Lesson 14

by Lorraine Gaudio

Lesson generated on August 21, 2025

BOISE STATE UNIVERSITY

Contents

1. (maybe a Winter Icon) Welcome Back to R!

In lesson 14, We explore thematic coding and data filtering. Many datasets include text—movie genres, country names, survey answers. You often need to *find* rows that mention certain wordsor exact terms. Today you'll learn to combine str_detect() with dplyr::filter() on both **row names** and **character columns**, using word boundaries (\\b) and alternation (|) so your patterns are precise.

To begin Lesson 14, follow these steps:

1. Open your course project for RStudio
2. Create a new file. From the file types we have used so far, pick which file type you want to use. (File > New File > ???).
3. Type in the code provided in this document as you follow along with the video. Pause the video at anytime to answer assignment questions, dig deeper or add memo notes.

Lesson Overview

By the end of Lesson 14 you will be able to:

1. □ Remember – Recall str_detect(), word boundaries \\b, and |.
2. □ Understand – Explain sapply() and lapply() and how regex boundaries match *whole* words.
3. □ Apply – Use filter() + str_detect() on data□ frame columns *and* on row.names().
4. □ Analyze – Combine boundary tokens and alternation for precise filters.
5. □ Evaluate – Judge which words give meaningful thematic signals.

Keep these goals in mind as you move through each section.

2. 📦 Packages

Install once (if needed): install.packages("dplyr"); install.packages("dslabs"); install.packages("stringr")

```
install.packages("stringr")
```

Load the packages at the start of every session:

```
library(dslabs)  # Data science labs package
library(dplyr)   # Data manipulation package
library(stringr) # For str_detect() function
```

3. Warm‑Up

What is str_detect?

```
?str_detect    # Arguments and return value (logical)
```

□ Check□ in: Why might ignore_case = TRUE be safer for text search?

4. Row name search

Start every plot by piping a data frame into ggplot().

```
View(USArrests)  # View USArrests dataset in RStudio viewer
row.names(USArrests)  # Check row names of USArrests dataset
```

```
USArrests %>%
  filter(str_detect(row.names(USArrests), "^New")) -> New_states
```

 Check in: Which *New* states were found? Use row.names(New_states).

5. Column search

```
View(movielens)
nrow(movielens)
```

```
movielens %>%
  filter(str_detect(title, "\\bLove\\b")) -> Love_films
```

□ Check□ in: Run nrow(Love_films). Surprised at the count?

6. Exact word match

\\b boundary & regex

region column holds strings like “Northeast”, “South”.

```
?stringr::regex
```

 Explanation: Regular expressions (regex) allow you to search, extract, and manipulate text based on patterns rather than exact matches.

```
# Case Insensitive Matching
str_detect("HELLO", regex("hello", ignore_case = TRUE))
```

```
# Word Boundary Matching
murders %>%
  filter(str_detect(region, regex("\\bSouth\\b"))) -> South_states
```

 Explanation: \\b anchors to word boundaries → avoids matching inside longer words, ensuring “South” is matched not “Southampton”.

```
murders %>%
  filter(str_detect(region, regex("\\b(North|South)\\b"))) -> NS_states
```

 Explanation: (A|B) acts like A **OR** B inside a single regex pattern, matches either “North” OR “South” in one operation.

7. Count Matches Quickly

```
match_counts <- c(
  nrow(New_states),
  nrow(Love_films),
  nrow(South_states),
  nrow(NS_states)
)
```

 The GOAL: Build a small data frame of simulated heights.

```
# Add names if needed
names(match_counts) <- c("New_states", "Love_films", "South_states", "NS_states")

# Display the results
match_counts
```

Alternatively, use sapply() to count rows in each data frame

```
?sapply # sapply() applies a function to each element of a list or vector
match_counts <- sapply(list(New_states, Love_films, South_states, NS_states), nrow)
match_counts
```

8. Inspect the Corpus

Load the lesson corpus (Resnik & Elliott passages):

```
load("Lesson_13_Workspace.RData")  # creates Science_Values
```

```
length(Science_Values)   # lines/pages in the article
head(Science_Values)
```

9. Build a Keyword Index

```
?lapply  # lapply() applies a function to each element of a list or vector
```

```
Index_Words <- c("values", "norms", "integrity", "ethics", "bias", "trust",
                 "transparency", "responsibility", "reproducibility", "evidence",
                 "policy", "funding", "conflict")
```

□ We compose the function(w). It takes each Index_Words as input.

```
Index <- lapply(Index_Words, function(w) {
  which(str_detect(Science_Values, regex(paste0("\\b", w, "\\b"),
                                         ignore_case = TRUE)))
})
```

```
names(Index) <- Index_Words
```

□ Check□ in: What does Index contain? Use names(Index) to see the words.

```
# Check the index for "trust"
Index[["trust"]]
```

□ Check□ in: How many lines mention “trust”? Use length(Index[[“trust”]])

10. Preview Context Lines

```
Science_Values[Index[["transparency"]]][1:2]   # first two transparency hits
```

```
# One-liner to get all sentences containing "trust"
trust_sentences <- Science_Values[Index[["trust"]]]
```

11. Practice Space

 Challenge:

1. Create **United_Year2000** with gapminder rows from year == 2000 *and* countries containing "United".
2. Write one regex that captures both “bias” and “biased”. Test on Science_vec and count lines.

 Practice:

a. Build a vector of three new ethics-related words. Or create objects that filter gapminder country names for each word.

b. Write a loop (lapply) that saves a separate data frame per word.

Add at least one check in predicting the total rows you expect.

12. ☒ Assignment

Replace each ____ placeholder (and any TODO comments) with working code or a short written answer. Run each section; be sure the requested objects appear in the Environment. When finished, save **BOTH** this script and your .RData workspace and upload.

When you're done, your workspace should contain SIX new objects: Merc_cars, United, Saint, Republics, Union_2000, Keyword_Counts

12.1 Task 0

☐ Packages

Attach stringr, dplyr, and dslabs and load Lesson_13_Workspace.RData

```
____(____)
____(____)
____(____)
load("Lesson_13_Workspace.RData")
```

12.2 Task 1

Row-name search

Replace each _____ with functions to find all rows in mtcars with "Mazda"

```
Mazda_cars <- mtcars %>%
  _____(_____(row.names(mtcars), "Mazda"))
```

12.3 Task 2

☐ Explain what Task 1's pipeline does.

☐ EXPLANATION: "___"

12.4 Task 3

“United” countries

Use gapminder rows whose country contains "United". Store as **United**.

```
____ <- _____
```

12.5 Task 4

Exact “St” countries

gapminder rows whose country is exactly the word "St". Hint: use \\b before and after. Store as **Saint**.

```
_____ <- _____
```

12.6 Task 5

Republic OR Rep

Use gapminder rows whose country contains "Republic" OR exactly "Rep". Hint: combine \\b with alternation (|). Store as **Republics**.

```
_____ <- _____
```

12.7 Task 6

Year-specific filter + ignore_case

Use gapminder year == 2000 AND country contains"union" (any case). Store as **Union_2000**.

```
_____ <- _____
```

12.8 Task 7

Build a keyword hit count

Using Science_Values (vector of text lines) and the vector keys of "bias", "ethics", "trust" produce **Keyword_Counts**: a named integer vector of line counts per word.

```
keys <- _____


_____ <- _____
```

```
# Quick check
print(Keyword_Counts)
```

12.9 Task 8

- ☐ Reflect:
- ☐ Write a short paragraph reflecting on when is adding \\b around a word **not** a good idea?
- ☐ EXPLANATION: "___"

13. Save and Upload

1. You will be submitting **both** the Quarto Document and the workspace file. The workspace file saves all the objects in your environment that you created in this lesson. You can save the workspace by running the following command in a code chunk of the Quarto Document document:

```
save.image("Assignment14_Workspace.RData")
```

Or you can click the "Save Workspace" button in the Environment pane.

 Always save the R documents before closing.

2. Find the assignment in this week's module in Canvas and upload **both** the RMD and the workspace file.

14. Today you practiced:

- Used word boundaries (\\b) and alternation (|) inside regex strings.
- Combined str_detect() with dplyr::filter() for flexible row selection.
- Searched both data-frame columns and row names.
- Built a keyword index for qualitative reading.