02 Make Friends with R

Jonathan Templin

Make Friends with R (EDF 9780: Lecture 2)

Installing R

- Download latest version from https://cran.rstudio.com
- Install executable on your local machine

Alternate Method: rig (R installation manager; terminal window needed)

- There is a free and open-source installation manager named rig, available from https://github.com/r-lib/rig (see page for instructions)

Installing R Studio

- Download latest version from https://posit.co/downloads/

Alternate Method: R Studio Server

- Clemson offers a web-based R Studio version via the HPC (although I'm not super familiar with how to access this…yet)
- Similarly, you can sign up for a free account on Posit Cloud (https://posit.cloud)-giving you 25 hours per month of compute time

Enabling GitHub Copilot

Requesting Access as a Clemson Student:

1. Create a GitHub account at https://github.com/join using your Clemson email address (ending in `@clemson.edu`).
2. Go to https://education.github.com/discount_requests/new and fill out the form using your Clemson email address.
3. Once authorized, you will have access to Copilot syntax completion (which will help you with R).

Sign Into to GitHub in R Studio

1. Go to the following menu location: Tools…Global Options…Copilot
2. Sign in with your GitHub account

Using R

- Live demo in lecture

Using R Studio

- Live demo in Lecture (see Make Friends with R syntax file)

Alternatives to R/R Studio Setup

`radian` R Terminal (not available for R Studio)

Radian is a modern, alternative console for R. It provides features often missing from the standard terminal R console, including:

- **Syntax Highlighting:** Real-time coloring of code.
- **Auto-completion:** Multi-line editing and rich completion.
- **Vi Mode:** Keybindings for those who prefer Vim style navigation.
- **Reticulate Integration:** Seamless Python and R interoperability.

This guide covers the installation and configuration of Radian on your system.

Prerequisites

Before installing Radian, ensure you have the following installed:

1. **R**: Use your system's package manager or download from CRAN.
2. **Python (3.6+)**: Radian is a Python package, so Python must be available.

Installation

Radian is distributed via PyPI (Python Package Index). You can install it using `pip` or `conda`.

Option 1: Using pip (Recommended)

Open your terminal and run:

```
pip install -U radian
```

Note: If you are using a system-managed Python environment, you may need to use ***`pip3`*** *or install it within a virtual environment.*

Option 2: Using Conda

If you manage environments with Conda:

```
conda install -c conda-forge radian
```

Configuring Your IDE/Terminal

1. VS Code (Visual Studio Code)

To make Radian the default R terminal in VS Code:

1. Find the path to your Radian executable.

- **Mac/Linux:** Run `which radian` in your terminal.
- **Windows:** Run `where radian` in Command Prompt.

2. Open VS Code Settings (`Cmd+,` or `Ctrl+,`).
3. Search for **"r.rterm"**.
4. Paste the path to the radian executable in the specific setting for your OS (e.g., `R > Rterm: Mac`).
5. *Optional:* To ensure plots render correctly, enable `httpgd` in VS Code settings by searching for **"r.plot.view"** and selecting `httpgd` (requires the `httpgd` R package).

2. Alias in Shell (Zsh/Bash)

If you want `r` to launch Radian instead of standard R in your terminal:

1. Open your shell configuration file (e.g., `~/.zshrc` or `~/.bashrc`).
2. Add the following alias:

```
alias r="radian"
```

3. Reload your configuration: `source ~/.zshrc`.

Customization (.radian_profile)

You can customize Radian's behavior by creating a `.radian_profile` file in your home directory. This file is just an R script that runs when Radian starts.

Example `.radian_profile`:

```
# Set options
options(width = 100)

# Custom prompt
options(radian.prompt = "\033[0;34mr$>\033[0m ")

# Key bindings (if you want to override defaults)
# See prompt_toolkit documentation for complex bindings
```

Alternate Integrated Development Environments:

Visual Studio Code (VS Code) has become a alternative to RStudio, offering a lightweight, customizable, and multi-language development environment. This primer guides you through transforming VS Code into a fully-featured IDE for R.

VS Code is available at https://code.visualstudio.com/download

1. Essential Extensions

To get started, install the following extensions from the VS Code Marketplace (Sidebar → Extensions icon or `Ctrl+Shift+X`):

- **R (vscode-R):** The core extension by Yuki Ueda. It provides syntax highlighting, code execution, and terminal integration.
- **R Debugger:** Enables debugging R scripts directly within VS Code.

- **Quarto:** Essential for rendering `.qmd` files and provides excellent support for RMarkdown.
- **Path Intellisense:** (Optional) helpful for auto-completing file paths.

2. Required R Packages

VS Code relies on specific R packages to communicate with the editor and render plots efficiently. Open your R terminal and install:

```
install.packages(c("languageserver", "httpgd", "jsonlite"))
```

- **`languageserver`:** Powers IntelliSense (auto-completion), definitions, and error checking.
- **`httpgd`:** A modern HTTP graphics device that allows plots to be viewed in a dedicated VS Code panel or browser window (svg-based and resize-friendly).

3. Configuration

To configure VS Code, open the Settings (`Cmd+,` or `Ctrl+,`) and search for the following identifiers, or edit your `settings.json` directly.

A. Enable `httpgd` for Plotting

Search for `r.plot.view` and set it to: `httpgd`

This ensures plots appear in a dedicated panel rather than a pop-up window.

B. Link Radian (Recommended)

If you installed **Radian** (from the previous guide), set it as your default terminal.

- **Search setting:** `r.rterm` (select your OS, e.g., Mac/Linux/Windows)
- **Value:** Paste the path to your radian executable (e.g., `/usr/local/bin/radian` or `C:\Program Files\Python39\Scripts\radian.exe`).

C. Bracketed Paste

To ensure large blocks of code paste correctly into the terminal:

- **Search setting:** `r.bracketedPaste`
- **Value:** Check (enable) this box.

4. The Workflow

Sending Code

- **Run Line/Selection:** `Cmd+Enter` (Mac) or `Ctrl+Enter` (Windows/Linux).
- **Run All:** `Cmd+Shift+S` (Mac) or `Ctrl+Shift+S`.

The Help Viewer

You can view R documentation directly in VS Code.

- **Command:** `?mean` or `help(mean)`
- **Result:** Opens in a dedicated VS Code panel, similar to RStudio.

Workspace Viewer

The R extension provides a **Workspace** icon in the activity bar (usually on the left). It lists:

- Global Environment (Data frames, lists, functions)
- Loaded Packages
- Help Topics

5. Sample `settings.json`

For power users, here is a snippet to add to your `settings.json` for a robust setup:

```json
{
  "r.rterm.mac": "/usr/local/bin/radian",
  "r.bracketedPaste": true,
  "r.plot.view": "httpgd",
  "r.lsp.diagnostics": true,
  "editor.wordWrap": "on",
  "[r]": {
    "editor.defaultFormatter": "REditorSupport.r"
  }
}
```

Positron

What is Positron?

Positron is a next-generation integrated development environment (IDE) built by **Posit** (the creators of RStudio).

It is built on the **Code OSS** (VS Code) engine, meaning it inherits the speed, extension ecosystem, and modern editor features of VS Code, but it comes pre-configured specifically for data science.

The "Elevator Pitch": * If RStudio and VS Code had a child, it would be Positron. * It offers the **Data Explorer** and **Variable View** you love in RStudio. * It offers the **Editor speed**, **Copilot integration**, and **Customizability** of VS Code.

Why Choose Positron?

1. **Language Agnostic:** Unlike RStudio (which is R-centric), Positron treats **R** and **Python** as first-class citizens with equal support.
2. **No "Setup Fatigue":** In the previous guides, you had to manually install extensions, configure `httpgd`, and link Radian. Positron comes with these data-science specific connections pre-wired.
3. **The Data Explorer:** The biggest pain point for R users moving to standard VS Code is the lack of a good data viewer. Positron includes a high-performance native data viewer that supports filtering and sorting large datasets out of the box.

Installation

Positron is a standalone application (not an extension).

1. Go to the Positron GitHub Releases page.
2. Download the installer for your OS (`.dmg` for Mac, `.exe` for Windows).
3. Install and launch.

Key Interface Features

1. The Interpreter Selector

In the top right (or bottom status bar), you can instantly switch between R versions or Python environments (Conda/Venv) without restarting the IDE.

2. The Console

Positron uses a modern console implementation that feels very similar to Radian. It supports syntax highlighting and multi-line editing by default.

3. The Layout

- **Left Sidebar:** Files, Search, Git (Standard VS Code).
- **Right Sidebar:** Variables, Help, Plots (Standard RStudio style).
- **Center:** Editor and Console.

Essential Shortcuts (VS Code Style)

Since Positron is based on VS Code, it uses the VS Code command palette and keybindings by default (though you can enable "RStudio Keymap" in settings).

- **Command Palette:** `Cmd+Shift+P` (Mac) or `Ctrl+Shift+P` (Win). Use this to find *any* setting or command.
- **Run Code:** `Cmd+Enter` (Mac) or `Ctrl+Enter` (Win).
- **Pipe Operator (`|>` or `%>%`):** You may need to set a custom keybinding if `Cmd+Shift+M` isn't mapped to your preferred pipe by default.

Migrating from RStudio

If you are used to RStudio, you can make Positron feel like home:

1. Open the **Command Palette**.
2. Type **"Preferences: Color Theme"** to change the look (e.g., "Tomorrow Night Blue").
3. Search for **"Keymap"** in the extensions marketplace to install the **IntelliJ** or **RStudio** keymap extensions if you struggle with VS Code shortcuts.

Summary Comparison

Feature	RStudio	VS Code (Raw)	Positron
Engine	Java/Web (GWT)	Electron/Web	Electron/Web
Setup	Zero config	High config	Low config
Data Viewer	Excellent	Basic/Plugins	Excellent
Polyglot	R focused	Anything	R & Python
Extension Market	Limited	Massive	Massive