UNIVERSITY OF CALIFORNIA, BERKELEY

STAT 133 FALL 2024

Lecture Notes Taken in the Class

October 23rd, 2024 – Wednesday

COURSE: MATH 133 – CONCEPTS IN COMPUTING WITH DATA

INSTRUCTOR: GASTON SANCHEZ

NAME: **ALDAN OU**

DATE: OCTOBER 23RD, 2024

❖ R CODING PREPARATION – LOADING LIBRARIES

```
{r}
library(tidyverse)
library(tidytext)
library(janeaustenr)
library(wordcloud)
library(RColorBrewer)
```

```
{r}
class(prideprejudice)
length(prideprejudice)
head(prideprejudice, n = 15)
```

```
[1] "character"
[1] 13030
 [1] "PRIDE AND PREJUDICE"                                              ""
 [3] "By Jane Austen"                                                   ""
 [5] ""                                                                 ""
 [7] "Chapter 1"                                                        ""
 [9] ""                                                                 "It is a truth
universally acknowledged, that a single man in possession"
[11] "of a good fortune, must be in want of a wife."                    ""
[13] "However little known the feelings or views of such a man may be on his"   "first
entering a neighbourhood, this truth is so well fixed in the minds"
[15] "of the surrounding families, that he is considered the rightful property"
```

❖ TOKENIZATION

In short, tokenization si a fundamental preprocessing step in text analysis that breaks down text into smaller, meaningful units called tokens. These tokens can be words, characters, sentences, or sub-words.

The purposes of doing tokenization include:

- Makes raw text more manageable for computational analysis;
- Converts unstructured text into a structured numerical format suitable for machine learning;
- Enables pattern recognition in text data;
- Improves computational efficiency by breaking complex text into simpler units.

```r
# text into a data frame
pride = data.frame("text" = prideprejudice)
head(pride)
```

Description: df [6 × 1]

	text <chr>
1	PRIDE AND PREJUDICE
2	
3	By Jane Austen
4	
5	
6	

6 rows

```r
# tokenization
pride_tokens = unnest_tokens(tbl = pride, input = text, output = word)
pride_tokens
```

Description: df [122,204 × 1]

	word <chr>
1	pride
2	and
3	prejudice
4	by
5	jane
6	austen
7	chapter
8	1
9	it
10	is

1–10 of 122,204 rows Previous 1 2 3 4 5 6 ... 100 Next

❖ FREQUENCY ANALYSIS (GET COUNTS)

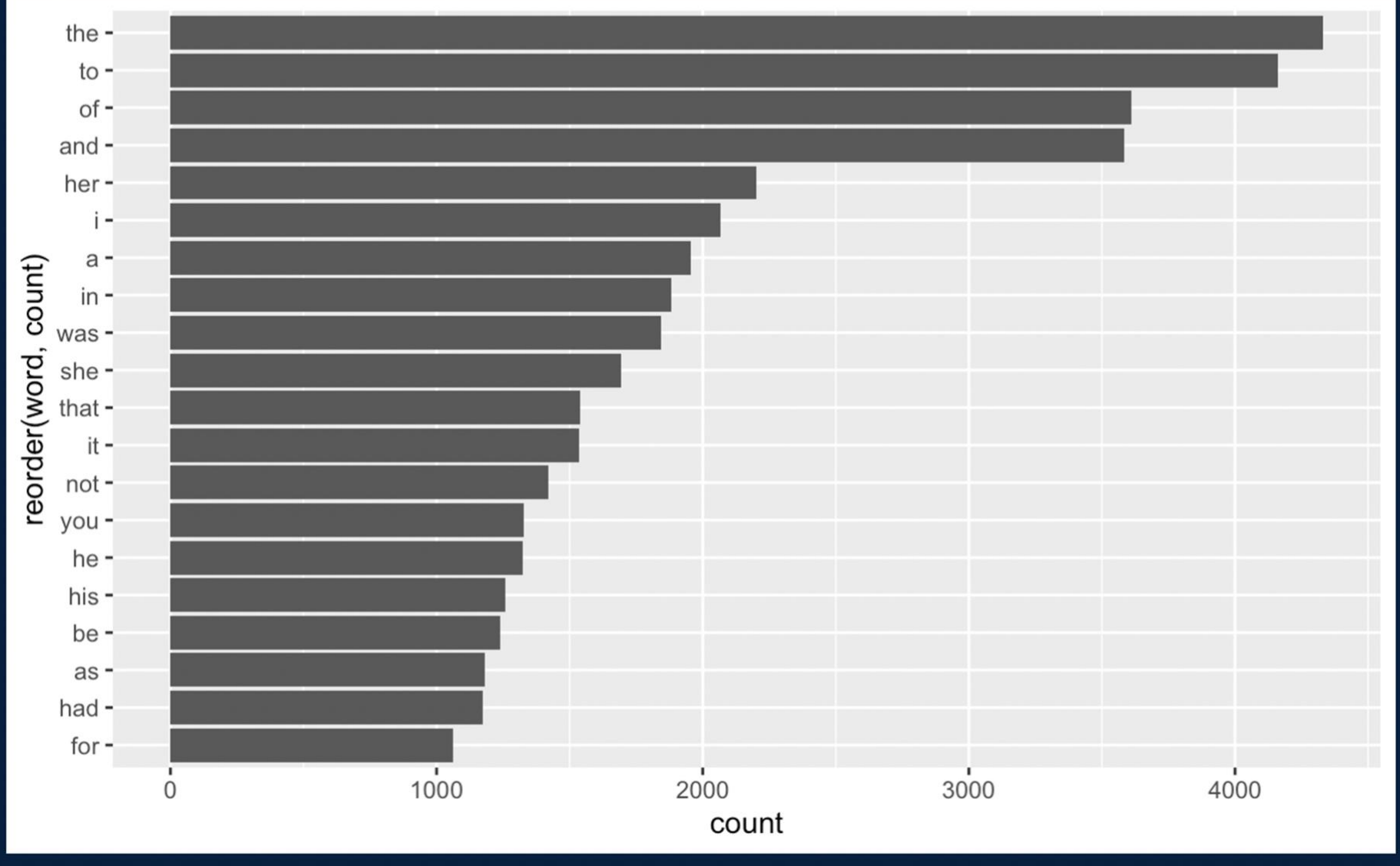
Frequency analysis is a fundamental method in text mining that examines how often words appear in text data. There are three main types of counts: Term Frequency (TF), Document Frequency, and Raw Counts.

The methods of getting counts include three steps.

1) Tokenization of text into individual words;
2) Counting occurrences of each token;
3) Calculating relative frequencies by normalizing counts.

See the following R coding example in R studio.

```r
pride_tokens %>%
  count(word, name = "count", sort = TRUE) %>%
  slice(1:20) %>%
  ggplot(aes(y= reorder(word, count), x = count)) +
  geom_col()
```



❖ REMOVE "STOP WORDS"

The removal of stop words serves several key purposes in text analysis and natural language processing. The benefits include:

- Reduces the size of text data significantly, making faster computations;
- Decreases the dimensionality of the data, resulting in more efficient processing;
- Focuses on meaningful and informative words by eliminating common words that carry little semantic value;
- Helps highlight keywords that contain essential meaning in the text;
- Creates more robust models by reducing the number of features they need to process.

Common stop words include “the”, “is”, “and”.

See the following R coding example from R studio.

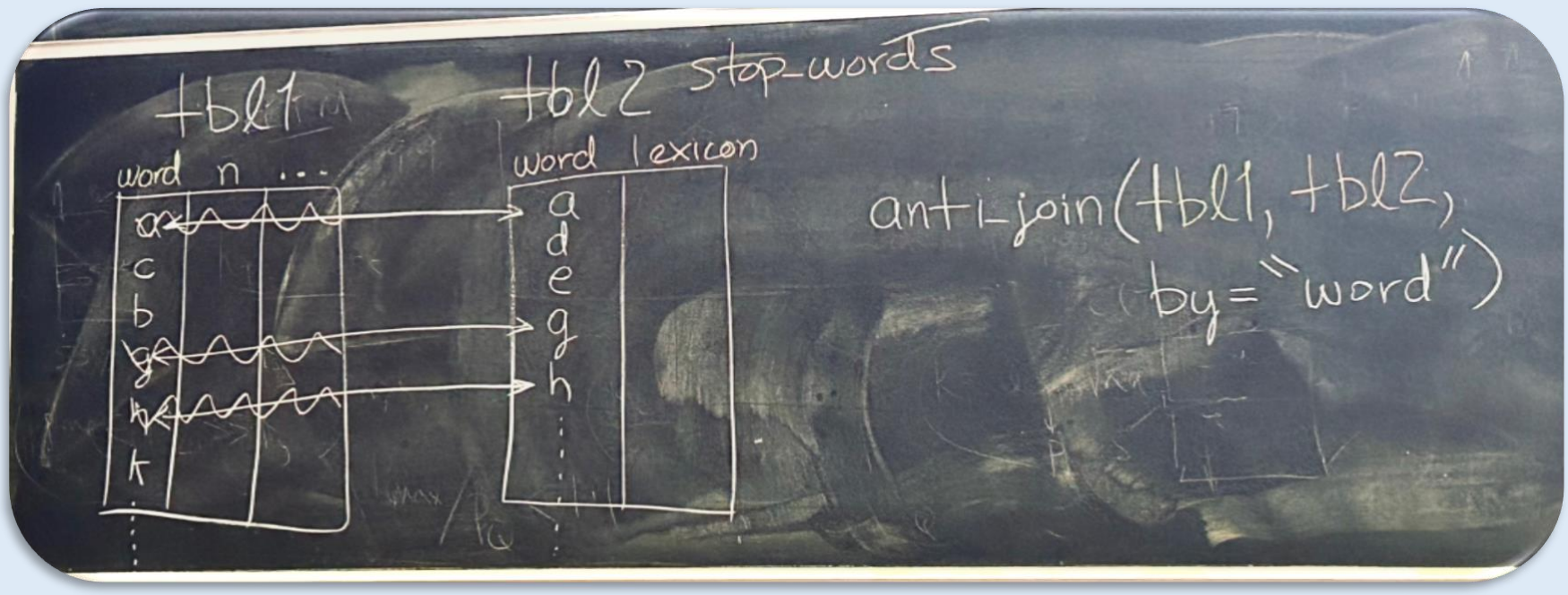
```r
# Lexicon of "Stop Words"
stop_words
```

A tibble: 1,149 × 2

word <chr>	lexicon <chr>
a	SMART
a's	SMART
able	SMART
about	SMART
above	SMART
according	SMART
accordingly	SMART
across	SMART
actually	SMART
after	SMART

1–10 of 1,149 rows Previous 1 2 3 4 5 6 ... 100 Next

The following photos are Prof. Sanchez’s lecture notes that were taken in the class.



- This diagram shown above is demonstrating how to perform text analysis using tidy data principles.

1) The first table “tbl1” represents a text document that has been tokenized into individual words or characters.
2) The second table “tbl2” represents a reference lexicon or dictionary.
3) The anti-join operation is being used to filter or remove certain words from the first table “tbl1” based on their presence in the lexicon table “tbl2”.

In short, the diagram illustrates a common text mining operation where we want to remove certain words (like stop words) from the text analysis by comparing against a reference list.

See the following R coding example from R studio.

```r
{r}
pride_tokens2 = anti_join(pride_tokens, stop_words, by = "word")
pride_tokens2 %>%
  count(word, name = "count", sort = TRUE) %>%
  slice(1:20) %>%
  ggplot(aes(y= reorder(word, count), x = count)) +
  geom_col()
```


❖ WORD CLOUDS

A word cloud is a visual representation of text data where words are displayed in varying sizes based on their frequency or importance within a given text. See the following R coding examples that Prof. Sanchez presented in the class.

```r
{r}
pride_tokens2_counts = pride_tokens2 %>%
  count(word, name = "count", sort = TRUE)
pride_tokens2_counts
```

Description: df [6,009 × 2]

word <chr>	co... <int>
elizabeth	597
darcy	373
bennet	294
miss	283
jane	264
bingley	257
time	203
lady	183
sister	180
wickham	162

1–10 of 6,009 rows Previous 1 2 3 4 5 6 ... 100 Next

```r
{r}
wordcloud(words = pride_tokens2_counts$word,
          freq = pride_tokens2_counts$count,
          max.words = 100)
```

```r
wordcloud(words = pride_tokens2_counts$word,
          freq = pride_tokens2_counts$count,
          max.words = 100,
          random.order =  FALSE,
          colors = brewer.pal(n = 8, name = "Dark2"))
```

❖ N-GRAMS

N-grams are sequences of “n” consecutive words or tokens taken from a text, used to model language patterns and predict word frequencies. Two main types of N-grams include “BIGRAMS” and “TRIGRAMS”.

N = 2: "brigrams"

```r
pride_bigrams = unnest_tokens(pride, input = text, output = bigram, token = "ngrams", n = 2)
pride_bigrams
```

Description: df [114,045 × 1]

bigram <chr>
pride and
and prejudice
NA
by jane
jane austen
NA
NA
NA
chapter 1
NA

1–10 of 114,045 rows Previous 1 2 3 4 5 6 ... 100 Next

Search for BIGRAMS that contain "Elizabeth"

```r
pride_bigrams %>%
  mutate(elizabeth = str_detect(bigram, "elizabeth")) %>%
  filter(elizabeth == TRUE) %>%
  count(bigram, name = "count", sort = TRUE) %>%
  slice(1:20) %>%
  ggplot(aes(y = reorder(bigram, count), x = count)) +
  geom_col()
```