Lesson 13

by Lorraine Gaudio

Lesson generated on August 21, 2025

BOISE STATE UNIVERSITY

Contents

1. (maybe a dice icon?) Welcome Back to R!

In lesson 12, we did simple data visualizations using R's built in function hist(). Today, we use the package ggplot2 to turn data frames into clear, publication quality plots. Specificaly, we'll look at the grammar of graphics.

To begin Lesson 13, 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 13 you will be able to:

1. Remember – State the steps: data aes() geom().
2. Understand – Describe how aes() maps variables to plot elements.
3. Apply – Build scatterplots, histograms, and boxplots.
4. Analyze – Use color or fill aesthetics to compare groups.
5. Evaluate – Choose suitable geoms and themes for a question.

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("ggplot2")
```

Load the packages at the start of every session:

```
library(dslabs)  # Data science labs package
library(dplyr)   # Data manipulation package
library(ggplot2) # Data visualization
```

3. Warm-Up

What is ggplot2?

```
?ggplot   # skim the Details section
```

- The SYNTAX (basic/general): ggplot(data = NULL, mapping = aes(), ..., environment = parent.frame())
- NOTICE: Which argument supplies mapping information?
- Explanation: In ggplot2, mapping refers to connecting your data variables to visual properties (aesthetics) of the plot. It's how you tell ggplot which column in your data should control which aspect of the visualization.

4. The Blank Canvas

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

df %>% ggplot()

OR without piping

ggplot(data = df)

```
divorce_margarine %>%
  ggplot()
```

The result is empty — we still need aesthetic mappings and a geom.

5. Mapping Aesthetics

```
?aes
```

- The SYNTAX (basic): ggplot(data = df, mapping = aes(x, y, other aesthetics))

OR with piping

df %>% ggplot(mapping = aes(x, y, other aesthetics))

- The GOAL: Map divorce rate to x and margarine consumption to y.

```
divorce_margarine %>%
  ggplot(aes(x = divorce_rate_maine, y = margarine_consumption_per_capita))
```

6. Scatterplot

geom_point() is a geometric object (or "geom") in ggplot2 that creates a scatter plot by placing points at the specified x and y coordinates.

```
divorce_margarine %>%
  ggplot(aes(divorce_rate_maine, margarine_consumption_per_capita)) +
  geom_point()
```

 Explanation: geom_point() understands these aesthetics:

- x, y: Position (required)
- alpha: Transparency
- color: Point color
- fill: Fill color (for shapes with fill)
- shape: Point shape (0-25)
- size: Point size
- stroke: Border thickness

```
divorce_margarine %>%
  ggplot(aes(x = divorce_rate_maine,
             y = margarine_consumption_per_capita)) +
  geom_point(aes(color = year,          # Color points by year
                 size = divorce_rate_maine, # Size points by divorce rate
                 alpha = margarine_consumption_per_capita), # Transparency by consumption
             shape = 16)
```

7. Histogram

```
?geom_histogram
```

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

```
set.seed(2025)
Tallness <- data.frame(
  Country = c(rep("USA", 1000), rep("CAN", 1000)),
  Height  = c(rnorm(1000, 67, 2.5), rnorm(1000, 66, 1.5))
)
```

 NOTICE: that the first country, "USA" will have a mean height of 67 inches

```
Tallness %>%
  ggplot(aes(Height)) +              # only x for histograms
  geom_histogram(fill = "steelblue", color = "white")
```

geom_histogram() understands these aesthetics:

- x: Position (required)
- fill: Fill color
- color: Border color
- binwidth: Width of each bin (default is 1)
- position: Position adjustment (e.g., "identity", "dodge", "fill")
- alpha: Transparency
- size: Border thickness
- weight: Weight for each observation (optional)
- linetype: Line type for borders (optional)

```
Tallness %>%
  ggplot(aes(Height)) +              # only x for histograms
  geom_histogram(fill = "steelblue",
                 color = "white",
                 binwidth = 0.5)       # Set bin width to 0.5
```

 Explore and Play: Change binwidth = 0.25 inside geom_histogram(). How does the distribution look?

 Break Things! Test out your hand at adding aesthetics geom_histogram() understands.

8. Group Comparison

 The GOAL: Overlay two histograms by mapping fill to Country.

```
View(Tallness)  # Recall that there are two countries: USA and CAN
```

```
Tallness %>%
  ggplot(aes(Height, fill = Country)) +
  geom_histogram(position = "identity", alpha = 0.6)
```

 Explanation:

1. position = "identity" lets bars overlap
2. alpha controls transparency: Sets the transparency of all bars to 0.6 (0 == transparent, 1 == opaque)

9. Boxplot

```
?geom_boxplot
```

```
Tallness %>%
  ggplot(aes(Country, Height, fill = Country)) +
  geom_boxplot(show.legend = FALSE)
```

□ Check□ in: Which country shows greater height variability?

10. Labels and Themes

```
Tallness %>%
  ggplot(aes(Country, Height, fill = Country)) +
  geom_boxplot(show.legend = FALSE) +
  xlab("North American Country") +
  ylab("Height (inches)") +
  ggplot2::theme_minimal()
```

 Explanation:

- xlab() andylab()‘ set axis labels.
- ggplot2::theme_minimal() applies a clean, minimal theme to the plot.
- ggplot2:: prefix explicitly specifies that you’re using the theme_minimal() function from the ggplot2 package.

This is useful when multiple packages have functions with the same name.

11. ✍ Practice Space

 Practice & Explore and Play:

1. Build a density plot of the heights dataset (dslabs). Shade by sex.
2. Try geom_smooth() on divorce_margarine to add a trend line.
3. Use iris data to create side by side boxplots of Sepal.Width by Species and change the theme.

Add your code below and include at least one check in to predict an outcome.

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 FOUR new objects: Pretty_Flowers, Petal_Spread, CO2_Graph, Tall_Density

12.1 Task 0

□ Packages

Attach ggplot2, dplyr, and dslabs

```
____(____)
____(____)
____(____)
```

12.2 Task 1

□ Quick Recall

List the three core ggplot2 steps

□ EXPLANATION: "___"

□ EXPLANATION: "___"

□ EXPLANATION: "___"

12.3 Task 2

Scatterplot

Build **Pretty_Flowers**: iris data □ Sepal.Length vs Petal.Length, color points by Species, size = 3, add labels for both axes.

Hint: use ggplot2 pipe

```
_____ <- _____

print(Pretty_Flowers)
```

12.4 Task 3

Boxplot: group comparison

Create **Petal_Spread**: iris Petal.Width by Species as green boxplots (no legend). Use fill = Species, then show.legend = FALSE.

```
____ <- _____
```

12.5 Task 4

Density plot: split + transparency

Use CO2 data to make **CO2_Graph**:

- x = uptake
- fill by Type
- alpha = 0.5
- title = "CO2 Uptake by Plant Type"

```
_____ <- _____ %>%
 ggplot(_____(uptake, fill = _____)) +
 _____ +
 _____(title = "CO2 Uptake by Plant Type")
```

12.6 Task 5

 Histogram vs. Density

Build **Tall_Density** on heights (dslabs):

- map height only (x)
- add geom_histogram(binwidth = 1, fill = "grey70", color = "white")
- add geom_density(color = "red", size = 1)
- x-lab "Height (inches)"

```
Tall_Density <- heights %>%
 ggplot(_____(height)) +
 _____ +
 _____ +
 _____("Height (inches)")
```

12.7 Task 6

☐ Reflect:

☐ Write a short paragraph reflecting on when would a density curve be more useful than a histogram?

☐ 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("Assignment13_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:

- Created a blank canvas with ggplot().
- Mapped variables inside aes().
- Added geoms for scatter, histogram, and boxplot.
- Used color / fill to compare groups.
- Applied labels and a minimal theme for clarity.