Lab 1

Student Name

YYYY-MM-DD

Data

We'll work with the #tidytuesday data for 2019, specifically the #rstats dataset, containing nearly 500,000 tweets over a little more than a decade using that hashtag.

The data is in under Dataset tab of Week 3 module on Canvas.

You can import the dataset using the code below.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
```

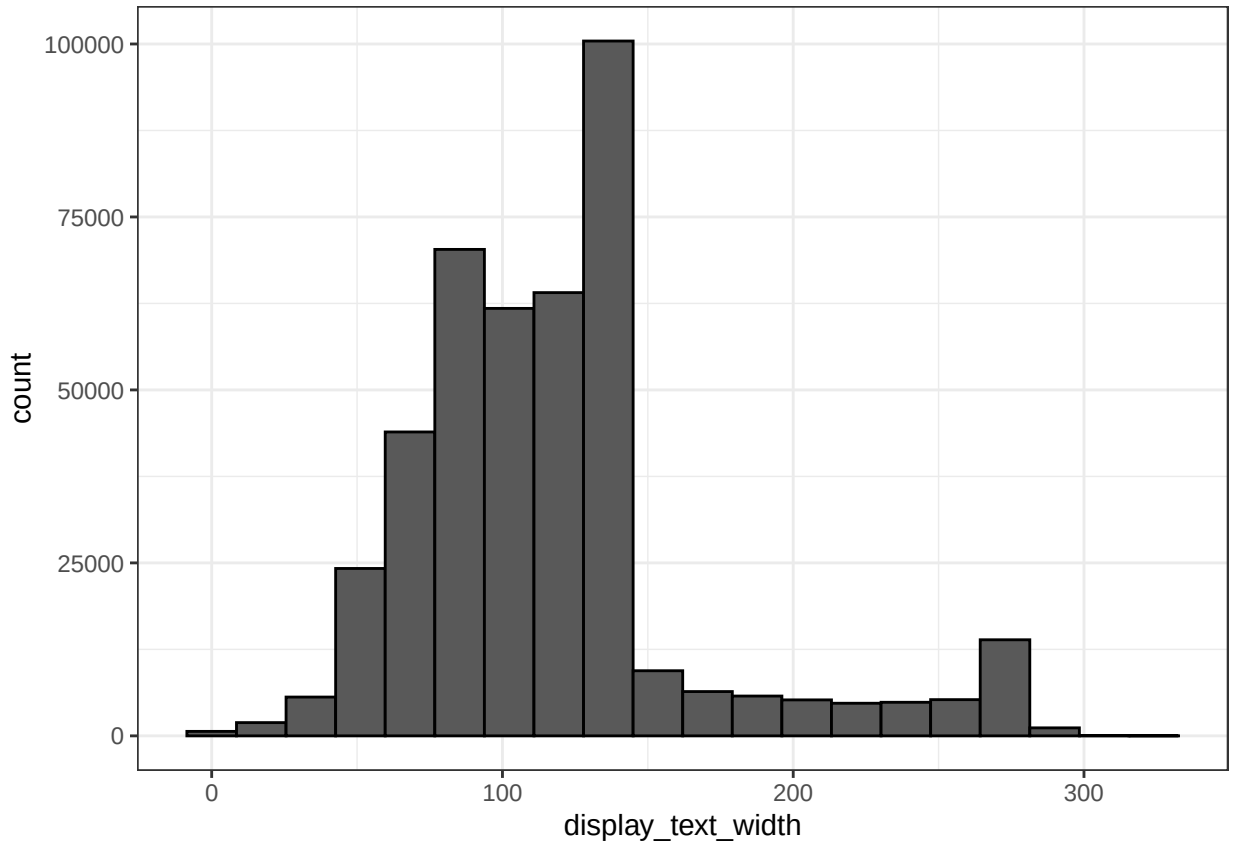
If you need help with processing text data, please revisit the notebook introduced in Week 1.

https://www.kaggle.com/code/uocoeeds/introduction-to-textual-data

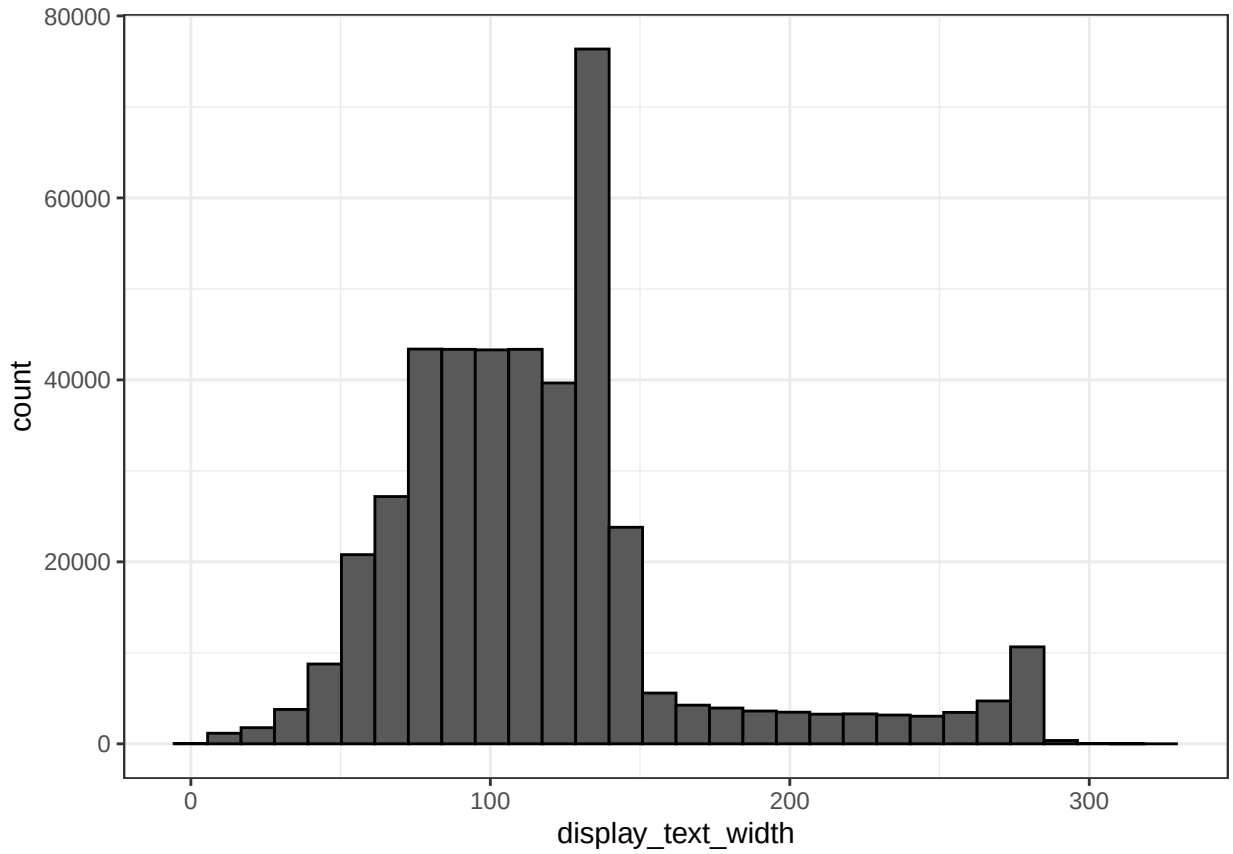
Histogram and Density plots

1. Create a histogram the column `display_text_width` using the `ggplot2` package and `geom_histogram()` function. Try at least four different numbers of bins (e.g., 20, 30, 40, 50) by manipulating the `bins=` argument. Select what you think best represents the data for each. Provide a brief justification for your decision. For all plots you created, change the default background color from grayish to white.

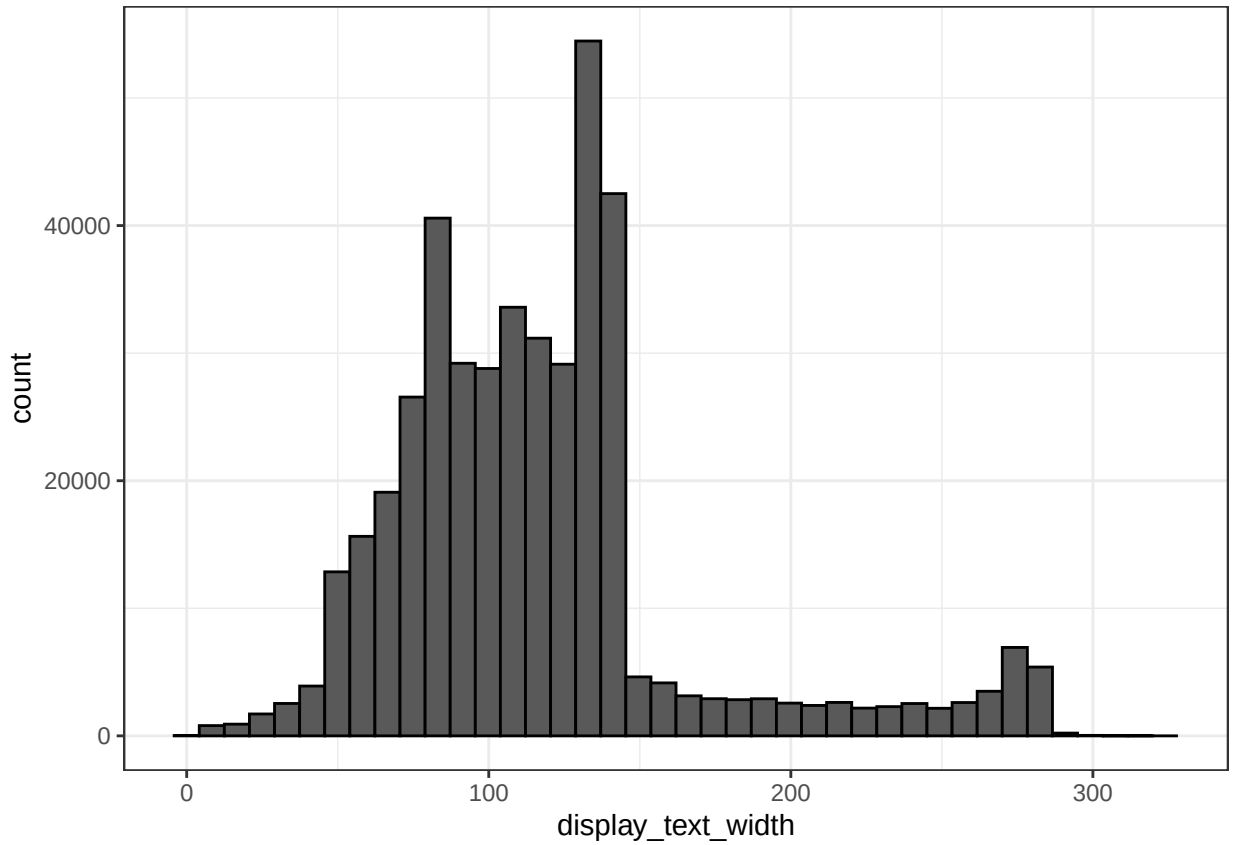
```
d %>%
  ggplot(aes(x = display_text_width))+ geom_histogram(bins = 20, color = "black") + theme_bw()
```



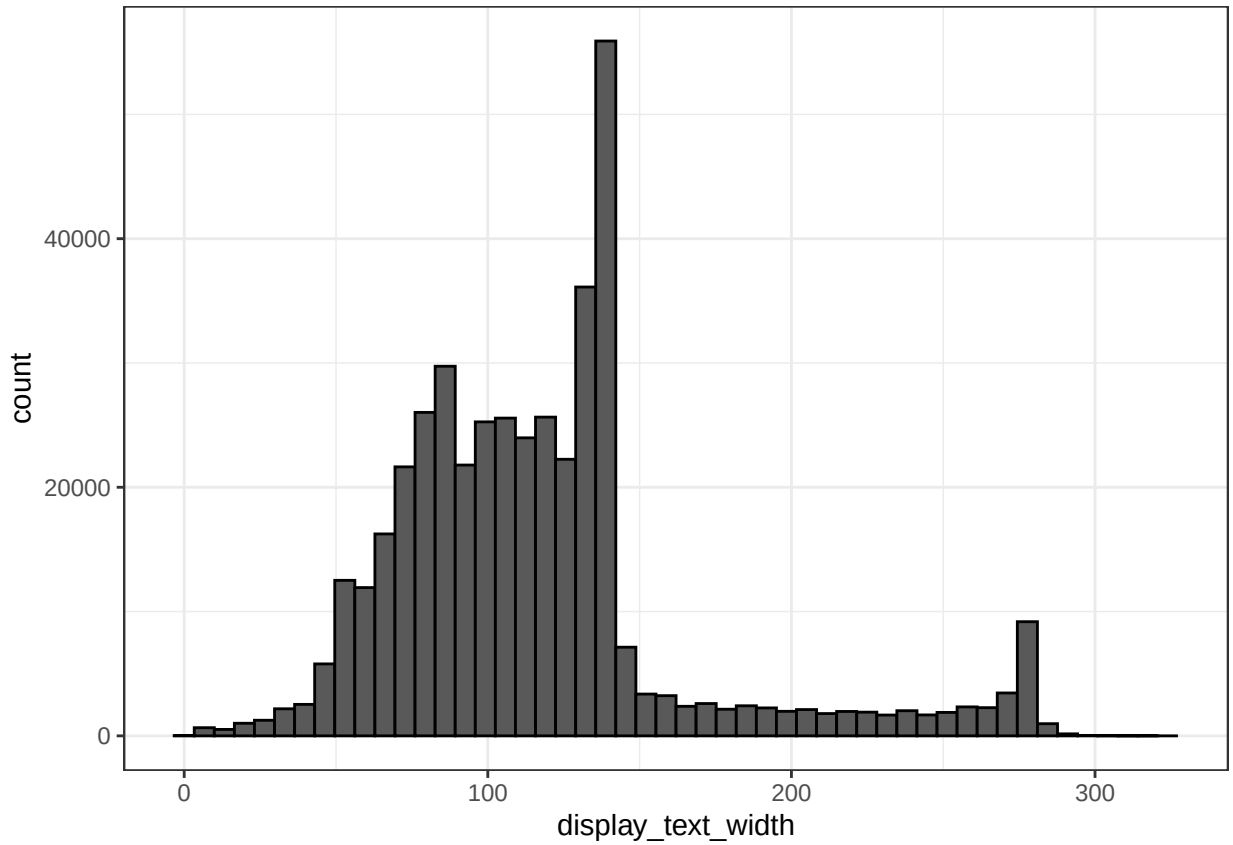

```
d %>%
  ggplot(aes(x = display_text_width))+ geom_histogram(bins = 30, color = "black") + theme_bw()
```




```
d %>%
  ggplot(aes(x = display_text_width))+ geom_histogram(bins = 40, color = "black") + theme_bw()
```

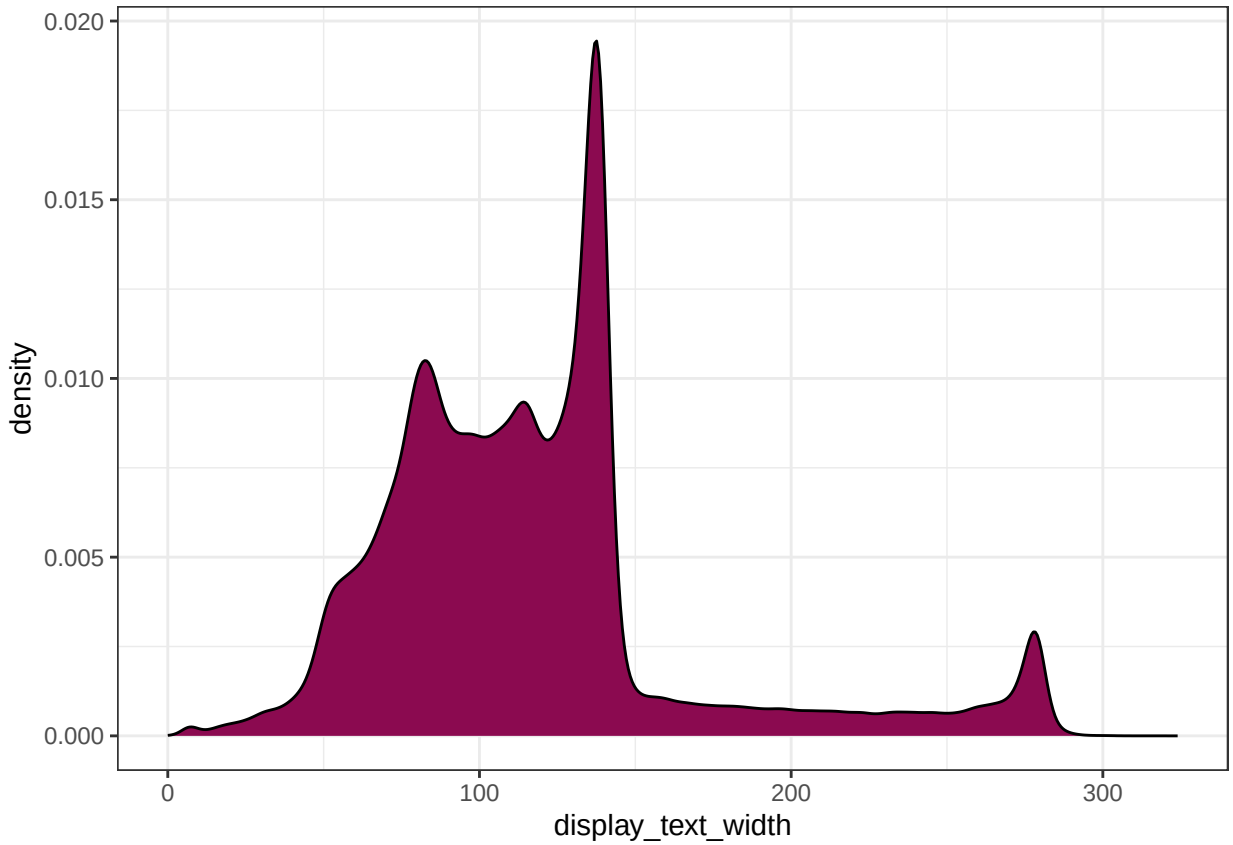



```
d %>%
  ggplot(aes(x = display_text_width))+ geom_histogram(bins = 50, color = "black") + theme_bw()
```



2. Create a density plot for the column `display_text_width` using the `ggplot2` package and `geom_density()` function. Fill the inside of density plot with a color using the `fill=` argument. Try at least four different numbers of smoothing badwith (e.g., 0.2, 1.5, 3, 5) by manipulating the `bw=` argument. Select what you think best represents the data for each. Provide a brief justification for your decision.

```
d %>%
  ggplot(aes(x = display_text_width))+ geom_density(fill = "deeppink4") + theme_bw()
```



Barplot

3. Using the information `text` column, create the following figure of the 15 most common words represented in these posts by using the `ggplot2()` package and `geom_col()` function. Remove the stop words, and also exclude the words such as 't.co','https','http','rt','rstats'.

```
# eugene_df <- tibble(
#   paragraph = seq_along(d$text),
#   description = d$text
# )
#
# ?seq_along
#
# eugene_df

names(d)
```

```
##  [1] "user_id"                "status_id"
##  [3] "created_at"             "screen_name"
##  [5] "text"                   "source"
##  [7] "display_text_width"     "reply_to_status_id"
##  [9] "reply_to_user_id"       "reply_to_screen_name"
## [11] "is_quote"               "is_retweet"
## [13] "favorite_count"         "retweet_count"
## [15] "hashtags"               "symbols"
## [17] "urls_url"               "urls_t.co"
## [19] "urls_expanded_url"      "media_url"
## [21] "media_t.co"             "media_expanded_url"
```

```
## [23] "media_type"              "ext_media_url"
## [25] "ext_media_t.co"          "ext_media_expanded_url"
## [27] "ext_media_type"          "mentions_user_id"
## [29] "mentions_screen_name"    "lang"
## [31] "quoted_status_id"        "quoted_text"
## [33] "quoted_created_at"       "quoted_source"
## [35] "quoted_favorite_count"   "quoted_retweet_count"
## [37] "quoted_user_id"          "quoted_screen_name"
## [39] "quoted_name"             "quoted_followers_count"
## [41] "quoted_friends_count"    "quoted_statuses_count"
## [43] "quoted_location"         "quoted_description"
## [45] "quoted_verified"         "retweet_status_id"
## [47] "retweet_text"            "retweet_created_at"
## [49] "retweet_source"          "retweet_favorite_count"
## [51] "retweet_retweet_count"   "retweet_user_id"
## [53] "retweet_screen_name"     "retweet_name"
## [55] "retweet_followers_count" "retweet_friends_count"
## [57] "retweet_statuses_count"  "retweet_location"
## [59] "retweet_description"     "retweet_verified"
## [61] "place_url"               "place_name"
## [63] "place_full_name"         "place_type"
## [65] "country"                 "country_code"
## [67] "geo_coords"              "coords_coords"
## [69] "bbox_coords"             "status_url"
## [71] "name"                    "location"
## [73] "description"             "url"
## [75] "protected"               "followers_count"
## [77] "friends_count"           "listed_count"
## [79] "statuses_count"          "favourites_count"
## [81] "account_created_at"      "verified"
## [83] "profile_url"             "profile_expanded_url"
## [85] "account_lang"            "profile_banner_url"
## [87] "profile_background_url"  "profile_image_url"
```

```
dat <- d %>%
  select(user_id, text)

dat <- dat %>%
unnest_tokens(word, text)

dat
```

```
## # A tibble: 7,736,204 x 2
##    user_id word
##    <chr>   <chr>
##  1 5685812 json
##  2 5685812 reading
##  3 5685812 in
##  4 5685812 python
##  5 5685812 using
##  6 5685812 rust
##  7 5685812 vs
##  8 5685812 c
##  9 5685812 backed
## 10 5685812 functions
```

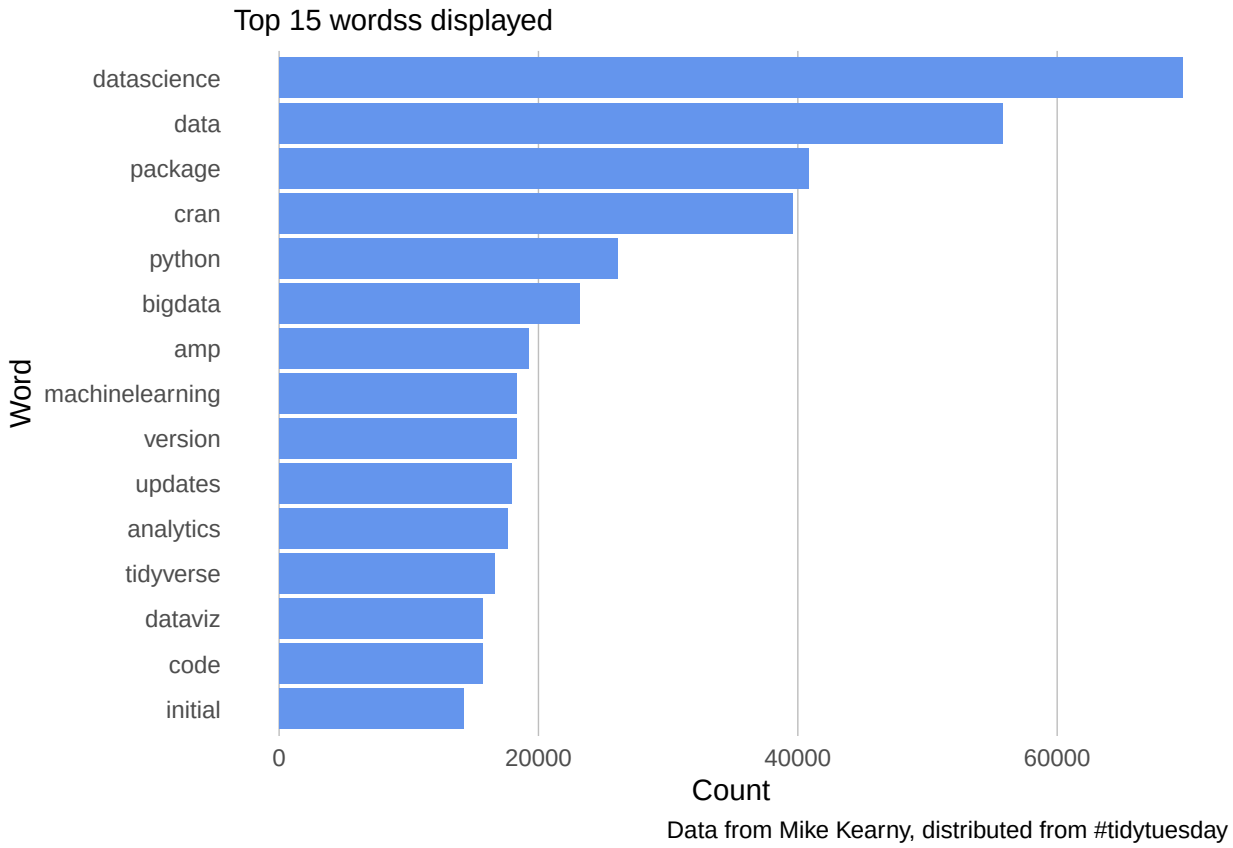
```
## # ... with 7,736,194 more rows
```

```
#dat_reduced <- dat[!dat$text %in% stop_words,]


dat %>%
anti_join(stop_words) %>%
filter(word != "t.co",word !="https",word != "http",word != "rt", word !="rstats") %>%
count(word, sort = TRUE) %>%
  mutate(word = reorder(word, n)) %>% # make y-axis ordered by n
  slice(1:15) %>% # select only the first 15 rows
  ggplot(aes(n, word)) +
  theme_minimal() +
    geom_col(fill = "cornflowerblue")+ theme(panel.grid.major.x = element_line(color = "grey",size = 0.1
```

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

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
```




```
  #theme(panel.grid.major = element_line(color = "black",
                                         # size = 0.5,
                                         # linetype = 1))
```

4. Style the plot so it (mostly) matches the below. It does not need to be exact, but it should be close.