Homework 6

Homework 6

Student Name: Jama Brookes

This homework builds on the effective visualization workshop with the Star Trek data. Below is what we completed in class. Output is suppressed for readability, but you can remove the supression on your code if you'd like.

```
invisible({
suppressPackageStartupMessages(library(tidyverse))

# Get the data.
dialogs <- read_csv(
  "https://raw.githubusercontent.com/Vincent-Toups/bios512/fcbc65a2696c7cff80d0f6ed1dd5c97abf0ef800/eff
  show_col_types = FALSE
)
head(dialogs, 10) # Showing first 10 observations

# Checkout the data.
names(dialogs)
dialogs %>% group_by(character) %>% tally() %>% arrange(desc(n))
dialogs %>% mutate(dialog_length=str_length(dialog)) %>% group_by(character) %>% summarize(mean_dialog_

# Fix weird data.
dialogs %>% filter(character=="BEVERLY'S")

dialogs_fixed <- dialogs %>%
  mutate(
    character = str_replace_all(character, "'S.*$", ""),
    character = str_replace_all(character, " VOICE", ""),
    character = str_replace_all(character, "\\.", ""),
    character = str_replace_all(character, "'", ""),
    character = str_replace_all(character, "S COM", ""),
    character = str_replace_all(character, " COM", ""),
    dialog_length = str_length(dialog)
  ) %>%
  filter(character %in% unlist(str_split("PICARD RIKER DATA TROI BEVERLY WORF WESLEY GEORDI", " ")))

dialogs_fixed %>% group_by(character) %>% summarize(mean_dialog_length = mean(dialog_length), std_dialog

dialog_len_per_ep <- dialogs_fixed %>% group_by(character, episode_number) %>% summarize(mean_dialog_le

dialog_len_per_ep

# Plot the data.
```

```r
ggplot(dialogs_fixed) + geom_density(aes(x=dialog_length))

for_factor <- dialog_len_per_ep %>% group_by(character) %>% summarise(m=mean(mean_dialog_length)) %>% ar
ggplot(dialog_len_per_ep, aes(factor(character,for_factor$character), mean_dialog_length)) + geom_boxplo

dialog_len_per_ep <- dialogs_fixed %>%
    group_by(character, episode_number) %>%
    summarize(mean_dialog_length = mean(dialog_length), dialog_count=n(), .groups = "drop") %>%
    arrange(desc(mean_dialog_length))

ggplot(dialog_len_per_ep, aes(dialog_count, mean_dialog_length)) + geom_point(aes(color=character)) + fa
})
```

Question 1

In class, we left off on the plot below, which shows the distribution of dialog count by mean dialog length, where each point represents an episode. Interpret these results. How can we tell the character's role in the story by their plot?

```r
ggplot(dialog_len_per_ep, aes(dialog_count, mean_dialog_length)) +
    geom_point(aes(color=character)) +
    facet_wrap(~character)
```

Using information from the plot, we can determine how many episodes these characters spoke in by how many total individual points plotted. We then can assess the importance of their roles by how

many times they spoke in each episode (dialog_count) and the mean length of dialog when they spoke (mean_dialog_length). According to the plot, Picard was in many episodes (observed by the amount of points on the plot) and spoke often in the episodes with quite long dialog lengths on average. This suggests that Picard was a main character who often pushed the plot forward with their speaking role. In contract, Wesley was in a lot fewer episodes (observed by the fewer total points) and overall spoke a smaller amount of times with smaller average dialog length. This would suggest that Wesley was potentially a minor character or had a less frequent, supporting role.

Question 2

a) Compare Beverly's mean dialog per episode vs. mean dialog count per episode from season 1 (episodes 102-126) to season 3 (episodes 149-174) in a table. *Hints*:
- First, use `filter()` to get - 1) the dialog from only Beverly's character and 2) the episodes within the ranges given. - Then, add a season variable using `mutate()` with `case_when()`. - To create the means per episode, after your `mutate()` step, you'll need to `group_by()` season and episode number, then you can do your `summarize()` step to get the means by episode. At the end of the `summary()` statement (inside the parenthesis), add `.groups="drop"`. - Then, to get the mean of means, you'll do the same as above, but only grouping by season.

```
str(dialog_len_per_ep)
```

```
## tibble [1,258 x 4] (S3: tbl_df/tbl/data.frame)
##  $ character         : chr [1:1258] "TROI" "GEORDI" "BEVERLY" "BEVERLY" ...
##  $ episode_number    : num [1:1258] 129 249 242 159 261 116 135 162 199 207 ...
##  $ mean_dialog_length: num [1:1258] 186 179 158 146 134 ...
##  $ dialog_count      : int [1:1258] 2 1 2 2 2 7 2 23 4 2 ...
```

```
#creating season variable
dialog_len_per_ep <- dialog_len_per_ep %>% mutate(season = case_when(
           episode_number %in% (102:126) ~ 1,
           episode_number %in% (127:148) ~ 2,
           episode_number %in% (149:174) ~ 3))

#Beverly's mean dialog length per season
Beverly_means_episode <- dialog_len_per_ep %>%
      filter(character == "BEVERLY" & episode_number %in% (102:174)) %>%
      group_by(season, episode_number) %>%
      summarize(mean_dialog = mean(mean_dialog_length), .groups = "drop") %>%
      group_by(season) %>%
      summarize(beverly_mean_per_ep = mean(mean_dialog), .groups = "drop")

#Overall mean dialog length per season
overall_means_per_episode <- dialog_len_per_ep %>%
      filter(episode_number %in% 102:174) %>%
      group_by(season, episode_number) %>%
      summarize(mean_dialog = mean(mean_dialog_length), .groups = "drop") %>%
      group_by(season) %>%
      summarize(overall_mean_per_ep = mean(mean_dialog), .groups = "drop")

#Combining tables
Beverly_vs_overall <- Beverly_means_episode %>%
      left_join(overall_means_per_episode, by = "season")
```

```
#Printing table
Beverly_vs_overall
```

```
## # A tibble: 3 x 3
##   season beverly_mean_per_ep overall_mean_per_ep
##    <dbl>               <dbl>               <dbl>
## 1      1                56.5                53.1
## 2      2                35.7                53.9
## 3      3                67.0                59.1
```

b) In class, we talked about this character saying the actress has stated that after she was fired and rehired, the writers began giving her storylines that made her feel like a male character. How is this reflected in our table? In the first season, Beverly spoke more often than the average by about 3 words.However, in the 3rd season, Beverly's mean length of dialog for each season increased by 8 words compared to the overall mean for season 3. This suggests that Beverly got longer dialog overall, which would suggest a leadership/main role in the show that typically would be for male actors.

Question 3

Let's compare the vocabulary richness (unique words / total words) of each character. #### a) Tokenize dialog into words, remove punctuation, convert to lowercase. Then filter out the stop words in the list below (from https://gist.github.com/sebleier/554280). *Hint*: Here's a template for that this step should look like:

```
stop_words <- c(
  "i","me","my","myself","we","our","ours","ourselves","you","your","yours","yourself",
  "yourselves","he","him","his","himself","she","her","hers","herself","it","its","itself",
  "they","them","their","theirs","themselves","what","which","who","whom","this","that",
  "these","those","am","is","are","was","were","be","been","being","have","has","had",
  "having","do","does","did","doing","a","an","the","and","but","if","or","because","as",
  "until","while","of","at","by","for","with","about","against","between","into","through",
  "during","before","after","above","below","to","from","up","down","in","out","on","off",
  "over","under","again","further","then","once","here","there","when","where","why","how",
  "all","any","both","each","few","more","most","other","some","such","no","nor","not",
  "only","own","same","so","than","too","very","s","t","can","will","just","don","should","now"
)


tokens <- dialogs_fixed %>%
  # Split each dialog into words
  mutate(word_list = str_split(dialog, "\\s+")) %>%

  # Unnest the list column so each word is a row
  unnest(word_list) %>%

  # Clean words
  mutate(
    word = str_remove_all(word_list, "[[:punct:]]"), # Remove punctuation
    word = str_to_lower(word) # Convert to lowercase
  ) %>%

  # Remove empty strings and stopwords
```

```
  filter(word != "", !word %in% stop_words)

head(tokens)
```

```
## # A tibble: 6 x 6
##   episode_number character dialog                   dialog_length word_list word
##            <dbl> <chr>     <chr>                           <int> <chr>     <chr>
## 1            102 PICARD    Captain's log, stardat~           128 Captain's capt~
## 2            102 PICARD    Captain's log, stardat~           128 log,      log
## 3            102 PICARD    Captain's log, stardat~           128 stardate  star~
## 4            102 PICARD    Captain's log, stardat~           128 42353.7.  4235~
## 5            102 PICARD    Captain's log, stardat~           128 destinat~ dest~
## 6            102 PICARD    Captain's log, stardat~           128 planet    plan~
```

b) Count unique words per character. Print a summary table with the following columns: character, total words, unique words, and vocabulary richness. *Hint*: Group by character, then use `summarize()` to get what you want. You'll use `n_distinct()` to get the unique word counts. Arrange in descending value of vocabulary richness.

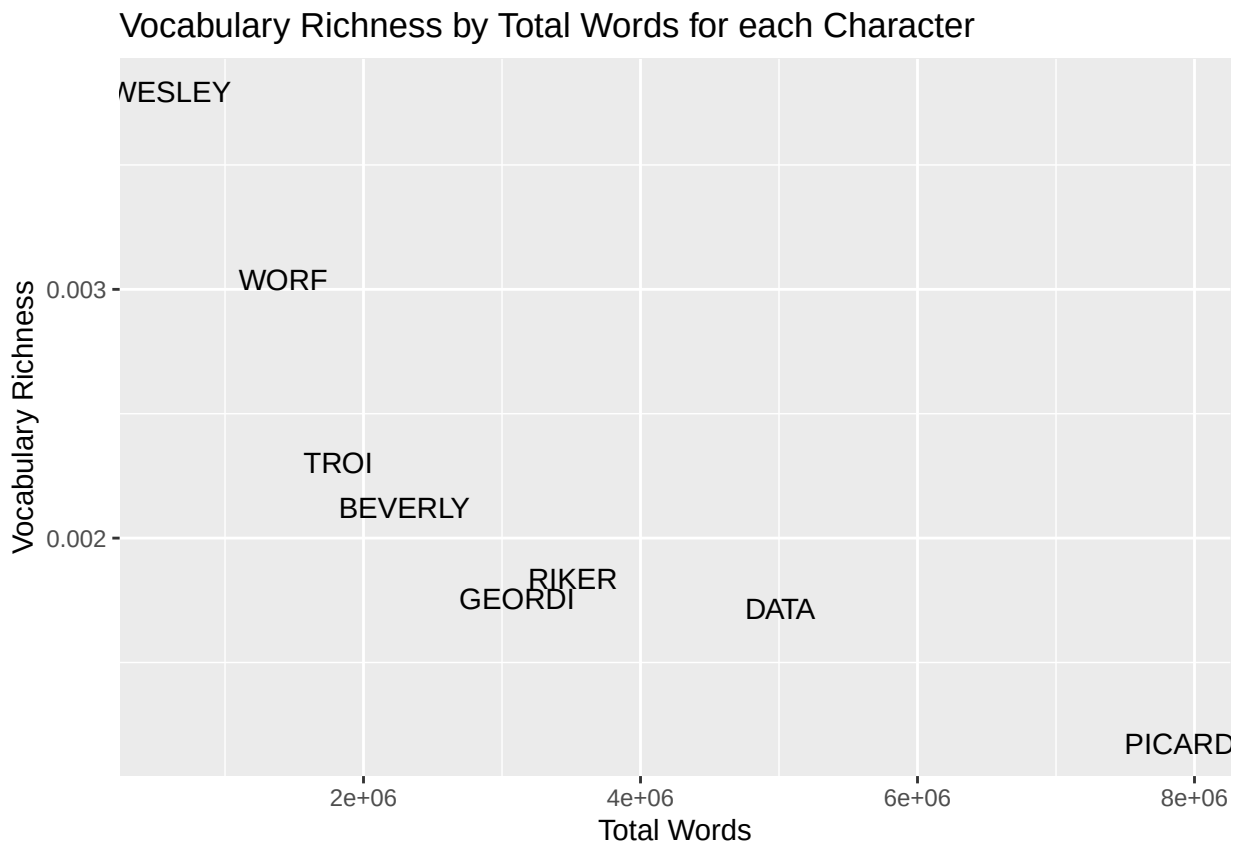
```
vocab_richness <- tokens %>%
  group_by(character) %>%
  summarize(total_words = sum(dialog_length),
            unique_words = n_distinct(word),
            vocabulary_richness = unique_words/total_words) %>%
  arrange(desc(vocabulary_richness))

vocab_richness
```

```
## # A tibble: 8 x 4
##   character total_words unique_words vocabulary_richness
##   <chr>           <int>        <int>               <dbl>
## 1 WESLEY         603644         2291             0.00380
## 2 WORF          1421059         4318             0.00304
## 3 TROI          1817372         4187             0.00230
## 4 BEVERLY       2294711         4875             0.00212
## 5 RIKER         3511683         6458             0.00184
## 6 GEORDI        3108907         5465             0.00176
## 7 DATA          5008971         8593             0.00172
## 8 PICARD        7894530         9272             0.00117
```

c) Plot total words versus vocab richness.

- Use the character names as the "points".
  - *Hint*: Use `geom_text()` to add the character names as the points.
- Do not include a legend.
  - *Hint*: Use `theme()` to remove the legend.
- Add a title and axis titles.
  - *Hint*: Use `labs()` to add titles.

```
vocab_richness %>% ggplot(aes(total_words, vocabulary_richness, group = character)) +
  geom_text(aes(label = character)) +
  theme(legend.position = "none") +
  labs(x = "Total Words",
       y = "Vocabulary Richness",
       title = "Vocabulary Richness by Total Words for each Character")
```



d) Interpret these results. There is an exponential decrease in vocabulary richness with the increase of total words. That means, as a character speaks many words, they are likely to have overall less unique words than characters who speak less often. Wesley has the highest vocabulary richness, likely because he appears less often, says less total words, but uses more unique words when speaking. This may suggest that Wesley has a large vocabulary, which may say something about his character's personality/role. However, having few total words spoken may inflate the "vocabualry richness" measure. On the other hand, Picard has the lowest vocabulary richness and highest amount of total words spoken. This may mean that Picard repeats a lot of phrases in the show that have high repetition.

Question 4

a) Find what episode Wesley left the show as a main character and state it explicitly. Meaning, find the first significant gap where he is not found in more than two episodes in a row. *Hint*: It's after season 3 (ended at episode 174), so you can filter out seasons 1-3 and print Wesley's dialog count per episode. Then, scan the table for the gap.

```
Wesley_leave <- dialog_len_per_ep %>%
  filter(character == "WESLEY") %>%
  arrange(episode_number)  %>%
  mutate(prev_episode = lag(episode_number),
    gap = episode_number - prev_episode)

Wesley_leave %>%
  filter(gap > 2) %>%
  subset(select = c(character, episode_number, prev_episode, gap)) %>%
  print()
```

```
## # A tibble: 6 x 4
##   character episode_number prev_episode   gap
##   <chr>              <dbl>        <dbl> <dbl>
## 1 WESLEY               122          119     3
## 2 WESLEY               126          123     3
## 3 WESLEY               206          183    23
## 4 WESLEY               219          206    13
## 5 WESLEY               263          219    44
## 6 WESLEY               272          263     9
```

Wesley first left the show as a main character in episode 183.

```
Wesley_leave %>% filter(episode_number >183) %>% print()
```

b) After Wesley leaves the main cast, in which episodes does he make cameo appearances?

```
## # A tibble: 4 x 7
##   character episode_number mean_dialog_length dialog_count season prev_episode
##   <chr>              <dbl>              <dbl>        <int>  <dbl>        <dbl>
## 1 WESLEY               206               46.9          131     NA          183
## 2 WESLEY               219               53.6           71     NA          206
## 3 WESLEY               263               66.3           18     NA          219
## 4 WESLEY               272               52.1           97     NA          263
## # i 1 more variable: gap <dbl>
```

He makes cameo appearances in episodes 206, 219, 263, and 272.

c) Dig back into the data. Print:

- Wesley's last piece of dialog before he left the main cast.
- Wesley's last piece of dialog ever.

Hint: To do this, you'll need to filter the `dialogs_fixed` data set to Welsey's lines and the episode number, and use `slice_tail(n = 1)` to get the last observation.

```
#last dialog before he left
dialogs_fixed %>%
  filter(character == "WESLEY", episode_number == c(183)) %>%
  slice_tail(n = 1) %>%
  print()
```

```
## # A tibble: 1 x 4
##   episode_number character dialog      dialog_length
##            <dbl> <chr>     <chr>               <int>
## 1            183 WESLEY    I can walk.            11
```

```
#last dialog ever
dialogs_fixed %>%
  filter(character == "WESLEY", episode_number == c(272)) %>%
  slice_tail(n = 1) %>%
  print()
```

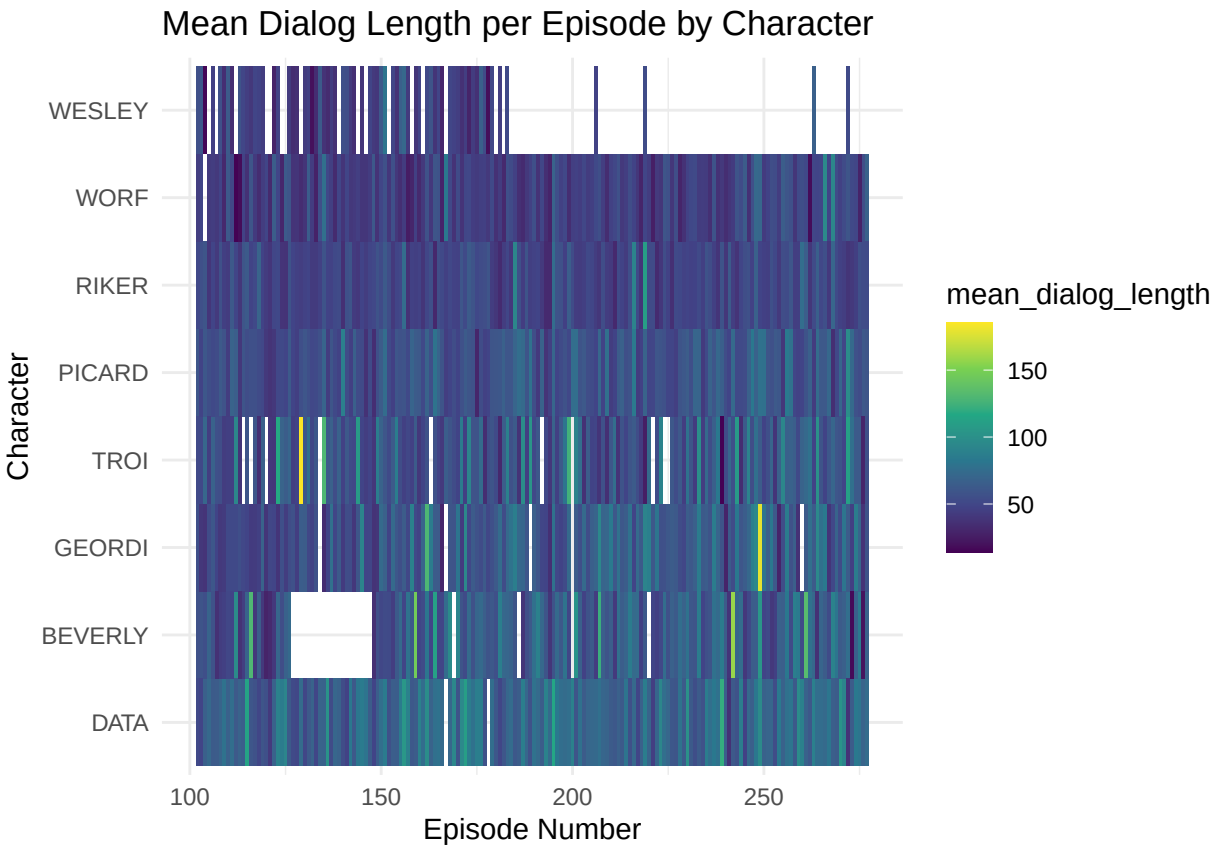
```
## # A tibble: 1 x 4
##   episode_number character dialog         dialog_length
##            <dbl> <chr>     <chr>                  <int>
## 1            272 WESLEY    Good-bye, Mom.            14
```

- Wesley's last piece of dialog before he left the main cast. "I can walk."
- Wesley's last piece of dialog ever. "Good-bye, Mom."

Question 5

Create a heatmap with `dialog_len_per_ep` showing mean dialog length per episode for each character. Sort the characters on the y-axis by their overall mean dialog length, with the lowest on top using a factor. Add a title and an axis title. *Hints*: For the factor: 1. Compute overall mean (mean of mean) dialog length per character (`group_by()` then `summarize()`), and arrange the overall mean in ascending order. Add `pull(character)` to the end of this step so that you can use character as a factor in the next step. Store all of this in a new tibble. 2. Convert character to factor with this order. On `dialog_len_per_ep`, you'll use a mutate statement to add the factor `(mutate(character = factor(character, levels = DATAFROMHINT1))`. 3. Create heatmap using `geom_tile()`. 4. If you want nicer colors, you can add `scale_fill_viridis_c()` (or another color scale) to your ggplot statement. **Not required**, but fun to mess around with!

```
character_order <- dialog_len_per_ep %>%
  group_by(character) %>%
  summarise(mean_of_means = mean(mean_dialog_length)) %>%
  arrange(desc(mean_of_means)) %>%
  pull(character)

dialog_len_per_ep <- dialog_len_per_ep %>%
  mutate(character = factor(character, levels = character_order))

dialog_len_per_ep %>% ggplot(aes(episode_number, character, fill = mean_dialog_length)) +
  geom_tile() +
  scale_fill_viridis_c() +
  labs(title = "Mean Dialog Length per Episode by Character",
    x = "Episode Number",
    y = "Character") +
  theme_minimal()
```


Mean Dialog Length per Episode by Character
WESLEY
WORF
RIKER
PICARD
TROI
GEORDI
BEVERLY
DATA
Character
100
150
200
250
Episode Number
mean_dialog_length
150
100
50