Webtoon Analysis

Bryce Wong
March 4, 2019

Exploratory analysis of the Webtoon Comment data

First reading in the data:

```
webtoons_data = read_csv(file = "./data/comments_april_24.csv")
```

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

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   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)

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

The total number of comments is 534.

The total number of episodes is 52.

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	25
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

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

episode	comment_txt	username	likes	reply	likes_per_ep
Heck of a Start	i love hamilton reference!	subpewds	124	FALSE	537
Heck of a Start	omg is that a-aron burr	saphirefan666	93	FALSE	537
Heck of a Start	Hamilton :^))	swirlixpuff	82	FALSE	537
This Could Be Bad	SHE HAD ONE DAMN JOB	GrimmZin	60	FALSE	246
Solution or More Problems?	I'm glad she was able to accept his help even if she can't forgive him. Please don't screw it up, dude. This is your last chance.	frowsy	59	FALSE	215
You Just Gonna Put My Business Out There?	Clearly tact and regard for patience privacy aren't a concern for this nurse. Fire her.	coyowolf TMT	55	FALSE	247
Brunchy Brunch	wait what..... what kinda crazy person just goes: HEY LET'S RAISE OUR FRIEND'S BROTHER!! WHEEE!!	happycat(:	54	FALSE	308

episode	comment_txt	username	likes	reply	likes_per_ep
WORLD AIDS DAY!!!	honestly this is my new favorite comic it talks about real stuff in the world and i love it.	just your avrageweeb	48	FALSE	288
Brunchy Brunch	There will be no hood-rat code-switching to improper English at brunch, young lady. Sounds like my dad.	gilleanfryingpan	47	FALSE	308
WORLD AIDS DAY!!!	Thanks to Featured, I discovered this comic and I love it! Keep up the amazing work, author, and keep on being realistic with the topics!	Gabby Gonzalez	42	FALSE	288

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: 11 x 2
##    username                    number_of_comments
##    <chr>                                    <int>
##  1 gilleanfryingpan                            45
##  2 AoiYeyi                                     21
##  3 sausage172000                               21
##  4 happycat(:                                  18
##  5 19danny15                                   16
##  6 Cheapthrill_Xo                              15
##  7 CopperMortar                                14
##  8 catberra                                    12
##  9 RedtheGreyFox                               12
## 10 "\xb0\x95Mariella\x95\xb0"                  11
## 11 neftana23                                   11
```

- 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	537
Uh oh	434
Flash Back	369
Doctor Visit	338
Work It Out	335

episode	likes_per_ep
Brunchy Brunch	308
WORLD AIDS DAY!!!	288
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
10.269	9.5	5.318

```
#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
2.825	1	4.69

```
#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
7.285	4	11.418

```
#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
```

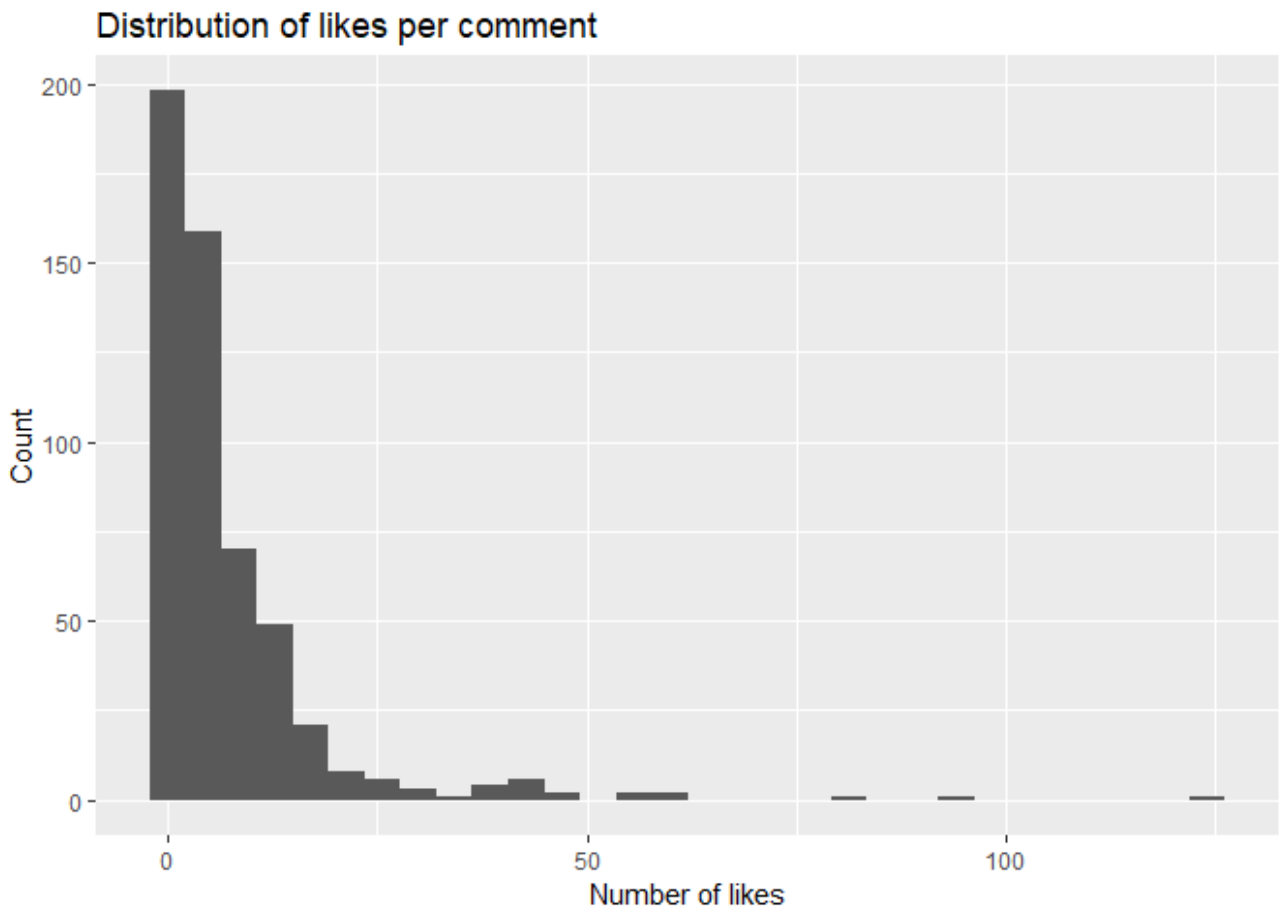
mean_total_likes	median_total_likes	sd_total_likes
74.808	58.5	65.581

```
#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
237.904	244	74.626

```
#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"
  )
```

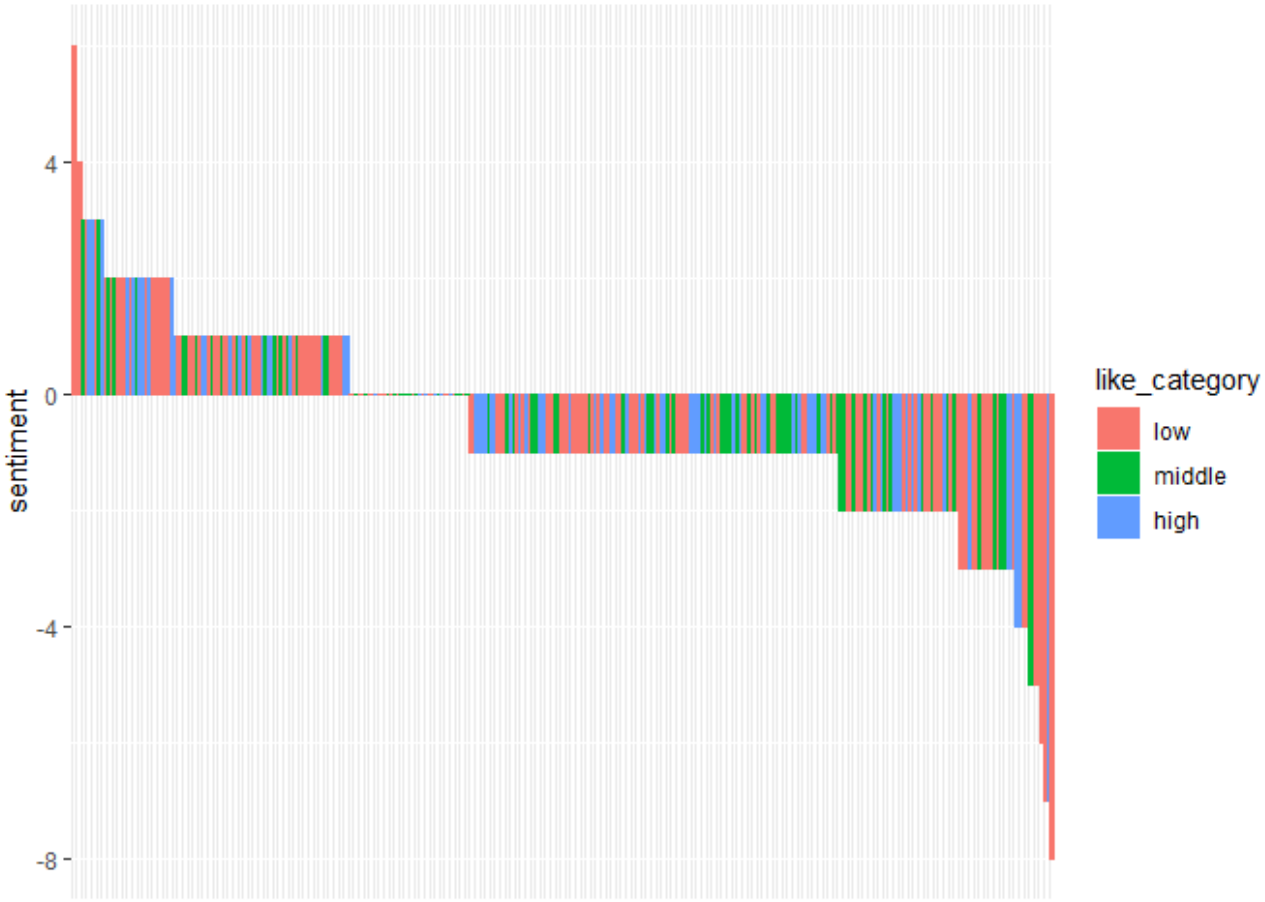
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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:

```
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.n
```