EDUCATIONAL FORUM

Creating Understandable Data Visualizations in R

Soundarya Soundararajan[1], Valliappan Muthu[2]

Received on: 01 January 2026; *Accepted on:* 25 April 2026; *Published on:* xxxx

Abstract

Data visualization is central to scientific communication, yet many clinicians and early-career researchers struggle to create clear and interpretable figures. This article presents a beginner-friendly approach to building charts in R using the grammar of graphics framework. Every element of a chart is a choice, and the default is rarely the best one. Using a simulated clinical dataset of systolic blood pressure across treatment groups, we demonstrate how plots are constructed explicitly by combining data, aesthetic mappings, and geometric representations. Through step-by-step examples, we show how deliberate layering of axis labels, titles, color palettes, and themes transforms a bare-minimum plot into a publication-ready figure. By emphasizing deliberate decision-making over reliance on software defaults, this work aims to lower barriers to effective data visualization and support transparent scientific reporting.

Keywords: Data visualization, ggplot, Grammar of graphics, R programming, Reproducibility.

Journal of Postgraduate Medicine, Education and Research (2026): 10.5005/jp-journals-10028-1816

Introduction

In the inaugural chapter of this series,[1] we concluded with a bonus figure that marked our first step into R-based visualization. In this chapter, we pick up from that point and begin building practical skills that allow beginners to create simple, clean, and reproducible charts. The focus is on starting small: one chart, one decision at a time.

Several data visualization principles exist, and numerous papers lay out these principles for different disciplines.[2,3] Rather than cataloging them all, we focus on one idea that will anchor everything in this chapter: building charts mindfully, layer by layer.[4,5]

The Problem We Are All Familiar With

Most of us have been there. A chart squinted at on a projected slide, the axis numbers too small to read from the back of the room. A default color palette was accepted without a second thought, the ones the software chose, not the ones the data needed. A figure packed with elements that the comparison we wanted to show is buried somewhere in the middle.

These are not failures of effort. They are failures of awareness. Nobody teaches us when we first start making charts that every element is a choice, and that the default is rarely the best one.

This chapter is about building that awareness, one layer at a time. We will not try to fix everything at once. We will start with the bare minimum that makes a chart a chart and add only what earns its place.

Setting Up

We use a simulated clinical dataset of 90 patients across three treatment groups, with systolic blood pressure (SBP), age, and BMI as variables. This mirrors data structures that health researchers encounter routinely. The dataset and the code used to generate it are available in the supplementary materials at https://github.com/soundarya24/ROAR-datasets/tree/main.

```
library(tidyverse) # Loads ggplot2, our main tool for charts
# Load the dataset directly from GitHub
health_data <- read_csv("https://raw.githubusercontent.com/soundarya24/ROAR-datasets/refs/heads/main/health_data.csv") |>
mutate(treatment = factor(treatment,
levels = c("Control," "Drug A," "Drug B")))
```

[1]Department of Health Sciences, Regional Occupational Health Centre, ICMR-National Institute of Occupational Health Research, Bengaluru, Karnataka, India

[2]Department of Pulmonary Medicine, Postgraduate Institute of Medical Education and Research, Chandigarh, India

Corresponding Author: Soundarya Soundararajan, Department of Health Sciences, Regional Occupational Health Centre, ICMR-National Institute of Occupational Health Research, Bengaluru, Karnataka, India, e-mail: soundarya24@gmail.com

How to cite this article: Soundararajan S, Muthu V. Creating Understandable Data Visualizations in R. J Postgrad Med Edu Res 2026;https://doi.org/10.5005/jp-journals-10028-1816.

Source of support: Nil

Conflict of interest: Dr Valliappan Muthu is associated with the Editorial Board member of this journal, and this manuscript was subjected to this journal's standard review procedures, with this peer review handled independently of this editorial board member and his research group.

The Very Basics: What Makes a Chart a Chart

Every ggplot chart needs exactly three things[6,7]:

1. Data: what you are plotting.
2. Aesthetics: which variables map to x and y.
3. Geometry: the type of chart.

The recipe is:

```
data |> ggplot(aes(x =?, y =?)) + geom_something()
```

The |> symbol, called a "pipe," means "and then".[1] Read it as: take this data, and then plot it, and then draw it as a boxplot. That is all. Everything else, labels, titles, colors, themes, comes later, added one at a time with a + sign.

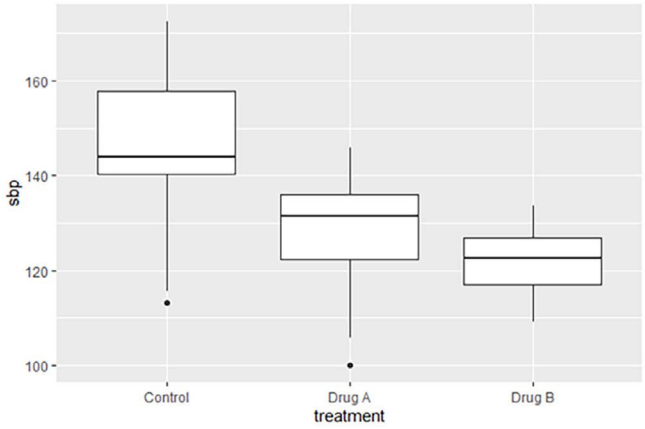

Fig. 1: Bare minimum plot showing data, aesthetics, and geometry, with no additional elements

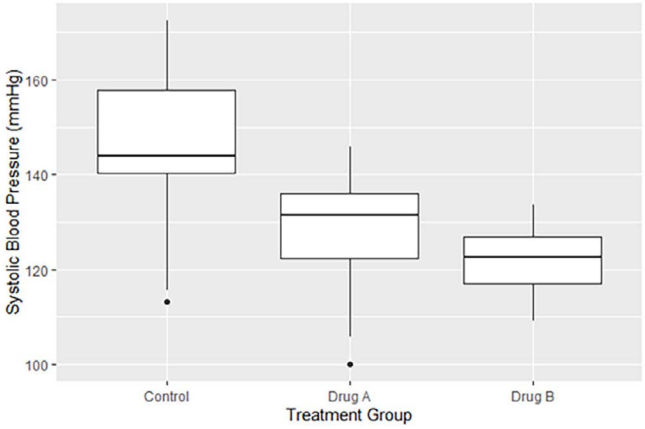

Fig. 2: Same chart with axis labels added using labs(), showing what the axes represent

Figure 1 shows what this looks like with nothing added yet.

```
health _ data |>
ggplot(aes(x = treatment, y = sbp)) +
#This simply means X axis will have
"treatment" and y will have "sbp" from the
"health _ data"
geom _ boxplot() #this will produce a box
plot…similarly we can write geom _ violin()
for violin
plot, geom _ histogram() for histograms
```

Statements written after "#" symbol do not run as commands; instead, they are notes to enable us to understand for future reference or to explain to others.

This is a complete working chart. It is also not yet ready to show anyone. The axis labels are variable names from the dataset, not descriptions that a reader can understand. We fix that next.

Adding Axis Labels

Axis labels tell the reader what the numbers actually mean. Without them, a chart cannot stand alone.

In Figure 2, we add them with labs(), naming each element directly.

```
health _ data |>
ggplot(aes(x = treatment, y = sbp)) +
```

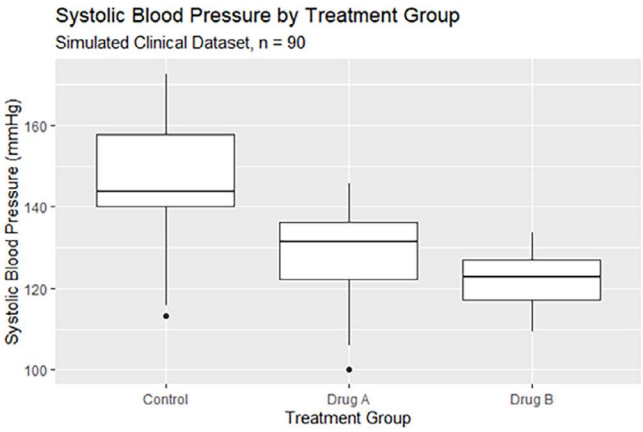

Fig. 3: Chart with title and subtitle added, allowing it to stand alone without surrounding explanatory text

```
geom _ boxplot() +
labs(
x = "Treatment Group,"
y = "Systolic Blood Pressure (mmHg)"
)#labs adds the desired labels
```

One addition, one + sign. The chart is already more readable.

Adding a Title and Subtitle

A title tells the reader what the chart is about. A subtitle can add context, the dataset, the time period, or a note about units. Both are optional, but a chart intended for a paper or a presentation almost always needs at least a title.

Figure 3 demonstrates this possibility.

```
health _ data |>
ggplot(aes(x = treatment, y = sbp)) +
geom _ boxplot() +
labs(
x = "Treatment Group,"
y = "Systolic Blood Pressure (mmHg),"
title = "Systolic Blood Pressure by Treatment
Group," subtitle = "Simulated Clinical Dataset,
n =90"
)
```

Choosing a Theme

The default grey background in ggplot2 is fine for exploring data on your own screen. It is not ideal for a paper or a presentation. theme_minimal() removes the grey and gives a cleaner look.

The base_size argument inside theme_minimal() scales all text elements proportionally, title, subtitle, axis labels, and tick marks, so you only need to change one number when preparing a chart for a different medium. Use base_ size =10 or 12 for papers, and base_ size =16 or larger for presentations. There is a natural hierarchy at work: title text is always the largest, followed by subtitle, then axis labels, then tick labels. base_size respects this hierarchy automatically (Fig. 4).

```
health _ data |>
ggplot(aes(x = treatment, y = sbp)) +
geom _ boxplot() +
labs(
x = "Treatment Group,"
y = "Systolic Blood Pressure (mmHg),"
title = "Systolic Blood Pressure by Treatment
```

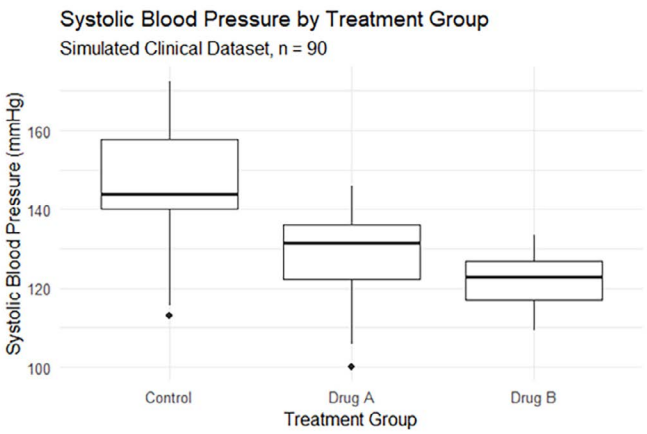

Fig. 4: Clean theme applied using theme_minimal(), with the base_size argument controlling all text proportionally

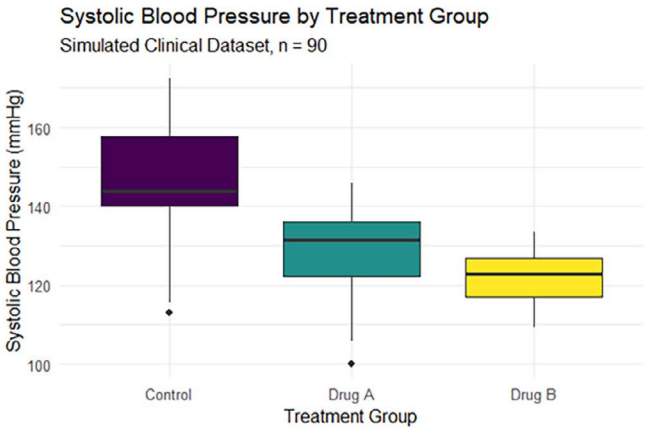

Fig. 5: Color added deliberately using scale_fill_viridis_d(), with the fill aesthetic mapping the treatment group to color

```
Group,"
subtitle = "Simulated Clinical Dataset, n =90"
) +
theme _ minimal(base _ size = 12) # Change to
16 for presentations
```

Choosing Colors Mindfully

Default colors are a choice made by someone else, for no particular reason related to your data. Making a deliberate color choice takes one line. scale_fill_viridis_d() is a good starting point: it is perceptually uniform, widely used in scientific publishing, and works across different display and print conditions.

Note the difference between color and fill in ggplot2. The color aesthetic controls outlines and borders, such as the lines around a boxplot or the border of a point, whereas fill controls the interior of shapes, such as the inside of a box or the body of a bar. In Figure 5, for boxplots, we map fill to the grouping variable and use scale_fill_viridis_d().

```
health _ data |>
ggplot(aes(x = treatment, y = sbp, fill =
treatment)) +
geom _ boxplot() +
scale _ fill _ viridis _ d() +
labs(
```

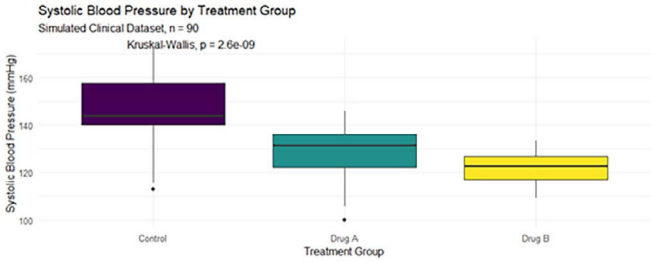

Fig. 6: Complete clinical figure built using the same grammar as Figure 1, showing age vs systolic blood pressure by treatment group, with a Kruskal–Wallis *p*-value

```
x = "Treatment Group,"
y = "Systolic Blood Pressure (mmHg),"
title = "Systolic Blood Pressure by Treatment
Group,"
subtitle = "Simulated Clinical Dataset, n
=90") +
theme _ minimal(base _ size = 12) +
theme(legend.position = "none") # this decides
where the legend goes, but since our groups
are already labeled, we can choose not to
have separate legends; the other options are
"top," "bottom" to place them where we want.
```

Compare Figure 5 with Figure 1. The data is identical. Every difference is a choice that was made deliberately, one addition at a time.

Putting It All Together

Figure 6 shows what becomes possible once the grammar of graphics is familiar. We move to a scatterplot of age versus SBP, add a fitted regression line per group, and separate each group into its own panel using facet_wrap(). A statistical comparison is added with ggpubr::stat_compare_means() (Fig. 6).

The only new functions here are geom_point(), geom_smooth(), facet_wrap(), and stat_compare_means(). The grammar, data, aesthetics, geometry, labels, theme, are exactly the same as Figure 1.

```
health _ data |>
ggplot(aes(x = treatment, y = sbp, fill =
treatment)) +
geom _ boxplot() +
scale _ fill _ viridis _ d() + ggpubr::stat _
compare _ means()+ #compare means
labs(
x = "Treatment Group,"
y = "Systolic Blood Pressure (mmHg),"
title = "Systolic Blood Pressure by Treatment
Group," subtitle = "Simulated Clinical
Dataset, n =90"
) +
theme _ minimal(base _ size = 12) +
theme(legend.position = "none") # x-axis
already labels the groups
```

This is only to show the destination. The path there is the six steps we have just walked through.

Exercise

Using health_data:

- Create a boxplot of BMI by treatment group.
- Add axis labels and a title.
- Apply theme_minimal() with base_ size =16 for a presentation.
- Add a deliberate color choice using scale_fill_viridis_d().

Remember the structure: data → aesthetics → geometry → labels → theme.

Pro Tips

```
# Always look at your data before plotting
summary(health _ data)
glimpse(health _ data)
# Save your plot at publication quality
ggsave("figure1.png," width = 8, height = 6,
dpi = 300)
# Build a reusable theme for consistency
across all figures in a paper
my _ theme <- theme _ minimal(base _ size = 12) +
theme(
plot.title = element _ text(hjust = 0.5),
legend.position = "bottom"
)
# Add it to any chart:
# your _ plot + my _ theme
```

Resources for Further Learning

- Principles of effective data visualization. https://pubmed.ncbi.nlm.nih.gov/33336199/
- Data visualization principles. https://rafalab.dfci.harvard.edu/dsbook/data-visualization-principles.html
- Understanding data visualization design practice. https://pubmed.ncbi.nlm.nih.gov/34596554/

Summary

- A chart needs three things: data, aesthetics, and geometry. Everything else is an addition.
- Each addition should earn its place: axis labels, title, theme, color, one at a time, one + at a time.
- Defaults are someone else's choice. Making deliberate decisions at each step is what separates a chart that communicates from one that merely displays.
- Happy visualizing!

Orcid

Soundarya Soundararajan https://orcid.org/0000-0003-0383-8389
Valliappan Muthu https://orcid.org/0000-0003-0410-8468

References

1. Muthu V, Soundararajan S. Reproducibility and objectivity in academic research with R. J Postgrad Med Edu Res 2025;59(4):192–198. DOI: 10.5005/jp-journals-10028-1741
2. Munzner T. Visualization Analysis and Design. New York: CRC Press; 2015.
3. Evergreen SDH. Effective Data Visualization: The Right Chart for the Right Data, 2nd edition. Thousand Oaks, CA: Sage; 2020.
4. Cairo A. The Truthful Art: Data, Charts, and Maps for Communication. Berkeley, CA: New Riders; 2016.
5. Schwabish JA. Better Data Visualizations: A Guide for Scholars, Researchers, and Wonks. New York: Columbia University Press; 2021.
6. Wickham H. ggplot2. Cham: Springer International Publishing; 2016.
7. Wilkinson L. The Grammar of Graphics, 2nd edition. Springer; 2005.