Lesson 10

by Lorraine Gaudio

Lesson generated on August 21, 2025

BOISE STATE UNIVERSITY

Contents

1. ⊠ Ready, Set, R!

In lesson 8 and 9, we explored several functions from dplyr (select(), filter(), mutate(), and ifelse()). This week, we will look at other functions in the dplyr package (group_by and summarize()) that will speed your ability to compare sub-groups and calculate descriptive statistics.

To begin Lesson 10, 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 10 you will be able to:

1. □ Remember – List the roles of group_by() and summarize().
2. □ Understand – Explain how grouping changes the behavior of summary functions.
3. □ Apply – Compute mean, standard deviation, and count for each group in one call.
4. □ Analyze – Combine multiple grouping variables.
5. □ Evaluate –Decide when to overwrite vs. keep original variables.

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

2. Packages

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

Load the packages at the start of every session:

```r
library(dslabs)  # Data science labs package
library(dplyr)   # Data manipulation package
```

3. Overall vs. Grouped Mean

□ Compare summary() with summarize()

```
?summary
data("mtcars")
summary(mtcars)  # Summarizes all columns at once
summary(mtcars$mpg)  # Summarizes just the mpg column
```

```
?summarize
# Overall mean mpg for all cars:
mtcars %>%
  summarize(mean_mpg = mean(mpg))
```

□ Reflect: How informative is a single overall mean? When might you need more detail? Create a memo note, demonstrate learning skill(s) used.

```
mtcars %>%
  summarize(mean_mpg = mean(mpg),
            max_mpg = max(mpg),
            count = n())
```

□ NOTICE: Why is there not piping %>% in the script above?□ Explain your answer in a memo to demonstrate learning skills.

4. The Basics

group_by() changes the behavior of summary functions.

□ The SYNTAX

data %>% group_by(variable1, variable2, ...) %>% some_operation()

□ The GOAL: mean mpg for each cylinder count.

```
cyl_means <- mtcars %>%
  group_by(cyl) %>%
  summarize(mean_mpg = mean(mpg))
```

□ Explanation: group_by(cyl) tells dplyr to treat each unique value of cyl separately and summarize() then runs mean(mpg) *per group*.

```
cyl_means  # print result
```

□ Check□ in: Which cylinder group is most fuel□ efficient?

5. Multiple Statistics

You can create several summary columns in one call, separated by commas.

```
unique(mtcars$cyl)

cyl_stats <- mtcars %>%
  group_by(cyl) %>%
  summarize(mean_mpg = mean(mpg),
            sd_mpg   = sd(mpg),
            n        = n())

cyl_stats
```

□ Notes:

- n() counts rows in each group.
- You *name* each new column using new_name = calculation.

6. Explicit vs. Default

Naming columns makes tables clearer—use it whenever you share results.

```
mtcars %>%
  group_by(cyl) %>%
  summarize(mean(mpg))      # default name is the code itself
```

7. Grouping Multiple Variables

1. group_by one column creates one group for each unique value.

```
unique(mtcars$cyl)

unique(mtcars$am)
```

2. group_by two columns creates one group for each unique combination.

```
mtcars %>%
  group_by(cyl, am) %>%
  summarize(avg_mpg = mean(mpg),
            count = n())
```

- Group 1: cyl = 4, am = 0 (4-cylinder automatic cars)
- Group 2: cyl = 4, am = 1 (4-cylinder manual cars)
- Group 3: cyl = 6, am = 0 (6-cylinder automatic cars)
- Group 4: cyl = 6, am = 1 (6-cylinder manual cars)
- Group 5: cyl = 8, am = 0 (8-cylinder automatic cars)
- Group 6: cyl = 8, am = 1 (8-cylinder manual cars)

Let's do another example using the dataset us_contagious_diseases.

```
data("us_contagious_diseases")
?us_contagious_diseases
```

```
state_disease_mean <- us_contagious_diseases %>%
  group_by(state, disease) %>%     # two grouping vars
  summarize(mean_count = mean(count))

head(state_disease_mean)
```

□ Reflect: How many rows would you expect if each state has 6 diseases?

□ Practice: Pipeline. Fill in the Blanks

For mtcars, compute minimum and maximum horsepower for each transmission type (0 = automatic, 1 = manual). Store as **transmission_hp**.

```
transmission_hp <- mtcars %>%
  ____(__) %>%
  ____(min_hp = min(____),
            max_hp = max(____))

transmission_hp
```

□ Explore and Play: Try changing horse power to miles per gallon or weight and rerun.

Create a memo note, demonstrate learning skill(s) used during □ Reflect, □ Practice, and □ Explore and Play.

8. ▯ Practice Space

□ Practice: Replace the blanks (___) and run.

You’ll use the heights dataset and use the functions group_by, summarize, mean, sd, n

```
data("heights")
?heights
```

```
heights %>%
  ___(sex) %>%
  ___(mean_height = ___(height),
          sd_height   = ___(height),
          n           = ___()) -> height_descriptives
View(height_descriptives)
```

9. ▯ 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.

9.1 Task 1

□ Library it up!

Make sure there is script in your document to load dplyr and dslabs packages so their functions / datasets load.

9.2 Task 2

Single grouping

□ Using stars, compute mean temperature **AND** mean magnitude for each type. Name the result **twinkle_twinkle**.

```
____ <- ______
```

```
# Quick check
head(twinkle_twinkle)
```

9.3 Task 3

Single grouping + three stats

□ Pick ONE numeric variable in olive (not region, area), group by region, return mean, sd, and n. Store as **Olive_Garden**.

```
____ <- ______
```

In a short comment, note the region with the highest mean value.

□ Comment: “___”

9.4 Task 4

Reinforce pattern

□ From heights, group_by sex and summarize mean_height, sd_height, n . Store as **height_descriptives**.

```
____ <- _____
```

9.5 Task 5

Multiple grouping

□ From us_contagious_diseases, group_by state AND disease then summarize the mean count. Keep only the three diseases with the highest national mean counts. Store as **Infectious_Burden**.

```
____ <- _____
```

9.6 Task 6

Create **Life_Decade**. Make a decade column (hint: floor(year/10)*10) and group_by continent, and decade. The summarize avg_lifeExp = mean(life_expectancy).

```
____ <- _____
```

□ Reflect

□ Write a short paragraph reflecting why we need to do group_by() come BEFORE summarize()?

□ EXPLANATION: "___"

10. 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:

```r
save.image("Assignment10_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.

11. Today you practiced:

- Compared overall vs. grouped summaries.
- Used group_by() + summarize() to calculate descriptive statistics.
- Named summary columns for clarity.
- Counted group sizes with n().
- Grouped by multiple variables for richer insights.

□ Practice writing one-line summaries on your own data and the syntax will quickly become second nature.