Problem Set 7

Ilayda Koca

Due Date: 2022/11/06 @ 11:59PM CST

Getting Set Up

If you haven't already, create a folder for this course, and then a subfolder within for the second lecture `Topic9_Clustering`, and two additional subfolders within `code` and `data`.

Open `RStudio` and create a new RMarkDown file (`.Rmd`) by going to `File -> New File -> R Markdown...`. Change the title to `"DS1000: Problem Set 7"` and the author to your full name. Save this file as `[LAST NAME]_ps7.Rmd` to your `code` folder.

If you haven't already, download the `CountyVote2004_2020.Rds` file from the course github page (https://github.com/jbisbee1/DS1000-F2022/blob/master/Lectures/Topic9_Clustering/data/CountyVote2004_2020.Rds) and save it to your `data` folder.

All of the following questions should be answered using

Require `tidyverse` and load the `CountyVote2004_2020.Rds` data to `dat`.

```
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   1.2.0      ✔ 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()
```

```
dat <- readRDS('../data/CountyVote2004_2020.Rds')
```

HINT: Questions 1 and 2 can be found in the slides and recording for Monday's lecture. Questions 3 - 5 + the extra credit can be found in the slides and recording for Wednesday's lecture. Pay particular attention to the pre-lecture handouts if you get stuck!

Question 1 [4 points]

Describe the columns `pct_rep_2020` and `pct_rep_2004`, following these steps:

1. Look at the data and identify missingness. Which states have missing values?
2. Visualize the data using univariate visualization for both measures, but put them on the same plot, differentiated by color. Do you notice any patterns?
3. Visualize the data using multivariate (or conditional) visualization where `pct_rep_2020` is the outcome and `pct_rep_2004` is the predictor. Use the `geom_abline()` to create a 45 degree line on these plots.

Do you notice any patterns? EXTRA CREDIT: Interpret this plot substantively and color the points by whether they are above or below the 45 degree line.

```
#1. Look at the data and identify missingness. Which states have missing values?
g0 <- dat %>%
  group_by(state) %>%
  filter(is.na(pct_rep_2004)) %>%
  summarize()

g0
```

```
## # A tibble: 2 × 1
##   state
##   <chr>
## 1 AK
## 2 ME
```

```
g00 <- dat %>%
  group_by(state) %>%
  filter(is.na(pct_rep_2020)) %>%
  summarize()

g00
```

```
## # A tibble: 2 × 1
##   state
##   <chr>
## 1 ME
## 2 VA
```

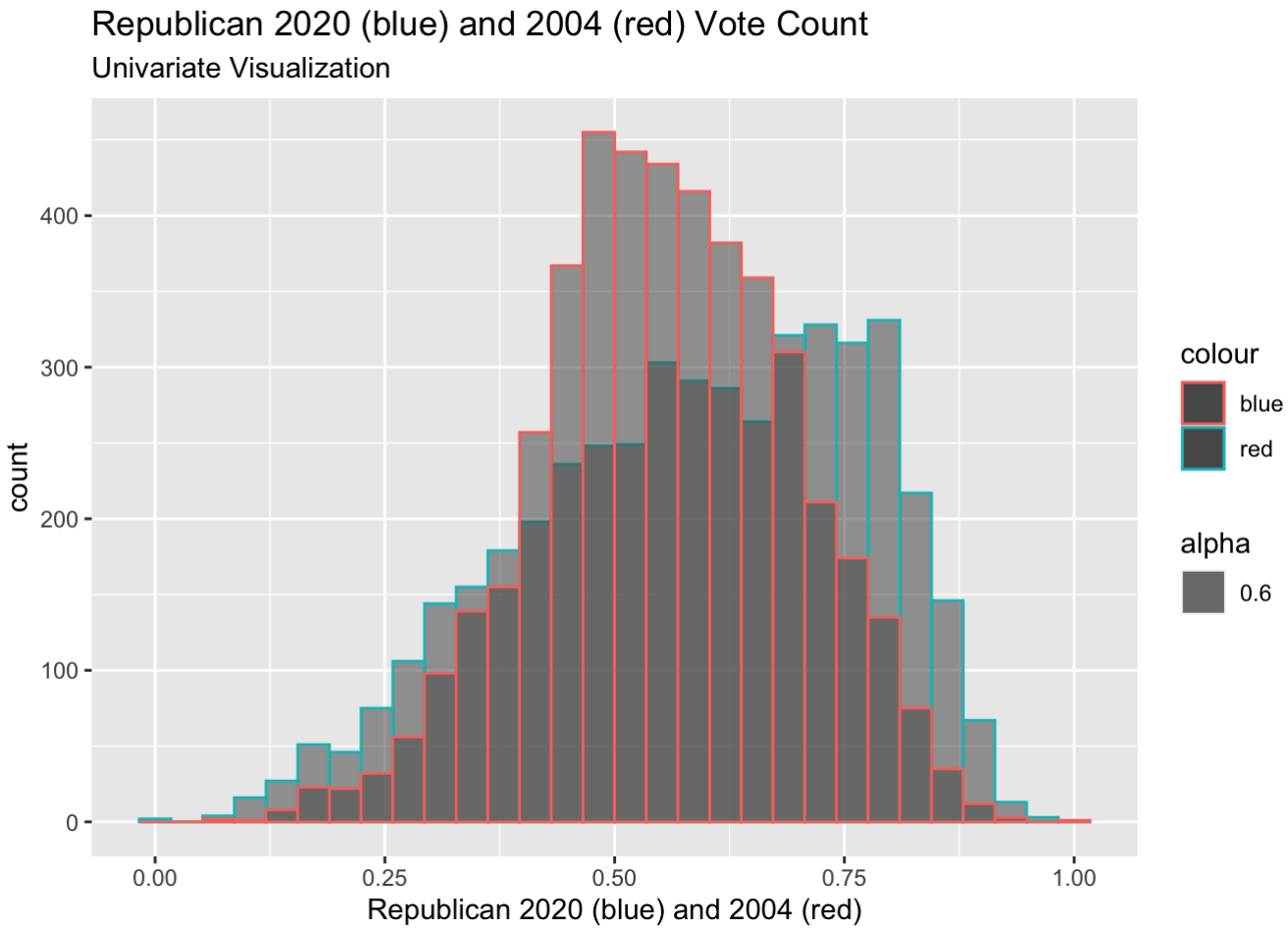
```
#2 Visualize the data using univariate visualization for both measures, but put them
 on the same plot, differentiated by color. Do you notice any patterns?
g1 <- dat %>%
  select(pct_rep_2020, pct_rep_2004) %>%
  ggplot() +
  geom_histogram(aes(x = pct_rep_2020, color = 'red', alpha = 0.6)) +
  geom_histogram(aes(x = pct_rep_2004, color = 'blue', alpha = 0.6)) +
  labs(title = "Republican 2020 (blue) and 2004 (red) Vote Count",
       subtitle = "Univariate Visualization",
       x = "Republican 2020 (blue) and 2004 (red)")

g1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

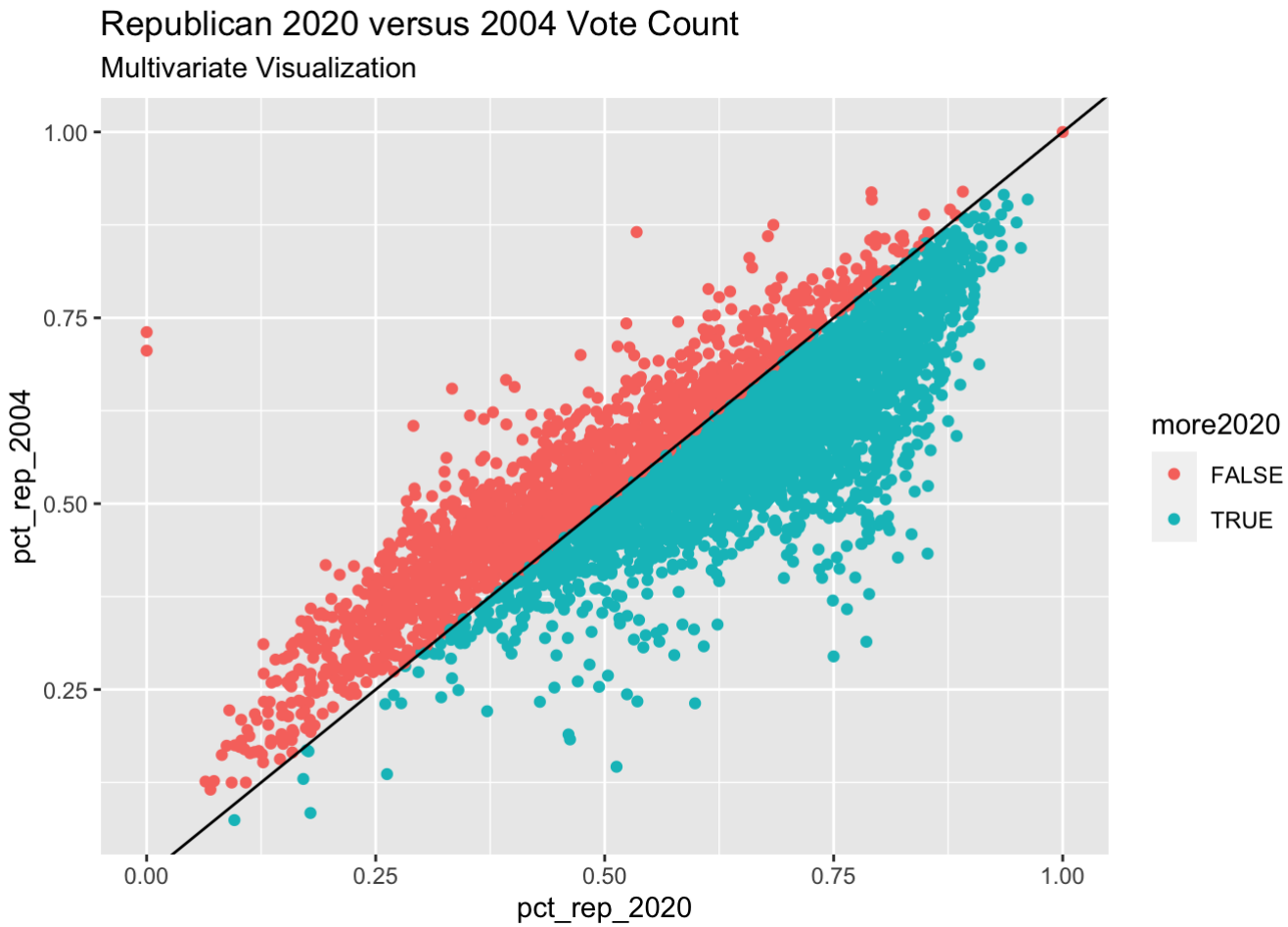
```
## Warning: Removed 35 rows containing non-finite values (stat_bin).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 54 rows containing non-finite values (stat_bin).
```




```
#3 Visualize the data using multivariate (or conditional) visualization where `pct_rep_2020` is the outcome and `pct_rep_2004` is the predictor. Use the `geom_abline()` to create a 45 degree line on these plots. Do you notice any patterns? EXTRA CREDIT: Interpret this plot substantively and color the points by whether they are above or below the 45 degree line.
g2 <- dat %>%
   select(c(pct_rep_2020, pct_rep_2004)) %>%
  drop_na() %>%
  mutate(more2020 = ifelse(pct_rep_2020>pct_rep_2004, TRUE, FALSE)) %>%
  ggplot() +
  geom_point(aes(x = pct_rep_2020, y = pct_rep_2004, color = more2020)) +
  geom_abline() +
  labs(title = "Republican 2020 versus 2004 Vote Count",
       subtitle = "Multivariate Visualization",
       x = "pct_rep_2020",
       y = "pct_rep_2004")

g2
```



- According to the summary of the dataset, pct_rep_2004 variable has missingness in states: AK and ME. pct_rep_2020 variable has missing data in the ME and VA states. According to the univariate analysis of the data, pct_rep_2020 variable is more left skewed than the pct_rep_2004 variable, meaning that in 2020, greater percentage of republican count had higher count compared to the lower percentage of republican vote count. According to the multivariate analysis of the data, more data points lie below the 45% line, meaning that, at average, republican vote share percentage was higher in 2020.

Question 2 [4 points]

Perform *k*-means analysis on these variables with $k = 2$, and then plot the results, coloring the points by cluster assignment. Then loop over values of *k* from 1 to 30 and plot the "elbow plot" with *k* on the x-axis and the total within sum of squares on the y-axis. What value of *k* would you choose? Re-calculate with that value, and then plot again. **NB: set `nstart = 25` to ensure replicability!** EXTRA CREDIT: Are you able to interpret these groups as a political scientist?

```
require(scales)
```

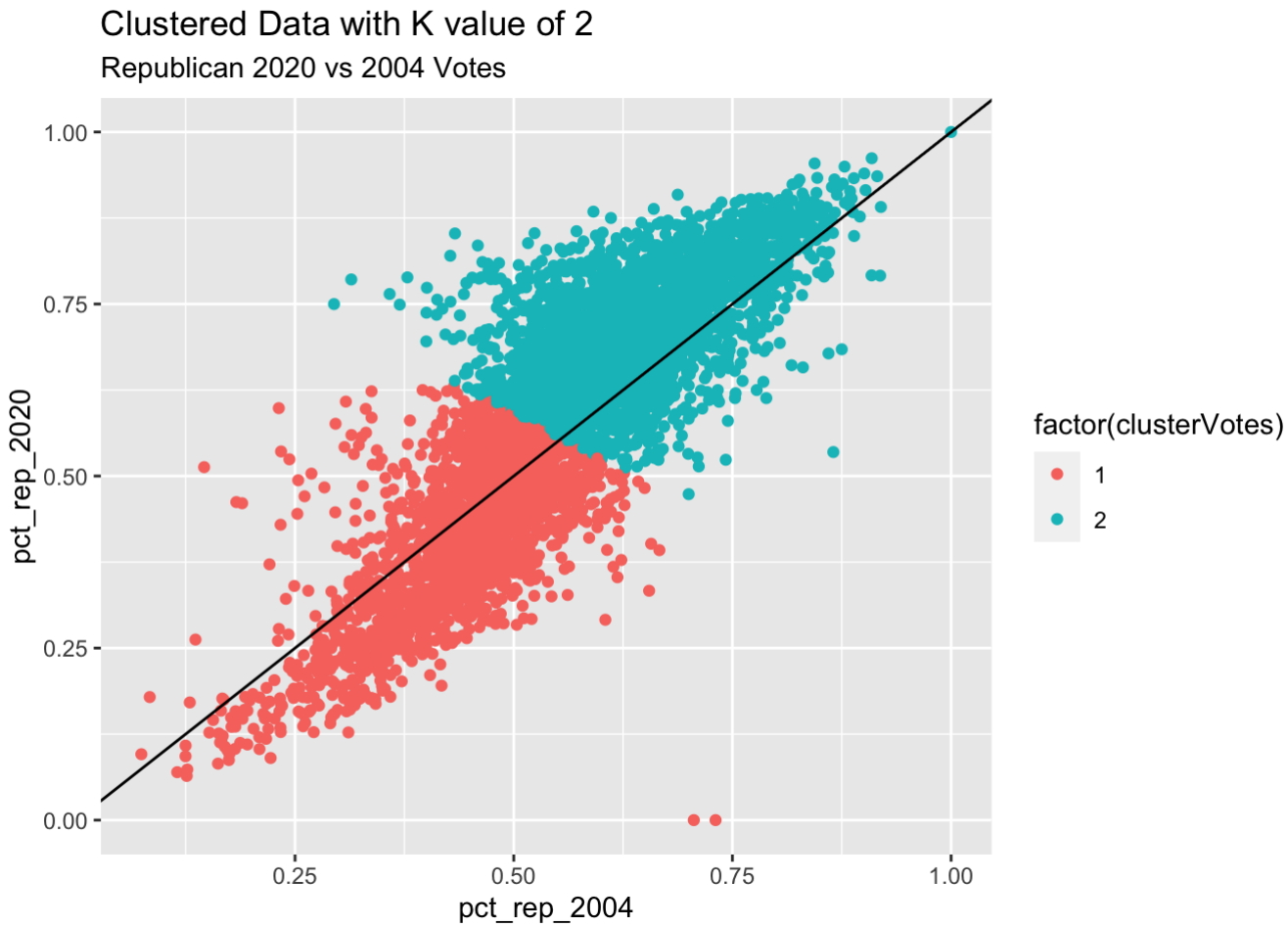
```
## Loading required package: scales
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

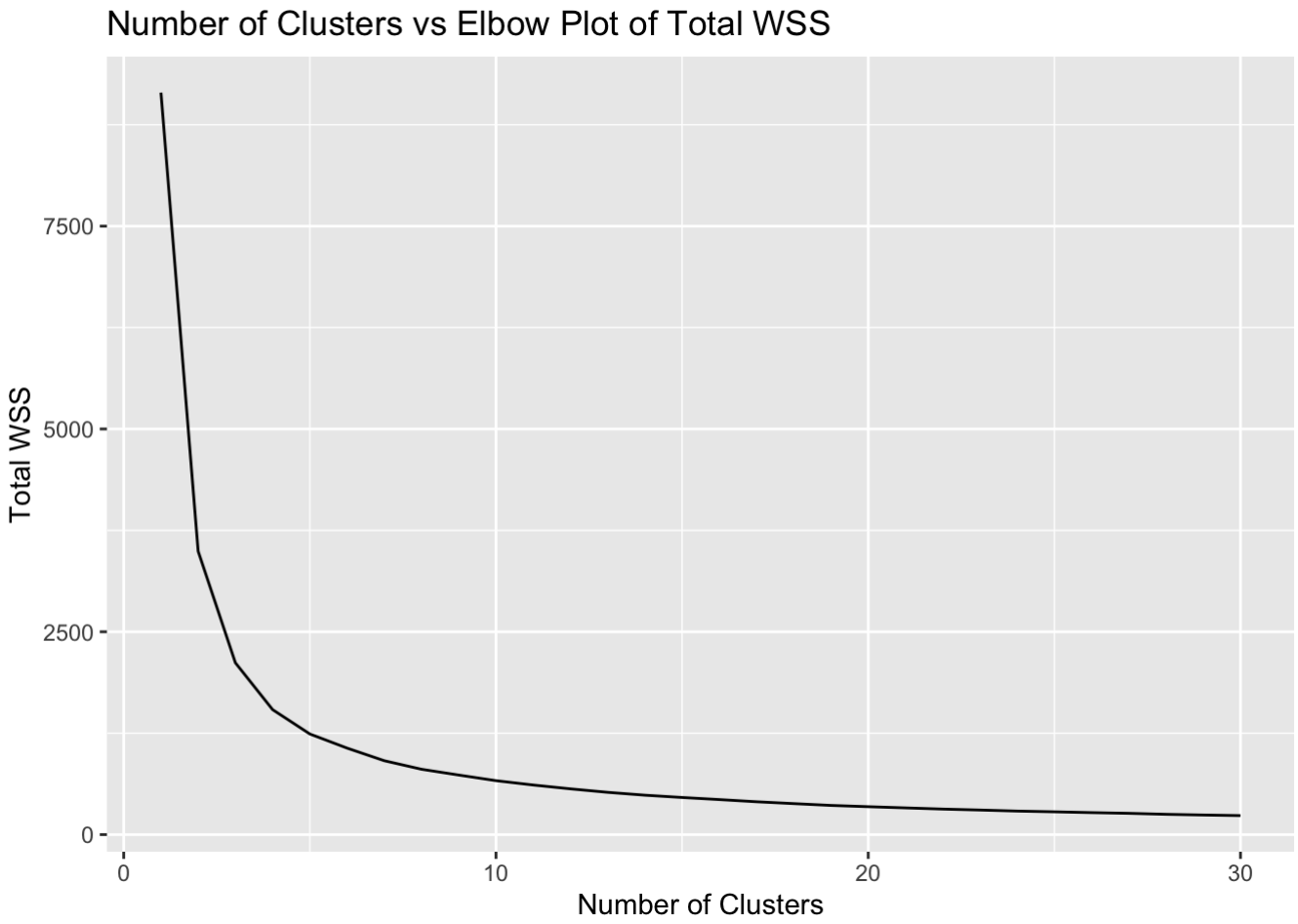
```
scaledDat <- scale(dat %>%
                   select(pct_rep_2004,pct_rep_2020) %>%
                   drop_na()) %>%
                   data.frame()

#Perform *k*-means analysis on these variables with *k* = 2
datClust <- dat %>% select(pct_rep_2004,pct_rep_2020) %>% drop_na()
cluster1 <- kmeans(datClust, centers = 2)

#and then plot the results, coloring the points by cluster assignment.
ggVotes <- datClust %>%
  mutate(clusterVotes = cluster1$cluster) %>%
  ggplot(aes(x = pct_rep_2004, y = pct_rep_2020, color = factor(clusterVotes))) +
  geom_point() +
  geom_abline() +
  labs(title = "Clustered Data with K value of 2",
       subtitle = "Republican 2020 vs 2004 Votes",
       x = "pct_rep_2004",
       y = "pct_rep_2020")

ggVotes
```




```
# Then loop over values of *k* from 1 to 30 and plot the "elbow plot" with *k* on the
x-axis and the total within sum of squares on the y-axis.
totWSS <- NULL
for(k in 1:30) {

  rep.clusterVotes <- scaledDat %>%
  kmeans(centers = k, nstart = 25, iter.max = 100)
  totWSS <- data.frame(totWSS = rep.clusterVotes$tot.withinss, k=k) %>%
  bind_rows(totWSS)
}
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 228700)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 228700)
```

```
totWSS %>%
  ggplot(aes(x = k, y = totWSS)) +
  geom_line() +
  labs(title= "Number of Clusters vs Elbow Plot of Total WSS",
       x = 'Number of Clusters',
       y = 'Total WSS')
```




```
# What value of *k* would you choose? Re-calculate with that value, and then plot aga
in.
datClust2 <- dat %>% select(pct_rep_2004,pct_rep_2020) %>% drop_na()
cluster2 <- kmeans(datClust2, centers = 4)

ggVotes2 <- datClust2 %>%
  mutate(clusterVotes = cluster2$cluster) %>%
  ggplot(aes(x = pct_rep_2004, y = pct_rep_2020, color = factor(clusterVotes))) +
  geom_point() +
  geom_abline() +
  labs(title = "Clustered Data with K value of 2",
       subtitle = "Republican 2020 vs 2004 Votes",
       x = "pct_rep_2004",
       y = "pct_rep_2020")

ggVotes2
```

- According to the Number of Clusters vs Elbow Plot of Total WSS Graph, the identified 'elbow point' in the plot coincides to the Cluster number of 4, where the point there the curve bends which is the optimal point that balances accuracy and parsimony. This means that there are 4 main clusters where the pct_rep_2004 and pct_rep_2020 data values were the most similar to each other.

Question 3 [4 points]

Now open the `FederalistPaperCorpusTidy.Rds` dataset (download from here (%5Bhttps://github.com/jbisbee1/DS1000-F2022/blob/master/Lectures/Topic9_Clustering/data/FederalistPaperCorpusTidy.Rds%5D)). Require the `tidytext` package (install it if you haven't yet) and tokenize the data via the `unnest_tokens()` function, stemming the words via the `token = "word_stems"` input. Remove stop words and then calculate the most frequently used words by author. Plot the top 10 words by author and interpret the results. Do you notice any patterns in how different authors write?

```
#Now open the `FederalistPaperCorpusTidy.Rds` dataset
dataset <- readRDS('../data/FederalistPaperCorpusTidy.Rds')

#Require the `tidytext` package (install it if you haven't yet)
require(tidytext)
```

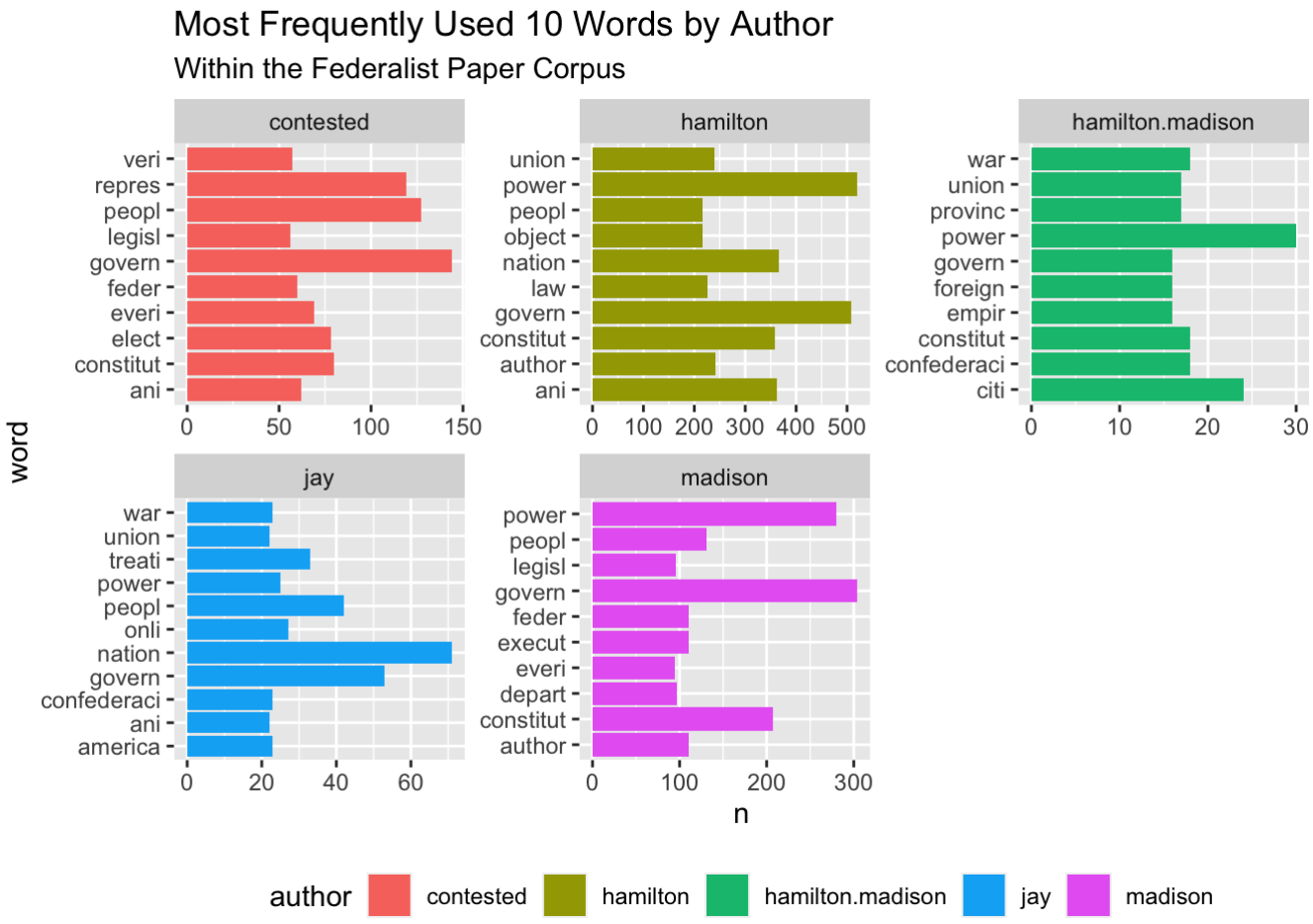
```
## Loading required package: tidytext
```

```
#tokenize the data via the `unnest_tokens()` function, stemming the words via the `to
ken = "word_stems"` input
tokens <- dataset %>%
  unnest_tokens(output = word, input = text, token = "word_stems") %>%
  mutate(word = str_replace_all(word, "\\d+", "")) %>%
  filter(word != c("of", "the", "to"))
```

```
## Warning: Outer names are only allowed for unnamed scalar atomic inputs
```

```
## Warning in word != c("of", "the", "to"): longer object length is not a multiple
## of shorter object length
```

```
data("stop_words", package = "tidytext")

tokens <- anti_join(tokens, stop_words, by = "word")

tokens <- tokens %>%
  filter(!word %in% stop_words$word)

g5 <- tokens %>%
  group_by(author) %>%
  count(word) %>%
  top_n(10, wt = n) %>%
  arrange(-n) %>%
  ggplot(aes(x = word, y = n, fill = author)) +
  geom_bar(stat = 'identity') +
  coord_flip() +
  facet_wrap(~author, scales = 'free') +
  theme(legend.position = 'bottom') +
  labs(title = "Most Frequently Used 10 Words by Author",
       subtitle = "Within the Federalist Paper Corpus")
g5
```

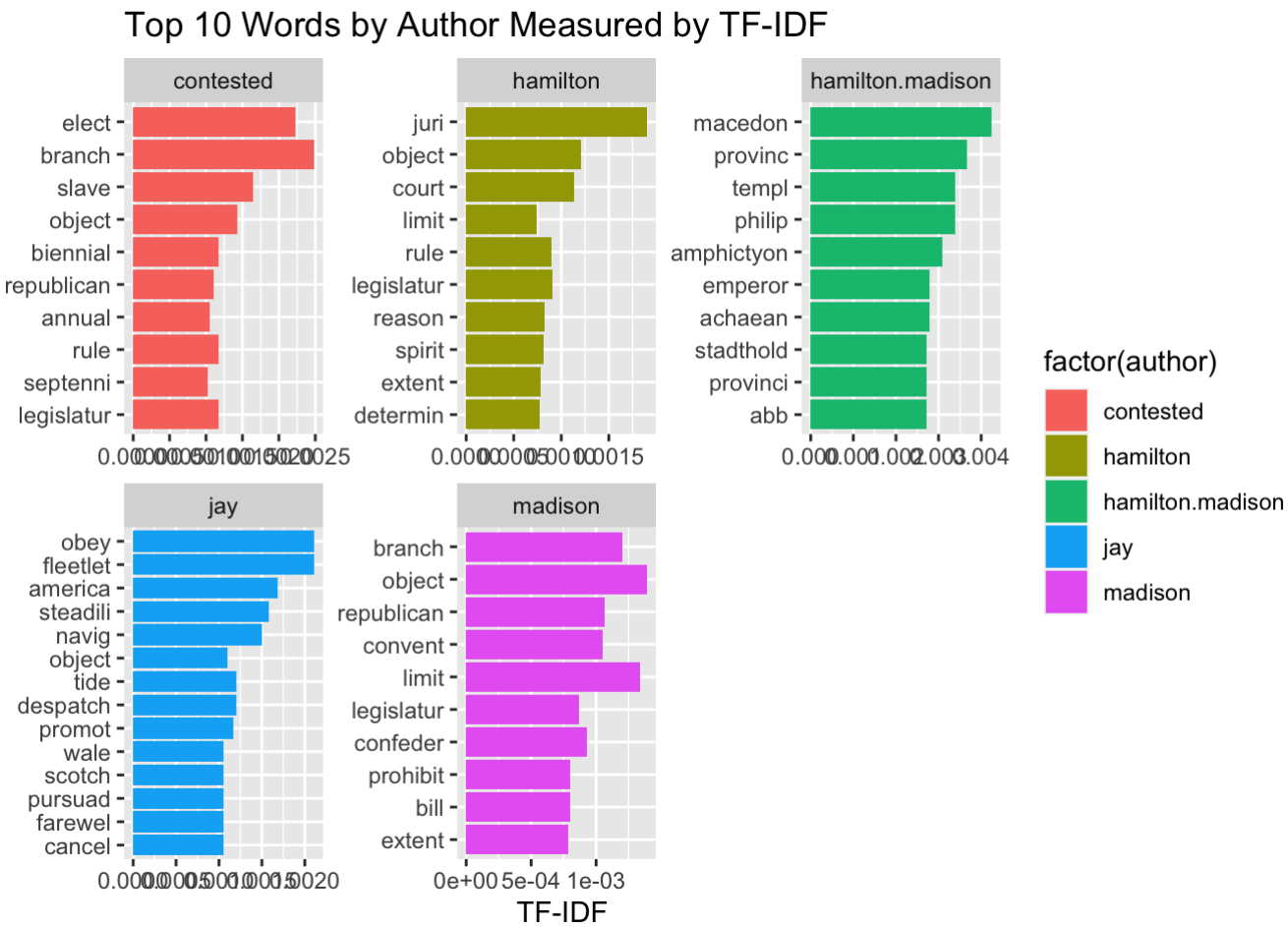


- While there are word that both Madison and Hamilton use in common (such as power, constitution, and people), there also are words that only one of them uses. For example, Hamilton uses the word law, nation, and union while Madison doesn't. Madison uses the words execute, legislation, and department. This difference indicates that the topic Madison focuses on is more oriented on the nation while the languages Madison uses is more oriented on the government.

Question 4 [4 points]

Create an author-term matrix (analogous to a document term matrix except organized by author). Then calculate the TF-IDF by author and plot the top 10 words by TF-IDF for each author. Do you observe any noticeable differences now?

```
atm <- tokens %>%
  count(author, word)

atm.tfidf <- bind_tf_idf(atm, word, author, n)

p <- atm.tfidf %>%
  group_by(author) %>%
  top_n(10, wt = tf_idf) %>%
  ggplot(aes(x = tf_idf, y = reorder(word, tf_idf), fill = factor(author))) +
  geom_bar(stat = 'identity') +
  labs(x = 'TF-IDF', y = NULL, title = "Top 10 Words by Author Measured by TF-IDF") +
  facet_wrap(~author, scales = 'free')

p
```




```
castdtm <- cast_dtm(atm.tfidf, author, word, tf_idf)
```

-Now, the difference is much more recognizable when the topics are taken into account. When we observe the use of the words of juri, court, rule, legislature, and object, we can conclude that Hamilton's documents were adressing issues regarding the court and jurisdiction. Madison's document has the words: object, prohibit, bill, and limit, indicating that her documents were on the topic of law-making. The documents written both by Hamilton and Madison involved words such as Macedon, Archean, Emperor, and Philip, indicating that the documents were written about Ancient Greek Emporers and their was of government.

Question 5 [4 points]

Now create a document term matrix (DTM) only using the documents we know were written by Hamilton. As above, calculate the TF-IDF and then use this to estimate *k*-means clustering on these text data. To do so, start by "casting" the data via the `cast_dtm()` function. Then calculate the *k*-means analysis using $k = 5$ and then visualize the top 10 words per cluster. Can you interpret them? (Hint: use the `tidy()` function from the `tidymodels` package to help here.) **NB: set `nstart = 25` to ensure replicability!**

```
dtm <- tokens %>%
  filter(author == 'hamilton') %>%
  count(document, word)

dtm.tfidf <- bind_tf_idf(dtm, word, document, n)

dtm.tfidf %>%
  filter(document <= 10) %>%
  top_n(10, wt=tf_idf)
```

```
## # A tibble: 10 × 6
##    document word            n      tf   idf tf_idf
##       <int> <chr>       <int>   <dbl> <dbl>  <dbl>
##  1        6 carthag         4 0.00529 3.93  0.0208
##  2        6 war            18 0.0238  0.713 0.0170
##  3        7 apportion       4 0.00480 2.83  0.0136
##  4        7 cession         3 0.00360 3.93  0.0142
##  5        7 connecticut     7 0.00840 2.55  0.0214
##  6        8 armi           15 0.0192  1.29  0.0248
##  7        8 disciplin       5 0.00639 2.55  0.0163
##  8        8 militari       12 0.0153  1.10  0.0169
##  9        9 montesquieu     4 0.00546 3.93  0.0215
## 10        9 republ         10 0.0137  1.37  0.0187
```

```
library(tm)
```

```
## Loading required package: NLP
```

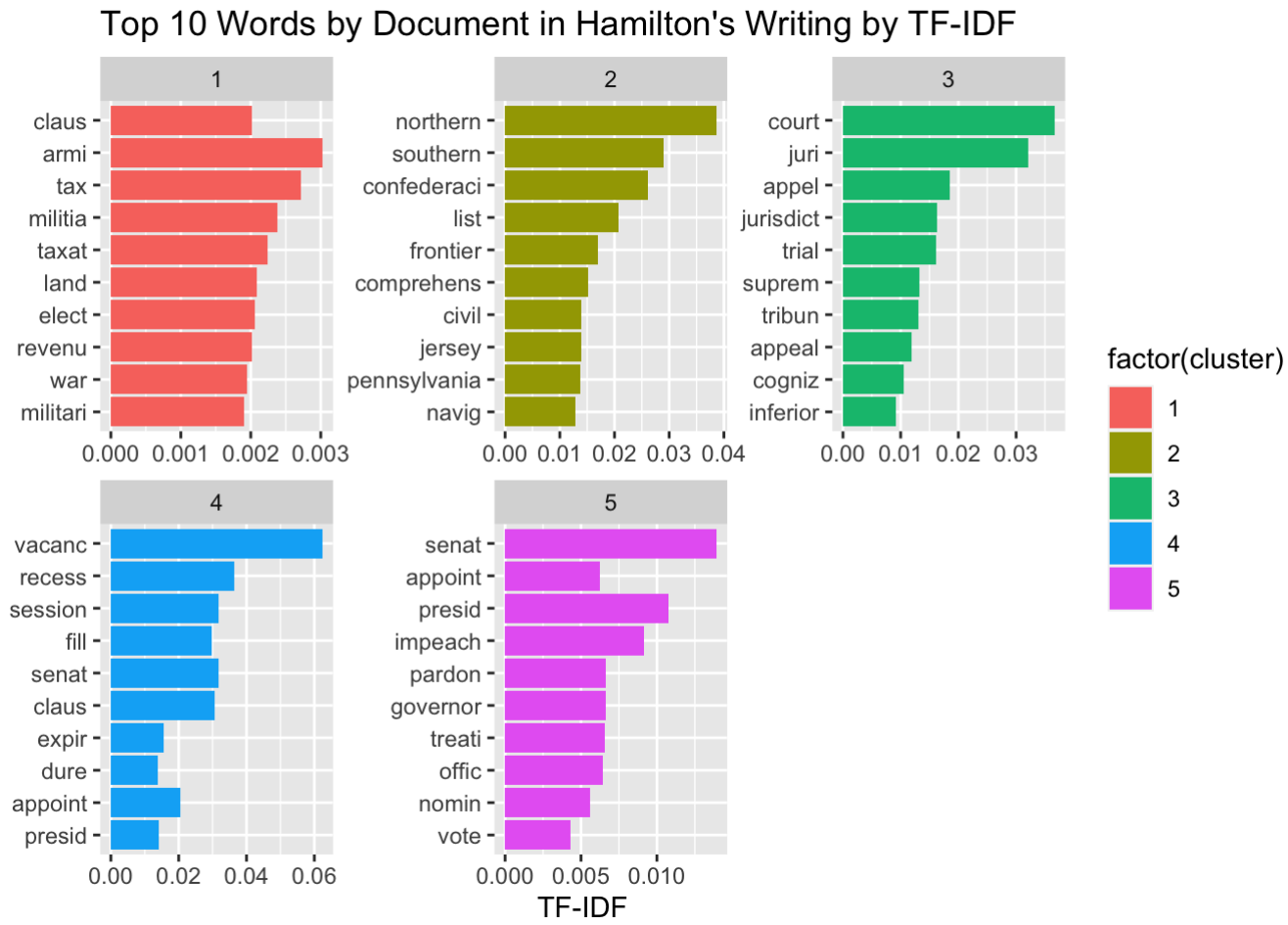
```
##
## Attaching package: 'NLP'
```

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

```
dtm.tfidf %>%
  filter(document <= 10) %>%
  top_n(10, wt=tf_idf)
```

```
## # A tibble: 10 × 6
##    document word            n      tf   idf tf_idf
##       <int> <chr>       <int>   <dbl> <dbl>  <dbl>
##  1        6 carthag         4 0.00529 3.93  0.0208
##  2        6 war            18 0.0238  0.713 0.0170
##  3        7 apportion       4 0.00480 2.83  0.0136
##  4        7 cession         3 0.00360 3.93  0.0142
##  5        7 connecticut     7 0.00840 2.55  0.0214
##  6        8 armi           15 0.0192  1.29  0.0248
##  7        8 disciplin       5 0.00639 2.55  0.0163
##  8        8 militari       12 0.0153  1.10  0.0169
##  9        9 montesquieu     4 0.00546 3.93  0.0215
## 10        9 republ         10 0.0137  1.37  0.0187
```

```
castdtm <- cast_dtm(dtm.tfidf, document, word, tf_idf)
km_out = kmeans(castdtm, centers = 5, nstart = 25)
km_out_tidy <- tidy(km_out) %>%
  gather(word, avg_tfidf, -size, -cluster, -withinss) %>%
  mutate(avg_tfidf = as.numeric(avg_tfidf))

km_out_tidy %>%
  group_by(cluster) %>%
  arrange(-avg_tfidf) %>%
  slice(1:10) %>%
  ggplot(aes(x = avg_tfidf,
             y = reorder(word, avg_tfidf),
             fill = factor(cluster))) +
  geom_bar(stat = 'identity') +
  facet_wrap(~cluster, scales = 'free') +
  labs(x = 'TF-IDF', y = NULL, title = "Top 10 Words by Document in Hamilton's Writin
g by TF-IDF")
```



- The blue colored topic is about the topics of elections.
- The pink colored words are about the topic of jurisdiction.
- The green words are about militia.
- The yellow words are about the northern vs southern comparison.
- The tomato red words are about governing.

EXTRA CREDIT [4 points]

Re-do question 5 but on Madison instead of Hamilton. Do you notice any differences between the clusters among essays written by Hamilton versus those written by Madison?

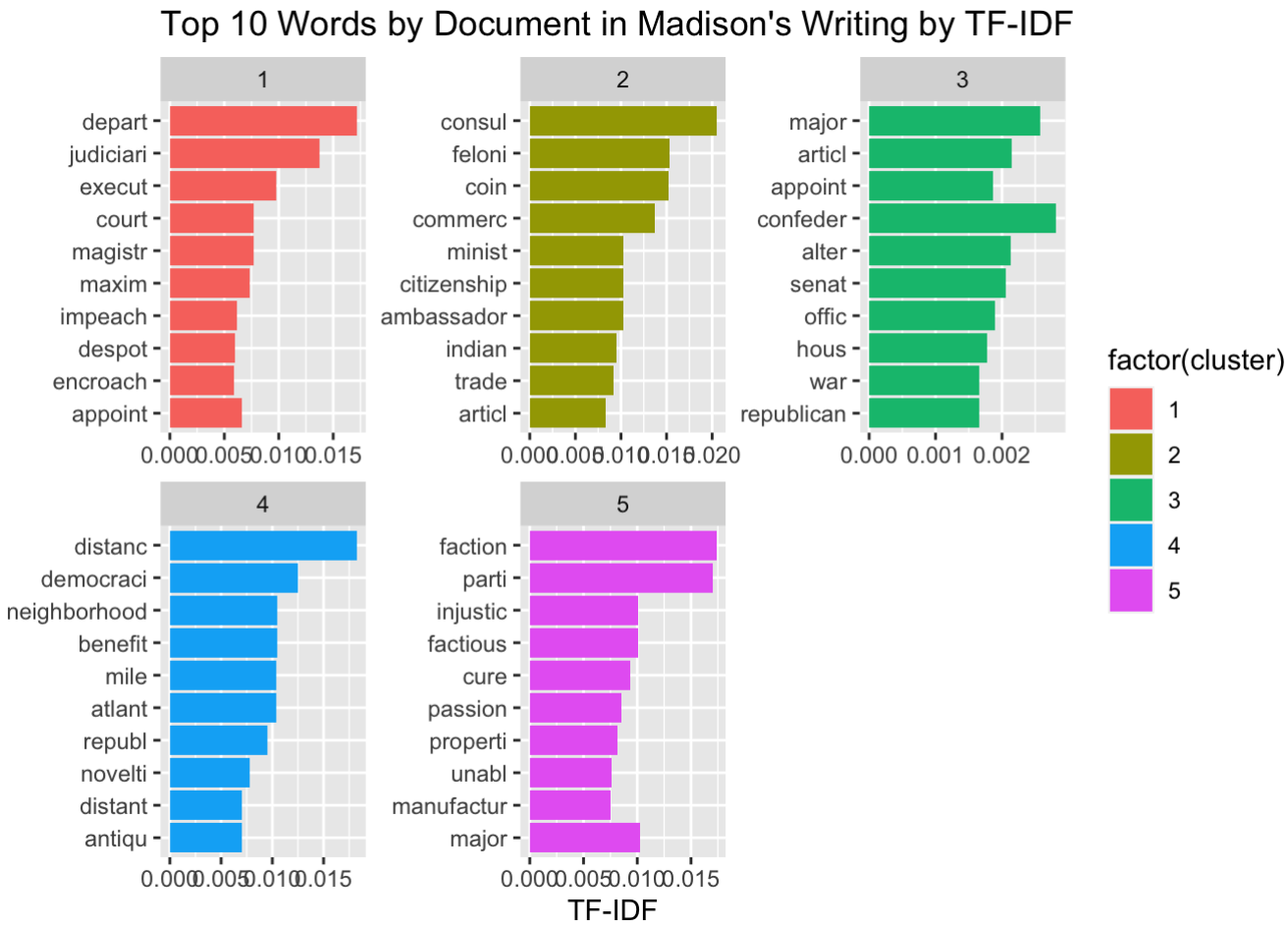
```
dtm <- tokens %>%
  filter(author == 'madison') %>%
  count(document, word)

dtm.tfidf <- bind_tf_idf(dtm, word, document, n)

dtm.tfidf %>%
  filter(document <= 10) %>%
  top_n(10, wt=tf_idf)
```

```
## # A tibble: 10 × 6
##    document word           n      tf   idf  tf_idf
##       <int> <chr>      <int>   <dbl> <dbl>   <dbl>
##  1       10 cure           5 0.00466 2.01  0.00938
##  2       10 faction       17 0.0158  1.10  0.0174
##  3       10 factious       4 0.00372 2.71  0.0101
##  4       10 injustic       4 0.00372 2.71  0.0101
##  5       10 major         12 0.0112  0.916 0.0102
##  6       10 manufactur     4 0.00372 2.01  0.00750
##  7       10 parti         20 0.0186  0.916 0.0171
##  8       10 passion       10 0.00931 0.916 0.00853
##  9       10 properti       8 0.00745 1.10  0.00818
## 10       10 unabl          3 0.00279 2.71  0.00756
```

```
library(tm)

dtm.tfidf %>%
  filter(document <= 10) %>%
  top_n(10, wt=tf_idf)
```

```
## # A tibble: 10 × 6
##    document word           n      tf   idf  tf_idf
##       <int> <chr>      <int>   <dbl> <dbl>   <dbl>
##  1       10 cure           5 0.00466 2.01  0.00938
##  2       10 faction       17 0.0158  1.10  0.0174
##  3       10 factious       4 0.00372 2.71  0.0101
##  4       10 injustic       4 0.00372 2.71  0.0101
##  5       10 major         12 0.0112  0.916 0.0102
##  6       10 manufactur     4 0.00372 2.01  0.00750
##  7       10 parti         20 0.0186  0.916 0.0171
##  8       10 passion       10 0.00931 0.916 0.00853
##  9       10 properti       8 0.00745 1.10  0.00818
## 10       10 unabl          3 0.00279 2.71  0.00756
```

```
castdtm <- cast_dtm(dtm.tfidf, document, word, tf_idf)
km_out = kmeans(castdtm, centers = 5, nstart = 25)
km_out_tidy <- tidy(km_out) %>%
  gather(word, avg_tfidf, -size, -cluster, -withinss) %>%
  mutate(avg_tfidf = as.numeric(avg_tfidf))

km_out_tidy %>%
  group_by(cluster) %>%
  arrange(-avg_tfidf) %>%
  slice(1:10) %>%
  ggplot(aes(x = avg_tfidf,
             y = reorder(word, avg_tfidf),
             fill = factor(cluster))) +
  geom_bar(stat = 'identity') +
  facet_wrap(~cluster, scales = 'free') +
  labs(x = 'TF-IDF', y = NULL, title = "Top 10 Words by Document in Madison's Writing
by TF-IDF")
```



- The blue colored topic is about the topics of elections.
- The pink colored words are about finances and trade.
- The green words are about faction of the country.
- The yellow words are about jurisdiction
- The tomato red words are about international relations.