Introduction to Tidy Text

SDS 322E
October 15, 2025

Dr. Lydia R. Lucchesi
Department of Statistics and Data Sciences
The University of Texas at Austin

Week 8

	Monday	Tuesday	Wednesday	Thursday	Friday
Lecture	Project 1 Workday		Introduction to Tidy Text		Text Data Analysis
Other	Lab 6	Office Hours (3-5PM)	Office Hours (3-5PM)	Project 1 Due	Pre-Lab 7 Quiz Due

Example text data:

survey responses
song lyrics
newspaper articles
policy documents
reddit threads
social media posts
interview transcripts

Example text data analyses:

word frequency
sentiment analysis
topic modeling

Let's say we have been tasked with analyzing 1,000 customer reviews for a chair from an online furniture company. The boss wants to understand why sales for the chair have been going down. Below are the last three reviews from the **reviews.txt** file containing all 1,000 reviews.

```
8/3/2025: The chair wobbles when you sit in it, so I am going
to return it. Do not recommend.

9/2/2025: It looks beautiful but it is not very sturdy :(

9/26/2025: It wobbles! Do not buy it!
```

1. Tidy the text data

Tidy text format: a table with one-token-per-row

Token: a meaningful unit of text, most often a word, that we are interested in using for further analysis

Tokenization: the process of splitting text into tokens

1. Tidy the text data

```r
library(tidyverse)

# Import customer reviews
reviews <- tibble(review = read_lines("reviews.txt"))
```

```
# A tibble: 6 × 1
  review
  <chr>
1 ""
2 "8/3/2025: The chair wobbles when you sit in it, so I am going to return it. Do not recommend."
3 ""
4 "9/2/2025: It looks beautiful but it is not very sturdy :("
5 ""
6 "9/26/2025: It wobbles! Do not buy it!"
```

1. Tidy the text data

```r
library(tidyverse)

reviews <- tibble(review = read_lines("reviews.txt"))

reviews |>
  # Keep lines that have at least one letter
  filter(str_detect(review, "[a-zA-Z]+")) |>
  # Split date and review into two variables
  separate_wider_delim(cols = c("review"),
                       delim = ":",
                       names = c("date", "review"),
                       too_many = "merge") |>
  # Format date object
  mutate(date = mdy(date))
```

```
# A tibble: 3 × 2
  date       review
  <date>     <chr>
1 2025-08-03 " The chair wobbles when you sit in it, so I am going to return it.…
2 2025-09-02 " It looks beautiful but it is not very sturdy :("
3 2025-09-26 " It wobbles! Do not buy it!"
```

Is this tidy, yet?

1. Tidy the text data

```r
library(tidyverse)
library(tidytext)

reviews <- tibble(review = read_lines("reviews.txt"))

reviews |>
  filter(str_detect(review, "[a-zA-Z]+")) |>
  separate_wider_delim(cols = c("review"),
                       delim = ":",
                       names = c("date", "review"),
                       too_many = "merge") |>
  mutate(date = mdy(date)) |>
  # Tokenization
  unnest_tokens(word, review)
```

1. Tidy the text data

```
# A tibble: 33 × 2
   date       word
   <date>     <chr>
 1 2025-08-03 the
 2 2025-08-03 chair
 3 2025-08-03 wobbles
 4 2025-08-03 when
 5 2025-08-03 you
 6 2025-08-03 sit
 7 2025-08-03 in
 8 2025-08-03 it
 9 2025-08-03 so
10 2025-08-03 i
```

What do you notice about punctuation and capitalization?

```r
reviews |>
  filter(str_detect(review, "[a-zA-Z]+")) |>
  separate_wider_delim(cols = c("review"),
                       delim = ":",
                       names = c("date", "review"),
                       too_many = "merge") |>
  mutate(date = mdy(date)) |>
  unnest_tokens(word, review) |>
  count(word, sort = TRUE)
```

```
# A tibble: 24 × 2
   word          n
   <chr>     <int>
 1 it            6
 2 not           3
 3 do            2
 4 wobbles       2
 5 am            1
 6 beautiful     1
 7 but           1
 8 buy           1
 9 chair         1
10 going         1
```

2. Remove stop words

stop_words {tidytext} R Documentation

Various lexicons for English stop words

Description

English stop words from three lexicons, as a data frame. The snowball and SMART sets are pulled from the tm package. Note that words with non-ASCII characters have been removed.

Usage

```
stop_words
```

Format

A data frame with 1149 rows and 2 variables:

word

An English word

lexicon

The source of the stop word. Either "onix", "SMART", or "snowball"

2. Remove stop words

```
> stop_words
# A tibble: 1,149 × 2
   word        lexicon
   <chr>       <chr>
 1 a           SMART
 2 a's         SMART
 3 able        SMART
 4 about       SMART
 5 above       SMART
 6 according   SMART
 7 accordingly SMART
 8 across      SMART
 9 actually    SMART
10 after       SMART
```

```
stop_words |>
  count(lexicon, sort = TRUE)
```

```
# A tibble: 3 × 2
  lexicon      n
  <chr>    <int>
1 SMART      571
2 onix       404
3 snowball   174
```

2. Remove stop words

If we want to keep all words in **reviews** that do not exist in the SMART lexicon in **stop_words**, what joining function should we use?

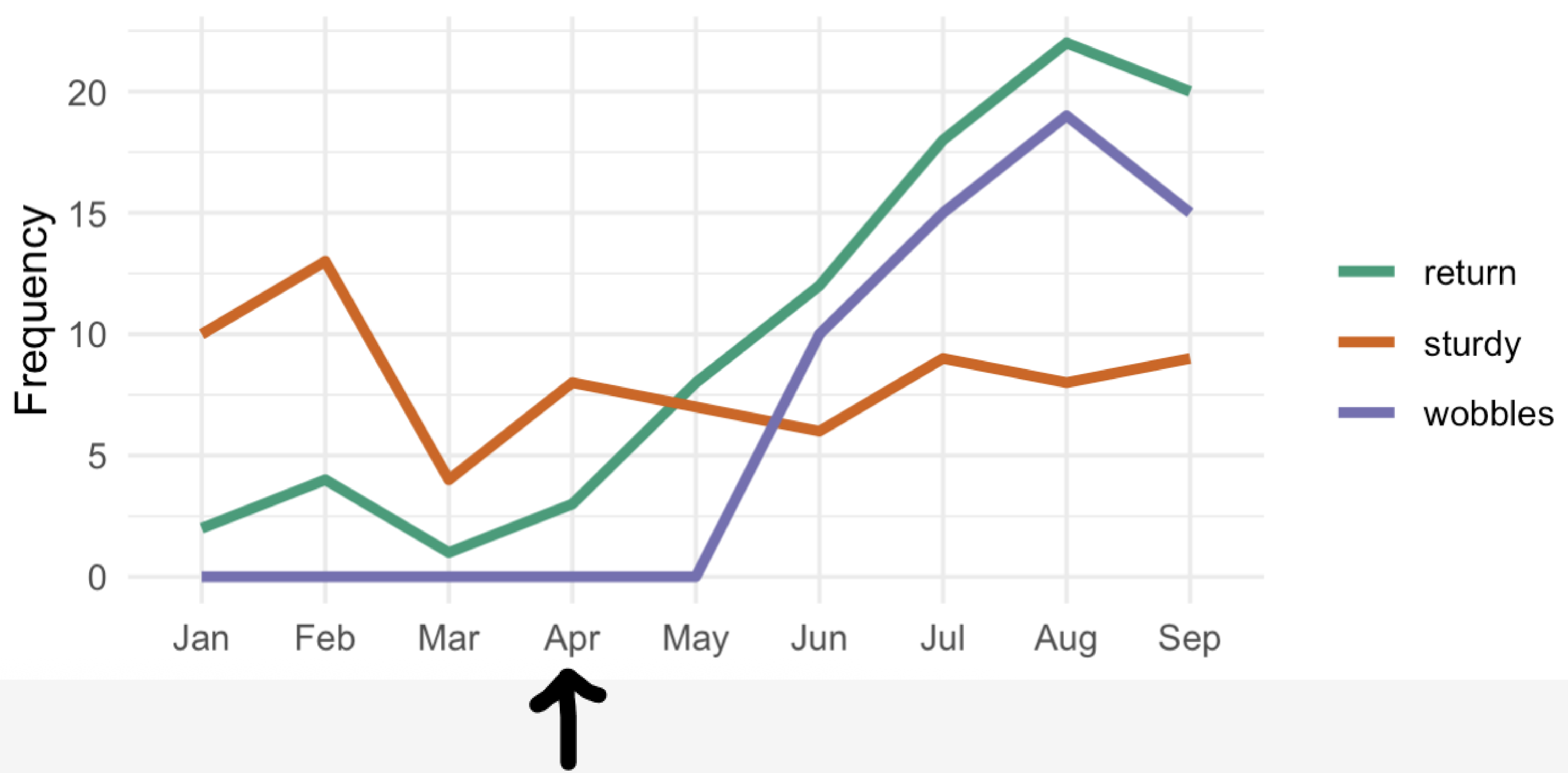
2. Remove stop words

```
smart_stop_words <- stop_words |>
  filter(lexicon == "SMART")

reviews |>
  filter(str_detect(review, "[a-zA-Z]+")) |>
  separate_wider_delim(cols = c("review"),
                       delim = ":",
                       names = c("date", "review"),
                       too_many = "merge") |>
  mutate(date = mdy(date)) |>
  unnest_tokens(word, review) |>
  # Remove stop words
  anti_join(smart_stop_words, join_by(word)) |>
  count(word, sort = TRUE)
```

```
# A tibble: 8 × 2
  word          n
  <chr>     <int>
1 wobbles       2
2 beautiful     1
3 buy           1
4 chair         1
5 recommend     1
6 return        1
7 sit           1
8 sturdy        1
```

Let's say after some initial exploration of the 1,000 reviews, we notice high word counts for *return*, *sturdy*, and *wobbles*. We decide to see if these words appear consistently in the reviews over time.



New machinery installed at the factory

Some other common preprocessing steps in text analysis

stemming – reducing a word to its most basic form

wobbles → wobbl

wobbling → wobbl

wobbly → wobbl

wobbled → wobbl

Some other common preprocessing steps in text analysis

n-gram inclusion – including contiguous sequences of tokens of length *n*

"The return shipping isn't free."

2-gram (bigram):

the return

<u>return shipping</u>

shipping isn't

<u>isn't free</u>

Worksheet background

Journal of Statistical Software

Register Login

Information on Mission Information for Authors Style Guide Volumes About Search

Established in 1996, the Journal of Statistical Software publishes articles on statistical software along with the source code of the software itself and replication code for all empirical results. Furthermore, shorter code snippets are published as well as book reviews and software reviews. All contents are freely available online under open licenses. We aim to present research that demonstrates the joint evolution of computational and statistical methods and facilitates their application in practice. Implementations can use languages and environments like R, Python, Julia, MATLAB, SAS, Stata, C, C++, Fortran, among others. See our mission statement for more details.

Worksheet background

https://doi.org/10.18637/jss.v105.i07

Home / Archives / Vol. 105 (2023) / Issue 7

Expanding Tidy Data Principles to Facilitate Missing Data Exploration, Visualization and Assessment of Imputations

Nicholas Tierney, **Dianne Cook**

Abstract

Despite the large body of research on missing value distributions and imputation, there is comparatively little literature with a focus on how to make it easy to handle, explore, and impute missing values in data. This paper addresses this gap. The new methodology builds upon tidy data principles, with the goal of integrating missing value handling as a key part of data analysis workflows. We define a new data structure, and a suite of new operations. Together, these provide a connected framework for handling, exploring, and imputing missing values. These methods are available in the R package naniar.

Worksheet background

JSS_papers {topicmodels} R Documentation

JSS Papers Dublin Core Metadata

Description

Dublin Core metadata for papers published in the Journal of Statistical Software (JSS) from 1996 until mid-2010.

Usage

```
data("JSS_papers")
```

Format

A list matrix of character vectors, with rows corresponding to papers and the 15 columns giving the respective Dublin Core elements (variables).