Lab 1

Emmanuel Maduneme

2023-01-23

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.

```
d <- rio::import(here::here("data", "rstats_tweets.rds"),
                 setclass = "tbl_df")
```

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

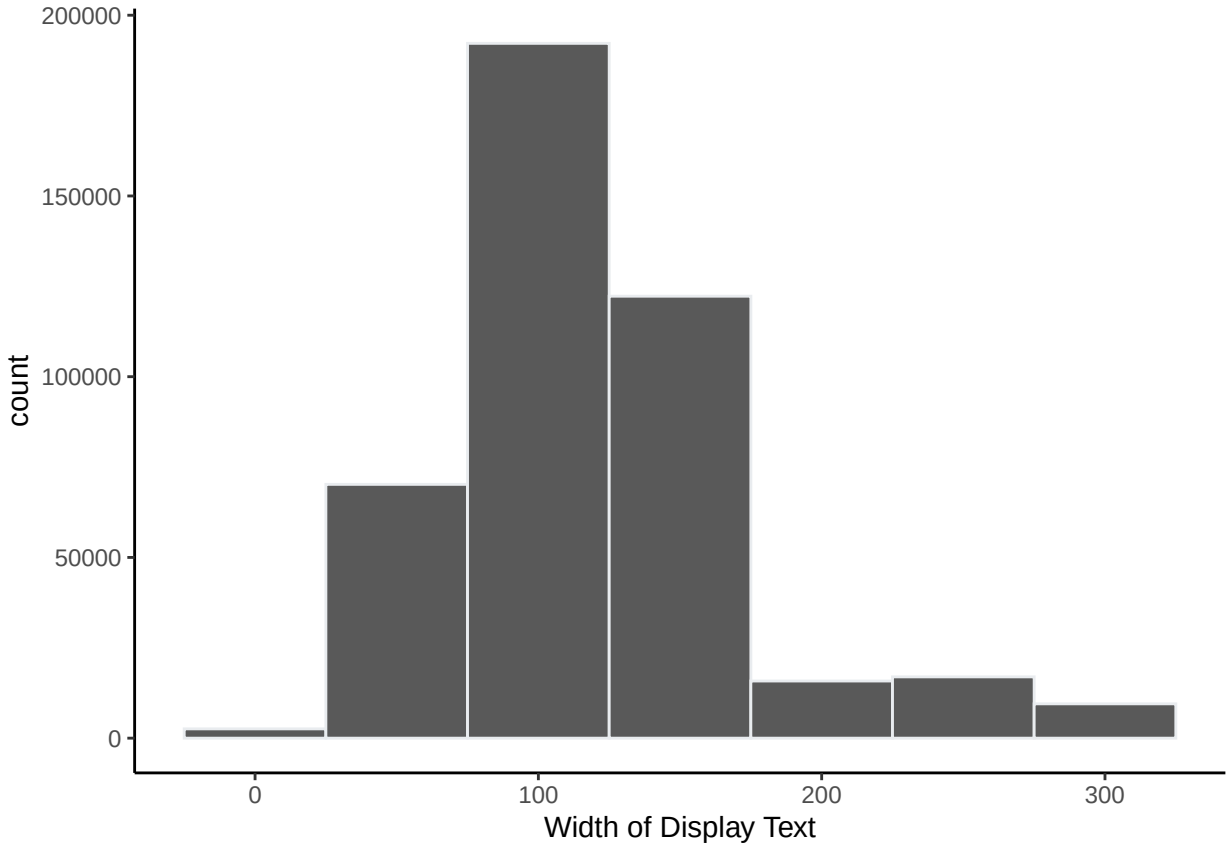
https://www.kaggle.com/code/uocoeeds/introduction-to-textual-data

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.

- I used a simple gray histogram because it is simple, easy to differentiate between widths of text after using the `color =` argument to specify and it makes it easy to identify which width number has the highest count.

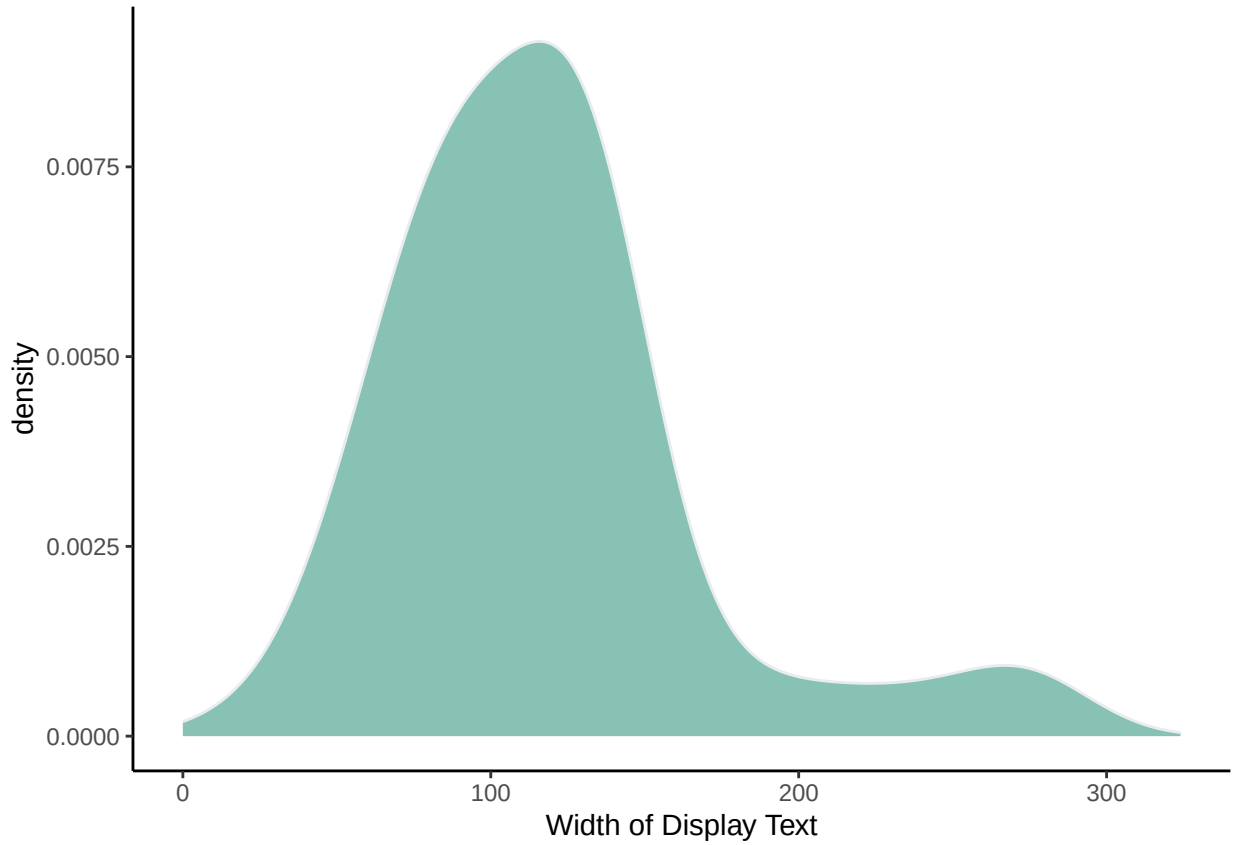
```
d_his <- d %>%
  ggplot(aes(display_text_width)) +
  geom_histogram(binwidth=50,color="#e9ecef", bins = 20)+
  theme_classic() +
  labs(x = "Width of Display Text")
d_his
```



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.

I also used a density plot with smooth bandwidth as it adheres to the assigment instructions while providing a clear distribution of the variable which appears to be skewed.

```
d %>%
  ggplot(aes(display_text_width)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8, bw = 20) +
  theme_classic() +
  labs(x = "Width of Display Text")
```



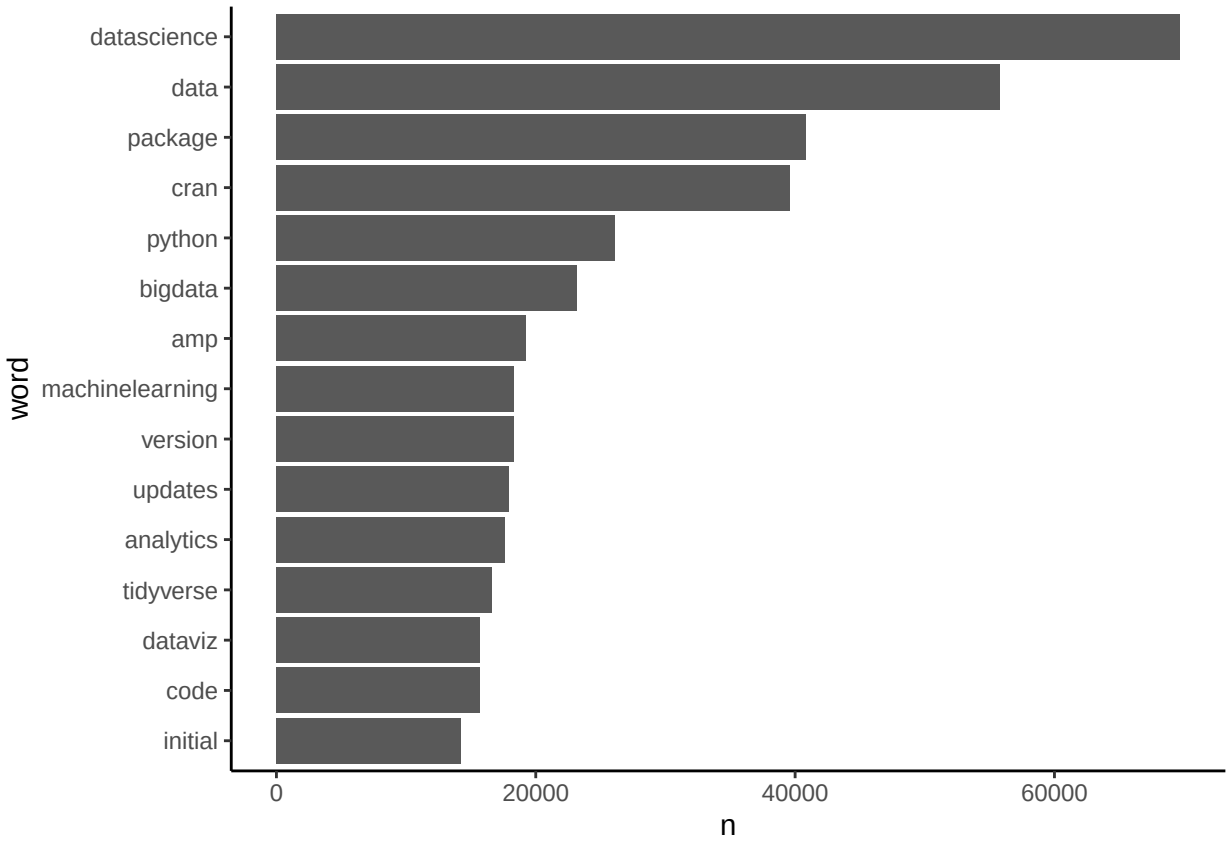
Barplot

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'.

```
library(tidytext)

d_tidy_words <- d %>%
  unnest_tokens(word,text) %>%
  select(user_id, word)
 d_tidy_words %>%
  anti_join(stop_words) %>%
  filter(!word %in% c("t.co","https","http","rt","rstats")) %>%
  count(word, sort = T) %>%
  slice(1:15) %>%
  mutate(word = fct_reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  theme_classic()
```

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



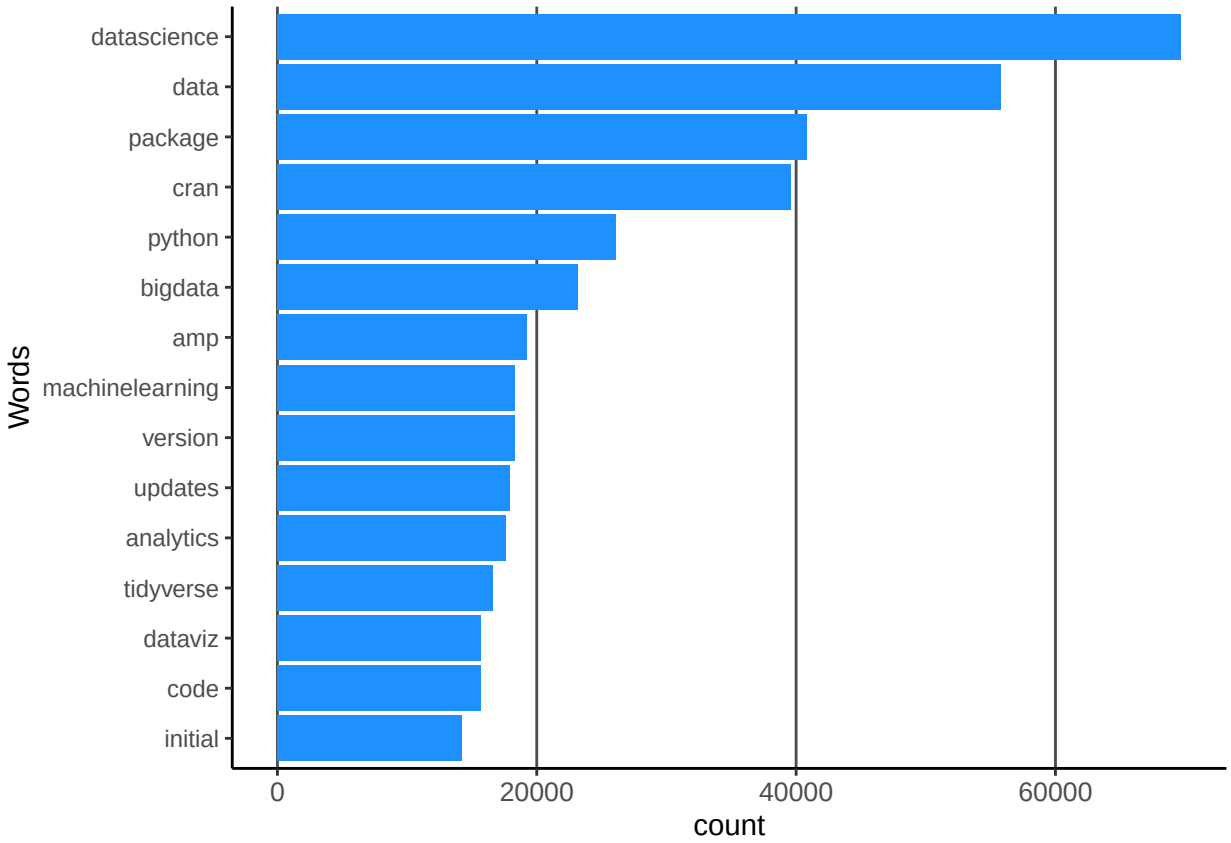
4. Style the plot so it (mostly) matches the below. It does not need to be exact, but it should be close.

```
d_plot <- d_tidy_words %>%
 anti_join(stop_words) %>%
 filter(!word %in% c("t.co","https","http","rt","rstats")) %>%
 count(word, sort = T) %>%
 slice(1:15) %>%
 mutate(word = fct_reorder(word, n))
```

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

```
 d_plot %>%
   ggplot(aes(n, word)) +
 geom_col(fill = "dodgerblue") +
 theme_classic() +
 labs(x = "count",
      y = "Words") +
 theme(axis.text.x = element_text(size = 10, color = "grey30"),
       panel.grid.major.x = element_line(color = "grey30",
                                         size = 0.5,
                                         linetype = 1))
```

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


datascience
data
package
cran
python
bigdata
amp
machinelearning
version
updates
analytics
tidyverse
dataviz
code
initial
Words
0
20000
40000
60000
count