100% ~

page 2

- op We can also do the sentiment ...
- T inner_join drops the unmatch...
- image 6
- T sentiment ← get_sentiments("...
- T wine_tasting %>% unnest_tok...
- T wine_tasting %>% unnest_tok...

T top_variety ← wine_t... \bullet •

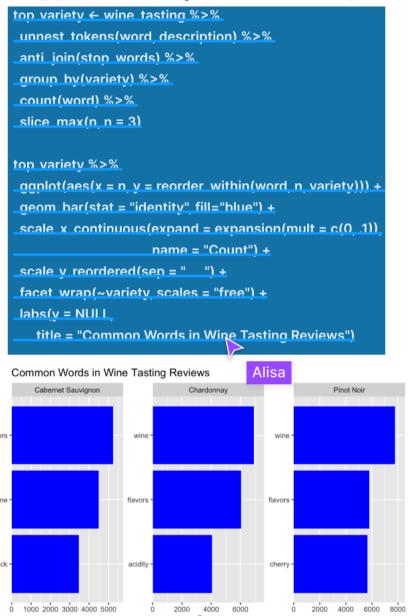
- □ Rectangle 9
- T We can also combine tidytext ...
- Rectangle 8
- T We can also do word sentime...
- Rectangle 7
- extstyle ext
- T reorder_within is a nifty functi...
- T By below code, we can visuali...
- image 5
- Screen Shot 2022-11-12 at 9....
- Rectangle 3
- image 3

page 1

page 2

By below code, we can visualize the top 3 word frenquency in each wine variety.

slice_max helps to choose the number of words we want, by changing that, we can get the top n word frequency. reorder_within is a nifty function within tidytext: it reorders a column before plotting with faceting, such that the values are ordered within each facet. This requires two functions: reorder_within applied to the column, then either scale_x_reordered or scale_y_reordered added to the plot.



We can also do word sentiment analysis using the built in sentiment list to get sentiment of each word and its frequency.

inner_join drops the unmatched observations

word	sentiment	n
rich	positive	4783
soft	positive	3552
sweet	positive	3228

sentiment ← get_sentiments("bing") word_sentiment ← wine_tasting %>% unnest_tokens(word, description) %>% anti_join(stop_words) %>% mutate(word = str_extract(word, "[a-z']+")) %>% inner_join(sentiment) %>% count(word, sentiment, sort = TRUE)

We can also combine tidytext with Wordcloud package. Larger words means higher frequency.



wine_tasting %>% unnest_tokens(word, description) %>% anti_join(stop_words) %>% count(word) %>% with(wordcloud(word, n, max.words = 100))

We can also do the sentiment analysis to tag positive and negative words using an inner join, then find the most common positive and negative words, and finally indicate the result by Wordcloud.



wine_tasting %>% unnest_tokens(word, description) %>% anti_join(stop_words) %>% inner_join(sentiment) %>% count(word, sentiment, sort = TRUE) %>% acast(word ~ sentiment, value.var = "n", fill = 0) %>% comparison.cloud(colors = c("blue", "darkcyan"), max.words = 100)

