Lesson 4

by Lorraine Gaudio

Lesson generated on September 15, 2025

BOISE STATE UNIVERSITY

Contents

1. Welcome back to R!

In lesson one, two and three you learned about objects, vectors, and functions. Now you'll learn more about handling missing data in R.

To begin Lesson 4, follow these steps:

1. Open your course project for RStudio
2. Create a new file. Today, let's try "R Notebook" (File > New File > R Notebook).
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 4 you will be able to:

1. Remember – Define NA vs. NaN in R.
2. Understand – Use anyNA() to ask "Is *anything* missing here?".
3. Apply – Pinpoint missing values with is.na() and which().
4. Analyze – Count and summarize how much data is missing.
5. Evaluate – Choose a simple strategy: keep, drop, or impute.

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

2. Quick Warm-up

Before diving into new territory, let's flex a skill you met last time: opening R help pages. Being comfortable with documentation means you can teach yourself any function you meet in the future.

Create an R script chunk by clicking the green C on the top right of the script window. Then type in the following code and run.

```
?gsub
```

- After viewing the page, close it or leave it open for reference.
- Nice! Summon help whenever you need it.

3. NA vs NaN

What & Why Both are special markers, not ordinary numbers.

- NA ("Not Available") signals *missing* information: the value simply was not recorded.
- NaN ("Not a Number") signals an *undefined* numeric result, such as 0 divided by 0.

```
0 / 0      # Produces NaN – undefined arithmetic

NA         # Built-in constant representing a missing value
```

 NOTICE: R prints NA and NaN in the Console so you can spot them quickly. Even though they look similar, treat them differently: NA means "value absent"; NaN means "math error".

4. Detecting Missing Values

What & Why

- anyNA(x) returns TRUE if *any* element in x is NA or NaN.
- A quick yes/no check prevents surprises later in your workflow.

Type the following code in a new code chunk.

```
Vector_NA <- c(3, 7, NA, 12)  # Our sample data – one value is missing

anyNA(Vector_NA)              # TRUE means "Something is missing!"
```

 Reflect: If you saw FALSE, you could relax; the vector has no gaps. Seeing TRUE tells you to investigate further. What are causes for NA in real data and how might it impact your research? How is data that is not available different from not possible or Null?

5. Locating Missing Values

What & Why

- is.na(x) returns a logical vector: TRUE wherever x is NA or NaN.
- which(logical_vec) converts TRUE positions into numeric indices (handy for slicing or replacing values).
- Precise locations let you decide row-by-row what to do.

Type the following into a R code chunk then run.

```
is.na(Vector_NA)          # Notice that it is TRUE at position 3

which(is.na(Vector_NA))   # Returns index 3 directly

mean(values, trim = 0.2, na.rm = TRUE)  # drop 20% from each end first
```

```
set.seed(100)

LargeVec <- sample(c(1:10, rep(NA, 10)))  # 20 vals, ~10 NA

which(is.na(LargeVec))                    #  Where are the gaps?
```

 Link: We used the c() function in lesson 2 and the sample() function in lesson 3. Here we have nested c() inside of sample(). Explain how this nesting is acts on each function. Create a memo note, demonstrate learning skill(s) used.

6. Summarizing “Missingness”

 WHY COUNT? Knowing *how much* data is missing guides your next step:

- A single NA might be harmless.
- 50% missing will bias results and needs attention.

Type the following R script in your document script and run.

```
# COMMON TOOLS
sum(is.na(LargeVec))

mean(is.na(LargeVec))
```

 Interpret: What does sum() tell us?

 Reflect: How does mean() calculate TRUE and FALSE? How do you interpret the fraction?

Create a memo note, demonstrate learning skill(s) used to answer Interpret and Reflect to potentially earn leaning skills points.

7. Action Strategies

Now that you can *detect* and *locate* gaps, what will you *do* about them? Here are two beginner friendly options. (Data scientists debate this a lot!)

1. Remove rows containing NA ("complete case analysis")

```
clean_vec <- LargeVec[!is.na(LargeVec)]   # Keeps only observed values
```

Link: The brackets are using indexing and the logical operator ! before the function is.na. Explain how this nesting acts on each function. (Hint: Review lesson 2.) Create a memo note, demonstrate learning skill(s) used.

2. Impute: replace NA with the mean of present values

```
imputed <- LargeVec                    #  Make a copy to preserve original

imputed[is.na(imputed)] <- mean(imputed, na.rm = TRUE)
```

Look deeper: What are other ways data analysts can handle NA? What R code demonstrates these alternative methods?

8. ☒ Assignment

Now it's your turn to practice creating and using vector objects. Follow the tasks below to complete part of the **technical skill practice assignment**.

1. Work through each task in order. Replace the ___ placeholder with your code or short written answer.
2. Run each completed line to be sure no errors appear and objects show in the Environment.
3. When finished, save your workspace and submit this R Notebook file (RMD) plus the .RData file.

8.1 Task 0

Theme setup: Choose ONE theme (□ Football or □ River) to use for Tasks 1–5.

```
# Boise State Football — Crowd Noise by Game Minute
set.seed(42)
BlueTurf_Noise_60 <- sample(c(sample(1:5, 36, replace = TRUE), rep(NA, 24)))
Did_Reading_Register <- sample(c(1:60, NA), 20)
```

```
# Sustainability — Boise River Weekly Water Quality
River_Quality_52 <- sample(c(sample(1:5, 31, replace = TRUE), rep(NA, 21)))
Did_Sample_Arrive <- sample(c(1:52, NA), 20)
```

Theme Key (what to use later)

- If you choose Football:
- DATA vector: BlueTurf_Noise_60
- ID vector: Did_Reading_Register
- If you choose River:
- DATA vector: River_Quality_52
- ID vector: Did_Sample_Arrive

In the tasks below, when you see **DATA vector**, use the one that matches your theme. When you see **ID vector**, use the one that matches your theme.

8.2 Task 1

 Comment: In one statement each, write your own definition of NA and NaN.

NA: ___

NaN: ___

8.3 Task 2

 Use anyNA() to test your **ID vector** for missing values. Write one line of code that returns TRUE/FALSE and saves it in Made_NA.

```
Made_NA <- "__"
```

8.4 Task 3

 Locate Missing & Complete Values:

1. Get the indices of missing values in your **ID vector**.

```
Missing_Idx <- "__"
```

2. Get the indices of NON-missing values in your **DATA vector**.

```
Keep_Idx <- "__"
```

8.5 Task 4

 Quantify "Missingness"

For each vector (your **ID vector** and your **DATA vector**), calculate the total number and proportion of missing values.

```
Total_Missing_IDs <- "__"

Prop_Missing_IDs  <- "__"

Total_Missing_Data  <- "__"

Prop_Missing_Data   <- "__"
```

Hints:

- sum(is.na(x)) counts missing values.
- mean(is.na(x)) gives the proportion missing.

8.6 Task 5

 Compare Handling Strategies

Focus on your **DATA vector**. Create two cleaned versions:

1. vec_removed – drop all NA values.
2. vec_imputed – replace NA with the mean of observed values.

```
vec_removed <- "__"

vec_imputed <- "__"
```

3. Compute and compare the mean of each cleaned vector.

```
Mean_Removed <- "__"

Mean_Imputed <- "__"
```

4. Comment: Which strategy would you choose if ~40% of values are missing, and why? Consider how dropping NAs changes your sample vs. how mean-imputation might bias results for this theme.

Choice & reason: ___

9. Save and Upload

1. You will be submitting **both** the R Notebook (RMD) 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 (RData) by clicking the "Save Workspace" button in the Environment pane.

 Always save before closing.

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

10. Today you practiced:

- Discovered the difference between NA and NaN.
- Ran anyNA() to test for “missingness” quickly.
- Used is.na() + which() to locate gaps precisely.
- Counted and summarized missing data to gauge its impact.
- Practiced two basic strategies: dropping or mean imputing values.

 Excellent progress! Missing data is no longer a mystery.