Introduction to R and RStudio

Lecture 8

Dirk Eddelbuettel

STAT 430: Data Science Programming Methods (Fall 2019)

Department of Statistics, University of Illinois

Topics

- Basic RStudio tour:
    - console,
    - editor,
    - environment/build/git,
    - help/packages/plot/viewer
- Help system, where to get help, cheatsheets, sites
- Installing packages, basics of repositories, versioning
- Resources:
    - RStudio IDE cheatsheet
    - And other items in *Help* menu of RStudio IDE
    - And of course the website

RStudio

Access Modes

Three ways to access RStudio

- RStudio Desktop
- RStudio Server
- RStudio Cloud

RStudio Desktop

Key features of RStudio Desktop:

- runs locally as an application
    - if your operating system is supported
- good: independent of network access
- possibly tricky: *you* need to manage packages and add-ons

RStudio Server

Key features of RStudio Server

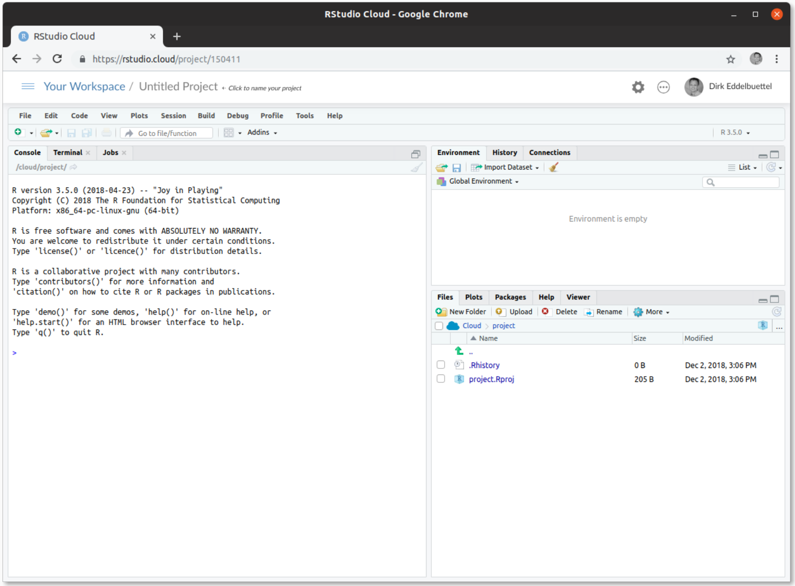
- Locally or remotely: can be the same machine
    - if your operating system is supported
- good:
    - access port 8787 on a network accessible machine,
    - only needs browser
- maybe tricky: needs network access
- at most one session per server
    - unless ‘pro’ version of server used

RStudio Cloud:

Key features of RStudio Cloud

- Remotely, only need browser access
- Requires network access
- Good:
    - Least amount of administration or maintenance
    - But new sessions are ‘empty’ so some local customization
- This is what we use here in this course

RStudio Cloud


RStudio Cloud - Google Chrome
RStudio Cloud
https://rstudio.cloud/project/150411
Your Workspace / Untitled Project ← Click to name your project
Dirk Eddelbuettel
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
R 3.5.0
Console Terminal Jobs
/cloud/project/
Environment History Connections
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
New Folder Upload Delete Rename More
Cloud > project
Name Size Modified
.Rhistory 0 B Dec 2, 2018, 3:06 PM
project.Rproj 205 B Dec 2, 2018, 3:06 PM

```
R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

RStudio Cloud

ILLINOIS

Generally four panes

- First pane: File
- Second pane: R Console
- Third pane: Environment
- Fourth pane: Files

RStudio Cloud: First Pane

ILLINOIS

Files (top left)

- Often an R file (and we will get there soon)
- Also any other supported files
- Try shell: we can save a textfile with extension `.sh`
    - proper hightlighting appears
    - in the bottom right right corner ‘Shell’ should appear
    - ‘Run’ button leads to execution
- Try markdown: save with extension `.md`
    - bottom right shows Markdown
    - Preview button appears
- Try SQL as we did already in the SQL lessons.

RStudio Cloud: Second Pane

Environment / History / Connections (top right)

- Environment shows current variables and functions
    - Very helpful to inspect data
- History allows browse and search of past commands
- Connections less relevant for us, more import in commerical setting with DB drivers
- *Added only in project mode*: git
- The git feature is very useful when using git

RStudio Cloud: Third Pane

Console / Terminal / Jobs (bottom left)

- Console is generally our R prompt where run R code
- Terminal is a newer addition with a full-feature shell
    - good for shell commands
- Jobs is newer for job control, we will not use this
- RMarkdown outlog can appear here too

RStudio Cloud: Fourth Pane

Files / Plots / Packages / Help / Viewer

- ‘Files’ lets you view, select, alter, open, … files and visit directories
- ‘Plots’ is where our (standard) R plots appears
- ‘Packages’ is the interface to R packages and lets us browse, install, …
- ‘Help’ is the very useful help browser and viewer
- ‘Viewer’ is where interactive graphics and displays are shown

GIT WITH RSTUDIO

R Project from Git

One example: data-gapminder

- One can use the repository at GitHub, *e.g.* either one of
  https://github.com/eddelbuettel/data-examples
  https://github.com/stat430dspm/data-examples
  - Select the green ‘Clone or download’ button
  - Select https, it should show ‘Clone with https’
  - Click the ‘copy’ icon (little folder with arrow)
- Then in RStudio Cloud:
  - Under ‘New Project’ select ‘New Project from Git Repo’
  - Paste in the URL copied from GitHub, hit OK
  - A few second later a new (untitled) project should be created *with a git menu*

Authentication: Basics

- You can always access a remote repository by its URL, either
    - https so that you have to authenticate via a password
    - git using ssh which is more convenient but more advanced
- One trick is to 'cache' the https authentication credentials,
    - see the FAQ entry on the course site
    - and last part of Happy Git with R on Credentials Caching

GIT AND EXTERNAL REPOS

ILLINOIS

Authentication: ssh

- Do you know about ssh and creating keys?
    - yes: then upload the public key to they git server
    - no: maybe look into credential management / caching
    - This StackOverflow post at the URL below has more: `https://stackoverflow.com/questions/5343068/`
- In general this is harder
    - one possible reference is chapter 11 in Happy Git with R

Fork versus Clone

- *Cloning a repo* creates a *local* copy you can use, inspect, alter, ...
- In general, a *clone* creates a (local) copy of someone else's repo
- (Generally) You *cannot* write back to the original version
  - as that repo belongs to someone else
  - generally speaking you will not have write access
- *Forking a repo* creates a *remote* copy that is yours
- You then install a local version of your remote copy
- This is your: you generally *can* write back
- So:
  - read-only access to study, install, use, ...: *clone*
  - read-write access to modify etc: *fork*

GIT AND RSTUDIO

ILLINOIS

Your own repo

- Create a repo at GitHub
    - Either start one from scratch
    - Or *fork* an existing repo
- Bring it to RStudio
- Examine the history (the ‘hourglass icon’)
- Maybe make a change commit, push, ...
- Remember from lecture 3 and do this in ‘Terminal’:
    - `git config --global user.name ”Your name”`
    - `git config --global user.email ”you@you.com”`

Summary

RStudio

- convenient and powerful IDE
- can be used three different ways: local app, server, cloud
- we already encountered a number of features
    - shell terminal
    - SQL mode
    - RMarkdown
- and there will be more during the year