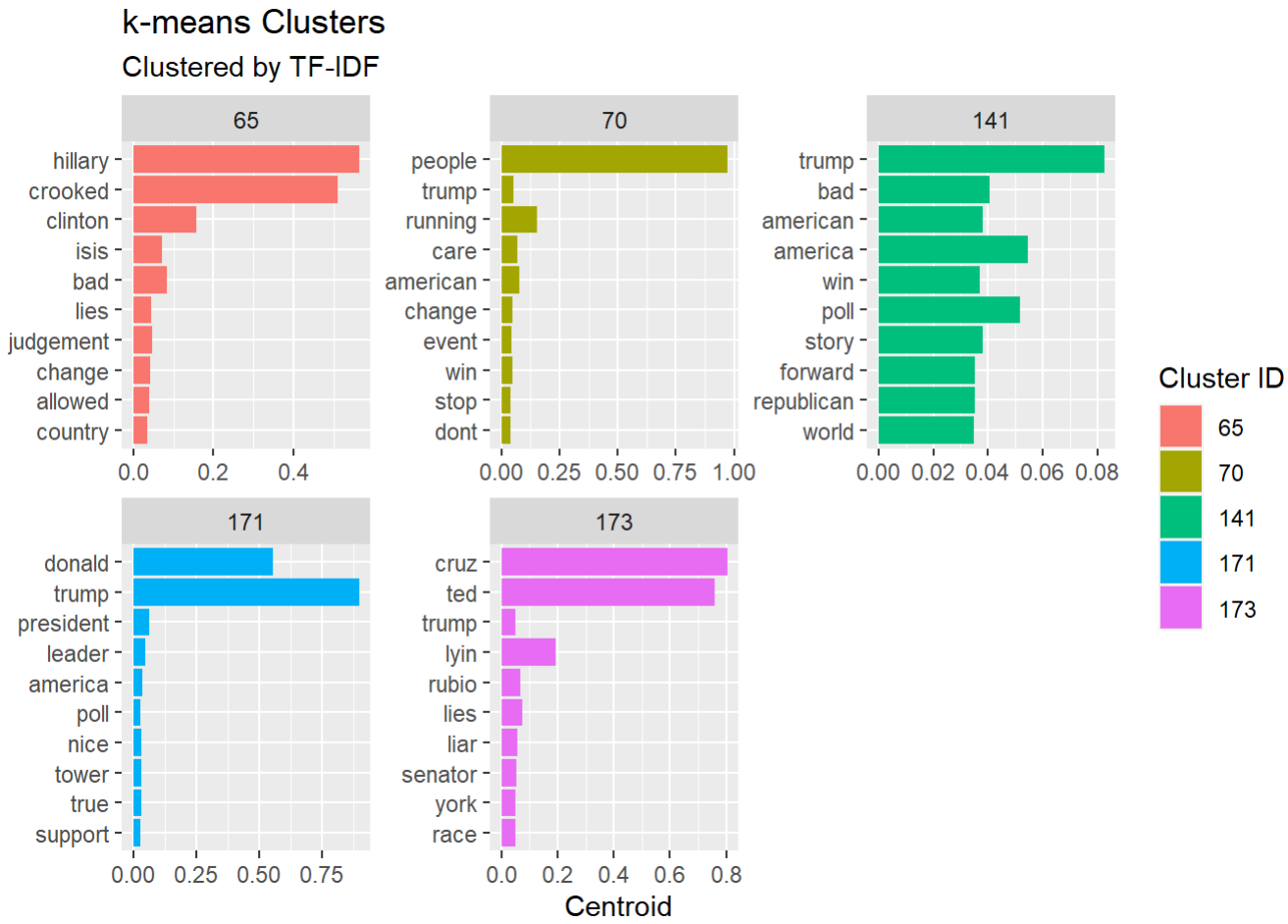
```
km_out <- kmeans(castdtm,
                 centers = 200,
                 nstart = 5)

km_out_tidy <- tidy(km_out) %>%
  gather(word,mean_tfidf,-size,-cluster,-withinss) %>%
  mutate(mean_tfidf = as.numeric(mean_tfidf))

#a.
(tops <- km_out_tidy %>%
  select(size,withinss,cluster) %>%
  distinct() %>%
  arrange(desc(size)) %>%
    slice(1:5))
```

```
## # A tibble: 5 × 3
##    size withinss cluster
##   <int>    <dbl> <fct>
## 1   894    5420. 141
## 2   229     651. 171
## 3   196     572. 65
## 4   135     429. 173
## 5   116     508. 70
```

```
#b.
km_out_tidy %>%
  filter(cluster %in% tops$cluster) %>%
  group_by(cluster) %>%
  arrange(-mean_tfidf) %>%
  slice(1:10) %>%
  ggplot(aes(x = mean_tfidf,y = reorder(word,mean_tfidf),
             fill = factor(cluster))) +
  geom_bar(stat = 'identity') +
  facet_wrap(~cluster,scales = 'free') +
  labs(title = 'k-means Clusters',
       subtitle = 'Clustered by TF-IDF',
       x = 'Centroid',
       y = NULL,
       fill = 'Cluster ID')
```



- These topics are basically about either Hillary Clinton, a group of Republican primary challengers, or a bunch of topics related to Trump (his performance in polls, his media appearances, etc.).