

Data Visualization Challenge

R Setup in Cursor + Penguins and Simpson's Paradox

Data Viz Challenge - Quick Start Guide

Challenge Overview

In this challenge, you'll set up R inside Cursor (VS Code-compatible), get comfortable with terminals, shells, and command lines, and then complete a real Simpson's Paradox analysis by forking and modifying a GitHub repository.

Your Mission: Fork the [Simpson's Paradox repository](#), use Cursor's AI agent to create a compelling side-by-side comparison plot, and render a professional README.md file.

By the end of this challenge, you will be able to run R code from Cursor, render Quarto documents, and collaborate on GitHub repositories.

Part 1: R in Cursor Setup

Follow these steps (adapted from the official R extension guidance) to get R working smoothly in Cursor.

1. Install R

- Download and install R (version 3.4.0) from [CRAN R downloads](#)
- During installation on Windows, select the option to **Write R path to the registry** (recommended)

2. Install the R extension in Cursor

- Open Extensions: **Ctrl+Shift+X**
- Search for “R” (publisher: REditorSupport) and install the extension named “R”
- Also install this/these extensions if you haven’t already:
 - “Quarto” (for .qmd editing and preview)

3. Install languageserver (enables R language features)

You can install directly from an R session. In Cursor:

- Open Command Palette: **Ctrl+Shift+P** → type “R: Create R Terminal” → Enter
- In the R console that appears, run:

```
install.packages("languageserver")
```

Note

Cursor may prompt to install `languageserver` for you automatically; accepting that prompt is fine.

4. Verify R works in Cursor

- 1) Create a new file `hello.R` and add a couple lines:

```
summary(cars)  
mean(cars$speed)
```

- 2) With the editor caret on a line, press **Ctrl+Enter** to send the current line to the active R terminal.

- 3) If Cursor can’t find R automatically on Windows, set the R path:

- Open Settings: **Ctrl+,**
- Search for “Rterm”
- Set R > Rterm: Windows to something like `C:\\Program Files\\R\\R-4.4.1\\bin\\x64\\R.exe` (adjust version/path as installed)

6. Running R chunks in Quarto (.qmd)

With the R extension (and Quarto extension installed), you can execute R code chunks inside Quarto documents.

- Create or open a .qmd file and insert an R chunk:

```
R.version.string
```

```
[1] "R version 4.5.1 (2025-06-13 ucrt)"
```

- Use the chunk toolbar (Run/Run Above/Run All) to execute. If the toolbar is hidden, place the cursor in the chunk and use **Ctrl+Shift+P** → “Quarto: Run Cell” or send lines with **Ctrl+Enter**.

Part 2: Terminals, Shells, and Command Lines

Putting It All Together

Here’s the hierarchy that helps when switching between the system terminal and the R terminal:

- **Terminal** → the app that gives you a text window (e.g., Cursor’s Integrated Terminal)
- **Shell** → the program that runs inside that window
 - Examples (Windows): PowerShell, `cmd.exe`
 - Examples (macOS/Linux): `bash`, `zsh`
- **Command line** → the interaction style: typing commands into the shell
- **R Terminal** → a specialized program you run from the shell, which then gives you an R-only command line

In Cursor: - Open the integrated terminal: **Ctrl+``** (backtick) - this gives you the system shell - Create an R terminal: **Ctrl+Shift+P** → “R: Create R Terminal” - this runs R inside the terminal - Use the R terminal for R commands; use the system shell (PowerShell/bash) for tools like `git` and `quarto`

Part 3: The Real Challenge - Simpson's Paradox Repository

Now that you have R set up, it's time for the main challenge! You'll fork a repository, complete a data visualization task, and demonstrate your GitHub collaboration skills.

A. Fork the Simpson's Paradox Repository

1. **Go to the repository:** <https://github.com/flyaflya/simpsonParadox>
2. **Fork the repository:** Click the "Fork" button (top right) to create your own copy
3. **Clone your fork locally:** In Cursor, press `Ctrl+Shift+P` → "Git: Clone"
4. **Paste your fork URL:** https://github.com/YOUR_USERNAME/simpsonParadox
5. **Choose a location** to save it (e.g., `C:\`)

B. Explore the Repository

The repository contains a `README.qmd` file that needs to be completed. Your task is to:

1. **Open `README.qmd`** in Cursor
2. **Find the section** that says " Your Task: Use Cursor's AI assistant..."
3. **Use Cursor's AI agent** to create the side-by-side comparison plot
4. **Render the document** to create `README.md`

C. Complete the Side-by-Side Plot Task

Your Mission: Replace the placeholder instructions with actual code that creates a compelling side-by-side comparison using the `patchwork` package.

Steps: 1. **Ask Cursor AI:** *"Create a side-by-side comparison plot using patchwork that combines plot1 and plot2 with a title, subtitle, and caption explaining Simpson's Paradox"* 2. **Replace the placeholder** with the generated code 3. **Test the plot** by running the chunk 4. **Render the document:** `quarto render README.qmd`

D. Commit and Sync Your Work

1. **Save all files:** `Ctrl+Shift+S`
2. **Stage your changes:** `Ctrl+Shift+G` → click "+" next to modified files
3. **Commit with message:** "Complete Simpson's Paradox side-by-side comparison"
4. **Push to GitHub:** Click "Sync Changes" or "Push"
5. **Verify on GitHub:** Check that your `README.md` file shows the completed plot

! Important

No Pull Request Needed! Adam will check your GitHub repository directly for the completed README.md file with the side-by-side comparison plot.

E. Key Takeaways

The Power of Visualization:

- **Always visualize your data** before drawing conclusions from statistical models
- A single trendline can be dangerously misleading when it masks underlying group differences
- **P-values are not the answer** - statistical significance doesn't guarantee the relationship is meaningful or correctly interpreted

Simpson's Paradox Dangers:

- **Aggregated data can reverse relationships** - what appears to be a positive correlation overall might be negative within each group
- **Business implications:** Making decisions based on aggregate trends without considering subgroups can lead to costly mistakes
- **The "lurking variable" problem:** Always ask "What am I missing?" when relationships seem counterintuitive

Notable Simpson's Paradox Examples:

- **UC Berkeley admissions (1973):** Overall admission rates suggested gender bias against women, but within each department, women had equal or higher admission rates
- **Baseball batting averages:** A player can have a higher batting average than another in both halves of a season, yet a lower overall average
- **Medical studies:** A treatment can appear harmful overall but beneficial within each age group
- **Marketing campaigns:** A campaign might seem ineffective overall but highly successful within specific customer segments

Best Practices:

- **Stratify your analysis** by relevant grouping variables (demographics, categories, time periods)
- **Look for confounding variables** that might explain apparent relationships
- **Use multiple visualization approaches** (overall vs. faceted plots, side-by-side comparisons)

- **Question your assumptions** - if a relationship seems too good (or bad) to be true, it might be!
-

Submission Checklist

R Setup:

- ☐ R installed (Windows: path written to registry)
- ☐ R extension installed in Cursor (plus Quarto extension)
- ☐ `languageserver` installed
- ☐ Able to send lines from `hello.R` to the R terminal
- ☐ Able to run an R chunk inside a `.qmd`

Repository Challenge:

- ☐ Simpson's Paradox repository forked to personal GitHub account
- ☐ Repository cloned locally in Cursor
- ☐ `README.qmd` file opened and explored
- ☐ Side-by-side comparison plot created using Cursor AI agent
- ☐ `README.qmd` rendered to `README.md` successfully
- ☐ Changes committed and synced to GitHub repository
- ☐ `README.md` file visible on GitHub with completed plot

Understanding Simpson's Paradox:

- ☐ Can explain how overall vs. within-group relationships differ
 - ☐ Can identify the grouping variable (species) that reveals the paradox
 - ☐ Understands the dangers of relying solely on aggregate trendlines
 - ☐ Can describe at least one real-world example of Simpson's paradox
-

Need Help?

- R extension documentation: [VS Code R Extension Wiki](#)
- Quarto docs: [Quarto Guide](#)
- General tip: search for "VS Code" results—most answers apply to Cursor as well