

Introduction and R

John Muschelli

About Us

John Muschelli

Assistant Scientist, Department of Biostatistics

PhD in Biostatistics, ScM in Biostatistics

Email: jmusche1@jhu.edu

About Us

Kristin Linn

Assistant Professor, Department of Biostatistics and Epidemiology
Perelman School of Medicine, University of Pennsylvania

PhD in Biostatistics

Email: klinn@mail.med.upenn.edu

Learning Objectives

- ▶ Introduction to the Statistical Software R
- ▶ Reading and Writing Images
- ▶ Visualization of Images and Image Data
- ▶ Inhomogeneity Correction
- ▶ Brain Extraction
- ▶ Image Segmentation
- ▶ Coregistration Within and Between MRI Studies
- ▶ Intensity Normalization
- ▶ Segmentation of MS lesions

Course Website/Materials

The Course overview is located at (with slides):

http://johnmuschelli.com/ISBI_2017.html

All materials for this course (including source for the slides) is located at:

https://github.com/muschellij2/imaging_in_r

RStudio Server

For this course, we will use an RStudio Server because installing all the packages can be a lengthy process. Mostly all the code we show requires a Linux/Mac OSX platform for FSL and other systems:

http://johnmuschelli.com/rstudio_server.html

The code to make the server is located at: https://github.com/muschelli2/rneuro/blob/master/ms_rstudio_droplet.sh

Installing R: Local

If you want to install things locally:

- ▶ You can install the latest R from <http://cran.r-project.org/>
- ▶ Install RStudio
- ▶ Directions for a local setup is http://johnmuscchelli.com/imaging_in_r/installing_everything_locally/

Introduction to R

What is R?

- ▶ R is a language and environment for statistical computing and graphics
- ▶ R is the open source implementation of the S language, which was developed by Bell laboratories
- ▶ R is both open source and open development

(source: <http://www.r-project.org/>)

Why R?

- ▶ Powerful and flexible
- ▶ Free (open source)
- ▶ Extensive add-on software (packages)
- ▶ Designed for statistical computing
- ▶ High level language

Why not R?

- ▶ Fairly steep learning curve
 - ▶ “Programming” oriented
 - ▶ Minimal interface
- ▶ Little centralized support, relies on online community and package developers
- ▶ Annoying to update
- ▶ Slower, and more memory intensive, than the more traditional programming languages (C, Java, Perl, Python)

Working with R

- ▶ The R Console “interprets” whatever you type
 - ▶ Calculator
 - ▶ Creating variables
 - ▶ Applying functions
- ▶ “Analysis” Script + Interactive Exploration
 - ▶ Static copy of what you did (reproducibility)
 - ▶ Try things out interactively, then add to your script

R essentially is a command line with a set of functions loaded

R Uses Functions, in Packages

- ▶ R revolves around functions
 - ▶ Commands that take input, performs computations, and returns results
 - ▶ When you download R, it has a “base” set of functions/packages (**base R**)
- ▶ Functions are enclosed in packages
 - ▶ These written by R users/developers (like us) - **some are bad**
 - ▶ Think of them as “R Extensions”

RStudio (the software)

RStudio is an Integrated Development Environment (IDE) for R

- ▶ It helps the user effectively use R.
- ▶ Makes things easier
- ▶ Is NOT dropdown statistical tools (such as Stata)
 - ▶ See Rcmdr or Radiant
- ▶ All snapshots in these slides are taken from <http://ayeimanol-r.net/2013/04/21/289/>

Easier working with R

- ▶ Syntax highlighting, code completion, and smart indentation
- ▶ Easily manage multiple working directories and projects

More information

- ▶ Workspace browser and data viewer
- ▶ Plot history, zooming, and flexible image and PDF export
- ▶ Integrated R help and documentation
- ▶ Searchable command history

RStudio/R Console

The screenshot shows the RStudio interface with the R Console window active. The console displays the R startup message, which includes the version number (3.0.0), copyright information, and a list of installed packages. A large black oval is drawn around the console output, highlighting the startup message and the list of installed packages.

R version 3.0.0 (2015-04-03) -- "Masked Marvel"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

>

The right-hand pane shows the 'Workspace' and 'History' tabs. Below these, the 'Files' tab is selected, displaying a list of installed packages:

Package	Description	Version	Update Status
boot	Bootstrap Functions (originally by Angelo Canty for S)	1.3-9	○
car	Companion to Applied Regression	2.0-16	○
class	Functions for Classification	7.3-7	○
cluster	Cluster Analysis Extended Rousseeuw et al.	1.14.4	○
codetools	Code Analysis Tools for R	0.2-8	○
colorspace	Color Space Manipulation	1.2-2	○
compiler	The R Compiler Package	3.0.0	○
datasets	The R Datasets Package	3.0.0	○
dichromat	Color Schemes for Dichromats	2.0-0	○

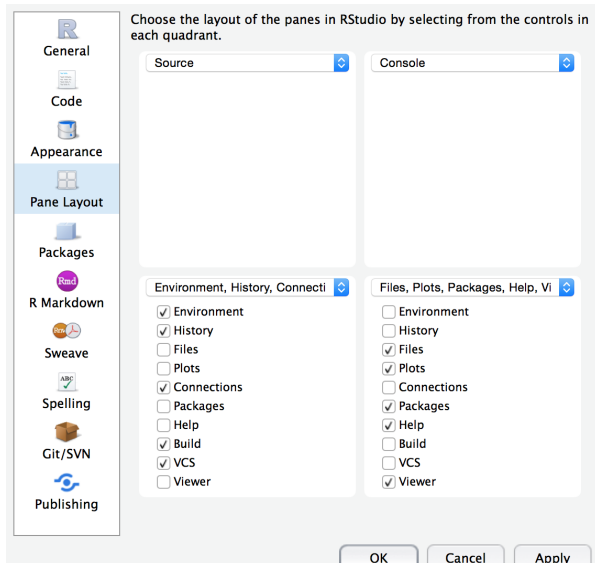
RStudio/R Console

- ▶ Where code is executed (where things happen)
- ▶ You can type here for things interactively
- ▶ Code is **not saved** on your disk

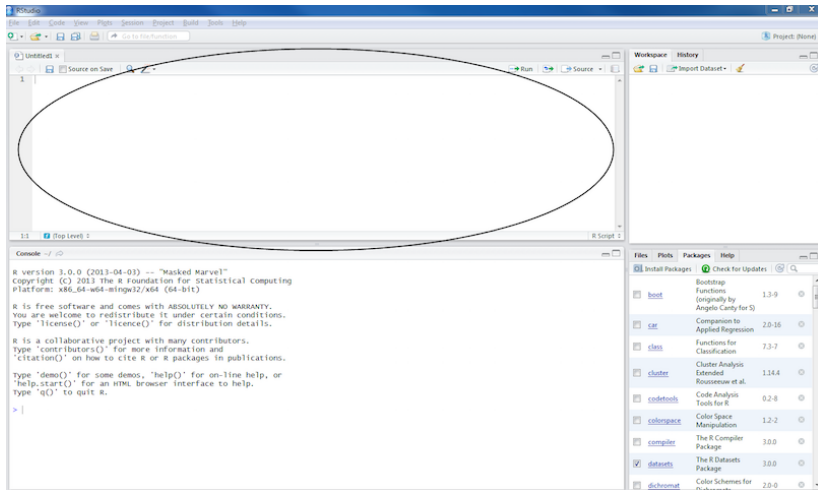
RStudio Layout

If RStudio doesn't look like this (or our RStudio), then do:

RStudio → Preferences → Pane Layout



Source/Editor



Source

- ▶ Where files open to
- ▶ Have R code and comments in them
- ▶ Can highlight and press (CMD+Enter (Mac) or Ctrl+Enter (Windows)) to run the code

In a .R file (we call a script), code is saved on your disk

Workspace/Environment

The screenshot displays the RStudio interface with three main panels:

- Source Editor:** Contains R code for loading packages and creating a ggplot. Lines 19 and 20 are highlighted in blue.
- Console:** Shows the execution of the code, including package attachment messages and the final plot command.
- Plots Panel:** Displays a scatter plot of mpg vs wt.

Source Editor Code:

```
1  
2  
3  
4 # load libraries of packages #####  
5  
6  
7 library(languageR)  
8 library(lme4)  
9 library(ggplot2)  
10 library(rms)  
11 library(plyr)  
12 library(reshape2)  
13 library(psych)  
14  
15  
16  
17 # plotting demonstration #####  
18  
19 p <- ggplot(mtcars, aes(wt, mpg))  
20 p + geom_point()  
21
```

Console Output:

```
Attaching package: 'plyr'  
  
The following object is masked from 'package:rmisc':  
  ls.discrete, summarize  
  
> library(reshape2)  
> library(psych)  
  
Attaching package: 'psych'  
  
The following object is masked from 'package:rmisc':  
  describe  
  
The following object is masked from 'package:ggplot2':  
  %>%  
  
> p <- ggplot(mtcars, aes(wt, mpg))  
> p + geom_point()  
>
```

Workspace/History Panel: A circle highlights the 'Values' section, showing the variable 'p' with the value 'gg[9]'.

Plots Panel: A scatter plot showing the relationship between weight (wt) on the x-axis and miles per gallon (mpg) on the y-axis. The plot shows a negative correlation, with mpg decreasing as wt increases.

Workspace/Environment

- ▶ Tells you what **objects** are in R
- ▶ What exists in memory/what is loaded?/what did I read in?

History

- ▶ Shows previous commands. Good to look at for debugging, but **don't rely** on it as a script. Make a script!
- ▶ Also type the “up” key in the Console to scroll through previous commands

Other Panes

- ▶ **Files** - shows the files on your computer of the directory you are working in
- ▶ **Viewer** - can view data or R objects
- ▶ **Help** - shows help of R commands
- ▶ **Plots** - pretty pictures
- ▶ **Packages** - list of R packages that are loaded in memory