Homework 6

Collin Stewart

https://github.com/collings512/BIOS512_Collin_Stewart

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.

In [1]:
```
library(tidyverse)

# Get the data.
dialogs <- read_csv(
  "https://raw.githubusercontent.com/Vincent-Toups/bios512/fcbc65a2696c7cff80d0f6ed
  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) %>% su

# 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 WESLE

dialogs_fixed %>% group_by(character) %>% summarize(mean_dialog_length = mean(dialo

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

dialog_len_per_ep

# Plot the data.
ggplot(dialogs_fixed) + geom_density(aes(x=dialog_length))

for_factor <- dialog_len_per_ep %>% group_by(character) %>% summarise(m=mean(mean_d
ggplot(dialog_len_per_ep, aes(factor(character,for_factor$character), mean_dialog_l
```

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

ggplot(dialog_len_per_ep, aes(dialog_count, mean_dialog_length)) + geom_point(aes(c
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.2     ✓ readr     2.1.4
✓ forcats   1.0.0     ✓ stringr   1.5.0
✓ ggplot2   3.4.2     ✓ tibble    3.2.1
✓ lubridate 1.9.2     ✓ tidyr     1.3.0
✓ purrr     1.0.1
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

A tibble: 10 × 3

episode_number	character	dialog
<dbl>	<chr>	<chr>
102	PICARD	Captain's log, stardate 42353.7. Our destination is planet Cygnus IV, beyond which lies the great unexplored mass of the galaxy.
102	PICARD	My orders are to examine Farpoint, a starbase built there by the inhabitants of that world. Meanwhile ...
102	PICARD	... I am becoming better acquainted with my new command, this Galaxy Class U.S.S. Enterprise.
102	PICARD	I am still somewhat in awe of its size and complexity.
102	PICARD	... my crew we are short in several key positions, most notably ...
102	PICARD	... a first officer, but I am informed that a highly experienced man, one Commander William Riker, will be waiting to join our ship when we reach our Cygnus IV destination.
102	PICARD	You will agree, Data, that Starfleet's instructions are difficult?
102	DATA	Difficult ... how so? Simply solve the mystery of Farpoint Station.
102	PICARD	As simple as that.
102	TROI	Farpoint Station. Even the name sounds mysterious.

'episode_number' · 'character' · 'dialog'

A tibble: 914 × 2

character	n
<chr>	<int>
PICARD	15180
RIKER	8843
DATA	7390
GEORDI	5498
WORF	4429
BEVERLY	4035
TROI	3922
WESLEY	1664
PULASKI	705
TASHA	615
GUINAN	610
Q	573
O'BRIEN	551
COMPUTER VOICE	505
RO	458
BARCLAY	421
LWAXANA	385
VASH	225
ALEXANDER	221
K'EHLEYR	221
JELLICO	202
LORE	183
MORIARTY	182
MRS. TROI	165
SOREN	155
AMANDA	140
GOWRON	139
LT. RIKER	137
SCOTT	137

character	n
<chr>	<int>
KURN	131
⋮	⋮
PICARD.	1
PORTAL FIGURE	1
Q SHADOW	1
RIKER AND TASHA	1
RIKER'S COMM VOICE	1
RISON	1
RONIN VOICE	1
SANTOS' COM VOICE	1
SECOND EDO BOY	1
SECOND LEADER	1
SECURITY GUARD COM VOICE	1
SECURITY MAN	1
SERGANT	1
SUPERNUMERARY	1
TARRANA	1
TEMAREK	1
THE ALIEN	1
THE GROUP	1
TOYA	1
TRANSPORTER CHIEF'S VOICE	1
TRANSPORTER OPERATOR	1
TRANSPORTER TECHNICIAN	1
TYLER'S VOICE	1
VOVAL	1
WAGNOR	1
WEAK VOICE	1
WESLEY COM VOICE	1

character	n
<chr>	<int>
WOUNDED CREWMEMBER	1
YOUNG REPOTER	1
YOUNG RIKER	1

A tibble: 914 × 2

character	mean_dialog_length
<chr>	<dbl>
BEVERLY'S	243.0000
MAXWELL AND O'BRIEN	226.0000
CAPTAIN ZAHEVA	225.0000
HOLO SPOCK	221.0000
HUMANOID	216.5000
ANTHARWA	195.0000
END OF TEASER	187.0000
END OF ACT ONE	185.0000
VARLEY	181.2000
ADMIRAL BRACKETT	172.6667
KWAN'S VOICE	169.0000
SHELIAK VOICE	167.0000
CONNAUGHT	163.5000
DUANE	161.0000
O'BRIEN'S VOICE	160.0000
HAFTEL	157.5000
ADMIRAL HADEN	149.5000
MALE VOICE	144.3333
GEORI	143.0000
JARALAN	143.0000
BATES	142.5000
ACT ONE	142.0000
INDIANS	142.0000
TOQ'S VOICE	141.0000
ANTHWARA	137.2273
HOLO-SUSANNA	127.0000
MALE COM VOICE	125.5000
HOLO-LEAH	123.0000
TAGGERT	121.5000

character	mean_dialog_length
<chr>	<dbl>
PICARD'S RECORDED VOICE	119.0000
⋮	⋮
RIKER AND TASHA	15.00000
VOLNOTH	14.66667
HOLO-WESLEY	14.33333
ENSIGN CRAIG	14.00000
JACK'S VOICE	14.00000
PICARD.	14.00000
SALAZAR'S VOICE	13.50000
AIDE	13.00000
AMERICAN INDIAN	13.00000
PICARD'S INTERCOM VOICE	12.00000
SKORAN'S VOICE	12.00000
KORAL	11.20000
BENSON	11.00000
KELLER	11.00000
MAURICE'S VOICE	11.00000
N.D.	11.00000
YOUNG RIKER	11.00000
ALEXANDRA	10.50000
ANNA'S VOICE	10.00000
CRUSHER	10.00000
KAMALA'S VOICE	10.00000
N.D. VOICE	10.00000
CON OFFICER'S VOICE	9.00000
GAINES' COM VOICE	9.00000
BOY	7.00000
DATA FRANK	7.00000
TARRANA	7.00000

character	mean_dialog_length
<chr>	<dbl>
MOLLY'S VOICE	6.00000
JONO'S VOICE	5.00000
KAHLESS' VOICE	5.00000

A spec_tbl_df: 1 × 3

episode_number	character	dialog
<dbl>	<chr>	<chr>
253	BEVERLY'S	Acting Captain's Log, Supplemental: The skeleton crew left on board the Enterprise is unable to help in the search for Commander Data. The planet's unusual EM field is interfering with the ship's sensors, severely limiting their effectiveness.

A tibble: 8 × 3

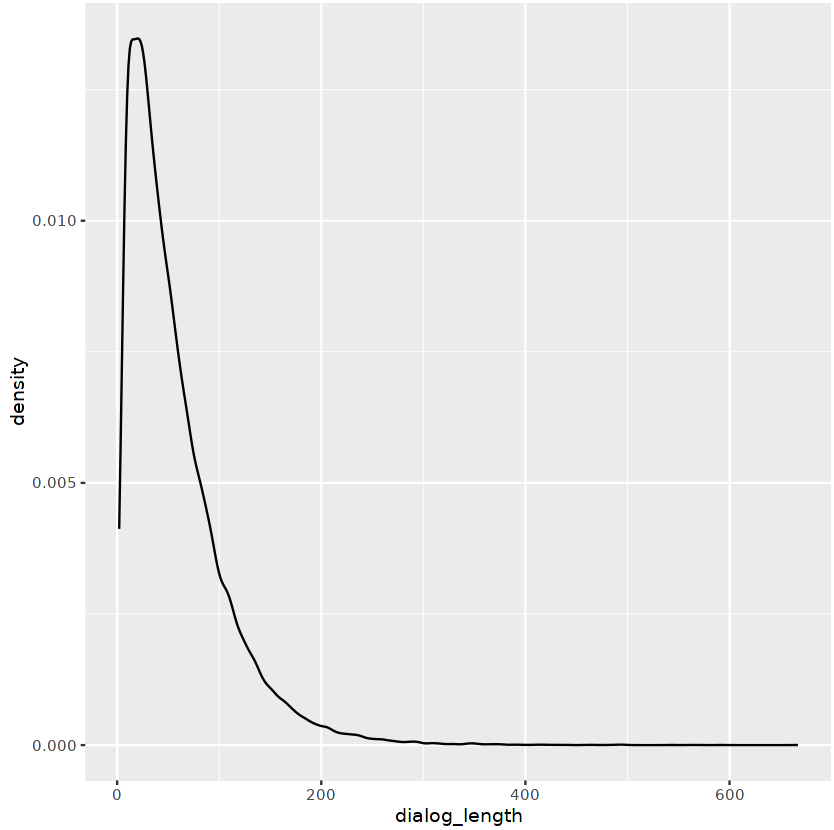
character	mean_dialog_length	std_dialog_length
<chr>	<dbl>	<dbl>
DATA	69.44862	54.14817
BEVERLY	61.79965	49.80177
GEORDI	60.57158	48.48001
PICARD	57.32696	50.98378
TROI	56.06494	46.65127
RIKER	49.76248	42.53393
WESLEY	47.11038	40.57180
WORF	47.05002	37.92466

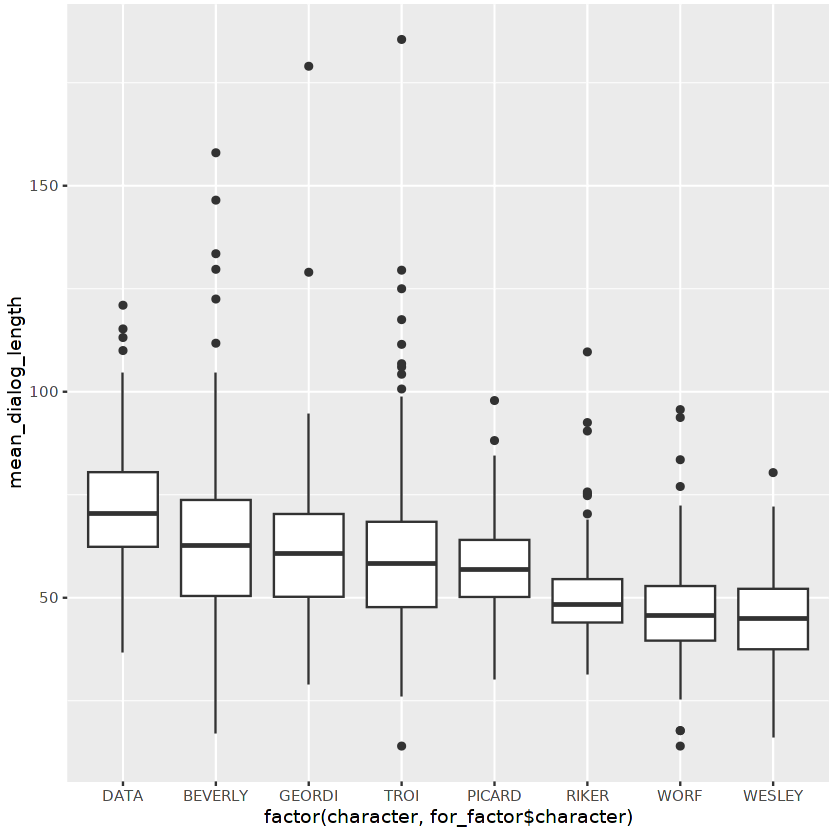
A tibble: 1258 × 4

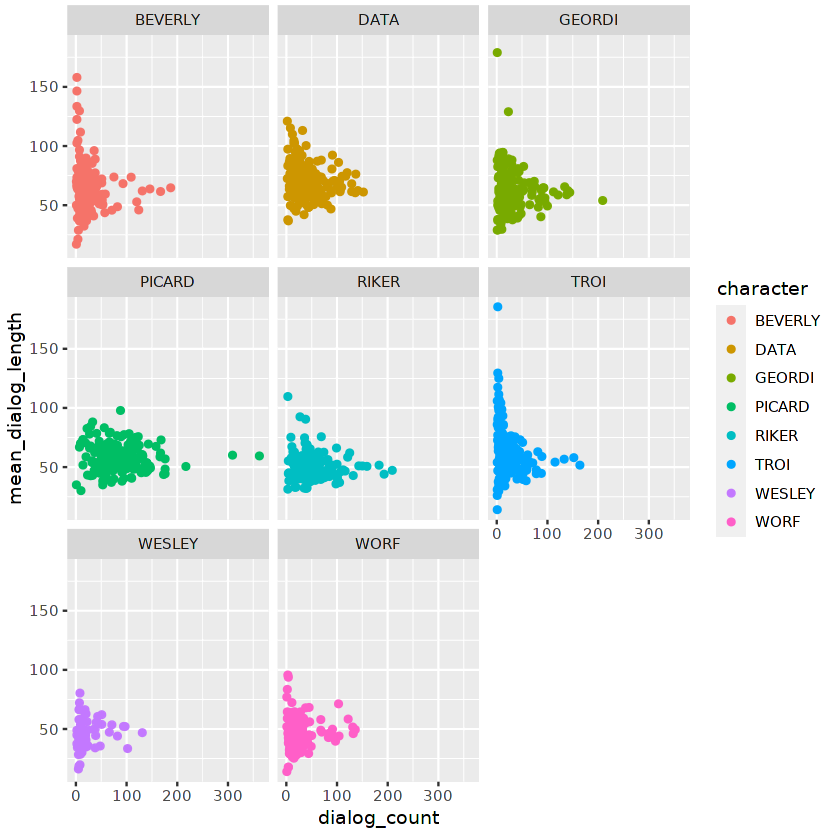
character	episode_number	mean_dialog_length	std_dialog_length
<chr>	<dbl>	<dbl>	<dbl>
TROI	129	185.5000	40.30509
GEORDI	249	179.0000	NA
BEVERLY	242	158.0000	57.98276
BEVERLY	159	146.5000	86.97413
BEVERLY	261	133.5000	20.50610
BEVERLY	116	129.7143	127.83807
TROI	135	129.5000	23.33452
GEORDI	162	129.0000	96.12728
TROI	199	125.0000	30.65942
BEVERLY	207	122.5000	75.66043
DATA	239	121.0000	94.75231
TROI	123	117.5000	89.80256
DATA	195	115.2500	68.82223
DATA	115	113.1562	96.89273
BEVERLY	164	111.7778	86.44475
TROI	272	111.5000	69.09173
DATA	172	110.0000	45.05754
RIKER	219	109.6667	77.70028
TROI	144	106.8000	41.82344
TROI	243	106.0000	NA
BEVERLY	249	104.7500	71.12138
DATA	156	104.6000	69.24573
TROI	189	104.2500	62.26155
DATA	270	102.6250	69.72219
BEVERLY	201	102.5000	84.14571
DATA	136	102.1333	80.01863
TROI	264	100.6667	20.84067
DATA	230	100.3846	52.92975
TROI	233	98.8000	51.46477

character	episode_number	mean_dialog_length	std_dialog_length
<chr>	<dbl>	<dbl>	<dbl>
TROI	187	98.0000	38.68678
⋮	⋮	⋮	⋮
TROI	181	31.00000	NA
WORF	124	30.26087	23.831258
WESLEY	166	30.25000	20.036898
PICARD	175	30.10000	28.321370
WORF	158	30.03448	20.127243
TROI	276	29.75000	15.671099
GEORDI	166	29.60000	16.194649
WORF	228	29.33333	21.667179
WORF	218	29.27273	21.961238
WORF	166	29.16667	22.513969
GEORDI	240	29.00000	2.828427
GEORDI	254	29.00000	NA
WORF	194	28.85714	26.251531
BEVERLY	120	28.80000	19.601020
WESLEY	173	28.66667	10.641898
WORF	275	28.57143	15.177521
WESLEY	122	28.40000	16.546903
WORF	133	27.63158	14.396028
WORF	221	26.33333	32.642230
TROI	219	26.00000	NA
WORF	157	25.33333	14.960265
BEVERLY	276	21.25000	5.909033
WESLEY	132	19.75000	11.461114
WESLEY	178	18.66667	11.201190
WORF	113	17.75000	14.384598
WORF	262	17.75000	7.804913
BEVERLY	273	17.00000	NA

character	episode_number	mean_dialog_length	std_dialog_length
<chr>	<dbl>	<dbl>	<dbl>
WESLEY	104	16.20000	20.644612
TROI	239	14.00000	NA
WORF	112	14.00000	NA



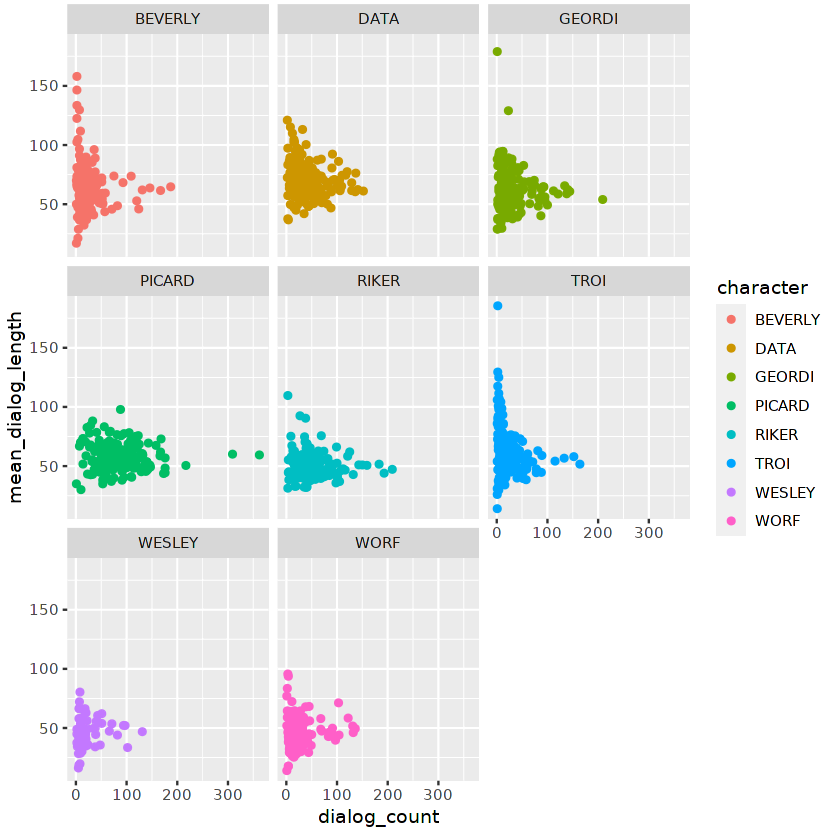

mean_dialog_length
150
100
50
DATA
BEVERLY
GEORDI
TROI
PICARD
RIKER
WORF
WESLEY
factor(character, for_factor$character)


BEVERLY
DATA
GEORDI
PICARD
RIKER
TROI
WESLEY
WORF
mean_dialog_length
150
100
50
0
100
200
300
dialog_count
character
BEVERLY
DATA
GEORDI
PICARD
RIKER
TROI
WESLEY
WORF

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?

In [2]:
```
ggplot(dialog_len_per_ep, aes(dialog_count, mean_dialog_length)) +
    geom_point(aes(color=character)) +
    facet_wrap(~character)
```



Each character's role can be understood by how far their points are from the origin (0,0) along the line y=x. Essentially, the more points (episodes) a character has that are high in either dialog count or length (or both), the more total time they spend talking or having screentime, signalling that they are more important to the plot. Based on these plots, it appears that Data, Riker, and Picard have the greatest average speaking time per episode.

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.

In [3]:
```
summary(dialog_len_per_ep)
```

```
 character         episode_number  mean_dialog_length  dialog_count   
Length:1258        Min.   :102.0   Min.   : 14.00     Min.   :  1.0  
Class :character   1st Qu.:145.0   1st Qu.: 46.72     1st Qu.: 12.0  
Mode  :character   Median :187.0   Median : 55.54     Median : 28.0  
                   Mean   :188.4   Mean   : 58.30     Mean   : 40.6  
                   3rd Qu.:233.0   3rd Qu.: 67.42     3rd Qu.: 55.0  
                   Max.   :277.0   Max.   :185.50     Max.   :362.0  
```

In [4]:
```
library(dplyr)

summary(dialog_len_per_ep)
beverly_diag <- dialog_len_per_ep %>%
    filter(character == "BEVERLY",
        (episode_number >= 102 & episode_number <= 126) |
        (episode_number >= 149 & episode_number <= 174)) %>%

    mutate(season = case_when(
        episode_number >= 102 & episode_number <= 126 ~ "Season 1",
        episode_number >= 149 & episode_number <= 174 ~ "Season 3")) %>%

  group_by(season, episode_number) %>%
  summarize(mean_dialog_length = mean(mean_dialog_length, na.rm = TRUE),
      dialog_count = mean(dialog_count, na.rm = TRUE),
      .groups = "drop") %>%

  group_by(season) %>%
  summarize(mean_dialog_length = mean(mean_dialog_length, na.rm = TRUE),
            dialog_count = mean(dialog_count, na.rm = TRUE),
            .groups = "drop")

beverly_diag
```

```
 character         episode_number  mean_dialog_length  dialog_count   
Length:1258        Min.   :102.0   Min.   : 14.00     Min.   :  1.0  
Class :character   1st Qu.:145.0   1st Qu.: 46.72     1st Qu.: 12.0  
Mode  :character   Median :187.0   Median : 55.54     Median : 28.0  
                   Mean   :188.4   Mean   : 58.30     Mean   : 40.6  
                   3rd Qu.:233.0   3rd Qu.: 67.42     3rd Qu.: 55.0  
                   Max.   :277.0   Max.   :185.50     Max.   :362.0  
```

A tibble: 2 × 3

season	mean_dialog_length	dialog_count
<chr>	<dbl>	<dbl>
Season 1	56.48460	25.40
Season 3	67.04817	19.64

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?

This is reflected in the table by the increase in mean dialog length from Season 1 to Season 3, but a decrease in dialog count. This can potentially be understood as the writers giving her "storylines that made her feel like a male character" by trying to reconstruct her character to have "deeper" or "thought-provoking" lines that sound more intelligent, rather than having more frequent lines that are shorter or lacking in substance.

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:

```
tokens <- YOUR_DATASET %>%
  # Split each dialog into words
  mutate(word_list = str_split(DIALOG_COLUMN, "\\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% STOPWORDS)
```

In [5]:
```
stop_words <- c(
  "i","me","my","myself","we","our","ours","ourselves","you","your","yours","yourse
  "yourselves","he","him","his","himself","she","her","hers","herself","it","its","
```

```
  "they","them","their","theirs","themselves","what","which","who","whom","this","t
  "these","those","am","is","are","was","were","be","been","being","have","has","ha
  "having","do","does","did","doing","a","an","the","and","but","if","or","because"
  "until","while","of","at","by","for","with","about","against","between","into","t
  "during","before","after","above","below","to","from","up","down","in","out","on"
  "over","under","again","further","then","once","here","there","when","where","why
  "all","any","both","each","few","more","most","other","some","such","no","nor","n
  "only","own","same","so","than","too","very","s","t","can","will","just","don","s
)

tokens <- dialogs_fixed %>%
    mutate(word_list = str_split(dialog, "\\s+")) %>%
    unnest(word_list) %>%

    mutate (
        word = str_remove_all(word_list, "[[:punct:]]"),
        word = str_to_lower(word)
    ) %>%

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

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.

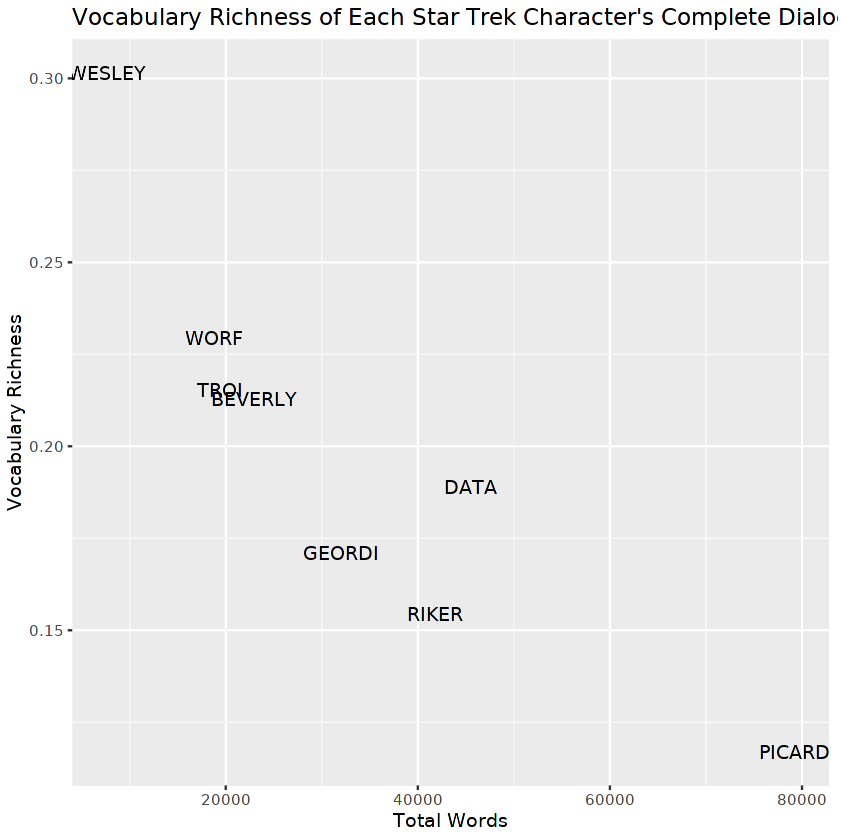
In [6]:
```
unique_counter <- tokens%>%
    group_by(character) %>%
    summarize(
        total_words = n(),
        unique_words = n_distinct(word),
        vocab_richness = unique_words / total_words
      ) %>%
    arrange(desc(vocab_richness))

print(unique_counter)
```

```
# A tibble: 8 × 4
  character total_words unique_words vocab_richness
  <chr>           <int>        <int>          <dbl>
1 WESLEY           7601         2291          0.301
2 WORF            18820         4318          0.229
3 TROI            19450         4187          0.215
4 BEVERLY         22900         4875          0.213
5 DATA            45462         8593          0.189
6 GEORDI          31978         5465          0.171
7 RIKER           41827         6458          0.154
8 PICARD          79214         9272          0.117
```

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.

In [7]:
```
ggplot(unique_counter, aes(x = total_words, y = vocab_richness, label = character))
  geom_text() +
  labs(
    title = "Vocabulary Richness of Each Star Trek Character's Complete Dialog",
    x = "Total Words",
    y = "Vocabulary Richness"
  ) +
  theme()
```



d) Interpret these results.

Among this smaller list of main characters, most of them have a total dialog length of >= 20000 words spoken. They have all have a relatively low vocabulary richness score (<= 0.30) because the sheer volume of their dialog almost necessitates that they reuse and repeat words throughout their time on the show. The more a character speaks (like Picard), the lower their vocabulary richness score likely is, suggesting a negative linear association.

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.

In [8]:
```
wes_leave_range <- dialog_len_per_ep %>%
    filter(character == "WESLEY", episode_number > 174) %>%
    select(episode_number, dialog_count) %>%
    arrange(episode_number)

wes_leave_range
```

A tibble: 11 × 2

episode_number	dialog_count
<dbl>	<int>
175	18
176	6
177	9
178	6
179	38
181	2
183	94
206	131
219	71
263	18
272	97

Wesley leaves the show as a main character after episode 183, where he doesn't appear for 23 consecutive episodes.

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

After leaving the main cast, Wesley makes came 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.

In [9]:
```
wes_last_maincast <- dialogs_fixed %>%
      filter(character == "WESLEY", episode_number == 183) %>%
      slice_tail(n=1)

wes_last_ever <- dialogs_fixed %>%
      filter(character == "WESLEY", episode_number == 272) %>%
      slice_tail(n=1)

wes_last_maincast
wes_last_ever
```

A tibble: 1 × 4

episode_number	character	dialog	dialog_length
<dbl>	<chr>	<chr>	<int>
183	WESLEY	I can walk.	11

A tibble: 1 × 4

episode_number	character	dialog	dialog_length
<dbl>	<chr>	<chr>	<int>
272	WESLEY	Good-bye, Mom.	14

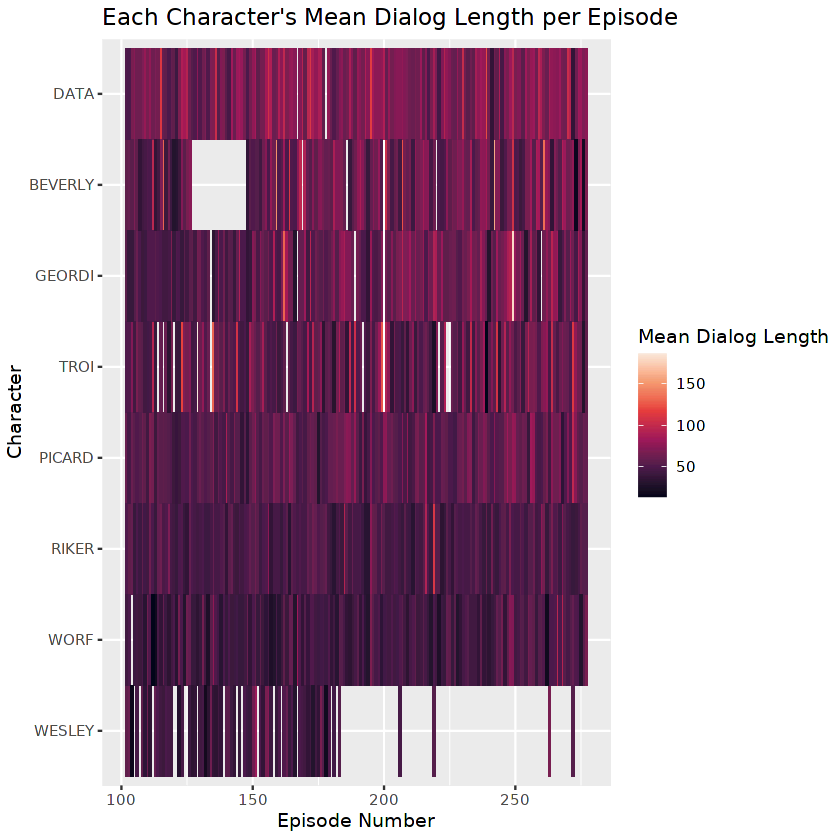
Wesley's last piece of dialogue before leaving the main case is "I can walk". Wesley's last piece of dialogue ever is "Good-bye, Mom."

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!

In [10]:
```
character_dialog_ranked <- dialog_len_per_ep %>%
    group_by(character) %>%
    summarize(overall_mean = mean(mean_dialog_length, na.rm = TRUE), .groups = "dro
    arrange(overall_mean) %>%
    pull(character)

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

ggplot(dialog_heatmap, aes(x=episode_number, y=character, fill=mean_dialog_length))
    geom_tile() +
    scale_fill_viridis_c(option = "rocket") +
    labs(
      title = "Each Character's Mean Dialog Length per Episode",
      x = "Episode Number",
      y = "Character",
      fill = "Mean Dialog Length"
    ) +
    theme()
```



In [ ]: