Webtoon Analysis

Bryce Wong
May 14, 2019

Exploratory analysis of the Webtoon Comment data

First reading in the data (updated as of May 10, 2019 - this was run BEFORE episode #55 had been posted):

```
webtoons_data = read_csv(file = "./data/comments_may_10.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   episode_num = col_character(),
##   episode = col_character(),
##   comment_txt = col_character(),
##   username = col_character(),
##   likes = col_integer(),
##   reply = col_logical(),
##   likes_per_ep = col_integer()
## )
```

```
webtoons_data = webtoons_data %>%
  filter(username != "TESTED @YGetIt on IG") %>%
  select(-X1) %>%
  mutate(
    episode_num = str_replace(episode_num, "#", ""),
    episode_num = as.numeric(episode_num),
    season = ifelse(episode_num %in% 1:12, "1",
                    ifelse(episode_num %in% 13:24, "2",
                           ifelse(episode_num %in% 25:37, "3", "4")))
  )

number_of_eps = webtoons_data %>%
  distinct(episode, .keep_all = TRUE)
```

The total number of comments is 543.

The total number of episodes is 54.

Now getting the number of comments per each episode:

- Outputting table of top 10 episodes by number of comments

```
#number of comments per each episode
num_eps = webtoons_data %>%
  count(episode) %>%
  arrange(desc(n))

#outputting table of top 10 episodes by number of comments
num_eps %>%
  top_n(10) %>%
  rename(number_of_comments = n) %>%
  knitr::kable(digits = 3)


## Selecting by n
```

episode	number_of_comments
WORLD AIDS DAY!!!	26
Heck of a Start	24
Brunchy Brunch	20
Sometimes People SUCK!!!	18
FIGHT!!!!	17
HAPPY NEW YEAR!!!!!	17
Doctor Visit	16
Prayers	16
This Could Be Bad	16
Further South	15

Now getting the number of likes per each comment:

- Outputting table of top 10 comments by number of likes (couldn't produce a pretty table)

```
#arranging comments by likes
arrange_by_likes = webtoons_data %>%
  arrange(desc(likes))

#outputting table of top 10 comments by number of likes
head(arrange(webtoons_data, desc(likes)), 10)
```

```
## # A tibble: 10 x 8
##    episode_num episode comment_txt userna~ likes reply likes_per_ep season
##          <dbl> <chr>   <chr>       <chr>   <int> <lgl>        <int> <chr>
##  1           1 Heck o~ i love ham~ sub<U+~   125 FALSE          540 1
##  2           1 Heck o~ omg is tha~ saphir~    93 FALSE          540 1
##  3           1 Heck o~ Hamilton :~ swirli~    82 FALSE          540 1
##  4          40 Soluti~ I'm glad s~ frowsy     65 FALSE          218 4
##  5          23 This C~ SHE HAD ON~ GrimmZ~    64 FALSE          246 2
##  6           6 Brunch~ wait what.~ happyc~    56 FALSE          308 1
##  7          30 You Ju~ Clearly ta~ coyowo~    56 FALSE          248 3
##  8          32 WORLD ~ honestly t~ just y~    50 FALSE          289 3
##  9           6 Brunch~ There will~ gillea~    49 FALSE          308 1
## 10          50 Tragedy "This one ~ pompou~    49 FALSE          149 4
```

Now getting the number of comments per each unique user:

- Outputting table of top 10 users by number of comments

```
#number of comments each unique user has posted
num_users = webtoons_data %>%
  count(username) %>%
  arrange(desc(n))

#outputting table of top 10 users by number of comments
#cannot output as a nice table, possibly because a user has UTF8 characters in their
num_users %>%
  top_n(10) %>%
  rename(number_of_comments = n)
```

```
## Selecting by n
```

```
## # A tibble: 10 x 2
##    username                  number_of_comments
##    <chr>                                  <int>
##  1 gilleanfryingpan                          45
##  2 sausage172000                             22
##  3 AoiYeyi                                   21
```

```
##  4 happycat(:                                18
##  5 19danny15                                 16
##  6 Cheapthrill_Xo                            16
##  7 CopperMortar                              14
##  8 "\xb0\x95Mariella\x95\xb0"                12
##  9 catberra                                  12
## 10 RedtheGreyFox                             12
```

- Outputting table of top 10 episodes by number of episode likes

```
#stats of likes per episode (likes of episode - NOT comments)
ep_likes = webtoons_data %>%
  distinct(episode, .keep_all = TRUE)

#removing other columns
ep_likes = ep_likes %>%
  select(episode, likes_per_ep)

#outputting table of top 10 comments by number of likes
head(arrange(ep_likes, desc(likes_per_ep)), 10) %>%
  knitr::kable(digits = 3)
```

episode	likes_per_ep
Heck of a Start	540
Uh oh	435
Flash Back	370
Doctor Visit	338
Work It Out	335
Brunchy Brunch	308
WORLD AIDS DAY!!!	289
Rough Start	279
It Goes Down in the Bathroom	267
Ape S#$%	262

Now a bunch of tables showing basic summary statistics for:

- comments across all episodes

- comments across all users
- likes across all comments

Also, one histogram at the end to show the distribution of likes.

(The histogram of the distribution of number of comments per episode was a bit funky and probably not worth viewing)

```
#stats of comments across all episodes
avg_num_comm = num_eps %>%
  summarize(mean_comments_per_ep = mean(n),
            median_comments_per_ep = median(n),
            sd_comments = sd(n)) %>%
  knitr::kable(digits = 3)

avg_num_comm
```

mean_comments_per_ep	median_comments_per_ep	sd_comments
10.056	9	5.272

```
#stats of commentators
avg_user = num_users %>%
  summarize(mean_comments_per_user = mean(n),
            median_comments_per_user = median(n),
            sd_comments = sd(n)) %>%
  knitr::kable(digits = 3)

avg_user
```

mean_comments_per_user	median_comments_per_user	sd_comments
2.828	1	4.705

```
#stats of likes
avg_likes = webtoons_data %>%
  summarize(mean_likes_per_comment = mean(likes),
            median_likes_per_comment = median(likes),
            sd_likes = sd(likes)) %>%
  knitr::kable(digits = 3)

avg_likes
```

mean_likes_per_comment	median_likes_per_comment	sd_likes

mean_likes_per_comment	median_likes_per_comment	sd_likes
7.576	4	11.791

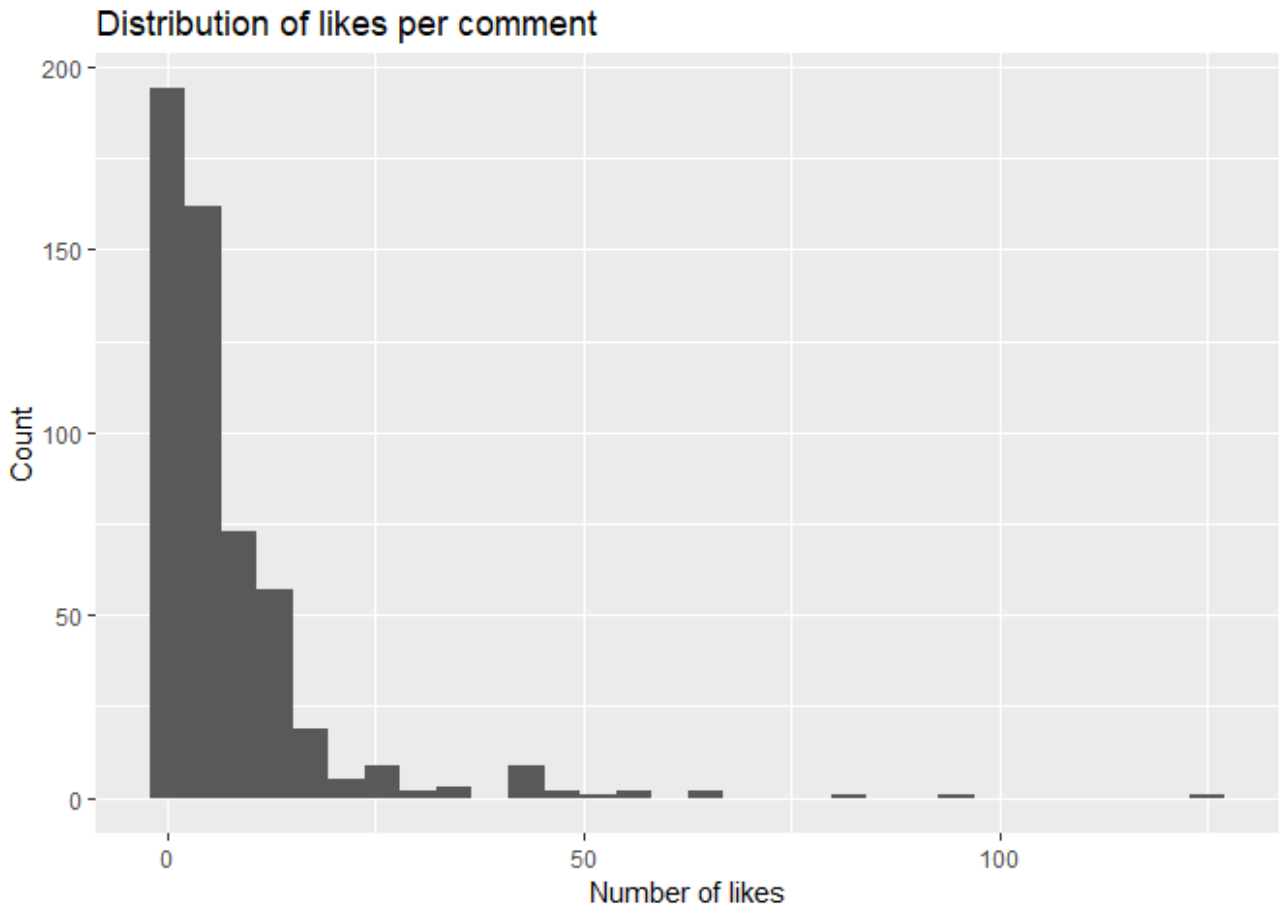
```
#stats of total comment likes
avg_total_likes = webtoons_data %>%
  group_by(episode) %>%
  summarise(likes = sum(likes)) %>%
  summarize(mean_total_likes = mean(likes),
            median_total_likes = median(likes),
            sd_total_likes = sd(likes)) %>%
  knitr::kable(digits = 3)

avg_total_likes
```

mean_total_likes	median_total_likes	sd_total_likes
76.185	58.5	66.315

```
#stats of likes per episode (likes of episode - NOT comments)
avg_likes_per_ep = webtoons_data %>%
  distinct(episode, .keep_all = TRUE) %>%
  summarize(mean_likes_per_ep = mean(likes_per_ep),
            median_likes_per_ep = median(likes_per_ep),
            sd_likes = sd(likes_per_ep)) %>%
  knitr::kable(digits = 3)

avg_likes_per_ep
```

mean_likes_per_ep	median_likes_per_ep	sd_likes
235.611	243.5	75.424

```
#visualizations

#distribution of likes
ggplot(webtoons_data, aes(x = likes)) +
  geom_histogram() +
  labs(
    title = "Distribution of likes per comment",
    x = "Number of likes",
    y = "Count"
  )


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



Descriptive Statistics By Season

Creating a variable that organizes the episodes by season - note that this code will not be extendable to organizing future episodes by season (there is no "season" identification marker built into the way the episodes were uploaded to Webtoons).

The assignment of an episode to a specific season had to be done by hand, by taking a look at the titles and figuring out which season they belong to.

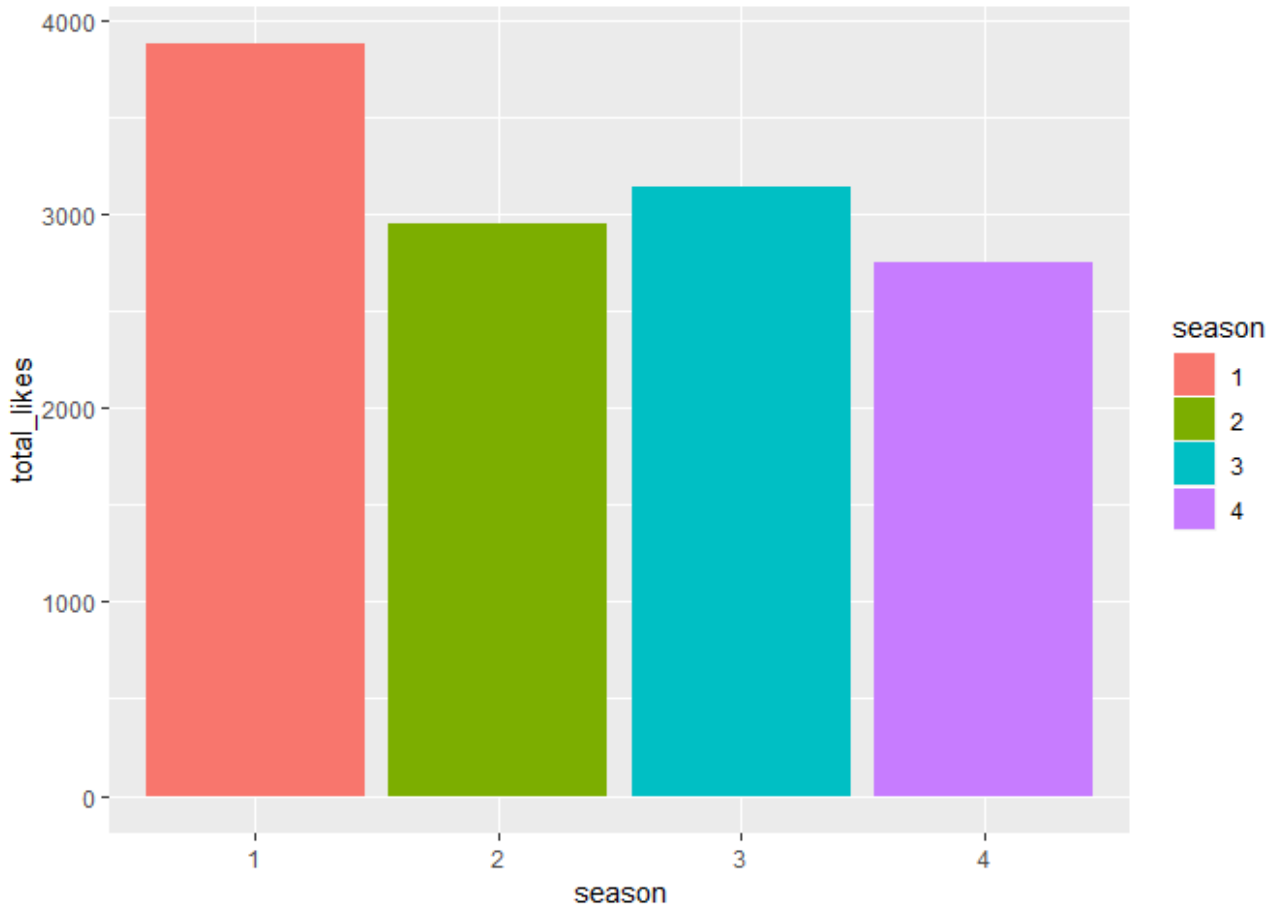
Note that not all seasons have the same number of episodes (some seasons have filler episodes) - thus, the average number of likes and comments per an episode in each season have also been calculated.

```
seasons = webtoons_data %>%
  select(season, episode_num, episode, comment_txt, likes_per_ep)

#total episode likes per season
likes_per_season = seasons %>%
  distinct(episode_num, .keep_all = TRUE) %>%
```

```
  group_by(season) %>%
  summarize(total_likes = sum(likes_per_ep))

likes_per_season %>%
  knitr::kable(digits = 3)
```

season	total_likes
1	3878
2	2955
3	3143
4	2747

```
#visualization
ggplot(likes_per_season, aes(x = season, y = total_likes, fill = season)) + geom_bar(
```

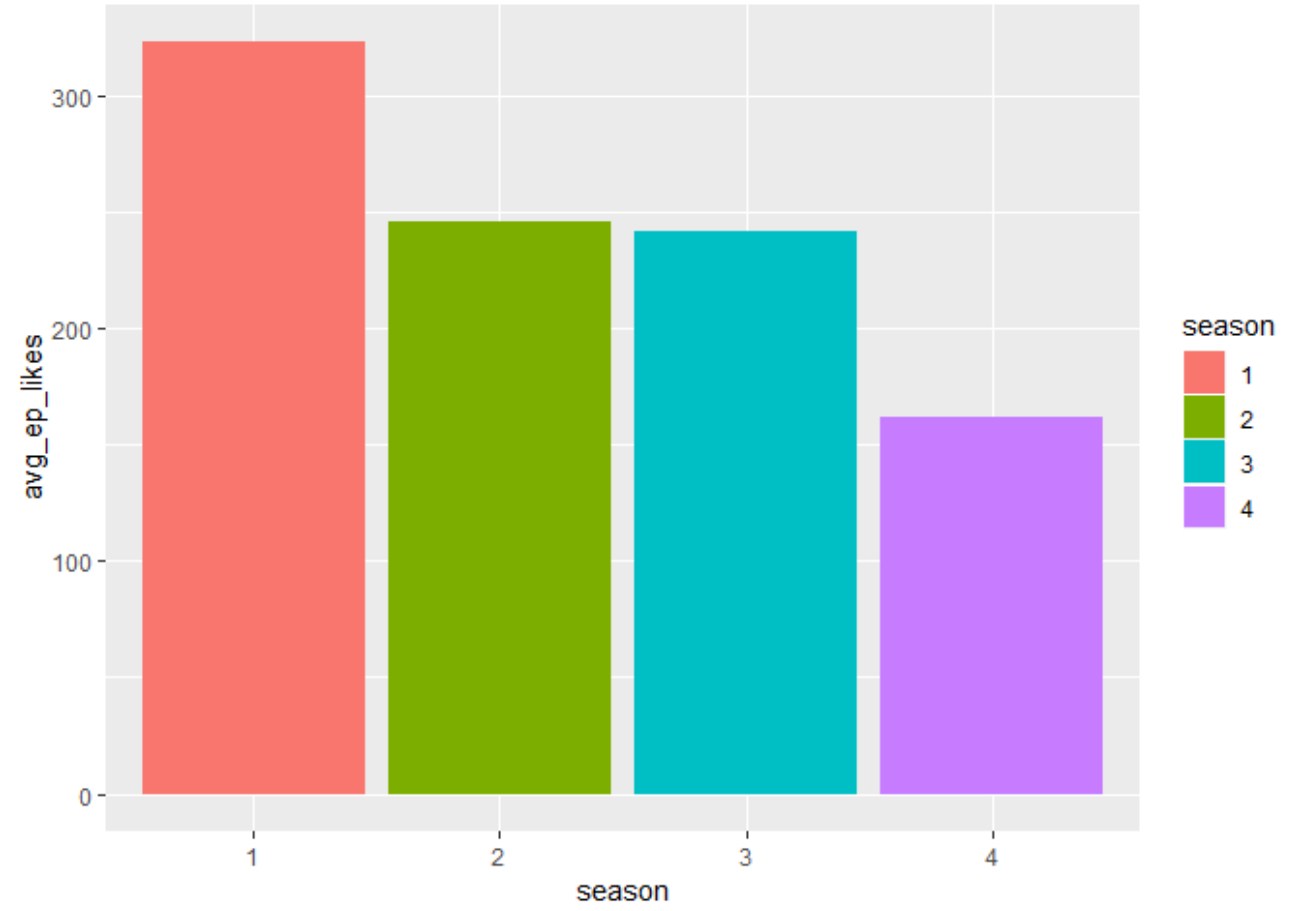



```
#avg likes per episode by season
avg_ep_likes_season = seasons %>%
```

```
  distinct(episode_num, .keep_all = TRUE) %>%
  group_by(season) %>%
  summarize(avg_ep_likes = mean(likes_per_ep))

avg_ep_likes_season %>%
  knitr::kable(digits = 3)
```

season	avg_ep_likes
1	323.167
2	246.250
3	241.769
4	161.588

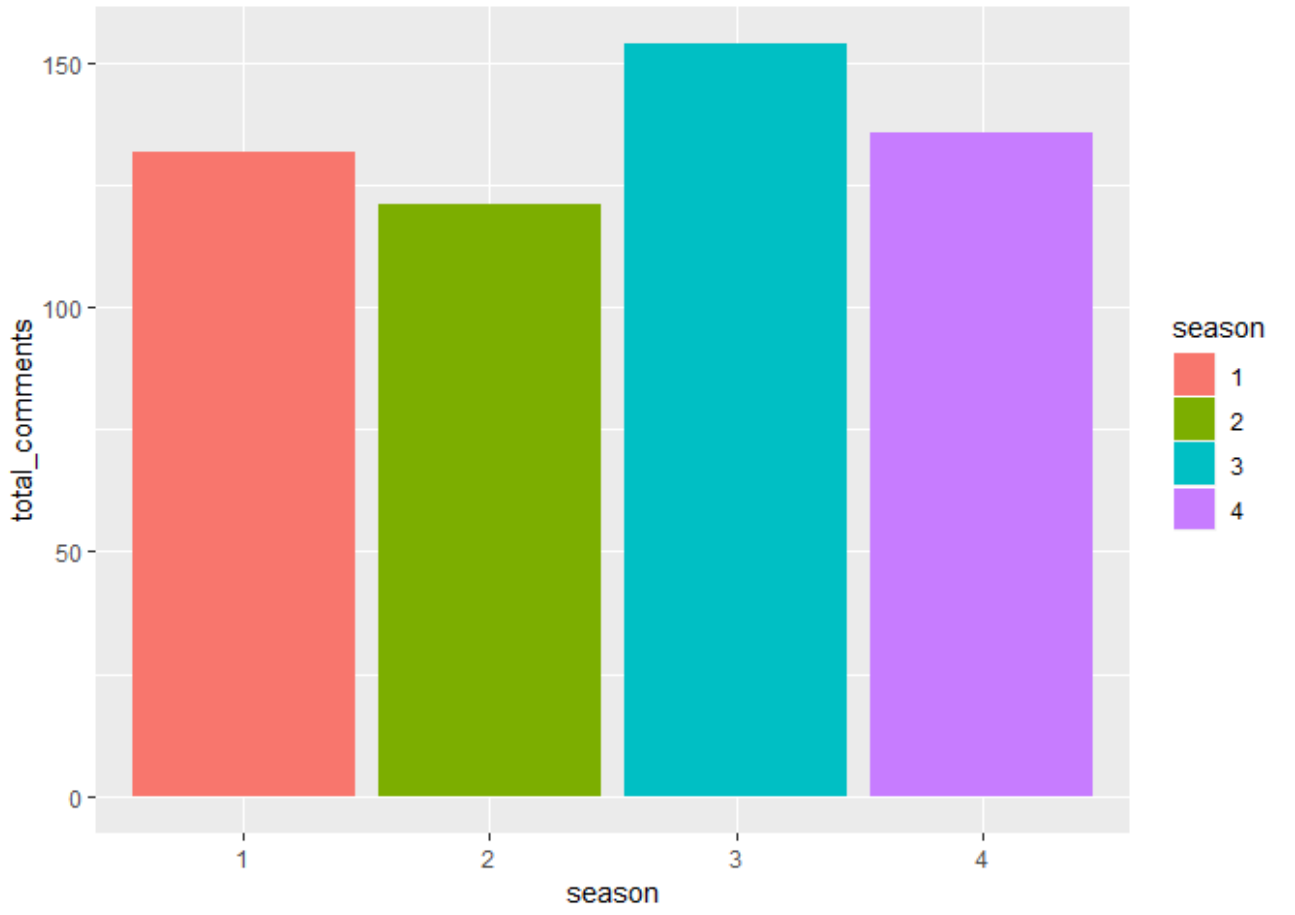
```
#visualization
ggplot(avg_ep_likes_season, aes(x = season, y = avg_ep_likes, fill = season)) + geom_
```



```r
#total comments per season
comments_per_season = seasons %>%
  count(season) %>%
  rename(total_comments = n)

comments_per_season %>%
  knitr::kable(digits = 3)
```

season	total_comments
1	132
2	121
3	154
4	136

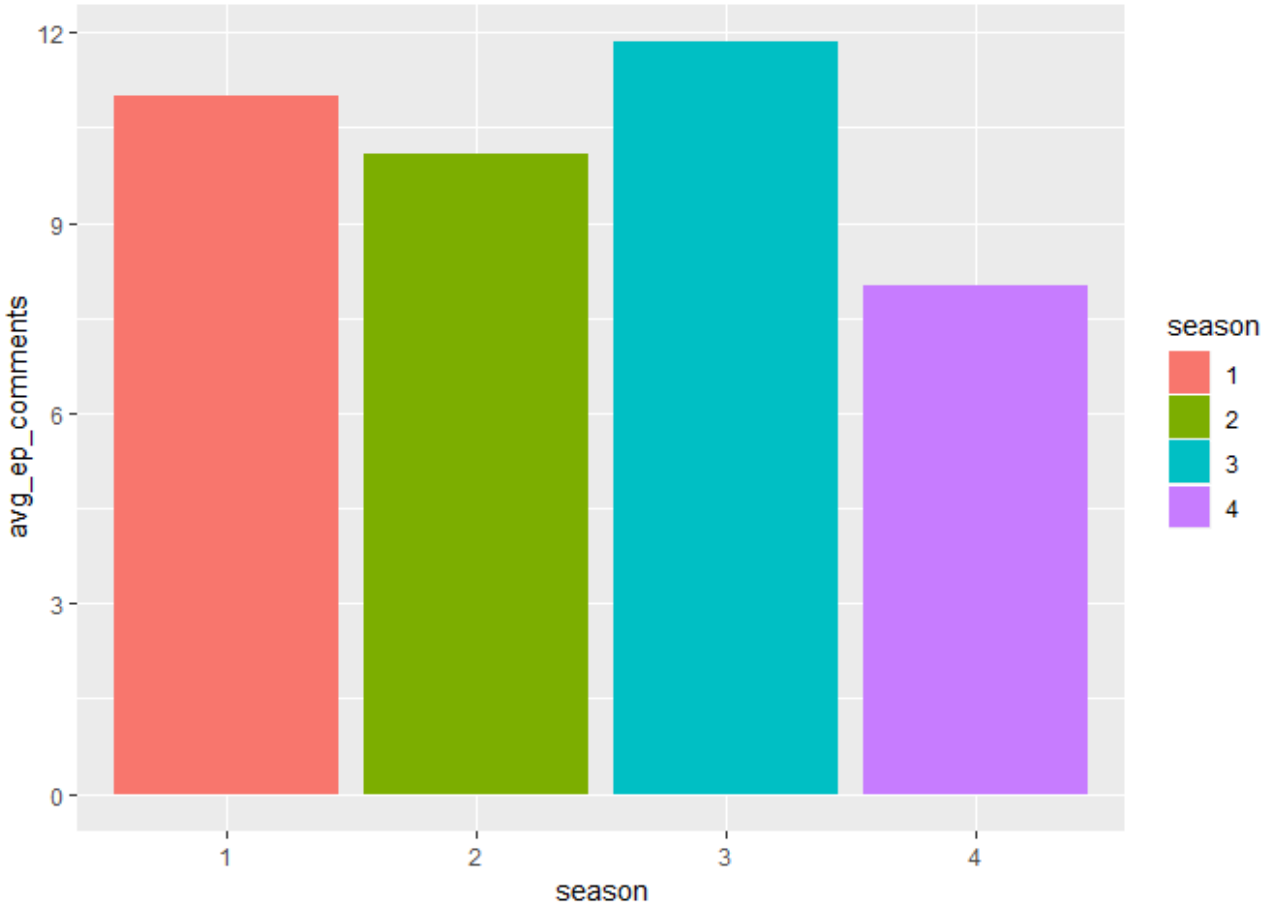
```r
#visualization
ggplot(comments_per_season, aes(x = season, y = total_comments, fill = season)) + geo
```



```r
#avg comments per episode by season
avg_ep_comments_season = seasons %>%
  count(season, episode) %>%
  group_by(season) %>%
  summarize(avg_ep_comments = mean(n))

avg_ep_comments_season %>%
  knitr::kable(digits = 3)
```

season	avg_ep_comments
1	11.000
2	10.083
3	11.846
4	8.000

```r
#visualization
ggplot(avg_ep_comments_season, aes(x = season, y = avg_ep_comments, fill = season)) +
```



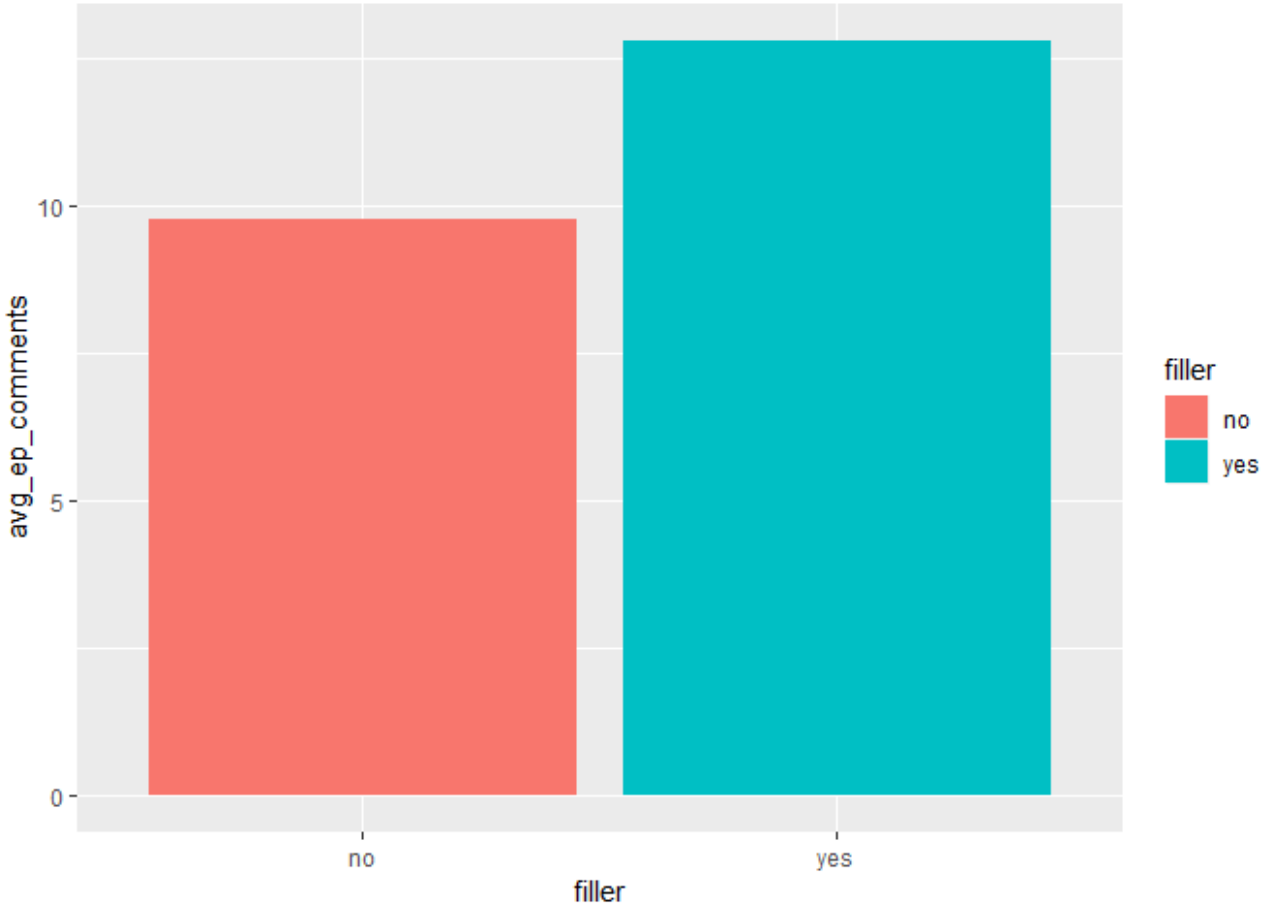
Comparing filler episodes to regular episodes

```
#assigning "filler" label to filler episodes
filler_data = seasons %>%
  mutate(
    filler = ifelse(episode_num %in% 36:38, "yes",
                    ifelse(episode_num == 32, "yes",
                           ifelse(episode_num == 54, "yes", "no")))
  )

#average comments for filler episodes versus for non-filler episodes
avg_ep_comments_filler = filler_data %>%
  count(season, episode_num, filler) %>%
  group_by(filler) %>%
  summarize(avg_ep_comments = mean(n))

avg_ep_comments_filler %>%
  knitr::kable(digits = 3)
```

filler	avg_ep_comments
no	9.776

filler	avg_ep_comments
yes	12.800

```
#visualization
ggplot(avg_ep_comments_filler, aes(x = filler, y = avg_ep_comments, fill = filler)) +
```

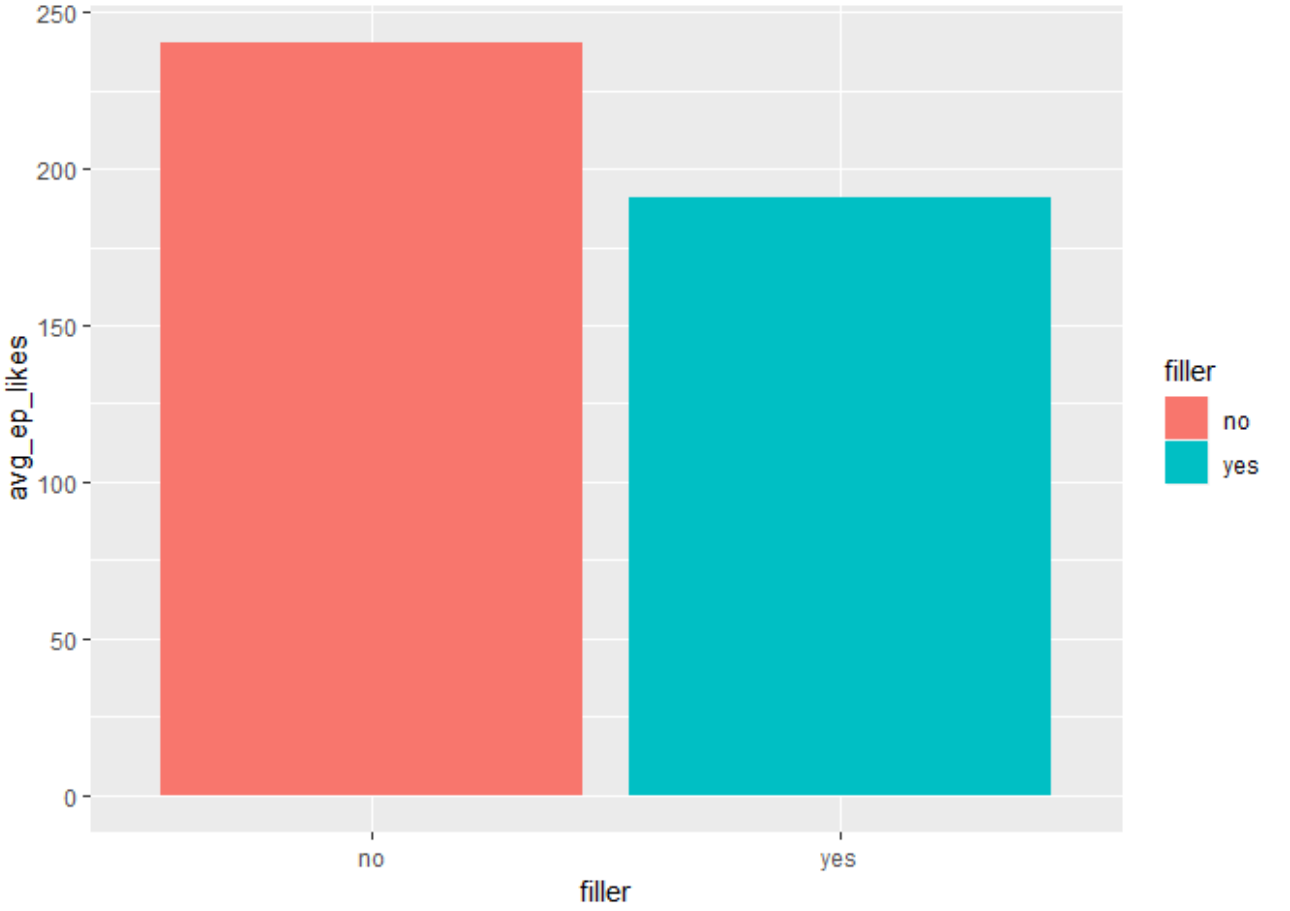



```
#average likes for filler episodes versus for non-filler episodes
avg_ep_likes_filler = filler_data %>%
  distinct(episode_num, .keep_all = TRUE) %>%
  group_by(filler) %>%
  summarize(avg_ep_likes = mean(likes_per_ep))

avg_ep_likes_filler %>%
  knitr::kable(digits = 3)
```

filler	avg_ep_likes
no	240.204

filler	avg_ep_likes
yes	190.600

```
#visualization
ggplot(avg_ep_likes_filler, aes(x = filler, y = avg_ep_likes, fill = filler)) + geom_
```

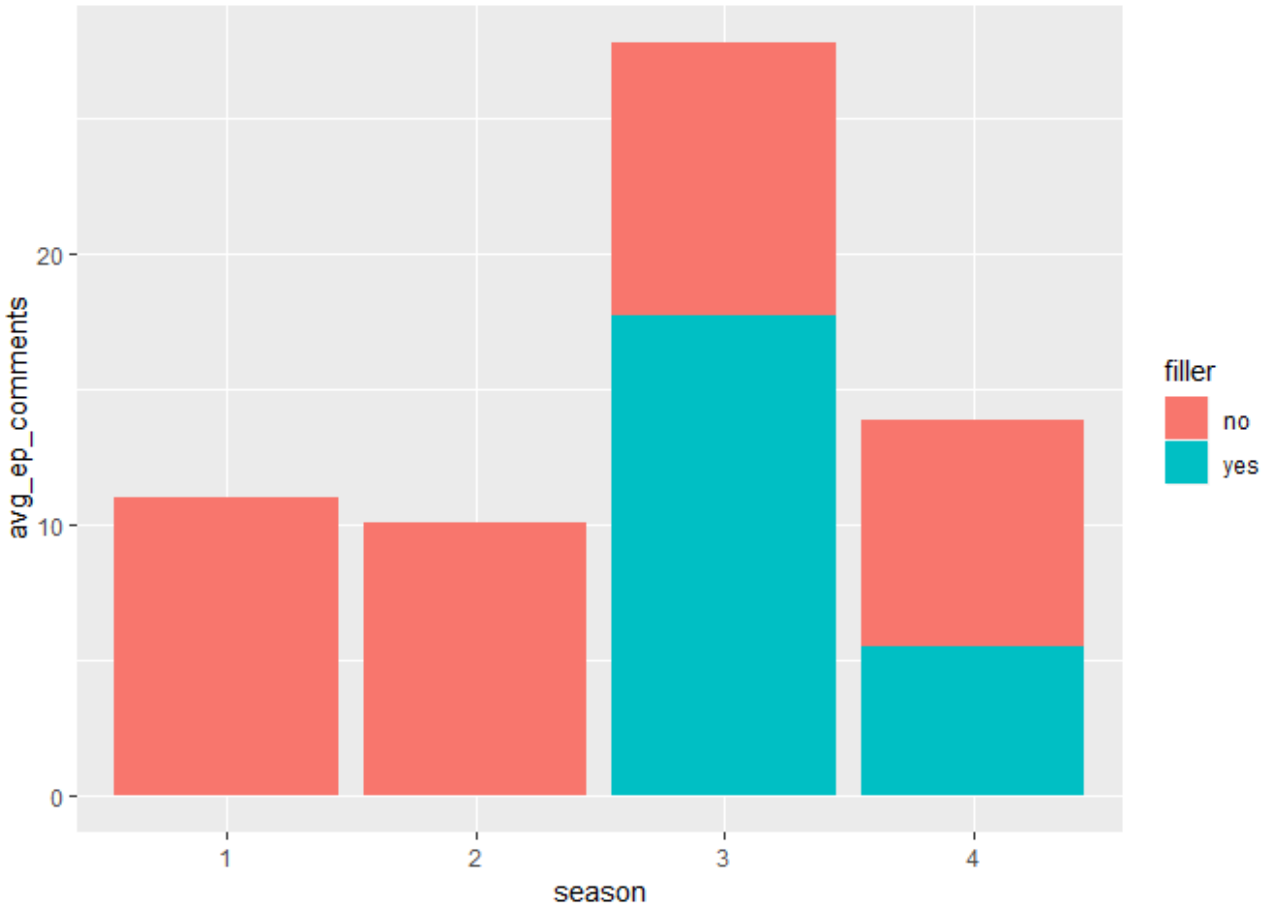



```
#average comments by season, filler vs non-filler
avg_comments_filler_season = filler_data %>%
  count(season, episode_num, filler) %>%
  group_by(season, filler) %>%
  summarize(avg_ep_comments = mean(n))

avg_comments_filler_season %>%
  knitr::kable(digits = 3)
```

season	filler	avg_ep_comments
1	no	11.000

season	filler	avg_ep_comments
2	no	10.083
3	no	10.100
3	yes	17.667
4	no	8.333
4	yes	5.500

```
#visualization
ggplot(avg_comments_filler_season, aes(x = season, y = avg_ep_comments, fill = filler
```

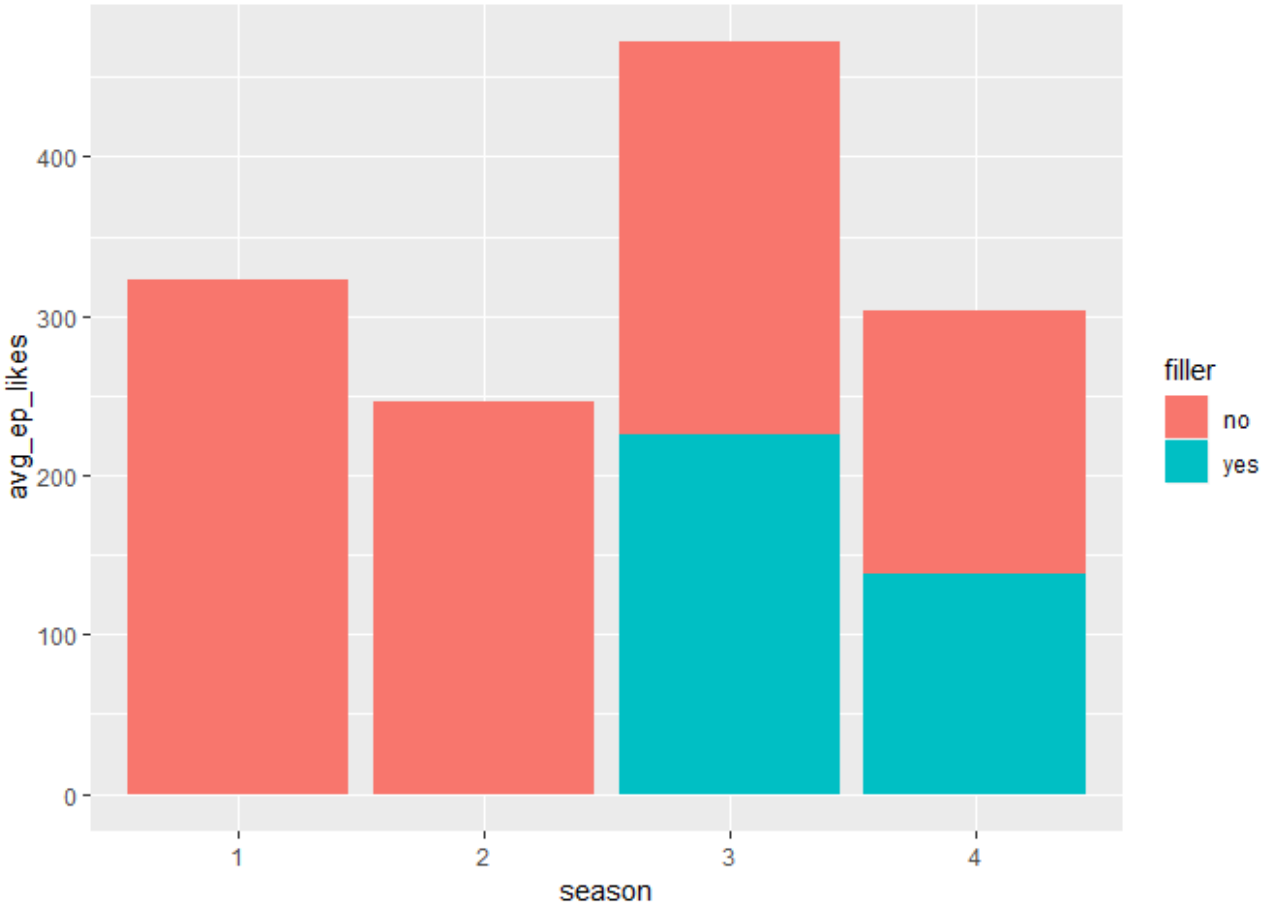



```
#average likes by season, filler vs non-filler
avg_likes_filler_season = filler_data %>%
  distinct(episode_num, .keep_all = TRUE) %>%
  group_by(season, filler) %>%
  summarize(avg_ep_likes = mean(likes_per_ep))
```

```
avg_likes_filler_season %>%
  knitr::kable(digits = 3)
```

season	filler	avg_ep_likes
1	no	323.167
2	no	246.250
3	no	246.700
3	yes	225.333
4	no	164.667
4	yes	138.500

```
#visualization
ggplot(avg_likes_filler_season, aes(x = season, y = avg_ep_likes, fill = filler)) + g
```



Sentiment analysis

Note: a lot of the code here is adapted from Jeff Goldsmith's TidyText lecture.

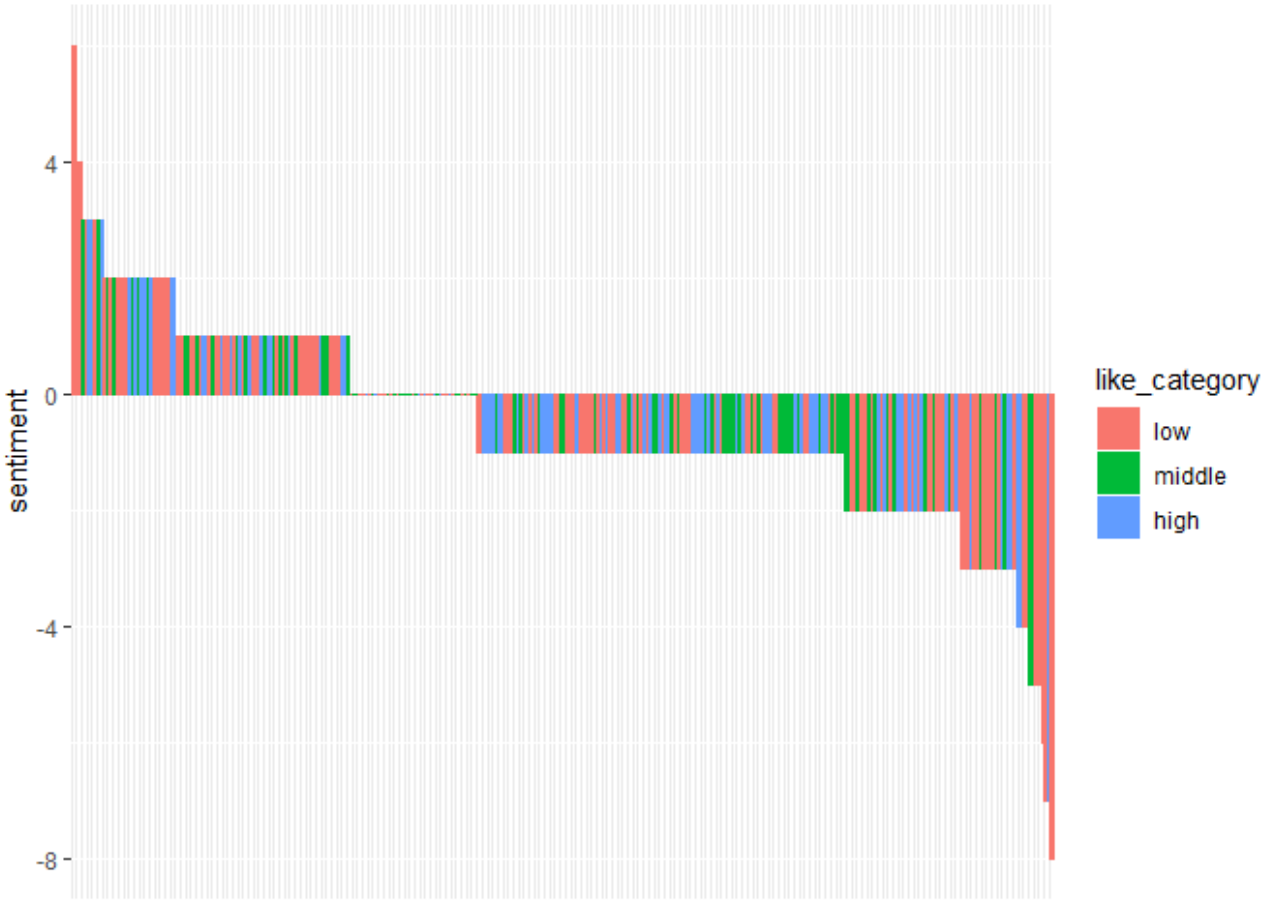
```
webtoon_comments =
  webtoons_data %>%
  mutate(comment_num = row_number(),
         like_category = cut(likes, breaks = c(-Inf, 4, 10, Inf),
                             labels = c("low","middle","high"))) %>%
  as_tibble()

data(stop_words)

comment_words =
  webtoon_comments %>%
  unnest_tokens(word, comment_txt) %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
comment_word_sentiments <- comment_words %>%
  inner_join(get_sentiments("bing")) %>%
  count(comment_num, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative) %>%
  left_join(webtoon_comments)
```

```
## Joining, by = "word"

## Joining, by = "comment_num"
```

```
ggplot(comment_word_sentiments,
       aes(x = reorder(comment_num, -sentiment),
           y = sentiment, fill = like_category, color = like_category)) +
  geom_bar(stat = "identity") +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
```



Most positive comment:

```
comment_word_sentiments %>%
  filter(sentiment == max(sentiment)) %>%
  pull(comment_txt)
```

```
## [1] "Thank you so much notgaybutnot straight thank you for listening so wish I can
```

Most negative comment:

```
comment_word_sentiments %>%
  filter(sentiment == min(sentiment)) %>%
  pull(comment_txt)
```

```
## [1] "When I say all of us I mean yes I was an addict and still am I have an addict
```

Interestingly, cannot find the text for the comment with the lowest/highest sentiment in a specific like_category - something to look into in the future.

Exporting a text file of the comments (commented out for now):

```
#just_comments = webtoons_data %>%
  #filter(username != "TESTED @YGetIt on IG") %>%
  #select(comment_txt)

#write.table(just_comments, file = "just_comments.txt", sep = ",", quote = TRUE, row.
```