R and RStudio

UNIVERSITY OF
SAN FRANCISCO

Abbie M Popa

BSDS 100 - Intro to Data Science with R

Class Organization

- In Class: Mix of Lecture and "Coding Challenges"
  - Not "graded" (though attendance is worth 20% of your grade), but you will do better on the homework if you attempt the activities!
- 8 Homework assignments (40% of your grade)
  - Based on in-class activities, completed **individually** at home
  - 30 points each, 10 points deducted automatically if turned in late
  - Attempt all questions! Even wrong answer may receive partial credit

Class Organization

- 2 Case Studies (20% of your grade)
  - Like homework assignments, but longer
  - Completed in pairs
  - You will receive **some** in-class time to work on these to help with scheduling issues
- Final Project (20% of your grade)
  - Similar to case studies, but more open-ended (you will pick the data and goals)
  - Completed in groups of 3 or 4

Outline

- The R Programming Language and RStudio
  - Comparison with other programming languages
  - Installation
  - Handy Shortcuts

What is R ?

[From the R website]: “R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, etc.) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source”

What is R?

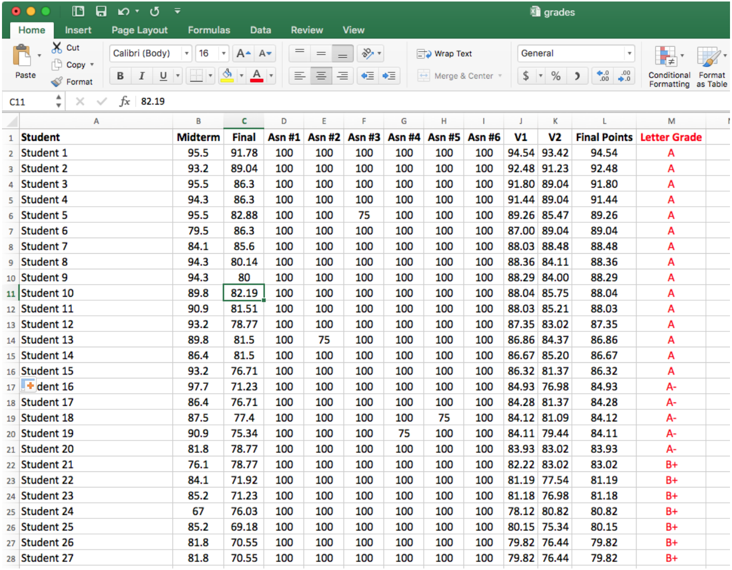
- R is a user-friendly integrated development environment (IDE) that is open source.
- Has many packages and functions for statistical analyses
- You can even run R from a terminal window if you wish (using BASH scripting)

Why Use R?

- Open source (free)
- Runs on just about any platform
- Great visualization capabilities (`ggplot2`)
- Read/write from/to various data sources
- Scripting language (interpreted)
- Massive library of data manipulation and statistical packages

But... what about Excel?

Excel is Great for Certain Things...



Student	Midterm	Final	Asn #1	Asn #2	Asn #3	Asn #4	Asn #5	Asn #6	V1	V2	Final Points	Letter Grade
Student 1	95.5	91.78	100	100	100	100	100	100	94.54	93.42	94.54	A
Student 2	93.2	89.04	100	100	100	100	100	100	92.48	91.23	92.48	A
Student 3	95.5	86.3	100	100	100	100	100	100	91.80	89.04	91.80	A
Student 4	94.3	86.3	100	100	100	100	100	100	91.44	89.04	91.44	A
Student 5	95.5	82.88	100	100	75	100	100	100	89.26	85.47	89.26	A
Student 6	79.5	86.3	100	100	100	100	100	100	87.00	89.04	89.04	A
Student 7	84.1	85.6	100	100	100	100	100	100	88.03	88.48	88.48	A
Student 8	94.3	80.14	100	100	100	100	100	100	88.36	84.11	88.36	A
Student 9	94.3	80	100	100	100	100	100	100	88.29	84.00	88.29	A
Student 10	89.8	82.19	100	100	100	100	100	100	88.04	85.75	88.04	A
Student 11	90.9	81.51	100	100	100	100	100	100	88.03	85.21	88.03	A
Student 12	93.2	78.77	100	100	100	100	100	100	87.35	83.02	87.35	A
Student 13	89.8	81.5	100	75	100	100	100	100	86.86	84.37	86.86	A
Student 14	86.4	81.5	100	100	100	100	100	100	86.67	85.20	86.67	A
Student 15	93.2	76.71	100	100	100	100	100	100	86.32	81.37	86.32	A
dent 16	97.7	71.23	100	100	100	100	100	100	84.93	76.98	84.93	A-
Student 17	86.4	76.71	100	100	100	100	100	100	84.28	81.37	84.28	A-
Student 18	87.5	77.4	100	100	100	100	75	100	84.12	81.09	84.12	A-
Student 19	90.9	75.34	100	100	100	75	100	100	84.11	79.44	84.11	A-
Student 20	81.8	78.77	100	100	100	100	100	100	83.93	83.02	83.93	A-
Student 21	76.1	78.77	100	100	100	100	100	100	82.22	83.02	83.02	B+
Student 22	84.1	71.92	100	100	100	100	100	100	81.19	77.54	81.19	B+
Student 23	85.2	71.23	100	100	100	100	100	100	81.18	76.98	81.18	B+
Student 24	67	76.03	100	100	100	100	100	100	78.12	80.82	80.82	B+
Student 25	85.2	69.18	100	100	100	100	100	100	80.15	75.34	80.15	B+
Student 26	81.8	70.55	100	100	100	100	100	100	79.82	76.44	79.82	B+
Student 27	81.8	70.55	100	100	100	100	100	100	79.82	76.44	79.82	B+

...but Not Everything

Sample Data

- Six columns of data with ~ 1.05 million rows
- Column 5: `startDate`
- Column 6: `endDate`
- **Objective**: test to see if `endDate` < `startDate`

RESULTS

- **Excel**: good luck...
- R: 33 min (poor coding technique)
- R: 58.5 sec (improved coding technique)

...but Not Everything

Sample Data

- Six columns of data with ~ 1.05 million rows
- Column 5: `startDate`
- Column 6: `endDate`
- **Objective**: test to see if `endDate` < `startDate`

RESULTS

- **Excel**: good luck...
- R: 33 min (poor coding technique)
- R: 58.5 sec (improved coding technique)

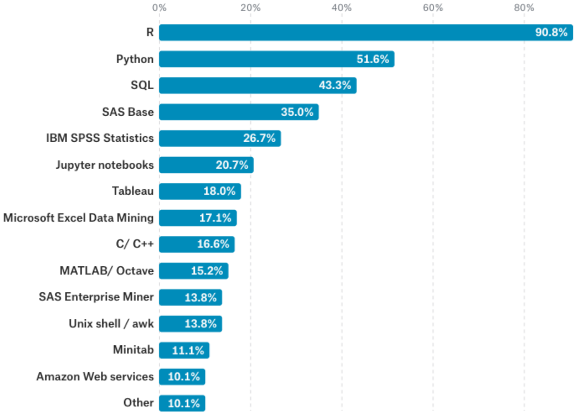
How about Python?

Vectorization in R

- Vectorization: a style of coding where an operation is applied to all elements in an array, rather than looping
- Vectorized code saves time asking `type` questions
- There is an optimized engine—a basic linear algebra system (BLAS)—that is highly efficient at solving linear algebra problems
- A lot or R functions are written in C (or variants)
- MATLAB, Mathematica and the NumPy package for Python are also vectorized

http://www.noamross.net/blog/2014/4/16/vectorization-in-r--why.html

And recall...



R wins at statistics

Installing R

- Download and install at this website:

  https://www.r-project.org

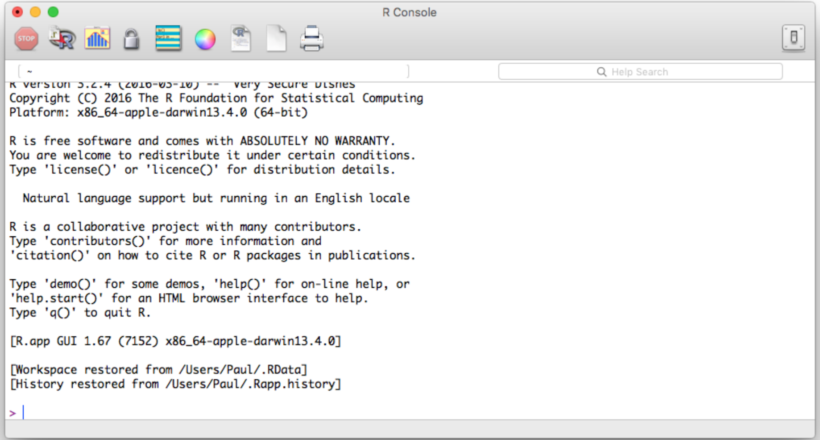
- **Important**: You will have to re-install R from time-to-time to maintain the newest version so that code remains compatible! New versions generally come out every 4 - 6 months.

Installing RStudio

- RStudio has a very nice graphical user interface (GUI) that is easier to use than base R
- We will be using this throughout the course
- Make sure that you have R installed first. Then, download and install at this website:

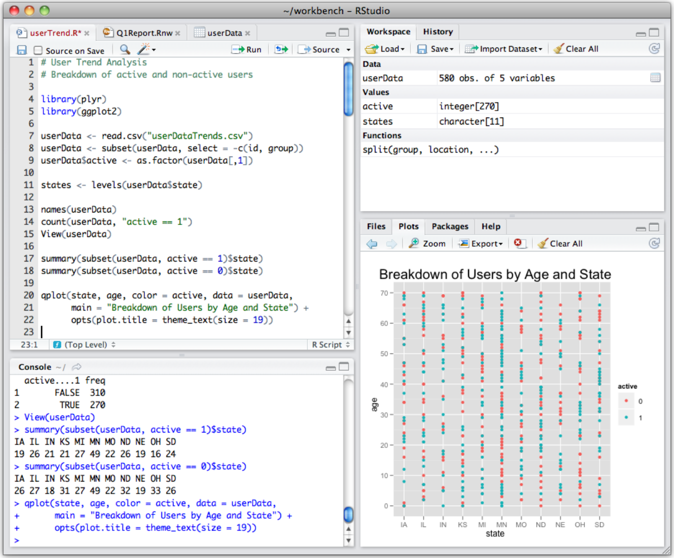
  https://www.rstudio.com/products/RStudio/

The R Graphical User Interface (GUI)



```
R version 3.2.4 (2016-03-10) -- "Very Secure Dishes"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (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.

  Natural language support but running in an English locale

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.

[R.app GUI 1.67 (7152) x86_64-apple-darwin13.4.0]

[Workspace restored from /Users/Paul/.RData]
[History restored from /Users/Paul/.Rapp.history]

> 
```

The RStudio GUI



```r
# User Trend Analysis
# Breakdown of active and non-active users

library(plyr)
library(ggplot2)

userData <- read.csv("userDataTrends.csv")
userData <- subset(userData, select = -c(id, group))
userData$active <- as.factor(userData[,1])

states <- levels(userData$state)

names(userData)
count(userData, "active == 1")
View(userData)

summary(subset(userData, active == 1)$state)
summary(subset(userData, active == 0)$state)

qplot(state, age, color = active, data = userData,
      main = "Breakdown of Users by Age and State") +
      opts(plot.title = theme_text(size = 19))

```

Data	
userData	580 obs. of 5 variables
Values	
active	integer[270]
states	character[11]
Functions	
split(group, location, ...)	

```
  active....1 freq
1       FALSE  310
2        TRUE  270
> View(userData)
> summary(subset(userData, active == 1)$state)
IA IL IN KS MI MN MO ND NE OH SD
19 26 21 21 27 49 22 26 19 16 24
> summary(subset(userData, active == 0)$state)
IA IL IN KS MI MN MO ND NE OH SD
26 27 18 31 27 49 22 32 19 33 26
> qplot(state, age, color = active, data = userData,
+       main = "Breakdown of Users by Age and State") +
+       opts(plot.title = theme_text(size = 19))
>
```

RStudio

RStudio has Four Panels

- Console (bottom left) - where all calculations are performed
- Scripting/Viewing (top left) - where writing of new functions / scripts should be done
- Files/Packages/Help/Plots (bottom right) - for easy analysis
- Variables/Data/Functions (top right) - what is stored in your current Rsession

Really Advanced Calculators

At their core, R and RStudio are just calculators! Try the following

$$3 + 2 = ? \qquad \log(10) = ? \qquad \sqrt{32} = ?$$

```
  > 3 + 2
```

```
> log(10)
```

```
> sqrt(32)
```

Some terms we'll be using

- command: tell R to do something (e.g., add, subtract, print)
- variable: assign a value to an identifier

```
> my_cool_integer <- 5
> my_cool_string <- "hedgehog"
```

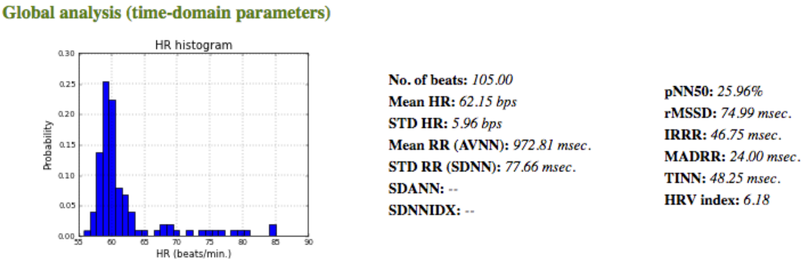
- object: often used interchangeably with variable
- script: a file documenting many commands, can be rerun

Why do we write scripts?

- reproducibility
- share-ability
- automate the boring stuff

Example:

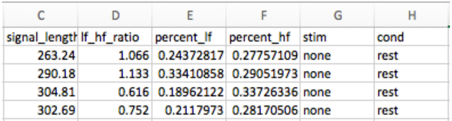
I compare 60 participants' heart rates before and after a stimulus



Global analysis (time-domain parameters)

No. of beats: *105.00*
Mean HR: *62.15 bps*
STD HR: *5.96 bps*
Mean RR (AVNN): *972.81 msec.*
STD RR (SDNN): *77.66 msec.*
SDANN: --
SDNNIDX: --

pNN50: *25.96%*
rMSSD: *74.99 msec.*
IRRR: *46.75 msec.*
MADRR: *24.00 msec.*
TINN: *48.25 msec.*
HRV index: *6.18*



C	D	E	F	G	H
signal_length	lf_hf_ratio	percent_lf	percent_hf	stim	cond
263.24	1.066	0.24372817	0.27757109	none	rest
290.18	1.133	0.33410858	0.29051973	none	rest
304.81	0.616	0.18962122	0.33726336	none	rest
302.69	0.752	0.2117973	0.28170506	none	rest

- function: often used interchangeably with "command"
    - though generally we would call "add(2, 3)" a function but wouldn’t call "2 + 3" a function
    - you will eventually be able to write your own functions!
    - very flexible
- argument: a function takes arguments to perform it’s task (e.g., in "add(2, 3)" the "2" and the "3" are arguments)
    - an argument can be a variable or an option (e.g., number_of_decimals = 2)
- working directory: where you are in the computers file structure

Notes on R

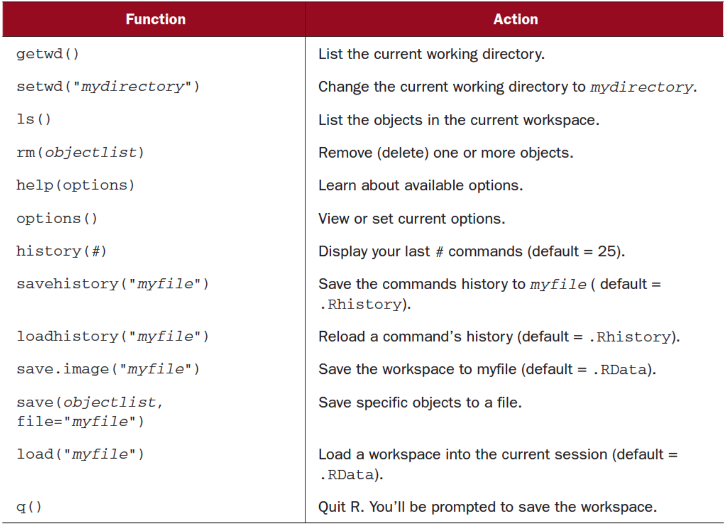
- R is case-sensitive
- I require you to use the assignment operator `'<-'` instead of the equality operator `'='` for all submitted code, even though both work, e.g.,

Syntax	*Comments*
`x <- 5`	standard syntax, required
`x = 5`	poor syntax, not permitted
`5 -> x`	awkward syntax, not permitted (but it works)

Basic R Help Functions

Function	Action
`?foo`	Help on the function `foo`
`??foo`	Search the help system for instances of the function `foo`
`data()`	List all available example datasets contained in currently loaded packages
`getwd()`	List the current working directory
`ls()`	List the objects in the current directory

Basic R Workspace Functions

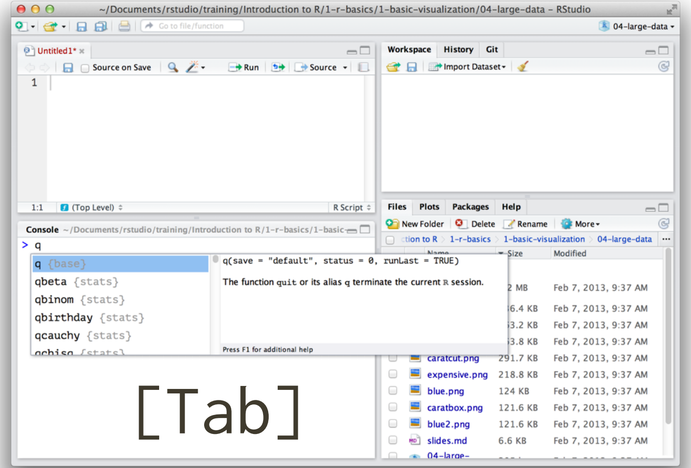


Function	Action
`getwd()`	List the current working directory.
`setwd("`*`mydirectory`*`")`	Change the current working directory to *`mydirectory`*.
`ls()`	List the objects in the current workspace.
`rm(`*`objectlist`*`)`	Remove (delete) one or more objects.
`help(options)`	Learn about available options.
`options()`	View or set current options.
`history(#)`	Display your last # commands (default = 25).
`savehistory("`*`myfile`*`")`	Save the commands history to *`myfile`* (default = `.Rhistory`).
`loadhistory("`*`myfile`*`")`	Reload a command's history (default = `.Rhistory`).
`save.image("`*`myfile`*`")`	Save the workspace to myfile (default = `.RData`).

`save(`*`objectlist`*`,`
`file="`*`myfile`*`")`
	Save specific objects to a file.

`load("`*`myfile`*`")`	Load a workspace into the current session (default = `.RData`).
`q()`	Quit R. You'll be prompted to save the workspace.

Useful R Keyboard Shortcuts: Autocomplete



```
> q
```

```
q {base}
qbeta {stats}
qbinom {stats}
qbirthday {stats}
qcauchy {stats}
qchisq {stats}
```

```
q(save = "default", status = 0, runLast = TRUE)
```

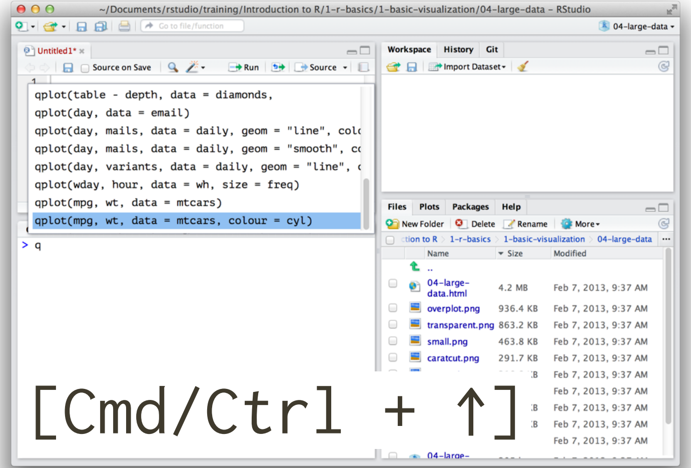
The function quit or its alias q terminate the current R session.

Press F1 for additional help

Name	Size	Modified
	2 MB	Feb 7, 2013, 9:37 AM
	6.4 KB	Feb 7, 2013, 9:37 AM
	3.2 KB	Feb 7, 2013, 9:37 AM
	3.8 KB	Feb 7, 2013, 9:37 AM
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM
expensive.png	218.8 KB	Feb 7, 2013, 9:37 AM
blue.png	124 KB	Feb 7, 2013, 9:37 AM
caratbox.png	121.6 KB	Feb 7, 2013, 9:37 AM
blue2.png	121.6 KB	Feb 7, 2013, 9:37 AM
slides.md	6.6 KB	Feb 7, 2013, 9:37 AM
04-large-		

[Tab]

Useful R Keyboard Shortcuts: History



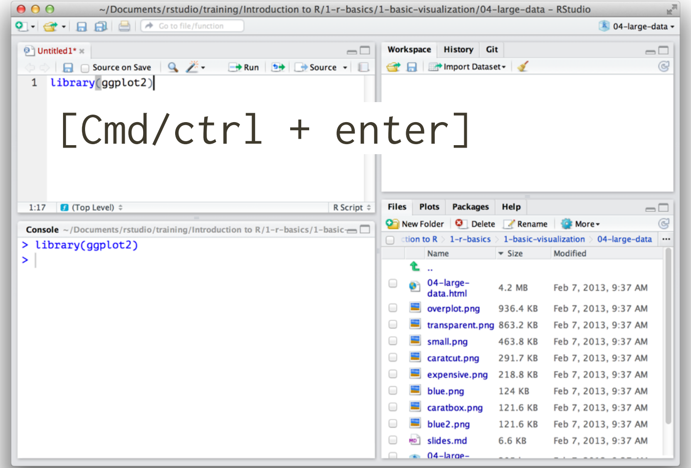
```
qplot(table - depth, data = diamonds,
qplot(day, data = email)
qplot(day, mails, data = daily, geom = "line", colo
qplot(day, mails, data = daily, geom = "smooth", co
qplot(day, variants, data = daily, geom = "line", c
qplot(wday, hour, data = wh, size = freq)
qplot(mpg, wt, data = mtcars)
qplot(mpg, wt, data = mtcars, colour = cyl)
```

```
> q
```

Name	Size	Modified
..		
04-large-data.html	4.2 MB	Feb 7, 2013, 9:37 AM
overplot.png	936.4 KB	Feb 7, 2013, 9:37 AM
transparent.png	863.2 KB	Feb 7, 2013, 9:37 AM
small.png	463.8 KB	Feb 7, 2013, 9:37 AM
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM

[Cmd/Ctrl + ↑]

Useful R Keyboard Shortcuts: Execute Code



~/Documents/rstudio/training/Introduction to R/1-r-basics/1-basic-visualization/04-large-data – RStudio

```
library(ggplot2)
```

[Cmd/ctrl + enter]

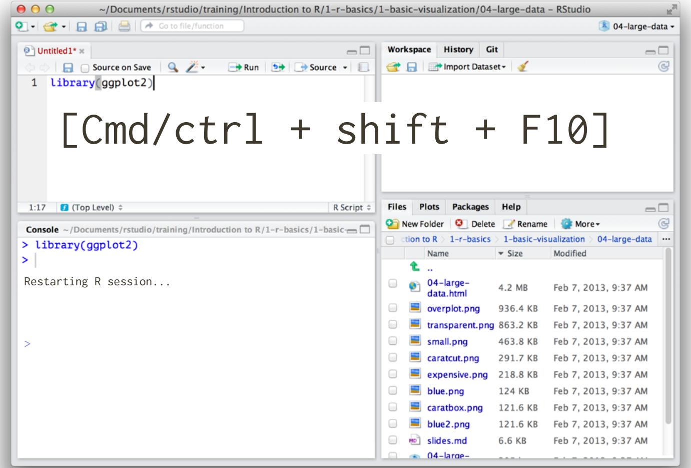
Console ~/Documents/rstudio/training/Introduction to R/1-r-basics/1-basic-

```
> library(ggplot2)
>
```

ction to R › 1-r-basics › 1-basic-visualization › 04-large-data

Name	Size	Modified
..		
04-large-data.html	4.2 MB	Feb 7, 2013, 9:37 AM
overplot.png	936.4 KB	Feb 7, 2013, 9:37 AM
transparent.png	863.2 KB	Feb 7, 2013, 9:37 AM
small.png	463.8 KB	Feb 7, 2013, 9:37 AM
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM
expensive.png	218.8 KB	Feb 7, 2013, 9:37 AM
blue.png	124 KB	Feb 7, 2013, 9:37 AM
caratbox.png	121.6 KB	Feb 7, 2013, 9:37 AM
blue2.png	121.6 KB	Feb 7, 2013, 9:37 AM
slides.md	6.6 KB	Feb 7, 2013, 9:37 AM
04-large-		

Useful R Keyboard Shortcuts: Restarting an R Sess



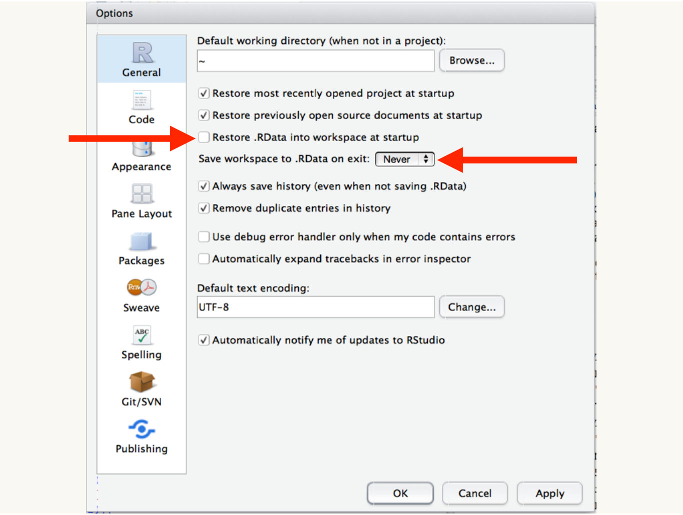
```
library(ggplot2)
```

[Cmd/ctrl + shift + F10]

```
> library(ggplot2)
>

Restarting R session...

>
```

Name	Size	Modified
..		
04-large-data.html	4.2 MB	Feb 7, 2013, 9:37 AM
overplot.png	936.4 KB	Feb 7, 2013, 9:37 AM
transparent.png	863.2 KB	Feb 7, 2013, 9:37 AM
small.png	463.8 KB	Feb 7, 2013, 9:37 AM
caratcut.png	291.7 KB	Feb 7, 2013, 9:37 AM
expensive.png	218.8 KB	Feb 7, 2013, 9:37 AM
blue.png	124 KB	Feb 7, 2013, 9:37 AM
caratbox.png	121.6 KB	Feb 7, 2013, 9:37 AM
blue2.png	121.6 KB	Feb 7, 2013, 9:37 AM
slides.md	6.6 KB	Feb 7, 2013, 9:37 AM
04-large-		

IMPORTANT R Setting



Options

Default working directory (when not in a project):

~

☑ Restore most recently opened project at startup

☑ Restore previously open source documents at startup

□ Restore .RData into workspace at startup

Save workspace to .RData on exit: Never

☑ Always save history (even when not saving .RData)

☑ Remove duplicate entries in history

□ Use debug error handler only when my code contains errors

□ Automatically expand tracebacks in error inspector

Default text encoding:

UTF-8

☑ Automatically notify me of updates to RStudio

A Brief Digression

- When writing code, you want to clear your environment to ensure the fidelity of your results
- If your previous setting is correct, one way to do this is to close and re-open your `R` environment
- You can also run the following two lines of code

```
rm(list=ls())
cat("\014")
```

1. `rm(list=ls())` removes all objects in the current environment
2. On a Mac, `cat("\014")` clear the console windows (same as `ctrl + l`)

If you are sharing code you may not want to include this in your script

Data Sets in R

- R comes built in with multiple data sets you can play with
- Many (most?) packages also have data sets
- `data()` will bring up a list of all data sets available across all loaded packages
- `help(<nameOfDataSet>)` will provide you a detailed description of the data set in question

How Big is *Big Data* in R?

- R holds data in memory, effectively limiting data to the amount of RAM a computer has access to
- It is not uncommon to work with a data set containing 100,000,000 elements (e.g., 100,000 observations of 1,000 variables or 1,000,000 observations of 100 variables) without difficulty
- Approximations depend on what type of data is contained in each variable, e.g., a data set with 2.2 million records and twenty variables, which takes approximately one minute to load into memory

How Big is *Big Data* in R?

- Also depends on what techniques and/or functions will be applied to the data
- The more complex and memory intensive the task, the smaller the data will be required to be
- Basic plotting requires far less computational exertion than a complex statistical learning model
- **Common Definition**: *Big Data* refers to any data set that cannot be loaded into working memory on your personal computer