Lab 1

Tian Walker

2023-01-29

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.

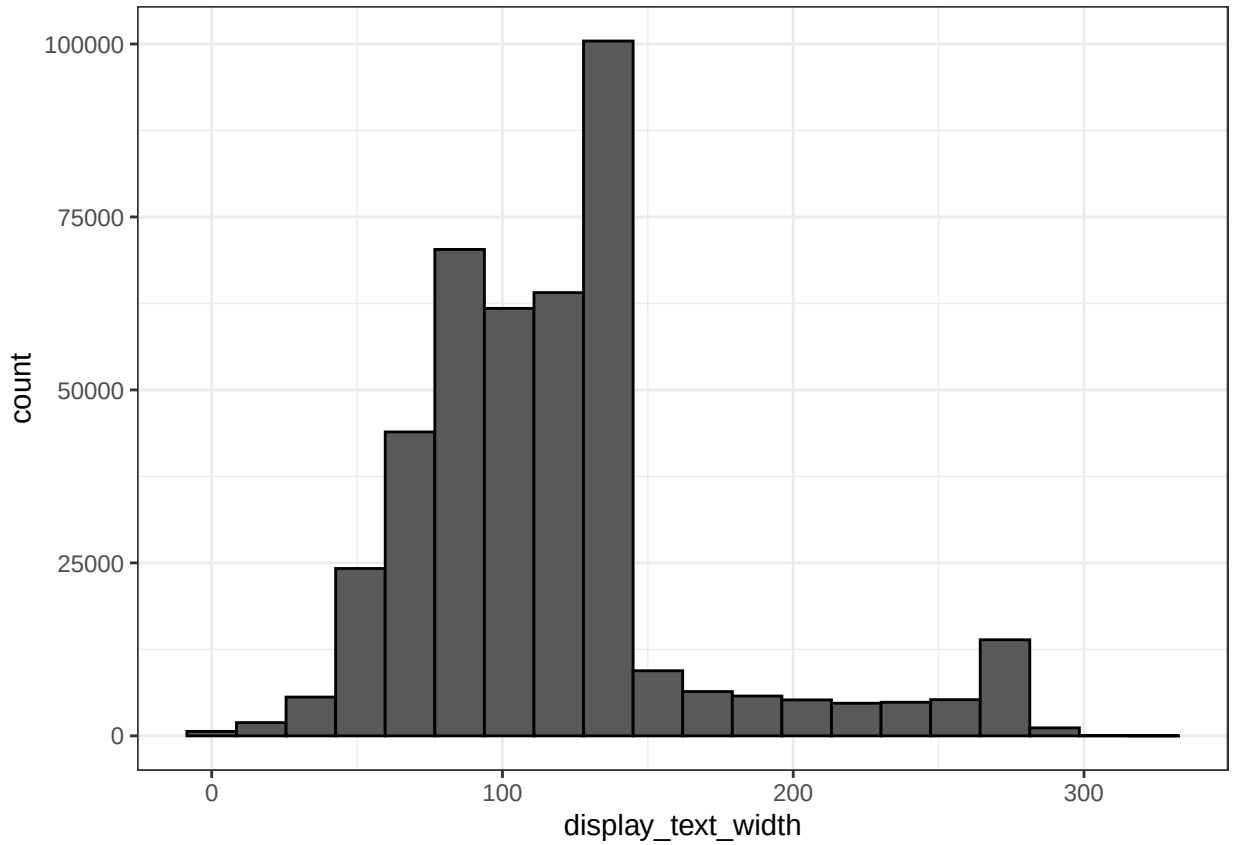
If you need help with processing text data, please revisit the notebook introduced in Week 1.

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

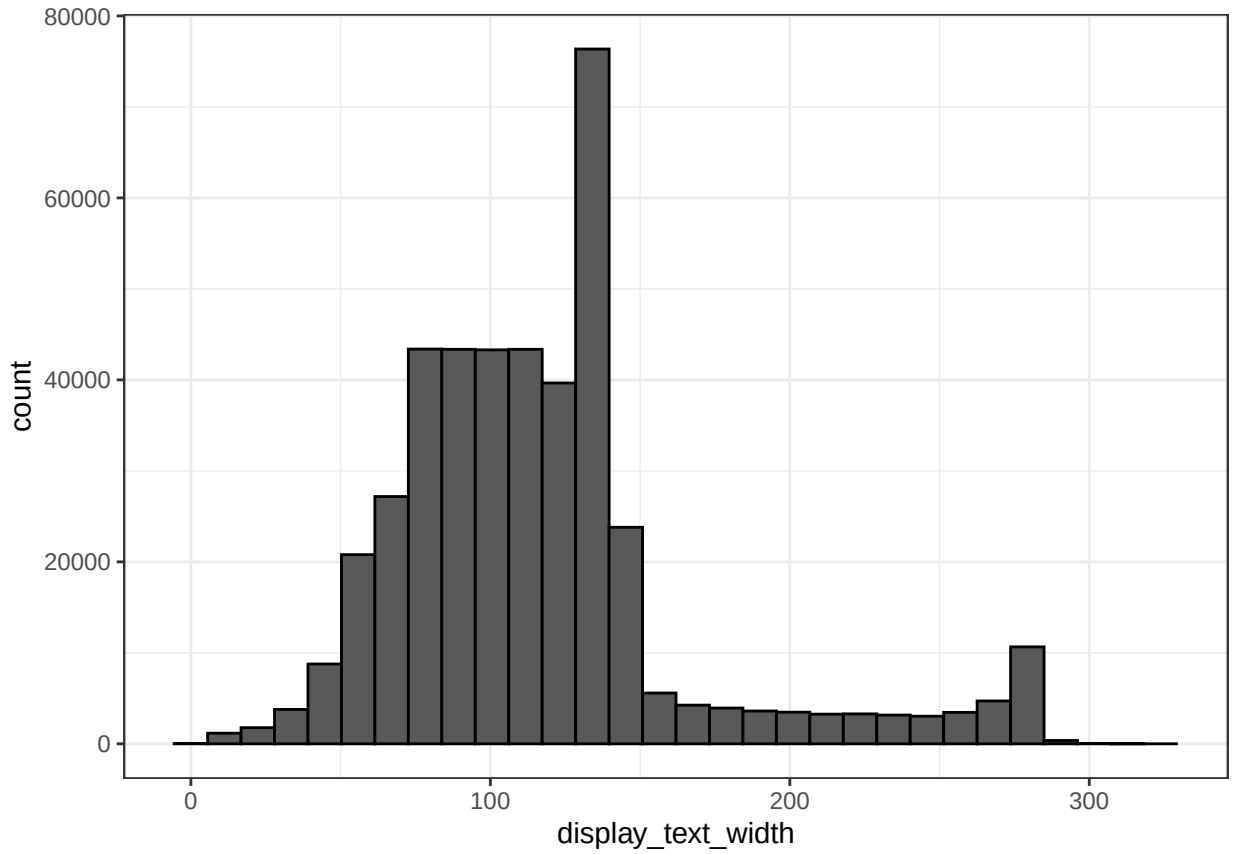
Histogram and Density plots

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.

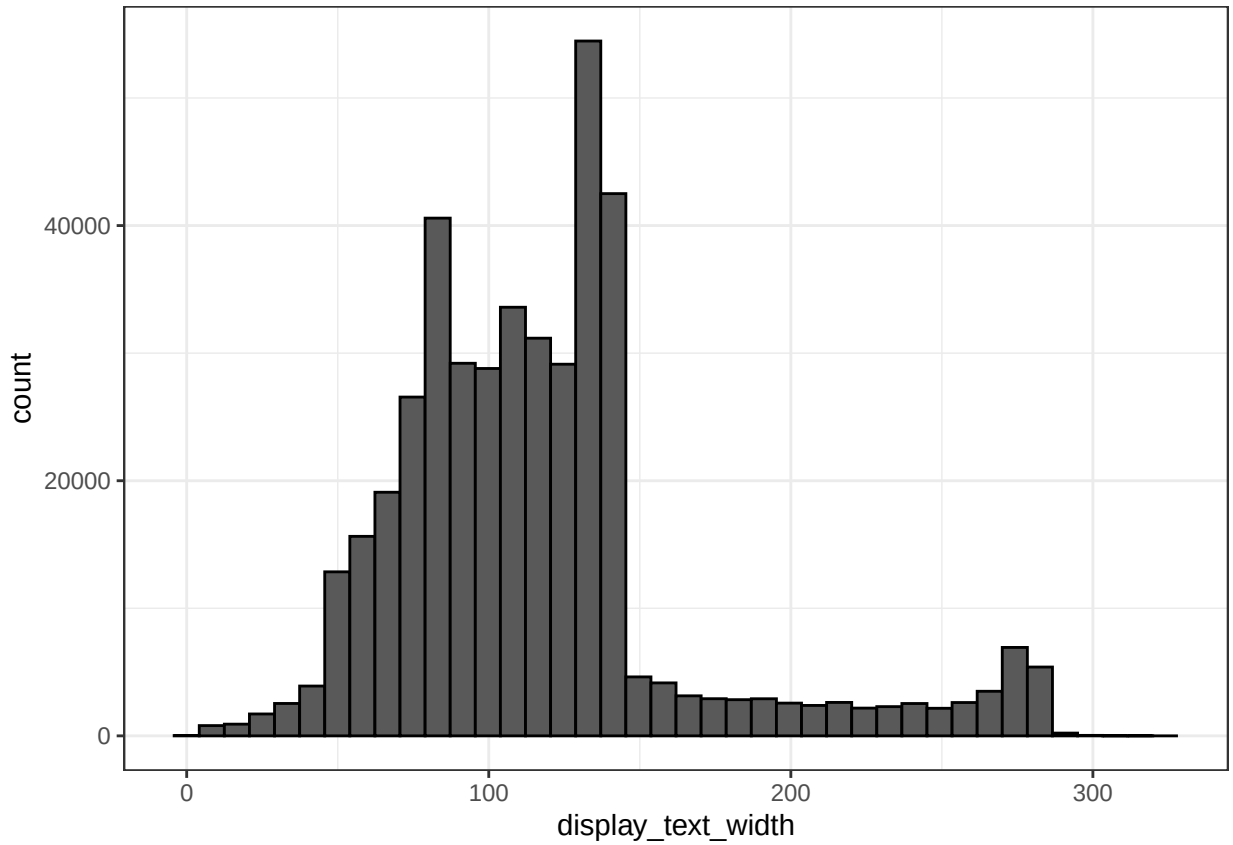
```
d %>%
  ggplot(aes(x = display_text_width))+ geom_histogram(bins = 20, color = "black") + theme_bw()
```



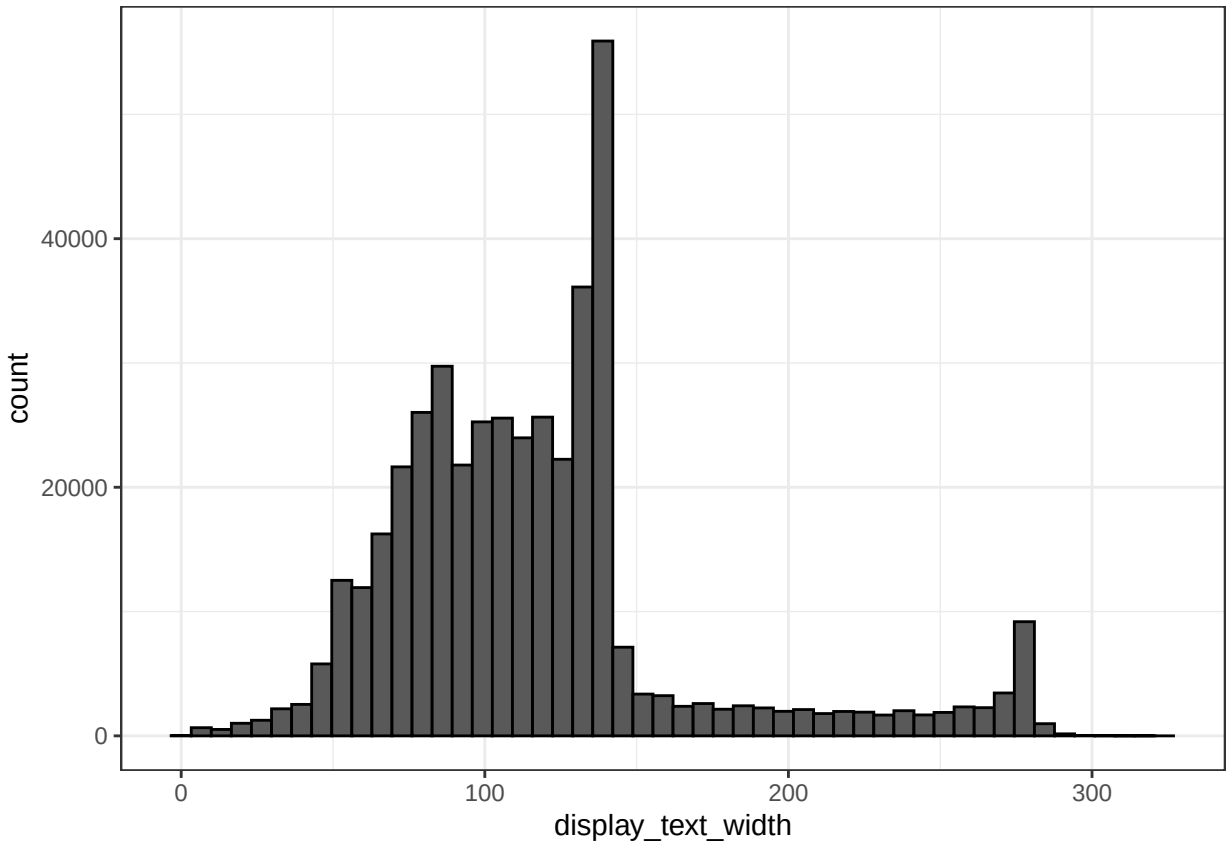

```
d %>%
  ggplot(aes(x = display_text_width))+ geom_histogram(bins = 30, color = "black") + theme_bw()
```




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



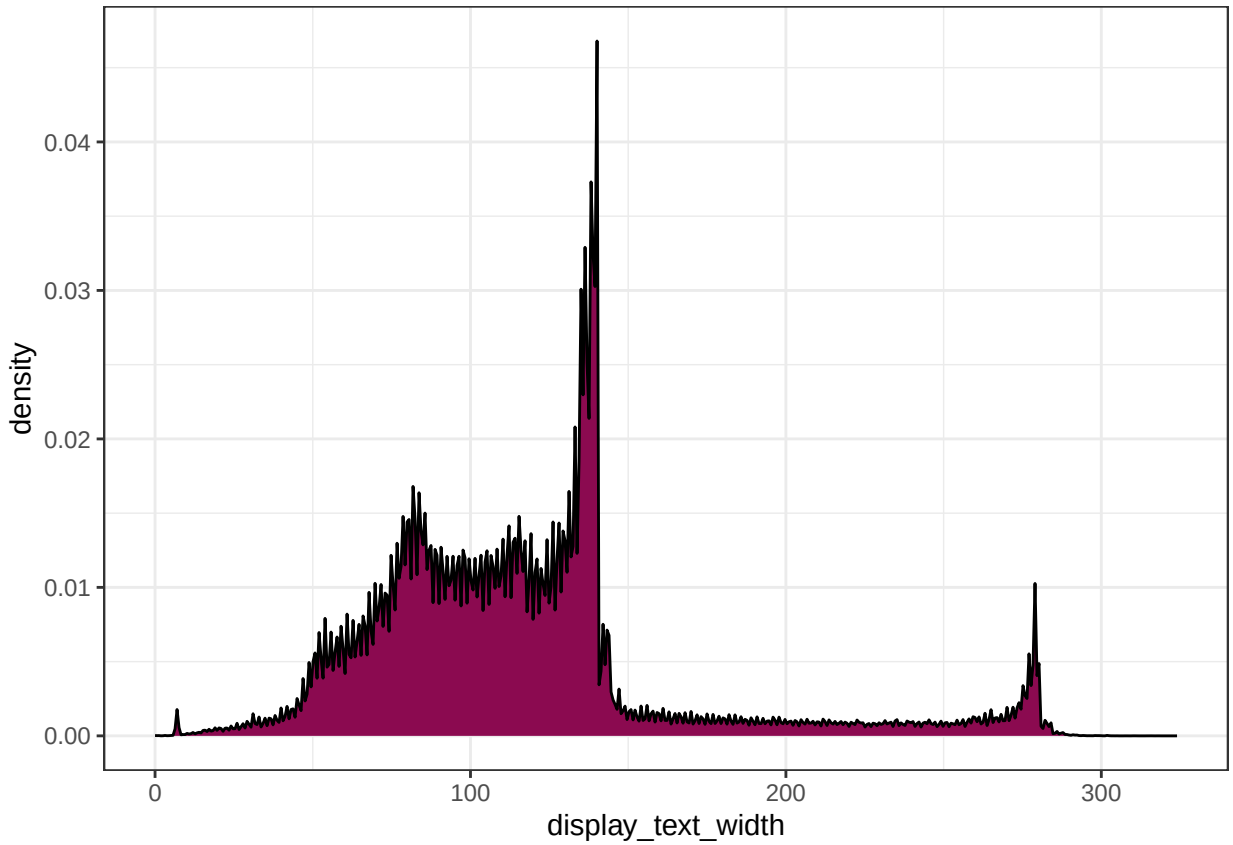

```
d %>%
  ggplot(aes(x = display_text_width))+ geom_histogram(bins = 50, color = "black") + theme_bw()
```



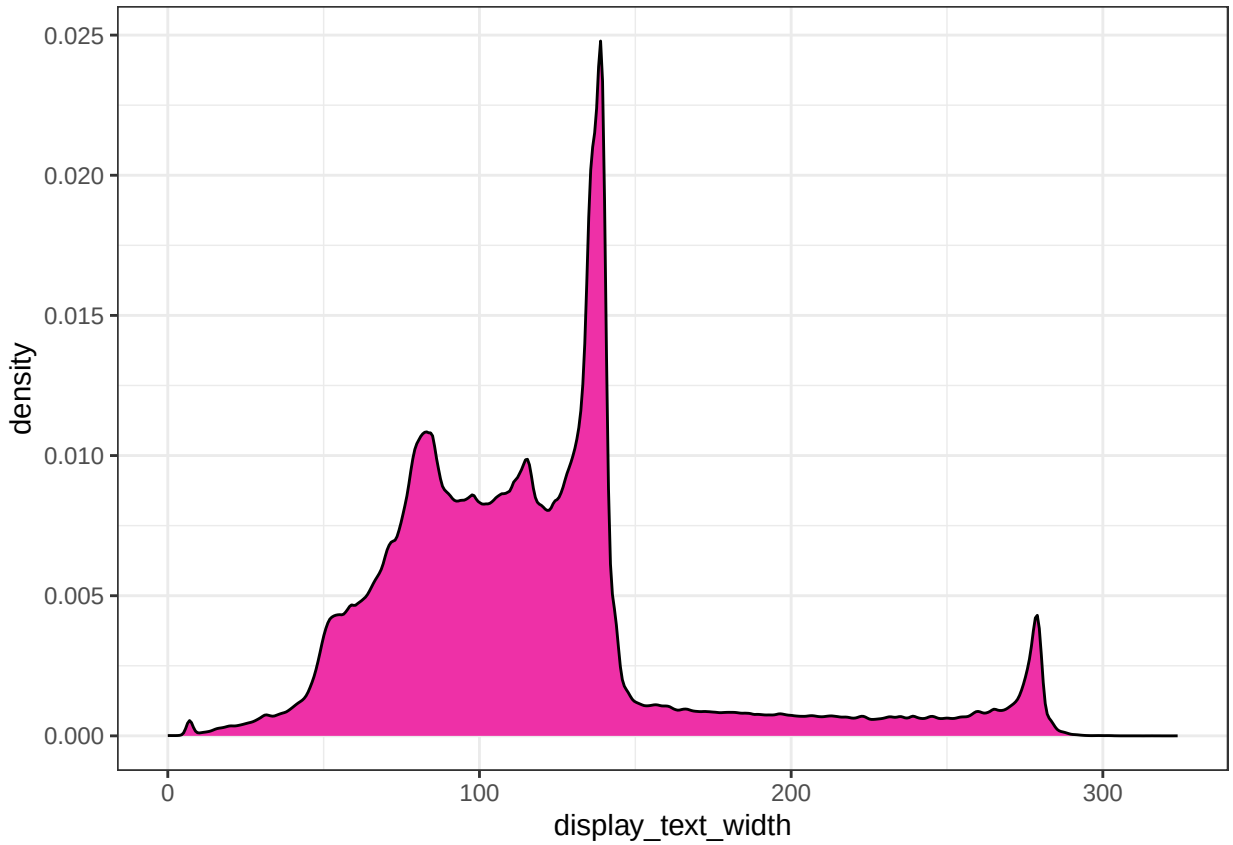
It is difficult to decide which bin number is the best. It would be helpful to have more details about the units of display text width and what they mean. The bin width changes what the graph communicates in terms of the variety of display widths it shows. Ultimately, I think that second bar chart with 30 bins that lumps more together and creates a y-axis of 80,000 instead of 40,000 offers the best "at-a-glance" picture of this variable. I think it is a nice middle ground between 20 bins and 40 or 50.

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.

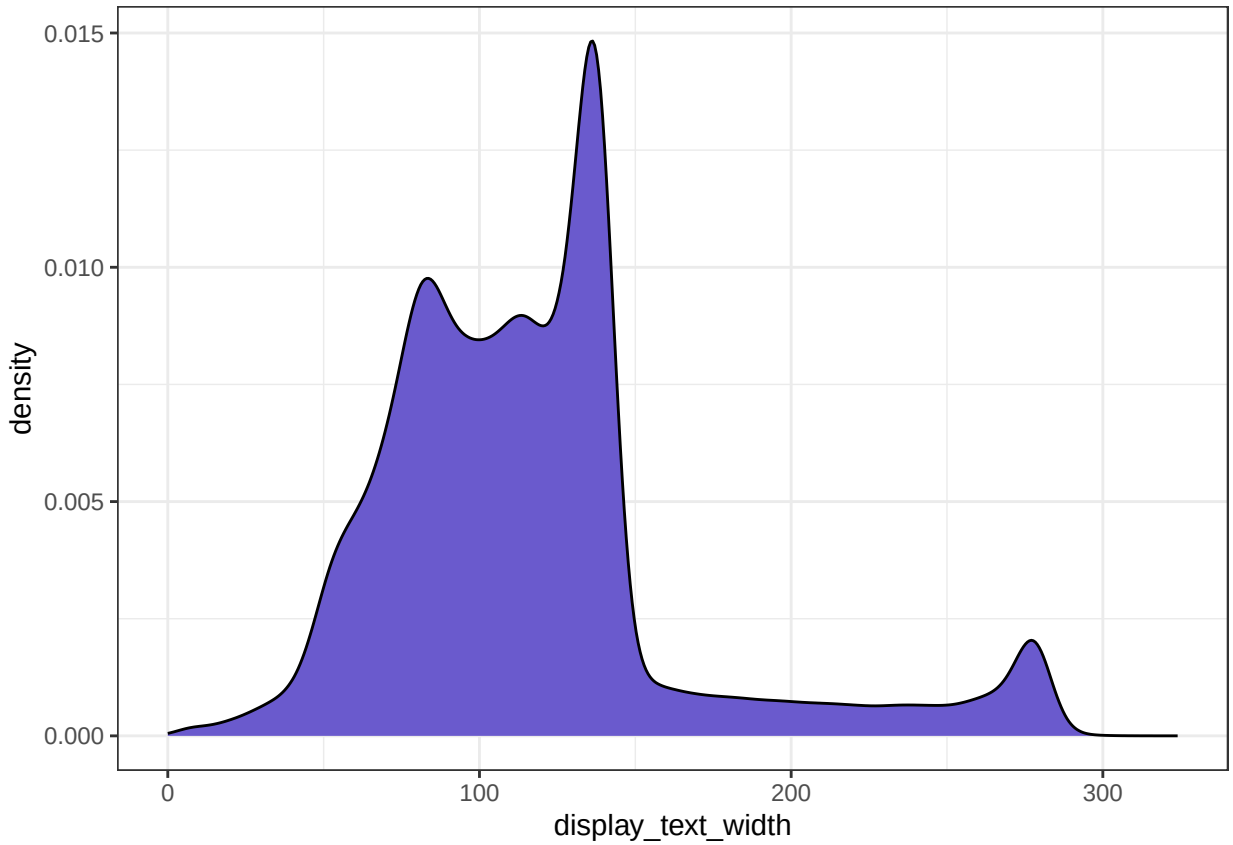
```
d %>%
  ggplot(aes(x = display_text_width))+ geom_density(bw = 0.2, fill = "deeppink4") + theme_bw()
```



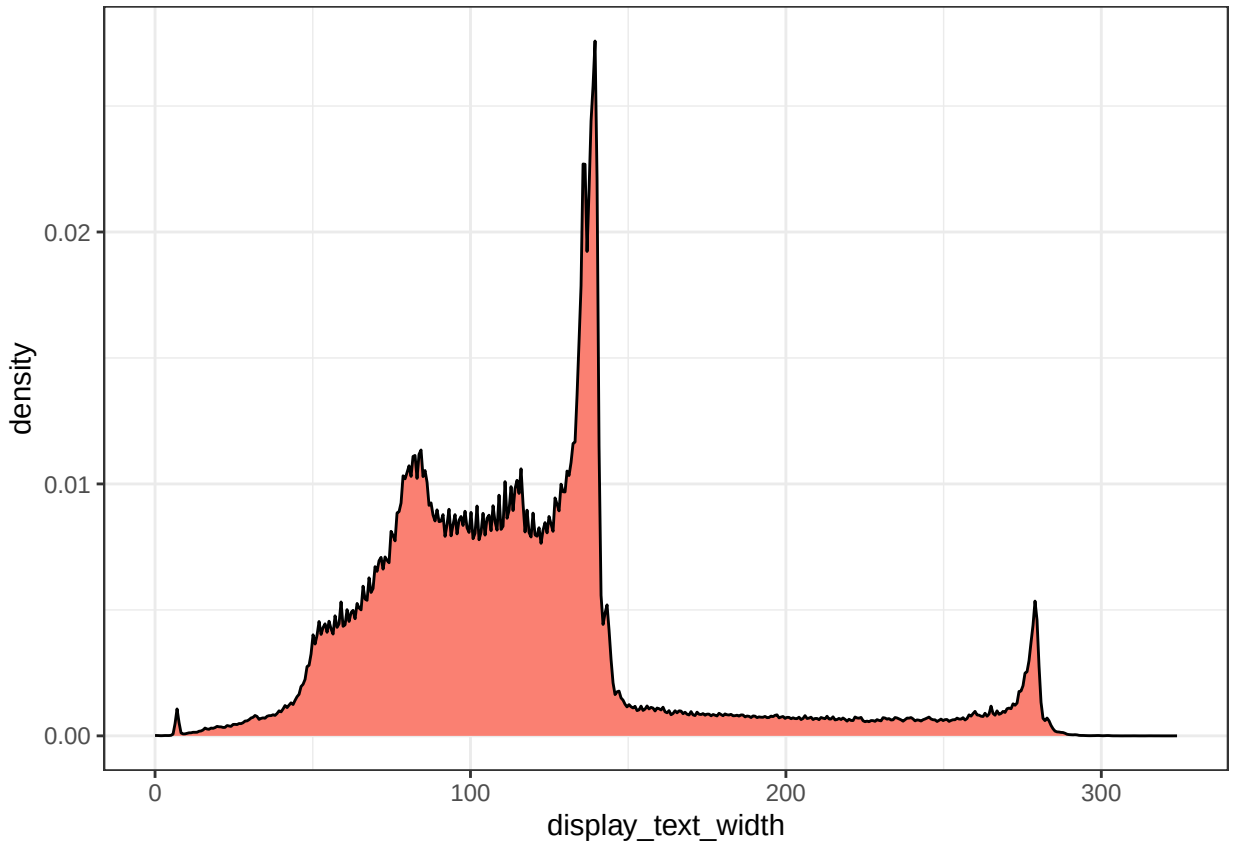

```
d %>%
  ggplot(aes(x = display_text_width))+ geom_density(bw = 1, fill = "maroon2") + theme_bw()
```



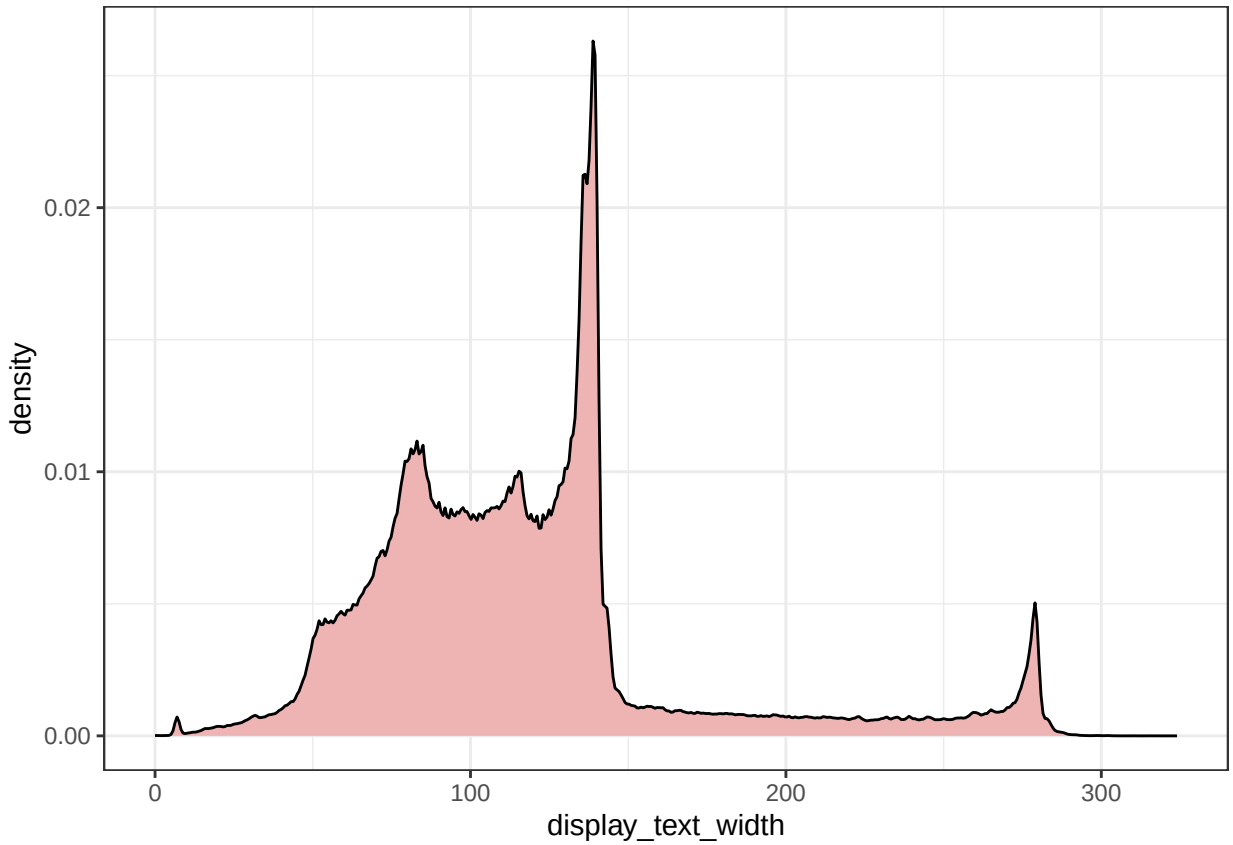

```
d %>%
  ggplot(aes(x = display_text_width))+ geom_density(bw = 5, fill = "slateblue") + theme_bw()
```



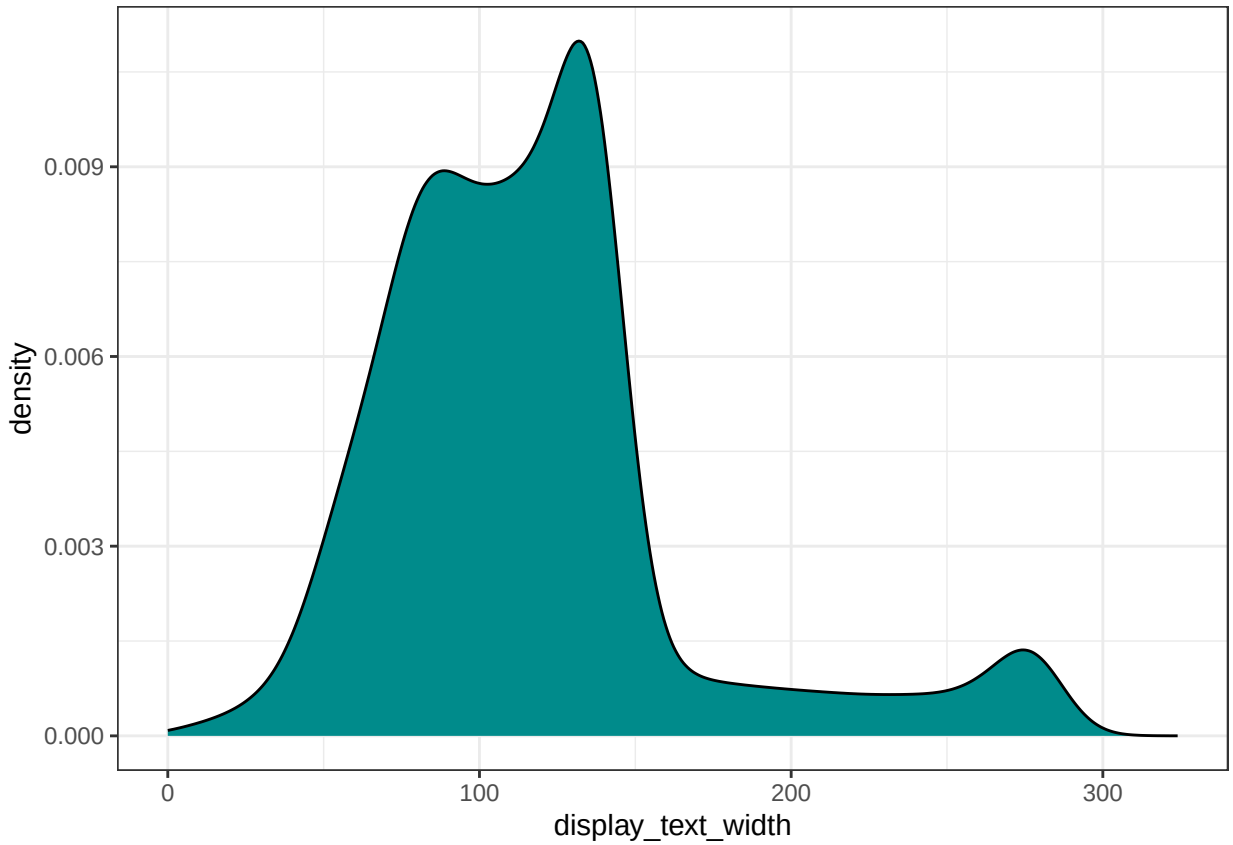

```
d %>%
  ggplot(aes(x = display_text_width))+ geom_density(bw = .5, fill = "salmon") + theme_bw()
```




```
d %>%
  ggplot(aes(x = display_text_width))+ geom_density(bw = 0.7, fill = "rosybrown2") + theme_bw()
```




```
d %>%
  ggplot(aes(x = display_text_width))+ geom_density(bw = 10, fill = "darkcyan") + theme_bw()
```



I think that a badwith of 5 works the best. It captures the shape of the data without showing unnecessary detail or overly rounding off edges (as seen with bw = 10). This is assuming that with a dataset this large the small differences are unimportant. If the small differences at each number mattered, then I would go with the 0.2. Color seems entirely aesthetic, but I like the deeppink4 the best.

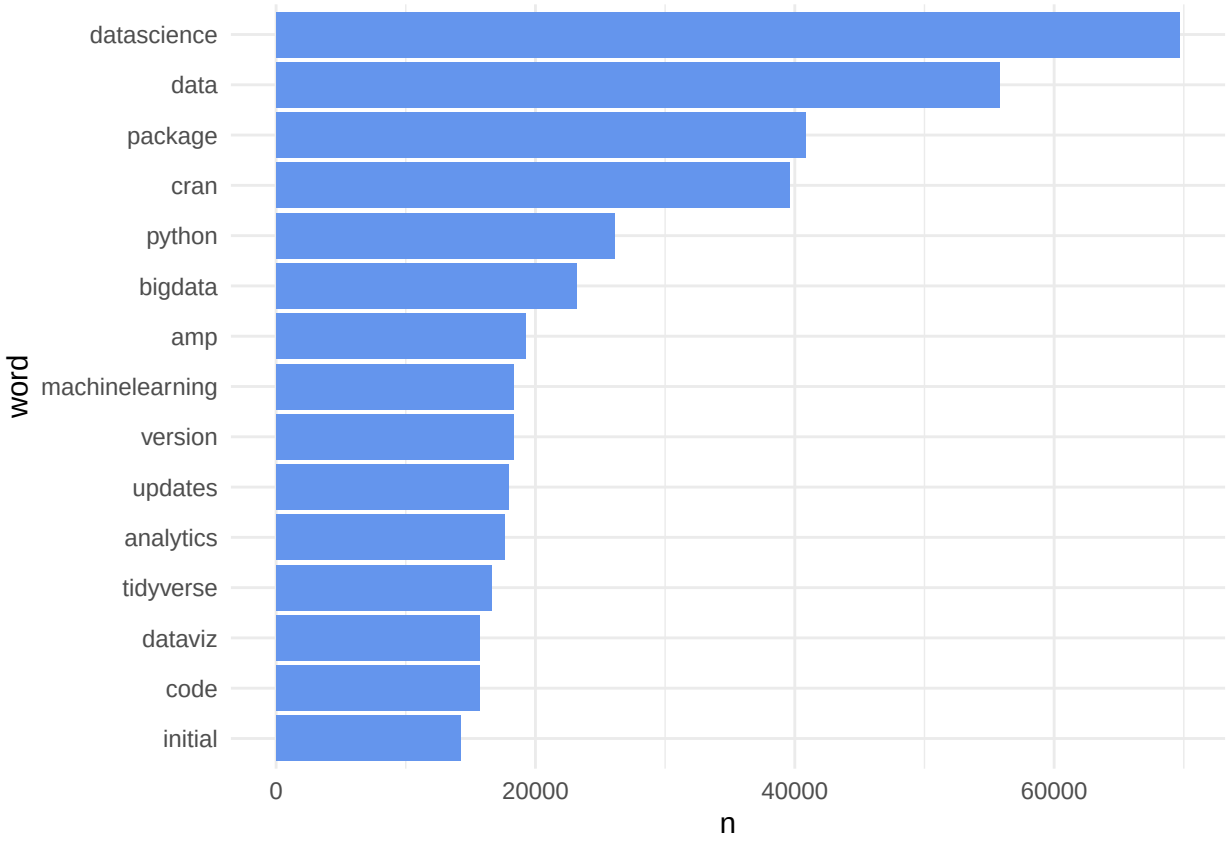
Barplot

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'.

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

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

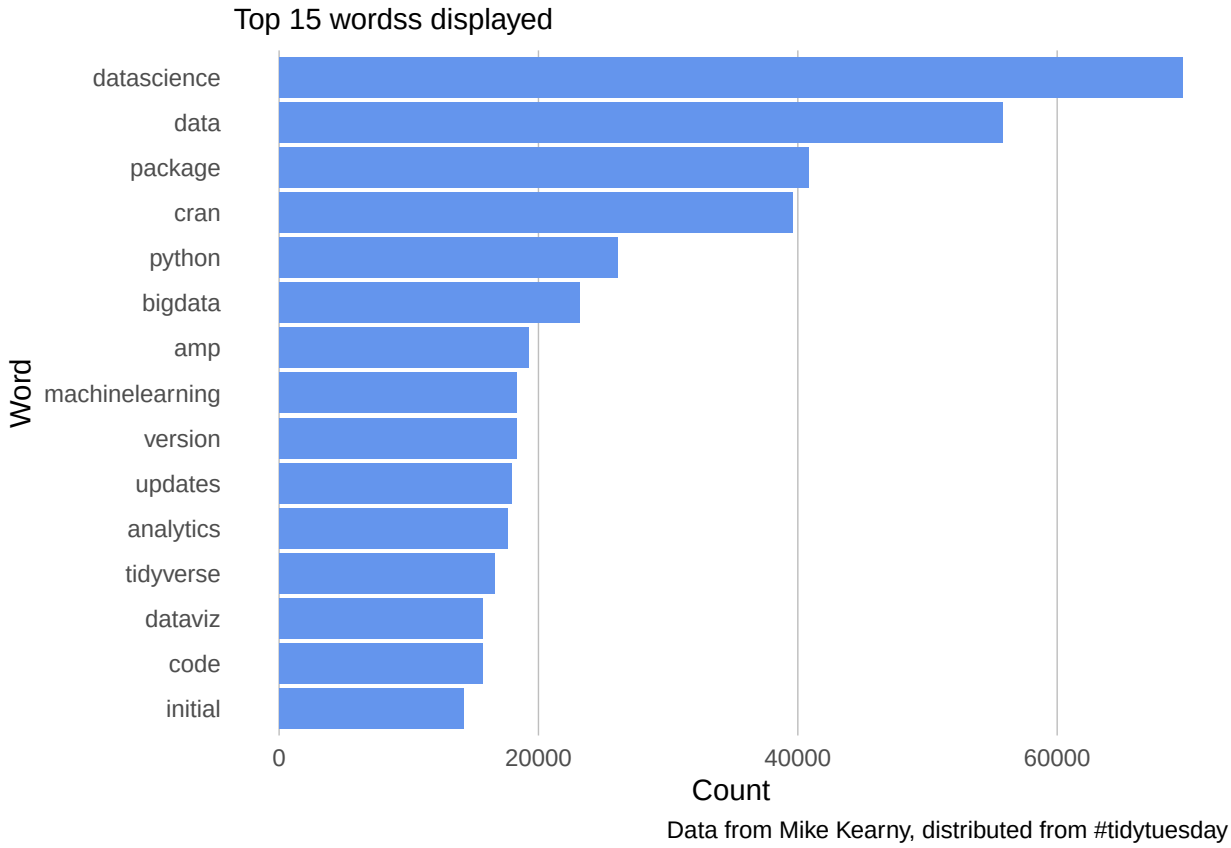
```
#dat_reduced <- dat[!dat$text %in% stop_words,]

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")
```

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



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

```
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
```


Top 15 wordss displayed
datascience
data
package
cran
python
bigdata
amp
machinelearning
version
updates
analytics
tidyverse
dataviz
code
initial
Word
0
20000
40000
60000
Count
Data from Mike Kearny, distributed from #tidytuesday