Getting the Most Out of RStudio

Josh Paulson

Product Manager
josh@rstudio.org

June 25, 2012

Before We Begin

1) Confirm Setup

2) Introductions

3) Overview of Training

Setup Instructions

- Install the latest version of RStudio (v0.96.304+)

  http://www.rstudio.org
- Install the knitr package (v0.5+)

  https://github.com/yihui/knitr
- Install a version control system (git or svn)

  http://www.rstudio.org/docs/version_control/overview
- Install TeX

  http://www.latex-project.org/ftp.html
- Obtain tutorial files:
  - GitHub - Download repository as Zip file
    https://github.com/jwpaulson/
  - USB Drive - Ask Josh

Introductions

- Who are you?
- How long have you been using R?
- What do you use R for?
- What other editors do or have you used?
- What do you hope to get out of this training?

Overview

- Lecture vs. hands-on
- Ask questions as they come up
- Follow along and experiment on your own
- Use the slides to take notes
- Collaborate with neighbors during hands-on sections

Overview - Day 1

- What is RStudio?
- RStudio Basics
- Customizing RStudio
- Managing Projects and Navigating Code
- Interactive Graphics with the **manipulate** Package
- Integrated Version Control with Git and Subversion
- Productivity Techniques

Overview - Day 2

- Customized Reports and Presentations (Sweave/knitr)
- Interactive R Documents (R Markdown/Notebook)
- Publishing Code, Analysis, and Results (RPubs)
- Debugging R Code
- Troubleshooting and Getting Help with R
- Best Practices for R Programming
- Introduction to RStudio Server

What is RStudio?

- An Integrated Development Environment for R
- Free and open source project to create a cross-platform IDE for R (AGPL v3 license)
- Focused on coding productivity rather than visual abstractions
- Desktop version for Windows, MacOS X, and Linux
- Server version for web access through a browser

Focus on Coding

- RStudio is an R coding tool, not an R GUI
- "Selection-and-response" user interfaces help with initial discovery, but eventually limit the user.
- Code is required for trustworthy and reliable analysis:
  - Repeatable
  - Inspectable
  - Reusable
  - Diffable

Helping Users Realize the Full Potential of R

- Expose and extend the capabilities of R
- Encourage the adoption of techniques and practices known to produce high quality, trustworthy analysis
- Incorporate the best ideas and practices from existing software development environments to help them gain wider use and application among users of R
- User feedback is essential

IDE Overview

- Four Pane Layout
- Console, Source, Workspace, and Files
- History, Plots, Packages, and Help
- Version Control, Sweave, and R Markdown
- RStudio Desktop and Server

Console

- Enter Commands
- **Up Arrow** for Command History
- **Ctrl + Up Arrow** for Filtered History
- **TAB** Key for Code Completion
- Title Bar

Source Editor

- Combine Commands into a Runnable Script
- Tight Integration of Source with Console
- Run Commands from Editor:
  - Step-wise (Line by Line)
  - Highlight Selections
- Source Scripts:
  - Source
  - Source with Echo
- Keyboard Shortcuts

Workspace

- Displays Global Environment
  - Data, Values, and Functions
- Save and Load Workspaces (.RData)
- Import and View Data
- Clear All = `rm(list=ls(all.names=TRUE))`

History

- Displays History of Commands
- Save and Load History (.Rhistory)
- Search and View History within its Context
- Send Commands to Console or Source
- Prune History and Clear All

Files Pane

- Displays Selected Directory
- Sort, Select, and Open Files
- Browse to New Directory
- Changing and Viewing Working Directories
  - Set Files Pane as Working Directory
  - View Current Working Directory in Files Pane
  - Set Source File Location as Working Directory
  - Choose Working Directory

Plots Pane

- View and Scroll History of Plots
- Zoom Plots
- Export and Manually Resize Plots
- Manage Plot History

Packages Pane

- View and Search Currently Installed Packages
- Load, Detach, and Manage Packages
- Package Manager to Install New Packages
- Configure Package Repositories
- Check and Install Updates

Help Pane

- Help Home
- Manuals, NEWS, and References
- Search For Help Topics
- Find in Topic
- History of Help Topics

Source Editor - Advanced

- Re-run Previous Region
- Extract Function and Code Transformations
- Code Folding
- Go To Function Definition
- Function Menu and Navigation
- Find and Replace
- Change File Type

Code Completion - `CodeCompletion.R`

- Open File: "`CodeCompletion.R`"
- Press **TAB** to Enable Code Completion
  - Commands Completion
  - Function Arguments
  - Directory Paths (Inside Quotes)
  - Objects in the Global Environment
  - Variables Inside data.frame

Hands On

- Review the Basics of the IDE by studying home prices in the Midwest
- http://rpubs.com/josh/activity-1
- Make use of features such as code completion and save your analysis in an R script.
- Ask Questions

Customizing RStudio

- Tools | Options (Windows/Linux)
  RStudio | Preferences (Mac)
- General R and Editor Options
- Editor Themes
- Customize Layout for Coding, Plotting, etc.
- Adjustable Pane Sizes and Layout

Hands On

- Briefly explore some of the customization options
- http://rpubs.com/josh/activity-2
- Ask Questions

Projects

- "Working-directory" Based Workflow
  (.R files, .Rprofile, .RData, .Rhistory, etc.)
- Creating a New Project
- Switching Projects
- Separate Source Documents, Workspace, Files, etc.
- Open Project in New Window
  (Multiple RStudio Sessions)
- Project Indexing

Project Level Options

- Global vs. Project Level Options
- Customize Options for Each Project
- Customize Pane Layout
- Configure TeX and Version Control

Hands On

- Turn this training into a project
- http://rpubs.com/josh/activity-3
- Explore project level options
- Switch between multiple projects
- Ask Questions

Code Navigation

- Find in Files
- Go to File/Function
- Go to Function Definition (**F2**)
- Navigate Back and Forth Between Cursor Positions
- Function Menu

Hands On

- Turn R's source code into a project
- http://rpubs.com/josh/activity-4
- Use code navigation to explore the source
- Ask Questions

Interactive Graphics with manipulate

- What is **manipulate**? (See example)
- Provides controls to dynamically change plot values
- `slider()` to manipulate variable along numeric range
- `picker()` to manipulate variable of fixed choices
- `checkbox()` to manipulate logical variable
- Various applications in the **mosaicManip** package

Hands On

- Experiment and create a manipulate plot
- See ?manipulate help and examples
- http://rpubs.com/josh/activity-5
- Use `slider()`, `picker()`, and `checkbox()`
- Check out examples from **mosaicManip**
- Ask Questions

Version Control - Motivation

Have you ever:

- Changed code, realized it was a mistake and wanted to go back?
- Lost code or had a backup that was too old?
- Had to maintain multiple versions of a product?
- Wanted to see the diff between two (or more) versions of code?
- Wanted to prove a particular change broke/fixed some code?
- Wanted to submit a change (patch) to someone else's code?
- Wanted to see what work is being done (where/when/who)?
- Wanted to test a new feature without breaking the working code?

http://stackoverflow.com/questions/1408450/why-should-i-use-version-control

Version Control

- Create a Dedicated Repository for Your Code
- Track Changes, Revert Mistakes, and Target Bugs
- Necessary For Project Collaboration
- Easily Compare Versions of Your Code (View Diff)
- "Version control is as fundamental to programming as accurate notes about lab procedures are to experimental science" (Greg Wilson)

Version Control - git or svn?

- RStudio has integrated support for git and svn

  (Can still use other vc systems normally)
- git - Distributed Version Control:
  - peer-to-peer approach (local repositories)
  - commit offline
  - pull, stage, commit, push
  - https://github.com/
- svn - Centralized Version Control
  - client-server approach
  - requires Internet to commit
  - checkout, update, commit

Version Control in RStudio

- Open New Project with Version Control
- Clone Existing Project
- View Changelist
- Diff Window
- Commit Directly Inside IDE
- View Commit History
- Pull/Push (git) and Checkout/Update (svn)

Hands On

- Clone a project and experiment with Version Control
- http://rpubs.com/josh/activity-6
- Example (git): I'll commit and you pull
- Ask Questions

Version Control - Resources

- git - http://git-scm.com/
  - http://git-scm.com/book
  - http://learn.github.com/p/intro.html
  - https://github.com/
- svn - http://subversion.apache.org/
  - http://svnbook.red-bean.com/en/1.7/index.html
  - http://subversion.apache.org/docs/community-guide/
- RStudio Documentation:
  - http://www.rstudio.org/docs/version_control/overview

Productivity Techniques - Reinforced

- Separate work into projects
- Code Navigation is essential
- Utilize Source Editor and IDE shortcuts
- Learning New Packages
- Using Version Control

Recap and Review

- RStudio is an R coding tool, not a GUI
- Exposing and extending features in R
- Basics, advanced, and customization
- Projects and code navigation
- The manipulate package
- Version control
- Questions on specific areas?

Setup For Day 2

- We'll need these for tomorrow, anyone need help?
- Install the knitr package (v0.5+)

  https://github.com/yihui/knitr
- Install TeX

  http://www.latex-project.org/ftp.html