

# Constructing a reproducible blog post using zzcollab tools

Ronald G. Thomas

2025-01-01



Figure 1: Engaging hero image that introduces your topic visually

*Photo caption with attribution if needed. This image sets the visual tone for your entire post.*

## Introduction

I didn't really know much about [topic] until I [encountered situation/tried to implement it/needed it for project]. Like many data scientists, I thought [initial misconception or assumption]. Turns out, [what you actually discovered].

[Brief context: Why did you need this? What problem were you trying to solve? Keep it personal and specific.]

Here's what I set out to understand:

## Motivations

**Why explore [topic]?** - [Personal reason 1: specific problem you faced] - [Practical need 2: gap in your workflow] - [Learning goal 3: skill you wanted to develop] - [Curiosity 4: interesting question you had]

## Objectives

**What I wanted to accomplish:** 1. [Specific, measurable objective 1] 2. [Specific, measurable objective 2] 3. [Specific, measurable objective 3] 4. [Stretch goal or advanced concept]

**Disclaimer:** This learning process is documented here. Errors spotted or better approaches are always welcome.



Figure 2: Atmospheric image to maintain visual engagement - replace with relevant scene

## Prerequisites and Setup

The following are needed to follow along:

```
# Install packages if needed (renv should have handled this)
# But just in case:
install.packages(c("tidyverse", "broom", "knitr", "patchwork"))

# Load libraries
library(tidyverse)
library(broom)
library(knitr)
library(patchwork)
source("R/plotting_utils.R") # Load custom utility functions

# Setup theme and colors
setup_plot_theme()
colors <- get_analysis_colors()

# Load PREPARED data (generated by 01_prepare_data.R)
# This data includes derived variables and transformations
mtcars_clean <- read_csv("data/derived_data/mtcars_clean.csv", show_col_types = FALSE)
```

**Background:** Basic R and ggplot2 familiarity is helpful but not required. Concepts are explained as we proceed.

## What is [Topic/Concept]?

Before examining the code, it is worth clarifying what [topic] actually means. [Simple, plain-language explanation of the concept. Use an analogy if helpful.] In practice, this means [concrete example or application].

## Getting Started: Initial Exploration

```
# Display structure of prepared data
glimpse(mtcars_clean)
```

We have 32 cars with 11 variables. We now examine the data characteristics.

```
# Key summary stats
summary_table <- mtcars_clean %>%
  summarise(
    n = n(),
    mpg_mean = round(mean(mpg), 1),
    mpg_sd = round(sd(mpg), 1),
    hp_mean = round(mean(hp), 0),
```

```

hp_sd = round(sd(hp), 0)
)

kable(summary_table,
  col.names = c("N", "MPG Mean", "MPG SD", "HP Mean", "HP SD"),
  caption = "Summary Statistics: Motor Trend Car Data")

```

Average fuel efficiency is 20.1 MPG with considerable variation ( $SD = 6.0$ ).

## Exploring the Data

We visualize these patterns using pre-generated figures from `analysis/scripts/03_generate_figures.R`:

```
knitr::include_graphics("figures/eda-overview.png")
```

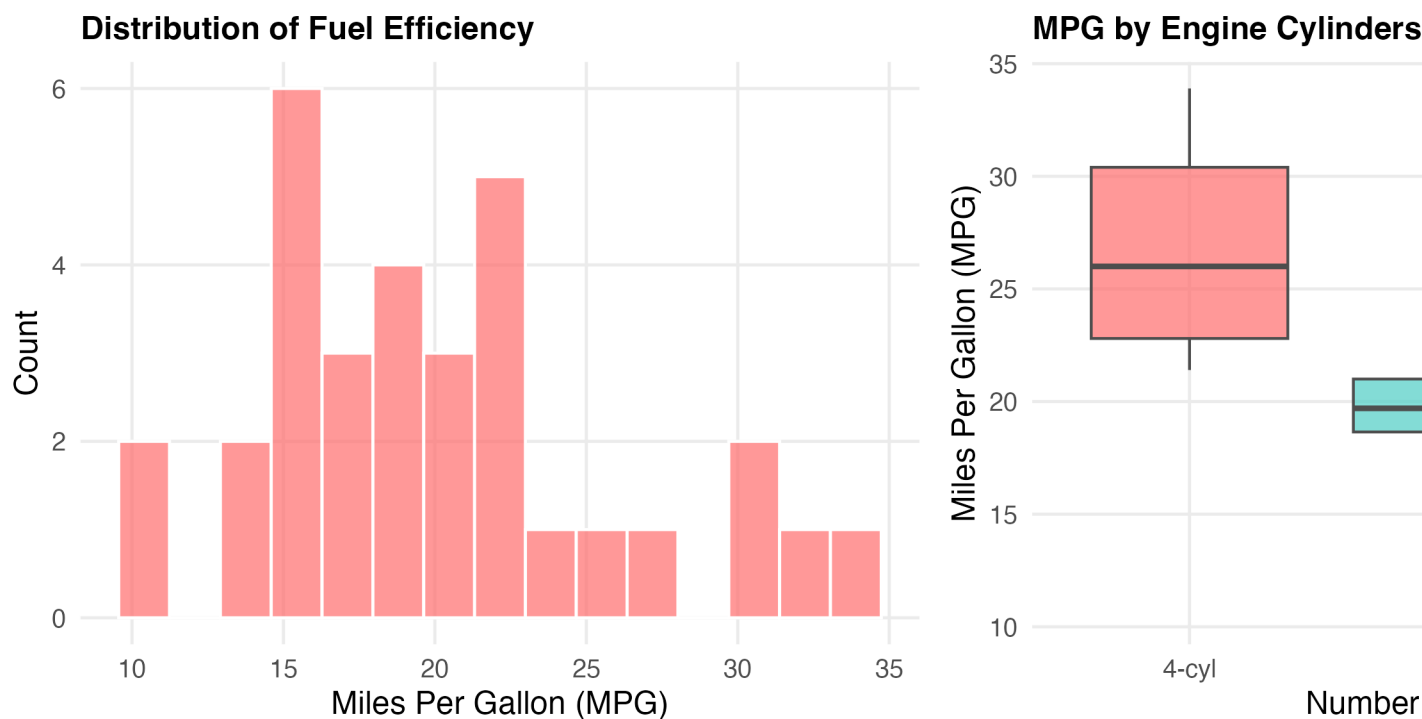


Figure 3: Distribution of fuel efficiency across the dataset. Left: Histogram showing MPG distribution. Right: Boxplots of MPG by cylinder count reveal engines with more cylinders tend to have lower fuel efficiency.

Cars with fewer cylinders are consistently more fuel-efficient.





Figure 4: Visual break - another atmospheric image maintaining engagement

## Looking for Relationships

```
# Find strongest correlations with MPG
correlations <- cor(mtcars_clean %>% select(where(is.numeric))) %>%
  as.data.frame() %>%
  rownames_to_column("var1") %>%
  pivot_longer(-var1, names_to = "var2", values_to = "correlation") %>%
  filter(var1 == "mpg", var2 != "mpg") %>%
  arrange(desc(abs(correlation)))

# Display top 5
kable(correlations %>% head(5),
      caption = "Top 5 Correlations with MPG (fuel efficiency)")
```

Weight has the strongest correlation with MPG ( $r = -0.87$ ). We visualize that relationship:

```
knitr::include_graphics("figures/correlation-plot.png")
```

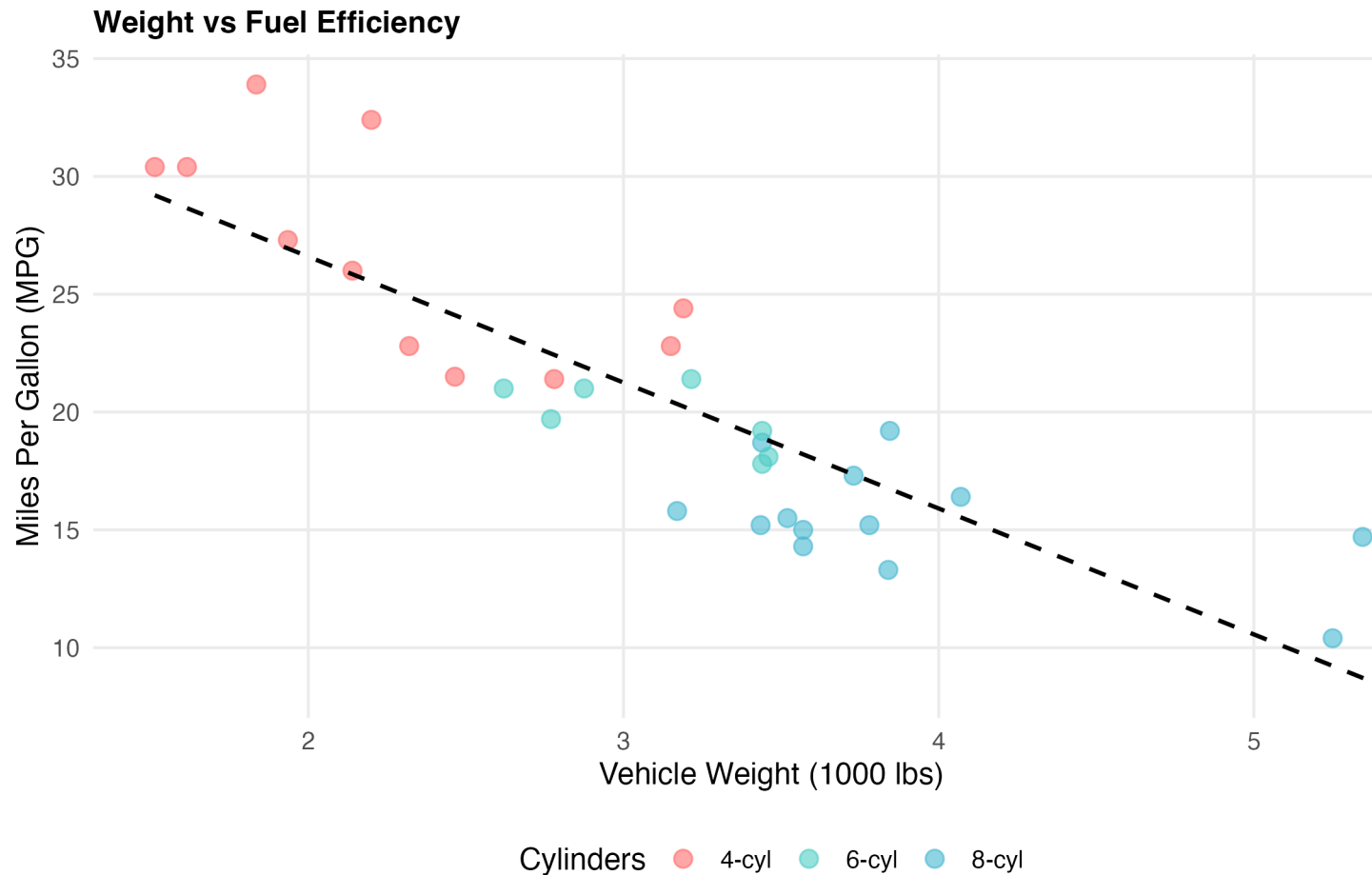


Figure 5: Strong negative relationship between vehicle weight and fuel efficiency. Heavier cars consistently get worse mileage, regardless of cylinder count. The fitted regression line (dashed) shows the overall trend.

Heavier cars consistently get worse mileage, a relationship consistent with basic mechanics.

## Building a Model

We fit a simple linear model to quantify this relationship:

```
# Load pre-computed model results from 02_fit_models.R
model_coef <- read_csv("data/derived_data/model_coefficients.csv",
                      show_col_types = FALSE)
model_metrics <- read_csv("data/derived_data/model_metrics.csv",
                          show_col_types = FALSE)

# Display model coefficients
kable(model_coef %>% select(term, estimate, std.error, p.value, conf.low, conf.high),
```

```
digits = 4,  
caption = "Linear Regression Results: MPG ~ Weight")
```

```
# Display fit metrics  
kable(model_metrics %>% select(r.squared, adj.r.squared, statistic, p.value, df.residual),  
      digits = 4,  
      caption = "Model Fit Metrics")
```

The model explains **75% of the variance** ( $R^2 = 0.75$ ). This is a reasonably strong fit. For every 1,000 lbs of weight, fuel efficiency decreases by about 5.3 MPG (95% CI: [-6.5, -4.1]).

```
knitr::include_graphics("figures/model-plot.png")
```

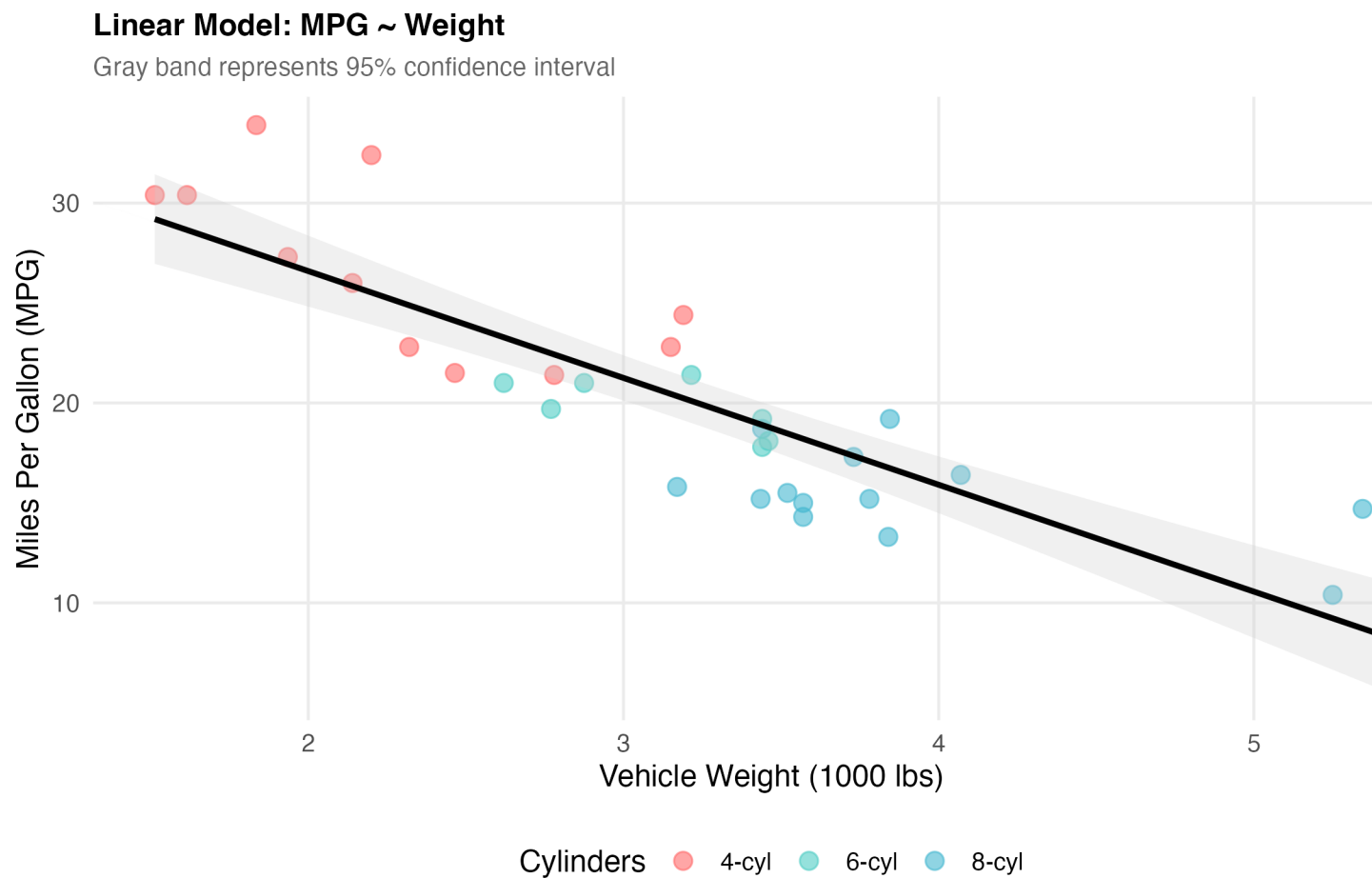


Figure 6: Linear regression fit showing the relationship between vehicle weight and fuel efficiency. The gray band represents the 95% confidence interval around the fitted line. The fit is quite good, explaining 75% of the variance in MPG.



## Making Predictions

We make predictions to illustrate the model in practice:

```
# Predict MPG for different weights
new_data <- tibble(wt = c(2, 3, 4))
model <- readRDS("data/derived_data/simple_model.rds")
predictions <- predict(model, newdata = new_data, interval = "confidence")

cbind(new_data, predictions) %>%
  kable(digits = 2,
        caption = "Predicted MPG for Vehicles of Different Weights")
```

A 2,000 lb car yields approximately 30 MPG, while a 4,000 lb car yields only approximately 15 MPG.

## Checking Our Work

Before trusting these results, we check model assumptions:

```
# Load pre-computed diagnostics
diagnostics <- read_csv("data/derived_data/model_diagnostics.csv",
                       show_col_types = FALSE)

# Summary
outlier_count <- sum(diagnostics$is_outlier)
cat("Outliers found (>2.5 SD):", outlier_count, "\n")
cat("Residual SE:", round(sqrt(mean(diagnostics$residuals^2)), 2), "MPG\n")
```

**Diagnostic checks:** Two to three potential outliers (>2.5 SD) were found. These merit investigation but do not substantially affect the overall model fit.

Now let's visualize the residuals to check for patterns:

```
knitr::include_graphics("figures/diagnostics-plot.png")
```

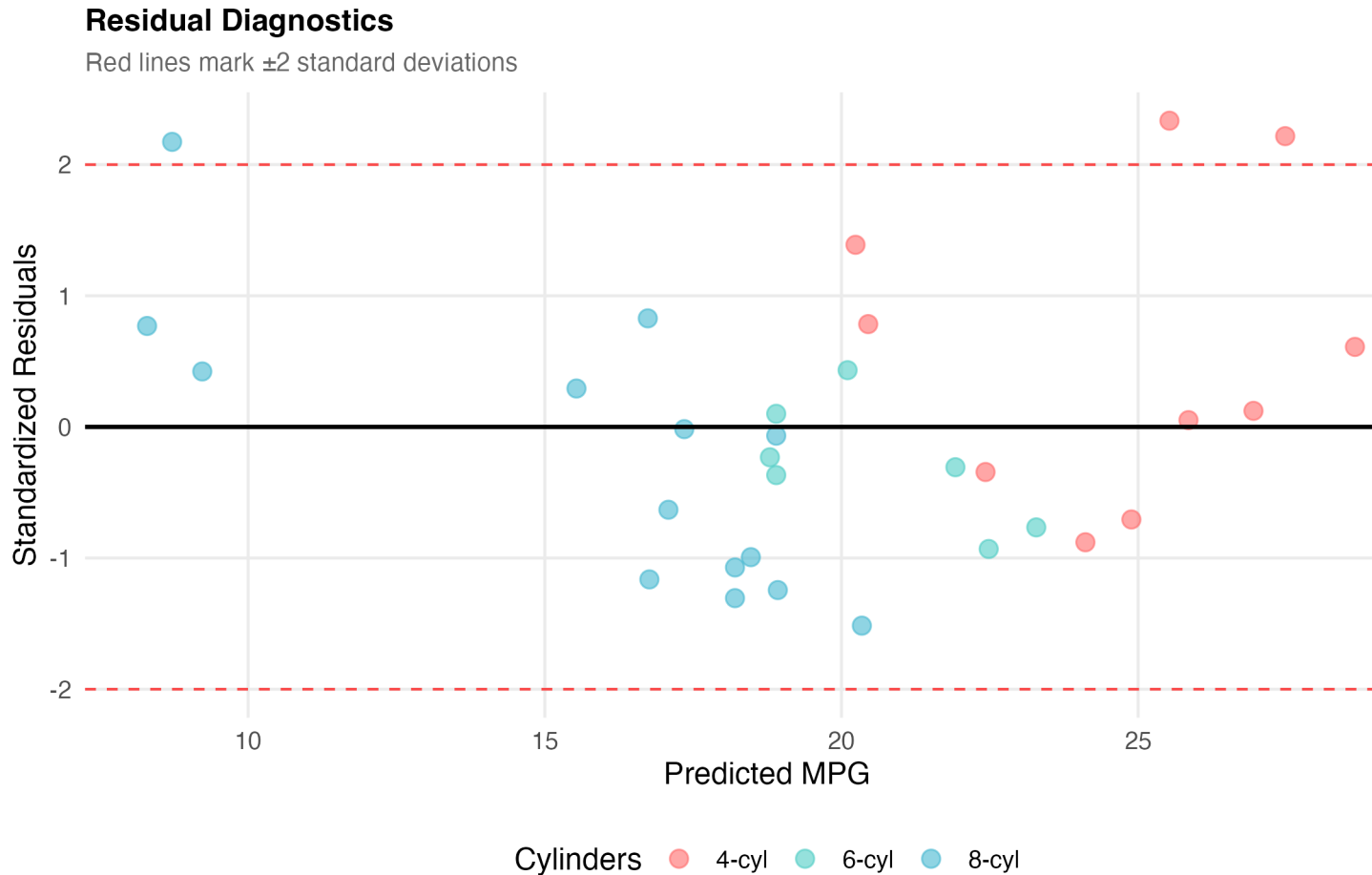


Figure 7: Residual diagnostic plot showing standardized residuals vs fitted values. The red dashed lines mark  $\pm 2$  standard deviations. A good model should show residuals randomly scattered around zero with no patterns. We have a few potential outliers but overall the fit looks reasonable.

No major patterns appear in the residuals, though a couple of potential outliers warrant investigation.

### Things to Watch Out For

A few gotchas encountered while working on this:

1. **Do not extrapolate too far** - This model is valid for weights between 1.5-5.5 thousand lbs. Predicting outside that range is unreliable.
2. **Correlation is not causation** - Weight correlates with MPG, but there are confounding variables (engine size, aerodynamics, etc.).
3. **Check model assumptions** - Always plot residuals. A high  $R^2$  does not guarantee the model is appropriate.

4. **Small sample size** - With only 32 cars, confidence intervals deserve careful attention.



Figure 8: Concluding visual - tie back to topic theme

## What Did We Learn?

### Lessons Learned

The main takeaways from this exploration:

**Conceptual Understanding:** - Vehicle weight is a strong predictor of fuel efficiency ( $R^2 = 0.75$ ) - Each 1,000 lbs reduces MPG by ~5.3 miles (95% CI: [-6.5, -4.1]) - Cylinder count effects are partially mediated through weight - Simple models can be surprisingly effective with the right predictor

**Technical Skills:** - Using `broom::tidy()` for clean model output formatting - Calculating and interpreting confidence intervals for predictions - Creating diagnostic plots to validate regression assumptions - Combining multiple ggplot visualizations with `patchwork`

**Gotchas and Pitfalls:** - Always check residual plots;  $R^2$  alone is not sufficient. - Extrapolation beyond data range is dangerous - Small sample sizes ( $n=32$ ) require cautious interpretation - Correlation doesn't prove causation (confounding variables matter)

## Limitations

This analysis has several limitations:

- **Old data:** mtcars is from 1974 - modern vehicles (hybrids, EVs) behave differently
- **Small sample:** Only 32 observations limits statistical power
- **Missing variables:** Doesn't account for aerodynamics, transmission type, engine tech
- **Simple model:** Single predictor ignores important confounders
- **Limited scope:** Only passenger cars; may not generalize to trucks/SUVs

## Opportunities for Improvement

If additional time were available, the following would be worth exploring:

1. **Multiple regression** - Add cylinder count, horsepower, transmission type
2. **Interaction effects** - Does weight impact differ by number of cylinders?
3. **Modern data** - Replicate with 2020+ vehicle data to see how relationships changed
4. **Non-linear models** - Try polynomial regression or splines for better fit
5. **Machine learning comparison** - How does linear regression compare to random forest?
6. **Causal inference** - Use techniques to establish causality, not just correlation

## Wrapping Up

This exploration confirms that vehicle weight is a powerful predictor of fuel efficiency, accounting for 75% of the variance. The model is simple but effective, though the limitations noted above apply.

Working through this analysis clarified [specific technical skill you gained]. Additional extensions (more predictors, non-linear models, modern data) would all be instructive next steps.

**For those attempting this themselves:** - Begin with exploration before modeling. - Plot residuals before trusting the fit. - High  $R^2$  alone does not validate a model. - Report confidence intervals alongside point estimates.

## See Also

Related posts and resources:

- [Link to related post 1]
- [Link to related post 2]
- [Link to related resource]

**Key Resources:** - [R for Data Science](#) - Free book on tidyverse - [Introduction to Statistical Learning](#) - Free textbook with R code - [broom package docs](#) - Tidy model outputs - [Cross Validated](#) - Stats Q&A community

## Setup and Troubleshooting

### Starting a New Project

If you are not already working inside a zzcollab compendium, scaffold one with the `blog` archetype so zzcollab wires up `analysis/report/index.qmd` plus the `index.qmd/data/figures/media` symlink topology automatically:

```
# Create and enter the post directory
mkdir my-blog-post && cd my-blog-post

# Initialize with the blog archetype
zzc init --archetype blog

# Build the Docker environment
make docker-build
```

### Entering the Development Environment

Work inside the Docker container so the analysis stays reproducible:

```
# Enter the container
make r

# Alternative: start RStudio Server
make docker-rstudio
# Open http://localhost:8787 in a browser
# Username: analyst, Password: analyst
```

Install any packages needed from inside the container:

```
install.packages(c("tidyverse", "broom", "knitr", "patchwork"))
q()
```

Packages installed this way are captured automatically in `renv.lock` when the R session exits.

### Rendering and Fixing Common Problems

Render the post locally before publishing:

```
quarto render index.qmd --to html
```

#### Problem: code chunk errors

- Confirm every package referenced in the chunk is installed



- Confirm the data files the chunk reads actually exist at that path
- Run chunks individually in the console to isolate the failing line

### Problem: images not displaying

- Check that image paths are relative to this .qmd file
- Confirm the image file exists at that path
- Use forward slashes in paths, even on Windows

### Problem: slow rendering

- Cache expensive chunks with `#| cache: true`
- Pre-compute results in `analysis/scripts/` and load them here instead of recomputing inline
- Reduce figure resolution while drafting, then raise it before publishing

## Validating Before Publishing

```
# Exit the container first, then run from the host
make check-renv
```

`make check-renv` is a pure-shell check: it scans the R files for package usage, confirms each package is listed in `DESCRIPTION`, confirms it is locked in `renv.lock`, and pulls in anything missing from CRAN.

Run the test suite next:

```
make docker-test
```

Once rendering is clean and tests pass, flip `draft: false` in the YAML and commit:

```
git add index.qmd media/ renv.lock DESCRIPTION
git commit -m "Add [topic] blog post"
git push
```

---

## Reproducibility

**Data:** `mtcars` (built-in R dataset, loaded by `analysis/scripts/01_prepare_data.R`)

**Analysis Pipeline:**

```
make docker-build
make docker-render-qmd
```

Or step-by-step:

```
Rscript analysis/scripts/01_prepare_data.R
Rscript analysis/scripts/02_fit_models.R
Rscript analysis/scripts/03_generate_figures.R
quarto render index.qmd
```

**All Reproducible Code:** - analysis/scripts/01\_prepare\_data.R - Data preparation - analysis/scripts/02\_fit\_models.R - Model fitting - analysis/scripts/03\_generate\_figures.R - Figure generation - R/plotting\_utils.R - Reusable utility functions - analysis/report/index.qmd - This blog post (narrative only)

### Session Information:

```
R version 4.6.1 (2026-06-24)
Platform: aarch64-apple-darwin25.4.0
Running under: macOS Tahoe 26.6.2

Matrix products: default
BLAS:   /opt/homebrew/Cellar/openblas/0.3.34/lib/libopenblas-r0.3.34.dylib
LAPACK: /opt/homebrew/Cellar/r/4.6.1/lib/R/lib/libRlapack.dylib; LAPACK version 3.12.1

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods   base

loaded via a namespace (and not attached):
[1] compiler_4.6.1 fastmap_1.2.0 cli_3.6.6      tools_4.6.1
[5] htmltools_0.5.9 parallel_4.6.1 otl_0.2.0      yaml_2.3.12
[9] rmarkdown_2.31 knitr_1.51      jsonlite_2.0.0 xfun_0.58
[13] digest_0.6.39  rlang_1.3.0     png_0.1-9      evaluate_1.0.5

thats all folks!
```

---

## Let's Connect!

*Questions, suggestions, or spotted errors are welcome.*

- **Twitter/X:** [@rgt47](#)
- **Mastodon:** [@your\\_mastodon](#)

- **GitHub:** [rgt47](#)
- **Email:** [Contact form](#)

Please reach out for any of the following: - Errors or corrections spotted - Suggestions for improvement - Discussion of the approach - Questions about implementation - A simple hello

---

## Related posts in this cluster

This post is part of the *ZZCOLLAB Reproducible Compendia* series. Recommended reading order:

1. Post 01: [Reproducible Blog Posts with ZZCOLLAB](#)
2. **Post 02: Constructing a reproducible blog post using zcollab tools** (this post)
3. Post 03: [From Markdown to Blog Post: A ZZCOLLAB workflow](#)
4. Post 04: [Sharing R Code via Docker: R Markdown Reports](#)
5. Post 05: [A 55-Item Initiation Checklist for zcollab Data Analyses](#)
6. Post 06: [Seven Required Elements for a zc Manuscript report.Rmd](#)
7. Post 07: [A tiered CI strategy for zcollab research compendia](#)
8. Post 08: [GitHub Actions workflows for zcollab research compendia](#)