Lesson 4

by Lorraine Gaudio

Lesson generated on August 16, 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 Markdown" (File > New File > R Markdown).
3. Type in the code provide 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(). Unpack how this nesting is acts on each function. Create a memo note, demonstrate learning skill(s) used.

6. Counting & Summarizing “Missingness”

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

- A single NA might be harmless.
- 40% missing could 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?

 Think: 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 Think 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. Unpack how this nesting is 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)
```

Reflect: Removal is simple but reduces sample size. Mean imputation keeps size but may shrink variability. Between these two options, which do you think is best practice and why?

Look deeper: What are other ways you can handle NA? What is the R code to act on your data set?

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 Markdown file (RMD) plus the .RData file.

8.1 Task 0

Run everything in this block first.

```
set.seed(3)

Ninety_Nine_Red_NAs <- sample(c(1:5, rep(NA, 99)))

Did_NA_Make_It      <- sample(c(1:99, NA), 20)
```

8.2 Task 1

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

NA: ___

NaN: ___

8.3 Task 2

☐ Detect "Missingness" Use anyNA() to test Did_NA_Make_It for missing values.

Made_NA <- ___

8.4 Task 3

 Locate Missing & Complete Values:

1. Get the indices of missing values in Did_NA_Make_It.

Missing_Idx <- ___

2. Get the indices of NON missing values in Ninety_Nine_Red_NAs.

Keep_Idx <- ___

8.5 Task 4

 Quantify "Missingness"

For each vector, calculate the total number and proportion of missing values.

Total_Missing_DidNA <- ___

Prop_Missing_DidNA <- ___

Total_Missing_99NA <- ___

Prop_Missing_99NA <- ___

8.6 Task 5

 Compare Handling Strategies

Focus on Ninety_Nine_Red_NAs. 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: State which strategy you would choose for a data set where 40% of the values are missing and why.

Choice & reason: ___

9. Save and Upload

1. You will be submitting **both** the R Markdown 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 R Markdown document:

```
save.image("Assignment3_Workspace.RData")
```

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

Always save the R documents before closing. If you don't, you will lose your work.

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