Supercharge your R Shiny Development Work flow

Tips, Tricks and Techniques for boosting Shiny Development Productivity and Efficiency

Mayank Agrawal

Senior R developer Consultant I

ProCogia

Sunday, September 24, 2023

Introduction

- Background: Computer Science Engineer and MBA in Business Analytics.
- Using R and R Shiny for 6+ years.
- Architected, built and managed more than 40+ R Shiny applications and dashboards.
- Built and led teams of 10+ R developers.
- Trained 30+ colleagues on R from diverse backgrounds.
- Have extensive hands on experience on all flavors of R products - Shiny, R Markdowns, Quarto, and Package development.
- Curious to learn.
- Working with **ProCogia**; an Official full service partner with Posit.

Overview

Hello There!

In this lightning talk, we will touch base on the following high level topics to boost your R development experience.

- Streamlining Workflow Essentials
- Shiny Development Tips
- Productivity Hacks
- Mastering Shortcuts

Streamlining Workflow Essentials

- Use RStudio Projects
  - Why Should I use it?
- Organize your project structure.
  - Use/Create a standard project template.
- Leverage package management tools
  - Utilize `renv` or `packrat` to manage package dependencies.
  - Ensure reproducibility and minimize version conflicts.
- Version Control with Git
  - Track changes, collaborate with team members, and manage project history effectively using Git.
  - Use Issues, Pull Requests and Connected Commits for efficient and effortless parallel development.

Using RStudio Projects

RStudio projects make it straightforward to divide your work into multiple contexts, each with their own working directory, workspace, history, and source documents.

Why should I use it?

- Isolates your development environment.
- Relative path referencing for your project files.
- Helps setup project package management via `renv` or `packrat`
- Segregate development flow and context.

How to create a new RStudio Project?

Use your RStudio IDE Toolbar:

- `File` –> `New Project` –> (Choose) `New Directory`/`Existing Directory`/`Version Control`

Organize your project structure

- Maintain a clear and logical directory structure.
- Group related files together for easy navigation.
- Create, Adopt and Enforce a standard development template for easier developer onboarding and faster development.
- Have a project template repository for initial cloning and usage.
- Please refer to the right image for an illustrative R project directory structure.

Shiny Development Tips

- Optimize Server logic
  - Minimize unnecessary computations and avoid redundant calculations.
  - Profile your code to identify bottlenecks with `profvis`
- Use General Coding principles
  - DRY (Don’t Repeat Yourself)
  - YAGNI (You aren’t going to need it)
  - KISS (Keep it Simple Silly)
  - Document your code for humans.
  - Clean Code at all costs
  - Separation of Concerns

Shiny Development Tips

- Utilize Reactive programming
  - Use **Waterfall** reactivity model.
  - Avoid **Spaghetti Reactivity mess**.
  - Leverage Shiny's reactivity to update UI elements dynamically based on user input.
  - This enhances interactivity and responsiveness.
- Implement efficient UI components
  - Prototype the UI layout using draw.io or similar tools.
  - Use conditional panels, modules, and well-designed layouts to create a user-friendly interface.
  - Iterate rapidly to test how different implementations enhance UI/UX

Productivity Hacks

- Debugging like a pro
  - Debug via `browser()`, `debug(function_name)`, and debug points.
  - Set breakpoints, step through code, and inspect variables to troubleshoot efficiently.
- Profiling and performance tuning
  - Use tools like `profvis` to identify performance bottlenecks and Optimize critical sections of your code for speed.
  - Use benchmarking techniques to compare computation speeds of various packages. Refer R function `rbenchmark::benchmark()`.
- Automate repetitive tasks with scripts
  - Write scripts to automate routine tasks such as data pre-processing, cleaning, visualization and report generation.
  - Refactor code into functions with parameters for versatile reusability across projects.

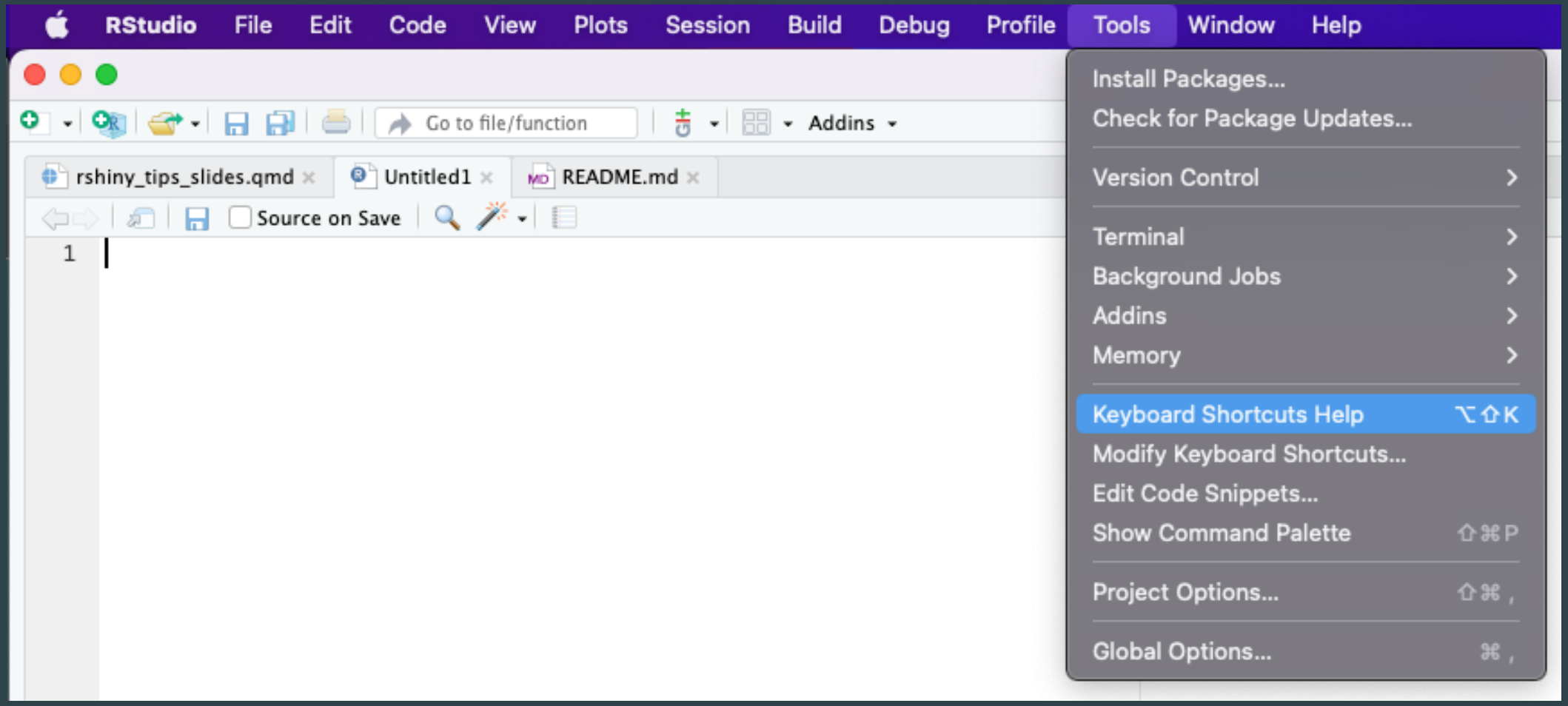
Mastering Shortcuts

- RStudio IDE shortcuts
  - Learn key shortcuts for common actions like running code, navigating files, and managing tabs.
  - Boost your coding speed and efficiency.
- Shiny-specific keyboard tricks
  - Use keyboard shortcuts for navigating the app, toggling between UI and server code, and inserting common Shiny functions.
- Time-saving code snippets
  - Store frequently used code snippets in RStudio's code snippets pane for quick access.

Keyboard Shortcuts at a glance

RStudio provides dozens of useful shortcuts that you can access through the menu at the top:

`Tools` > `Keyboard Shortcuts Help`.



Thank you

- Quarto presentation code available on GitHub at http://bit.ly/3ZHZTNf
- Slides available on GitHub Pages at https://bit.ly/3RsdExn
- Connect and/or send me a DM for a follow up question or catch up
  - Linkedin: mayank-agrawal-7jan
  - Twitter: mayank7jan
  - Mastodon: mayank7j

References

- RStudio Shortcuts Cheat sheet:
  - RStudio IDE Toolbar: `Help` –> `Keyboard Shortcuts Help`
  - Mac: `Option` + `Shift` + `K`
  - Windows: `Alt` + `Shift` + `K`
- Customizing the RStudio IDE: Link 1 Link 2
- Comprehensive documentation and self help guides for using RStudio IDE Link
- Using RStudio Projects. Link
- Posit Cheat Sheets Link
- Project package management packages - `renv` and `packrat`
- `profvis` for code profiling and time estimation

Appendix

My favourite Keyboard Shortcuts for R Development Workflow

Navigate between Window Panes

Knowing how to toggle between panes without touching your mouse to move your cursor will help save time and improve your workflow. Use the following shortcuts to switch between panes seamlessly (Mac/Windows):

- `Control/Ctrl + 1`: *Source editor* (your script)
- `Control/Ctrl + 2`: *Console*
- `Control/Ctrl + 3`: Help
- `Control/Ctrl + 4`: History
- `Control/Ctrl + 5`: *Files*
- `Control/Ctrl + 6`: *Plots*
- `Control/Ctrl + 7`: Packages
- `Control/Ctrl + 8`: Environment
- `Control/Ctrl + 9`: Viewer

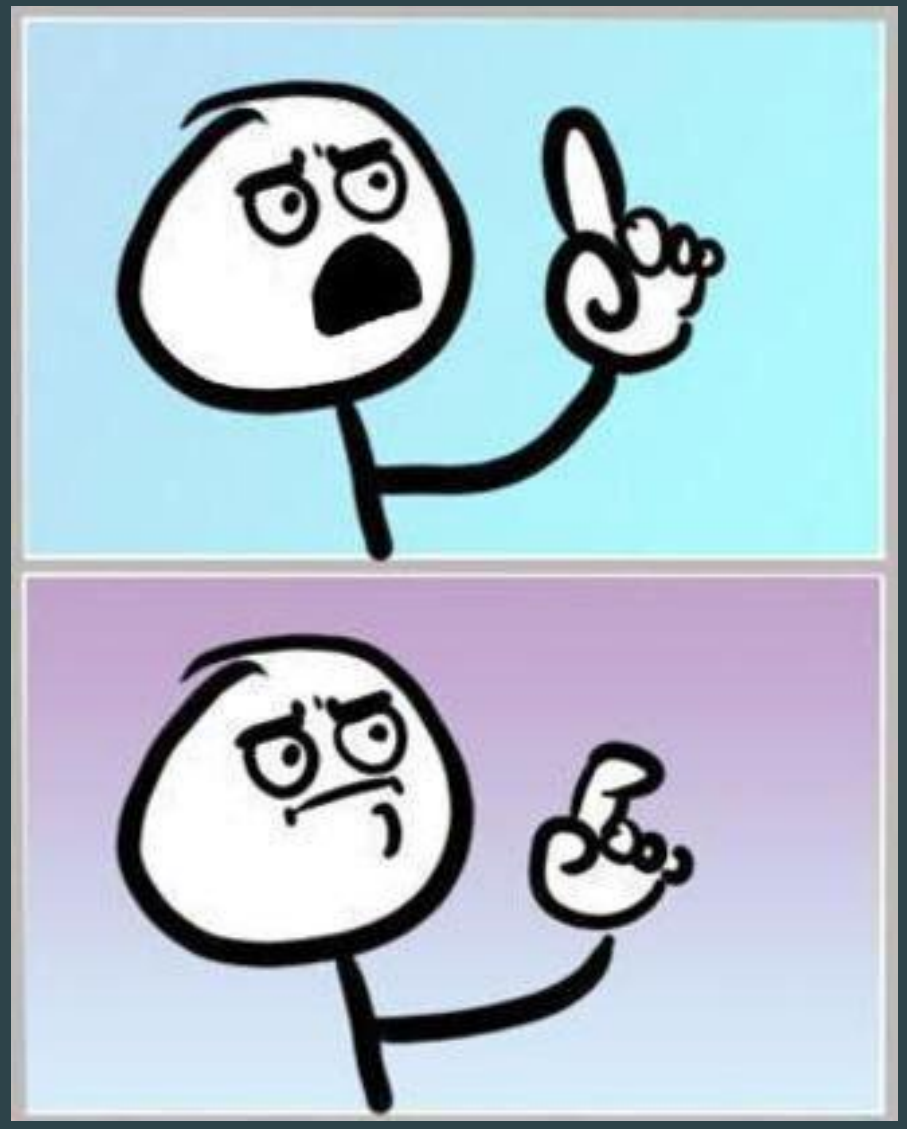
Single Pane View

If you prefer to only have one pane in view at a time, add `Shift` to any of the previous commands to maximize the pane. Use the following shortcuts to bring a panes into primary focus seamlessly.

- `Control/Ctrl + Shift + 1`: Source editor (your script) pane only
- `Control/Ctrl + Shift + 2`: Console pane only
- `Control/Ctrl + Shift + 3`: Help pane only
- `Control/Ctrl + Shift + 4`: History pane only
- `Control/Ctrl + Shift + 5`: Files pane only
- `Control/Ctrl + Shift + 6`: Plots pane only
- `Control/Ctrl + Shift + 7`: Packages pane only
- `Control/Ctrl + Shift + 8`: Environment pane only
- `Control/Ctrl + Shift + 9`: Viewer pane only

The Classic Four Pane View

Wait! How do I go back to the classic four-pane view?



`Control/Ctrl + Shift + 0` (Mac/Windows): Four-pane RStudio IDE View

Console

Description	Windows & Linux	Mac
Shift focus of cursor to Console	`Ctrl + 2`	`Control + 2`
Clear Console	`Ctrl + L`	`Control + L`
Move Cursor to beginning of a line	`Home`	`Cmd + Left`
Move Cursor to end of a line	`End`	`Cmd + Right`
Move Cursor to start/end of a word	`Alt + Left/Rigth`	`Option + Left/Right`
Navigate command history	`Up/Down`	`Up/Down`
Pop-up Command history	`Ctrl + Up`	`Cmd + Up`

Source Scripts

Description	Windows & Linux	Mac
Toggle document outline	`Ctrl + Shift + O`	`Cmd + Shift + O`
Clear Console	`Ctrl + L`	`Control + L`
New R Script	`Ctrl + Shift + N`	`Cmd + Shift + N`
Save active document	`Ctrl + S`	`Cmd + S`
Close active document	`Ctrl + W`	`Cmd + W`
Knit and Preview Document (knitr)	`Ctrl + Shift + K`	`Cmd + Shift + K`
Insert Chunk (Rmd)	`Ctrl + Alt + I`	`Cmd + Option + I`

Find

Description	Windows & Linux	Mac
Find and Replace	`Ctrl + F`	`Cmd + F`
Find in Files	`Ctrl + Shift + F`	`Cmd + Shift + F`

Insert

Description	Windows & Linux	Mac
Insert Assignment Operator (<-)	`Alt + -` (hyphen/minus)	`Option + -` (hyphen/minus)
Insert Pipe Operator (%>%)	`Ctrl + Shift + M`	`Cmd + Shift + M`

Code

Description	Windows & Linux	Mac
Re-indent lines	`Ctrl + I`	`Cmd + I`
Comment/un-comment current line/selection	`Ctrl + Shift + C`	`Cmd + Shift + C`
Reformat Selection	`Ctrl + Shift + A`	`Cmd + Shift + A`
Move Lines Up/Down	`Alt + Up/Down`	`Option + Up/Down`
Jump to Matching Brace/Parenthesis	`Ctrl + P`	`Control + P`
Add Cursor via drag	`Alt + drag mouse pointer`	`Option + drag mouse pointer`
Delete Current Line	`Ctrl + D`	`Cmd + D`
Insert Roxygen Skeleton	`Ctrl + Alt + Shift + R`	`Cmd + Option + Shift + R`

Terminal

Description	Windows & Linux	Mac
Move Focus to Terminal	`Alt + Shift + M`	`Shift + Option + M`
New Terminal	`Alt + Shift + R`	`Shift + Option + R`

Session

Description	Windows & Linux	Mac
Restart R session	`Ctrl + Shift + 0`	`Cmd + Shift + 0`

R Ladies NYC - September 24, 2023