Problem Set 4 (last one!)

Clustering & NLP

[YOUR NAME]

Due Date: Thursday, 2025-06-26

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]_ps4.Rmd` to your `code` folder.

Copy and paste the contents of this `.Rmd` file into your `[LAST NAME]_ps4.Rmd` file. Then change the `author: [Your Name]` 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. To submit, compile (i.e., `knit as pdf`) the completed problem set and upload the PDF file to Blackboard on Friday by midnight. Be sure to check your knitted PDF for mistakes before submitting!

This problem set is worth 9 total points, plus 1 extra credit point. **This is due on Thursday (6/26), the final day of the semester; not on Friday.** The point values for each question are indicated in brackets below. To receive full credit, you must have the correct code. In addition, some questions ask you to provide a written response in addition to the code.

You will be deducted 1 point for each day late the problem set is submitted, and 1 point for failing to submit in the correct format (i.e., not knitting as a PDF).

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.

Note that the professor will not respond to Campuswire posts after 2PM on Friday, so don't wait until the last minute to get started!

Good luck!

If you collaborated with a colleague and/or used AI for any help on this problem set, document here. Write the names of your classmates and/or upload a PDF of your AI prompt and output with your problem set:

Question 0

Require `tidyverse` and `tidymodels`, and then load the `pres_elec.rds` data to an object called `dat`.

Question 1 [0.5 points]

Describe the data. What is the unit of analysis? What information do the columns provide? What is the period described (i.e., how far back in time does the data go?). Is there any missing data? If so, "where" is it, in terms of both columns and in terms of the observations that have missing data?

```
# INSERT CODE HERE
```

- Write response here

Question 2 [1.5 point]

Perform k-means analysis on the Republican and Democrat votes with $k = 2$, and then plot the results, coloring the points by cluster assignment. Then predict the `GOP_win` binary outcome as a function of the cluster assignment using a logit regression. Make sure to `factor(cluster)` in the regression. What is the AUC for this model? Finally, use cross validation with an 80-20% split to re-calculate the AUC. Overall, would you say that the k-means algorithm helps you predict which counties vote Republican?

```
set.seed(123)
# K-means with k = 2
# INSERT CODE HERE

# Plotting the result
dat %>%
  select(...) %>%
  drop_na() %>%
  mutate(cluster = ...) %>%
  ggplot(aes(x = ...,y = ...,color = factor(...),group = 1)) +
  geom_point() +
  labs(x = '',
       y = '',
       color = '')
```

```
## Error in dat %>% select(...) %>% drop_na() %>% mutate(cluster = ...) %>% : could not find function "
```

```
# Create dataset for analysis
toanal <- dat %>%
  select(...,...,...) %>%
  drop_na() %>%
  mutate(cluster = ...)
```

```
## Error in dat %>% select(..., ..., ...) %>% drop_na() %>% mutate(cluster = ...): could not find functi
```

```
# Estimate logit model
summary(m <- glm(...,toanal,family = binomial))
```

```
## Error in summary(m <- glm(..., toanal, family = binomial)): '...' used in an incorrect context
```

```
# Calculate AUC
roc_auc(toanal %>%
          mutate(prob_win = ...,
                 truth = ...),
        truth,prob_win)
```

```
## Error in roc_auc(toanal %>% mutate(prob_win = ..., truth = ...), truth, : could not find function "ro
```

```
# Calculate cross-validated result
cvRes <- NULL
for(i in 1:100) {
  # INSERT CODE HERE
}

# Cross-validated AUC
# INSERT CODE HERE
```

- Write response here.

Question 3 [2 points]

Now create an elbow plot by looping over potential values of k from 1 to 30 and plotting the k on the x-axis and the total Within Sum of Squares (total WSS) on the y-axis. What value of k would you use? Then re-run the preceding analysis with that value of k and interpret the results. Does the model improve?

```
# Looking at multiple values of k
kRes <- NULL
for(k in 1:30) {
  # Calculate k-means cluster solution for given value of k
  kResTmp <- # INSERT CODE HERE

  # Save result including value of k and the total WSS
  kRes <- data.frame(withinSS = ...,
                     k = ...) %>%
    bind_rows(kRes)
}
```

```
## Error in data.frame(withinSS = ..., k = ...) %>% bind_rows(kRes): could not find function "%>%"
```

```
# Plotting the elbow plot. Looks like k=4 is the elbow?
# INSERT CODE HERE

# Rerunning with optimal k
# INSERT CODE HERE

# Plotting again
# INSERT CODE HERE

# Create dataset for analysis
# INSERT CODE HERE

# Estimate logit model
# INSERT CODE HERE

# Calculate AUC
# INSERT CODE HERE

# Calculate cross-validated result
# INSERT CODE HERE
```

```
# Cross-validated AUC
# INSERT CODE HERE
```

- Write response here

Question 0.B

Require `tidyverse`, `tidytext` and `tidymodels`, and then load the `Trump_tweet_words.Rds` data to an object called `tweet_words`.

```
# INSERT CODE HERE
```

Question 4 [1 point]

a. Plot the total number of times the word "trump" is used each year. Then, plot the proportion of times the word "trump" is used each year. Make sure to justify your choice of `geom_...()`!

b. Why are these plots so different? Which measure is better? Why?

```
# Total number of times
tweet_words %>%
  count(...) %>% # Calculate total number of times each word was used in each year
  filter(...) %>% # Filter to the word of interest
  ggplot(aes(x = ...,
             y = ...)) +  # Plot results
  geom_...() + # Choose appropriate geom_...()
  labs(x = '', # Provide descriptive labels
       y = '',
       title = '')
```

```
## Error in tweet_words %>% count(...) %>% filter(...) %>% ggplot(aes(x = ..., : could not find functio
```

```
# Proportion of times
tweet_words %>%
  count(...) %>% # Calculate total number of times each word was used in each year
  group_by(...) %>% # Calculate total number of words used each year
  mutate(...) %>%
  ungroup() %>%
  mutate(prop = ...) %>% # Calculate proportion
  filter(...) %>% # Filter to the word of interest
  ggplot(aes(x = ...,
             y = ...)) +  # Plot results
  geom_...() + # Choose appropriate geom_...()
  labs(x = '', # Provide descriptive labels
       y = '',
       title = '')
```

```
## Error in tweet_words %>% count(...) %>% group_by(...) %>% mutate(...) %>% : could not find function
```

- Write answer here

Question 5 [2 points]

We want to only look at tweets written during Trump's first year as president (January 20th, 2017 through December 31st, 2017), and are interested if there are patterns in what he talks about.

We will use k-means clustering to learn about this data. To do so, follow these steps.

a. Create a document-term matrix (`dtm`), dropping any words that appear fewer than 20 times total, and using the `document` column as the document indicator. **NB: Drop the word `'amp'`.**

b. Calculate the TF-IDF using the appropriate function from the `tidytext` package.

c. Cast the DTM to wide format using the `cast_dtm()` function, also from the `tidytext` package.

d. 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)` and set `nstart = 5` with `set.seed(123)`. (This will still take a little while to run so be patient!).

e. Using the optimal value from the elbow plot, run k-means on the data with `nstart` set to 5 and `set.seed(123)`.

f. Which are the top 3 most popular topics for Donald Trump in this period? Plot the top 10 highest scoring words for each of the top 3 most popular topics. What is each "about"?

```
# a.
dtm <- tweet_words %>%
  filter(..., # Filter to the correct period
         ...) %>%  # Drop the word 'amp'
  count(...) %>%  # Count the number of times each word appears in each document
  group_by(...) %>%  # Count the total number of times a word appears overall
  mutate(tot_n = sum(...)) %>%
  ungroup() %>%
  filter(...) # Filter to only words that appear more than 20 times in total
```

```
## Error in tweet_words %>% filter(..., ...) %>% count(...) %>% group_by(...) %>% : could not find funct
```

```
#b.
dtm.tfidf <- bind_tf_idf(tbl = ...,  # Calculate the TF-IDF metric
                         term = ...,
                         document = ...,
                         n = ...)
```

```
## Error in bind_tf_idf(tbl = ..., term = ..., document = ..., n = ...): could not find function "bind_
```

```
#c.
castdtm <- cast_dtm(data = ...,  # Cast to a DTM
                    document = ...,
                    term = ...,
                    value = ...)
```

```
## Error in cast_dtm(data = ..., document = ..., term = ..., value = ...): could not find function "cas
```

```
#d.
# INSERT CODE HERE (see pset 10 if you need a refresher)

#e.
# INSERT CODE HERE (see pset 10 if you need a refresher)

km_out_tidy <- tidy(...) %>% # Tidy the kmeans result
  gather(...) %>% # Pivot to long format
  mutate(...) # Convert the average TF-IDF value to numeric
```

```
## Error in tidy(...) %>% gather(...) %>% mutate(...): could not find function "%>%"
```

```
# For students who can't load tidymodels
# km_out_tidy <- as_tibble(km_out$centers) %>%
#   mutate(size = km_out$size,
#          withinss = km_out$withinss,
#          cluster = factor(row_number())) %>%
#   gather(word,mean_tfidf,-size,-cluster,-withinss)

#f. Find the top 3 topics tweeted by Trump
tops <- km_out_tidy %>%
  select(...) %>% # Select only the size and cluster
  distinct() %>% # Keep only distinct rows
  arrange(...) %>% # Arrange by size in descending order
    slice(...) # Get top 3 topics
```

```
## Error in km_out_tidy %>% select(...) %>% distinct() %>% arrange(...) %>% : could not find function "
```

```
# Visualize the top 10 words in the top 3 most used topics
km_out_tidy %>%
  filter(cluster %in% ...) %>% # Filter to the topics found above
  group_by(...) %>% # Arrange in descending order of average TF-IDF
  arrange(...) %>%
  slice(...) %>% # Get top 10 words
  ggplot(aes(x = ...,
             y = ..., # Reorder words by highest scoring TF-IDF
             fill = ...)) + # Fill by topic
  geom_...(...) + # Choose the appropriate geom_...()
  facet_wrap(~...,scales = 'free') + # Create facets by topics
  labs(title = '', # Good labels!
       subtitle = '',
       x = '',
       y = '',
       fill = '')
```

```
## Error in km_out_tidy %>% filter(cluster %in% ...) %>% group_by(...) %>% : could not find function "%
```

- Write answer here

Question 6 [2 points]

Now load the sentiment dictionary `nrc` from the `tidytext` package, and look at the clusters with sentiment scores by merging the `km_out_tidy` dataset with the `nrc` dataset using the `inner_join()` function. (If you

can't open the `nrc` object from the `tidytext` package, you can just load it from GitHub with this link: `https://github.com/rweldzius/PSC4175_SUM2025/raw/main/Data/nrc.Rds`)

Filter to only look at `positive` and `negative` categories and then `select()` only the `size`, `cluster`, `word`, `mean_tfidf`, and `sentiment` columns. Then use either `spread()` or `pivot_wider()` to create two columns of `mean_tfidf` values: one for `positive` and one for `negative`. Replace `NA` values with 0! Finally, filter to only look at clusters with more than 10 tweets in them. Save this processed data to an object named `cluster_sentiment`.

Using this data, plot the top 10 words for the three most positive clusters and the top 10 words for the three most negative clusters. Describe what you see. What are Trump's most positive and negative topics about?

```
# Load the NRC dictionary

cluster_sentiment <- km_out_tidy %>%
  inner_join(nrc %>% # Join on the words that appear in both (ignore the warning)
               filter(...)) %>% # Filter the nrc to only positive and negative labels
  select(...) %>% # Select the columns size, cluster, word, mean_tfidf, and sentiment
  spread(...) %>% # Spread the data into two columns, one for
  mutate(net_sentiment = ...) %>% # Calculate the net_sentiment as positive - negative
  group_by(cluster,size) %>% # Calculate the average sentiment by cluster and size
  summarise(net_sentiment = ...) %>%
  ungroup() %>%
  filter(...) # Drop clusters with fewer than 10 tweets
```

```
## Error in km_out_tidy %>% inner_join(nrc %>% filter(...)) %>% select(...) %>% : could not find functi
```

```
# Calculate the top 3 most positive clusters
top_sentiment <- cluster_sentiment %>%
  arrange(...) %>%  # Arrange in descending order of net sentiment
  slice(...) # Get top 3
```

```
## Error in cluster_sentiment %>% arrange(...) %>% slice(...): could not find function "%>%"
```

```
# Visualize the top 10 words in the top 3 most positive clusters
km_out_tidy %>%
  filter(cluster %in% ...) %>% # Filter to the topics found above
  group_by(...) %>% # Arrange in descending order of average TF-IDF
  arrange(...) %>%
  slice(...) %>% # Get top 10 words
  ggplot(aes(x = ...,
             y = ..., # Reorder words by highest scoring TF-IDF
             fill = ...)) + # Fill by topic
  geom_...(...) + # Choose the appropriate geom_...()
  facet_wrap(~...,scales = 'free') + # Create facets by topics
  labs(title = '', # Good labels!
       subtitle = '',
       x = '',
       y = '',
       fill = '')
```

```
## Error in km_out_tidy %>% filter(cluster %in% ...) %>% group_by(...) %>% : could not find function "%
```

```
# Calculate the bottom 3 most negative clusters
bottom_sentiment <- cluster_sentiment %>%
  arrange(...) %>%  # Arrange in ascending order of net sentiment
  slice(...) # Get bottom 3
```

```
## Error in cluster_sentiment %>% arrange(...) %>% slice(...): could not find function "%>%"
```

```
# Visualize the top 10 words in the bottom 3 most negative clusters
km_out_tidy %>%
  filter(cluster %in% ...) %>% # Filter to the topics found above
  group_by(...) %>% # Arrange in descending order of average TF-IDF
  arrange(...) %>%
  slice(...) %>% # Get top 10 words
  ggplot(aes(x = ...,
             y = ..., # Reorder words by highest scoring TF-IDF
             fill = ...)) + # Fill by topic
  geom_...(...) + # Choose the appropriate geom_...()
  facet_wrap(~...,scales = 'free') + # Create facets by topics
  labs(title = '', # Good labels!
       subtitle = '',
       x = '',
       y = '',
       fill = '')
```

```
## Error in km_out_tidy %>% filter(cluster %in% ...) %>% group_by(...) %>% : could not find function "%
```

- Write answer here

Extra Credit [1 point]

Which of Trump's topics are the most "popular", measured by total retweets? You will need to get creative with this final extra credit question. Broadly, you will need to link each tweet to the topic it was assigned as well as the number of retweets it received. This will require you to exploit the fact that the `km_out$cluster` data includes both the cluster to which each observation was assigned, as well as the tweet ID associated with that observation. Once you have created this lookup object, you can then link the original `tweet_words` dataset with the clusters. (NOTE: not every tweet will be assigned to a cluster, since we are dropping many of them.) This will require you to pay attention to object types (the names of the `km_out$cluster` are character, but the document IDs are stored as numeric in the `tweet_words` object), think creatively about how to merge the datasets, be aware of `NA`'s and think about how to deal with them, and then finally analyze the data once you have built it. Your end result should be, as above, the top 10 words associated with the top 3 most popular topics. Good luck!

```
# INSERT CODE HERE
```

- Write answer here.