# ST 555: Statistical Programming I

Intro to R

Dr. Reneé H. Moore

# Outline of Module

- What is R?
- How to Install R?
- R Packages
- Running your  R programs
  - At the Prompt
  - Using Rstudio
- Assigning an Object in R
- Basic Commands in R
- Use R as a Calculator

# Intro: What is R?

- A statistical language & environment for statistical computing and graphics
- FREE software, built and contributed by public users
  - Maintained by a small group of statisticians known as R Development Core Team
  - Large group of volunteers contribute add-on packages
- Available in UNIX, Windows, Linux, MacOS
- Based on S language (S-plus)

# The R Environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

# The R Environment

- The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

- Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

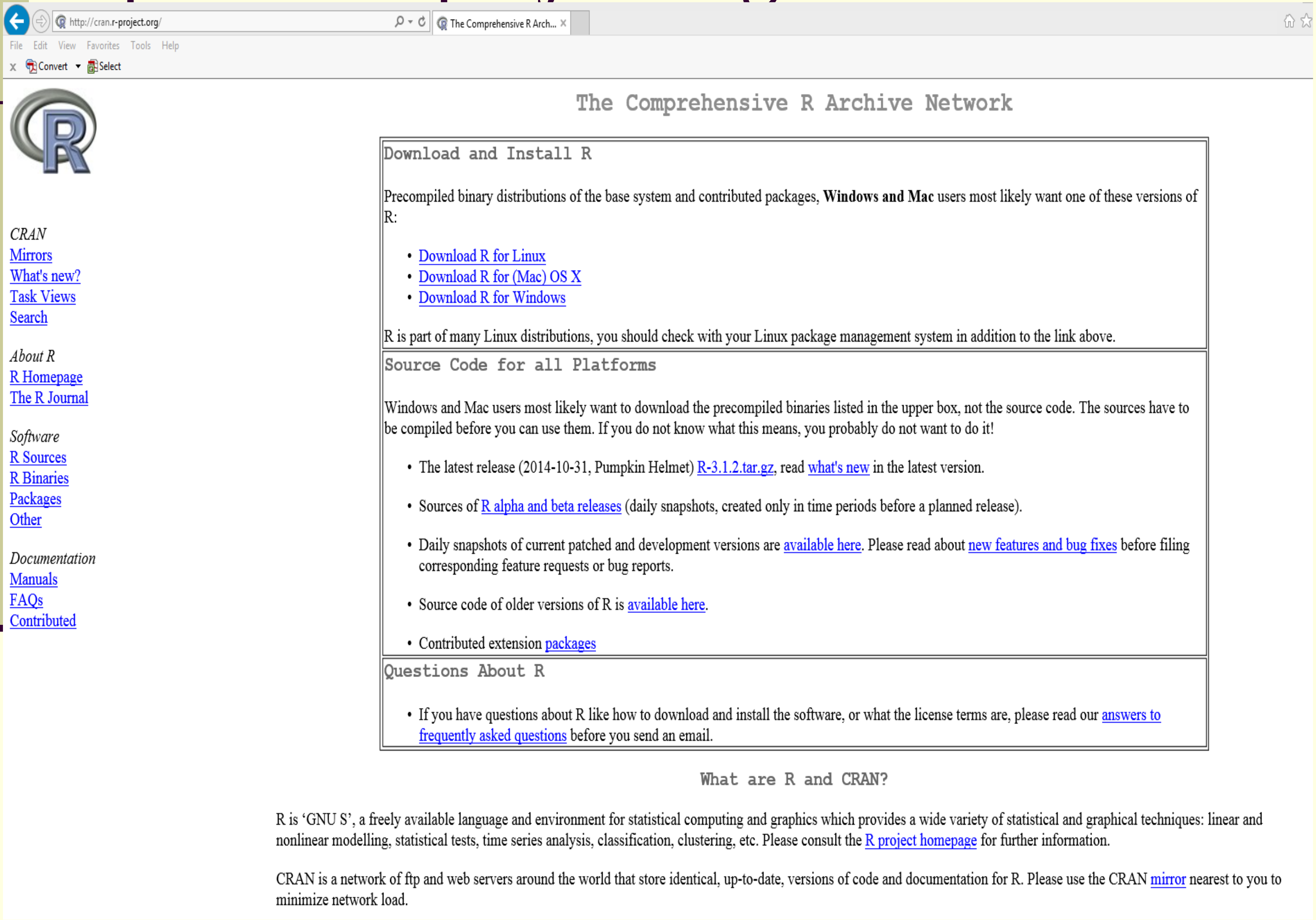
- CRAN= Comprehensive R Archive Network

# Intro: Installing R

- Two major parts: base program & add-on packages
- Download Comprehensive R Archive Network (CRAN) at

  http://CRAN.R-project.org
- Currently base is version R 3.1.2 (released 2014-10-31)
- Once downloaded to begin,
  - In Unix, type R on shell
  - In Windows, click on R icon (top of page)

# http://cran.r-project.org/



http://cran.r-project.org/
The Comprehensive R Arch...
File Edit View Favorites Tools Help
Convert
Select

## The Comprehensive R Archive Network

*CRAN*
Mirrors
What's new?
Task Views
Search

*About R*
R Homepage
The R Journal

*Software*
R Sources
R Binaries
Packages
Other

*Documentation*
Manuals
FAQs
Contributed

### Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

### Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2014-10-31, Pumpkin Helmet) R-3.1.2.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

### Questions About R

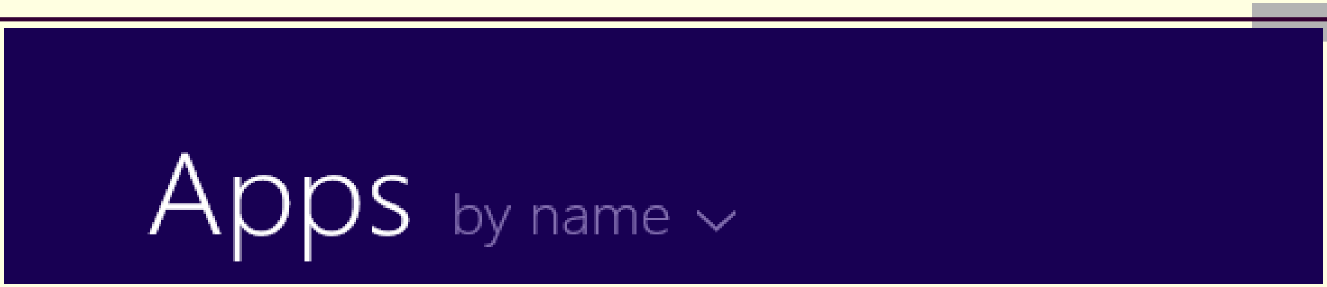
- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

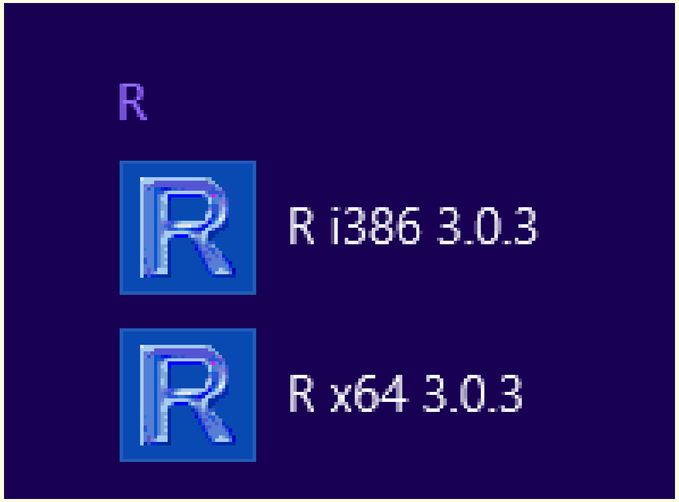
### What are R and CRAN?

R is ‘GNU S’, a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the R project homepage for further information.

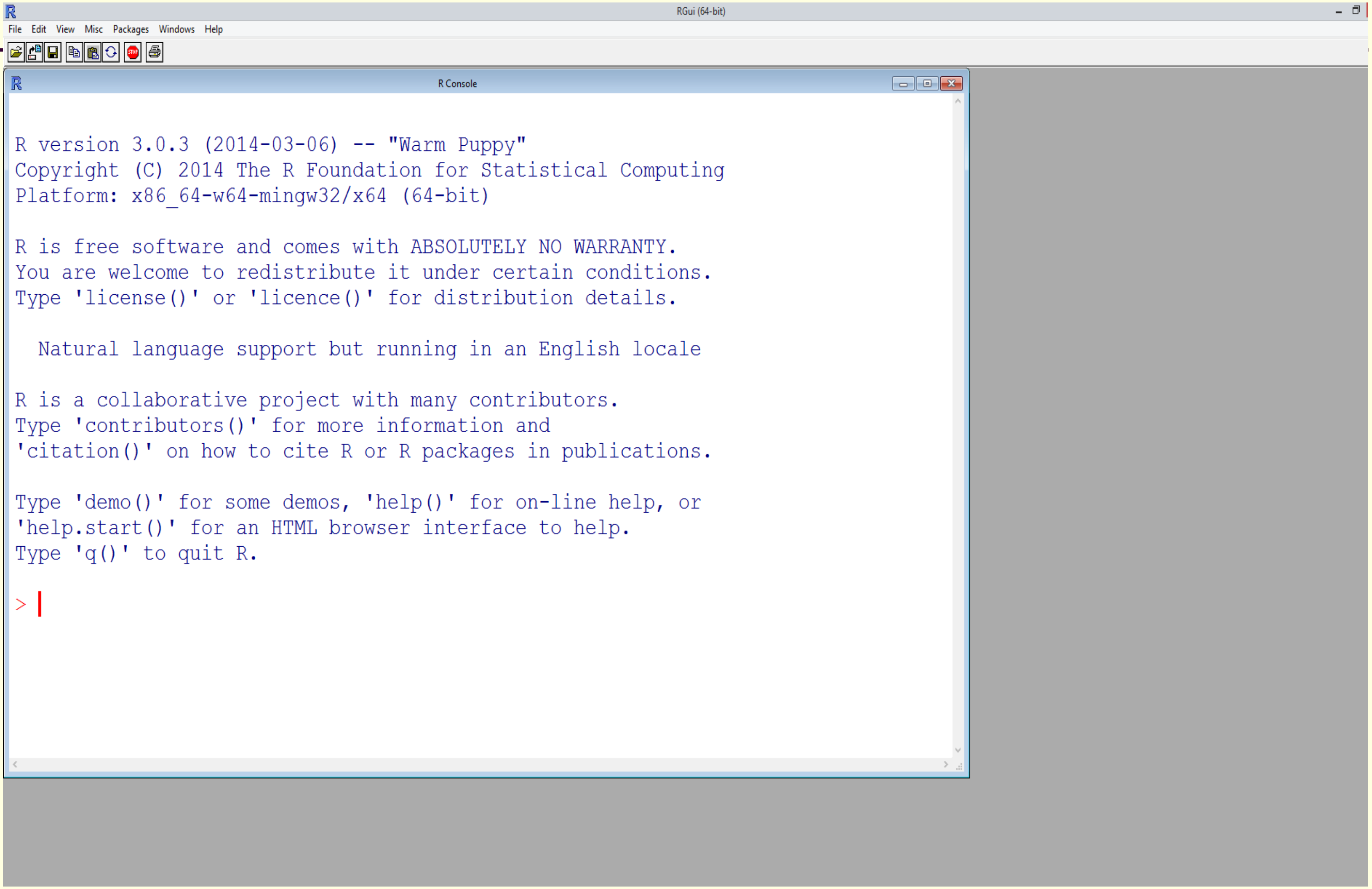
CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN mirror nearest to you to minimize network load.

# R Installed





# R Installed



```
R version 3.0.3 (2014-03-06) -- "Warm Puppy"
Copyright (C) 2014 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.

  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 Add-on packages

- Over 6062 packages (reported 11/13/2014)

To find out which packages are available on your system, type

- library( ) at the prompt

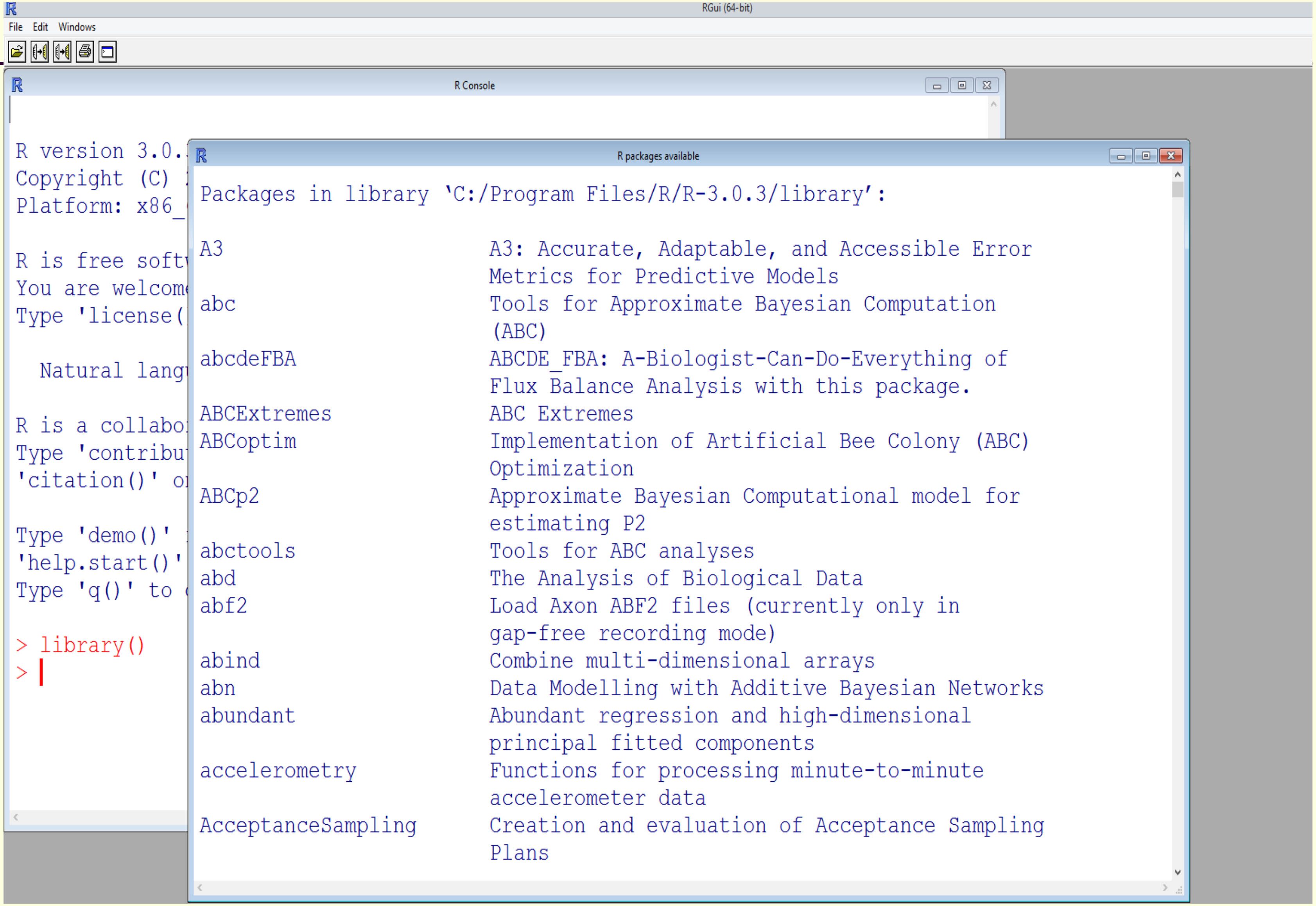
You can “load” an installed package “pkg” by

- library(pkg)

You can then find out which functions it provides by typing one

```
>library(help = pkg)
>help(package = pkg)
```

# R Add-on packages Installed

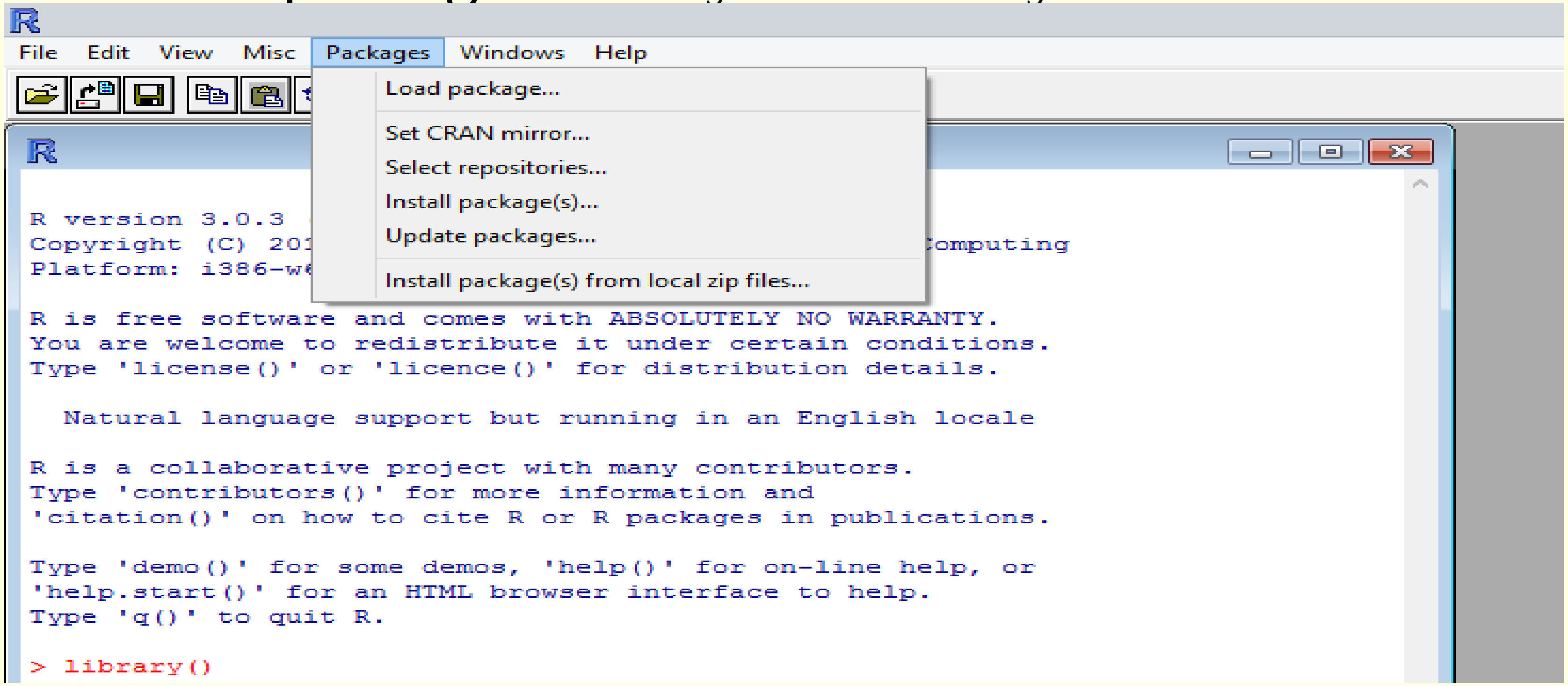


```
R version 3.0.
Copyright (C)
Platform: x86_

R is free soft
You are welcom
Type 'license(

  Natural lang

R is a collabo
Type 'contribu
'citation()' o

Type 'demo()'
'help.start()'
Type 'q()' to

> library()
> 
```

```
Packages in library 'C:/Program Files/R/R-3.0.3/library':

A3                      A3: Accurate, Adaptable, and Accessible Error
                        Metrics for Predictive Models
abc                     Tools for Approximate Bayesian Computation
                        (ABC)
abcdeFBA                ABCDE_FBA: A-Biologist-Can-Do-Everything of
                        Flux Balance Analysis with this package.
ABCExtremes             ABC Extremes
ABCoptim                Implementation of Artificial Bee Colony (ABC)
                        Optimization
ABCp2                   Approximate Bayesian Computational model for
                        estimating P2
abctools                Tools for ABC analyses
abd                     The Analysis of Biological Data
abf2                    Load Axon ABF2 files (currently only in
                        gap-free recording mode)
abind                   Combine multi-dimensional arrays
abn                     Data Modelling with Additive Bayesian Networks
abundant                Abundant regression and high-dimensional
                        principal fitted components
accelerometry           Functions for processing minute-to-minute
                        accelerometer data
AcceptanceSampling      Creation and evaluation of Acceptance Sampling
                        Plans
```

# R Add-on packages

- Need a package not in your Library

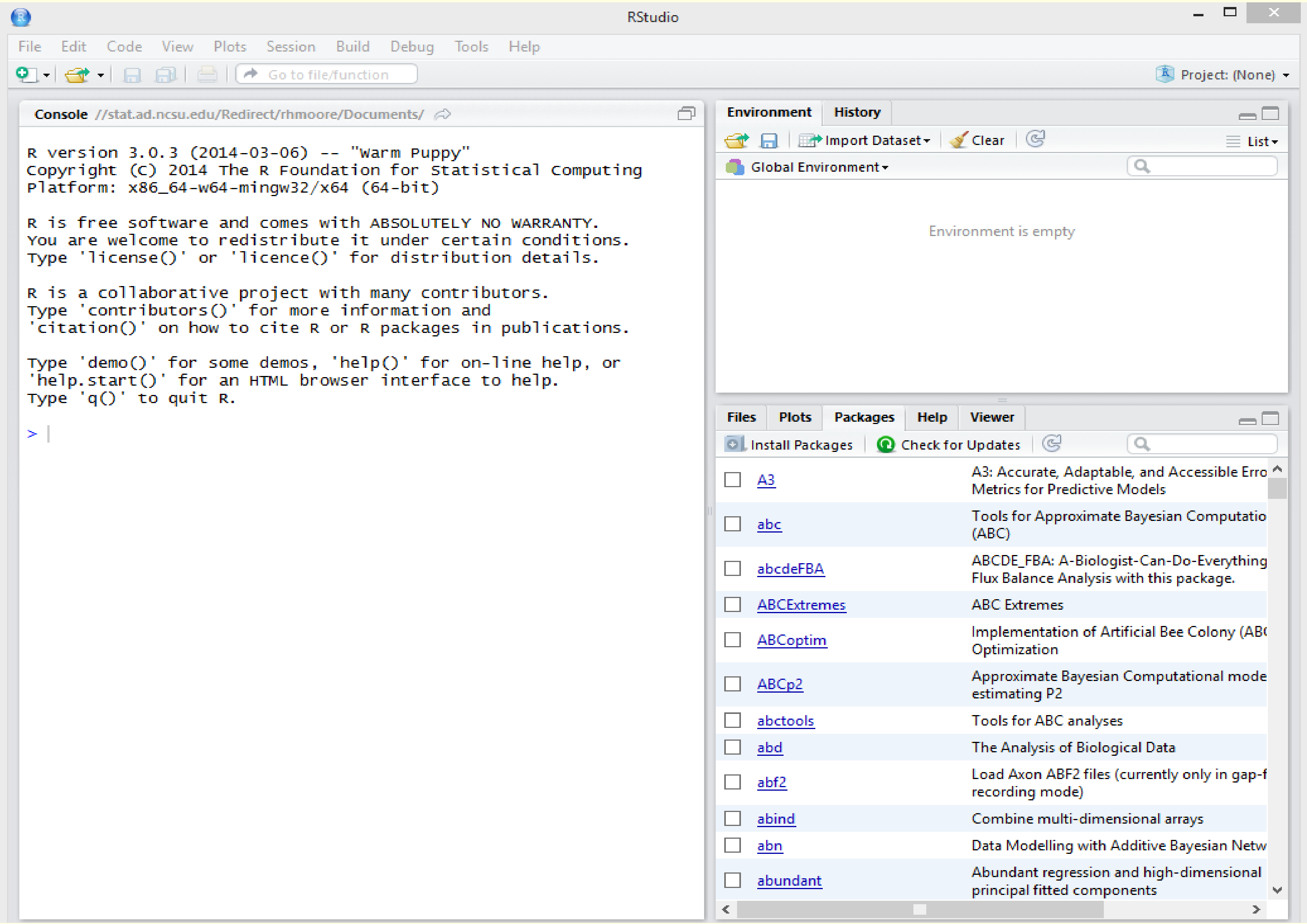


```
R version 3.0.3
Copyright (C) 201                                          omputing
Platform: i386-w6

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.

> library()
```

# Intro: Forums for Running your Program

- Run your “program of codes” by
  - Just type in prompt
  - a text file (e.g. Notepad, MikTek)
  - Open Script file
  - Latest “Sexy Platform”
    - Rcmdr (2009)
    - RStudio (2015):

# RStudio



```
R version 3.0.3 (2014-03-06) -- "Warm Puppy"
Copyright (C) 2014 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.

> 
```

| | | |
|---|---|---|
| □ | A3 | A3: Accurate, Adaptable, and Accessible Erro Metrics for Predictive Models |
| □ | abc | Tools for Approximate Bayesian Computatio (ABC) |
| □ | abcdeFBA | ABCDE_FBA: A-Biologist-Can-Do-Everything Flux Balance Analysis with this package. |
| □ | ABCExtremes | ABC Extremes |
| □ | ABCoptim | Implementation of Artificial Bee Colony (AB Optimization |
| □ | ABCp2 | Approximate Bayesian Computational mode estimating P2 |
| □ | abctools | Tools for ABC analyses |
| □ | abd | The Analysis of Biological Data |
| □ | abf2 | Load Axon ABF2 files (currently only in gap-f recording mode) |
| □ | abind | Combine multi-dimensional arrays |
| □ | abn | Data Modelling with Additive Bayesian Netw |
| □ | abundant | Abundant regression and high-dimensional principal fitted components |

# The Basics

- Case- Sensitive: $X \neq x$
  - Commands are in lower case
  - Logical Values (T or F) are in upper case
- Type commands after the prompt “>” or “R>”
- Results follow the return prompt “[1]”
- To type in a comment on the same line as a command, precede all comments with a pound (#)

Example: > X<- 3 ## here we assign X the value of 3

# Assigning an Object

- Two Ways
  - X <- 3  (Recommended)
  - X=3

- Use “<- “ to avoid confusion with arguments to function calls

# Useful Basic Commands

- help.search(“mean”)  ##another help command
  - Provides help files with alias or concept or title matching 'mean' using regular expression matching:
- ls()  ## lists data objects available in your session
- objects()  ## lists data objects available in your session
- search()  ## tells you which libraries and data frames are available
- attach()  ## make all variables in the dataset visible to R

# More Basic Commands

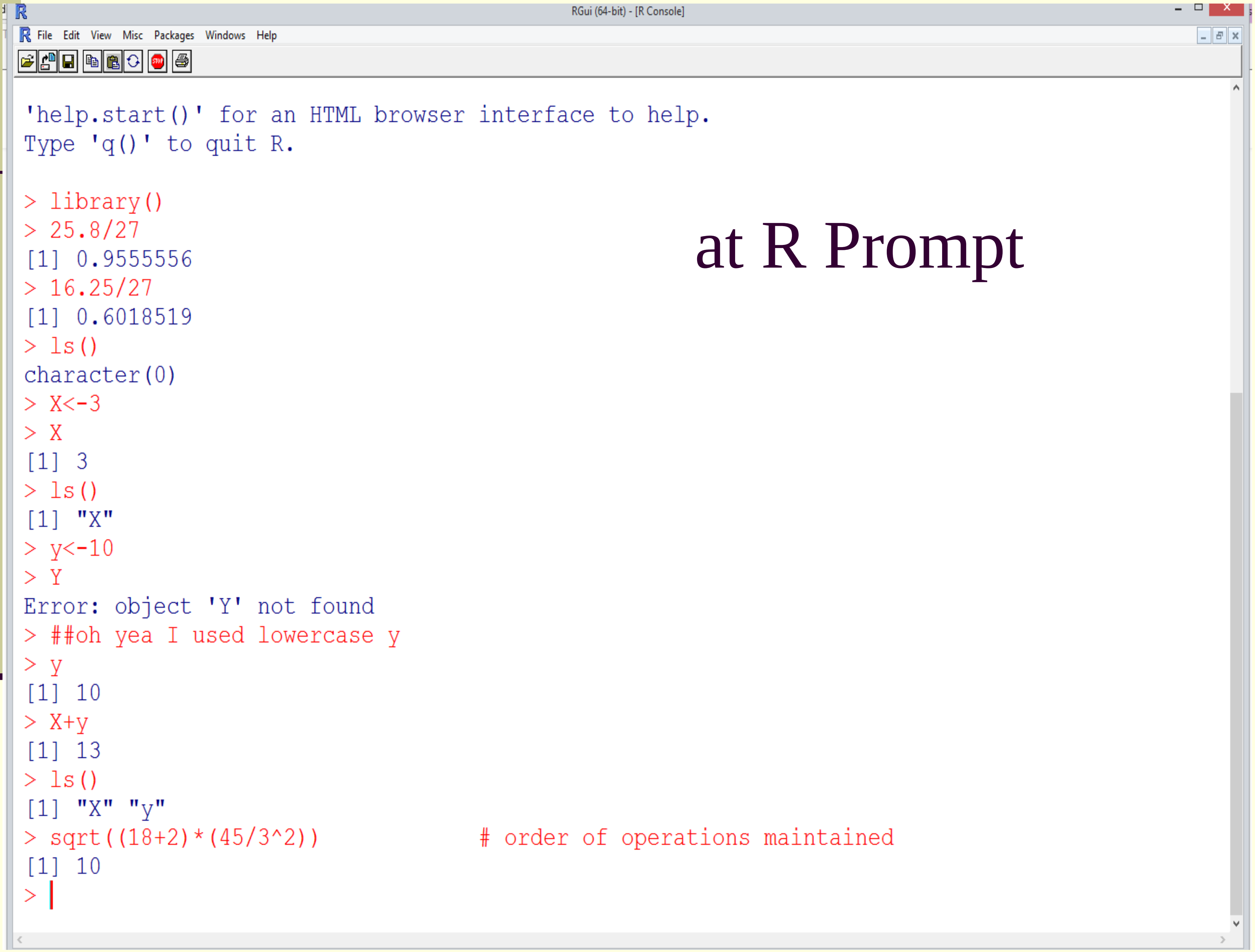
- detach() ## make the dataset unavailable again
- rm(object) ## removes/deletes the variable “object” from the session
- rm(list=ls()) ## removes ALL variables from session, so you have a clear new session
- q() ## Quit R/S-Plus , automatically saves all variables not removed
  - [Previously saved workspace restored] ## if get this notation after opening R it means there are variables saved in the workspace from the previous session

# Intro: R as a Calculator

| **Operation** | **Symbol** | **Example** |
|---|---|---|
| Arithmetic ops. | +, -, *, / | a + b |
| Exponentiation | ^ | b^2 |
| Square root | sqrt() | sqrt(b) |
| Matrix Multiplication | %*% | a %*% b |

Example:

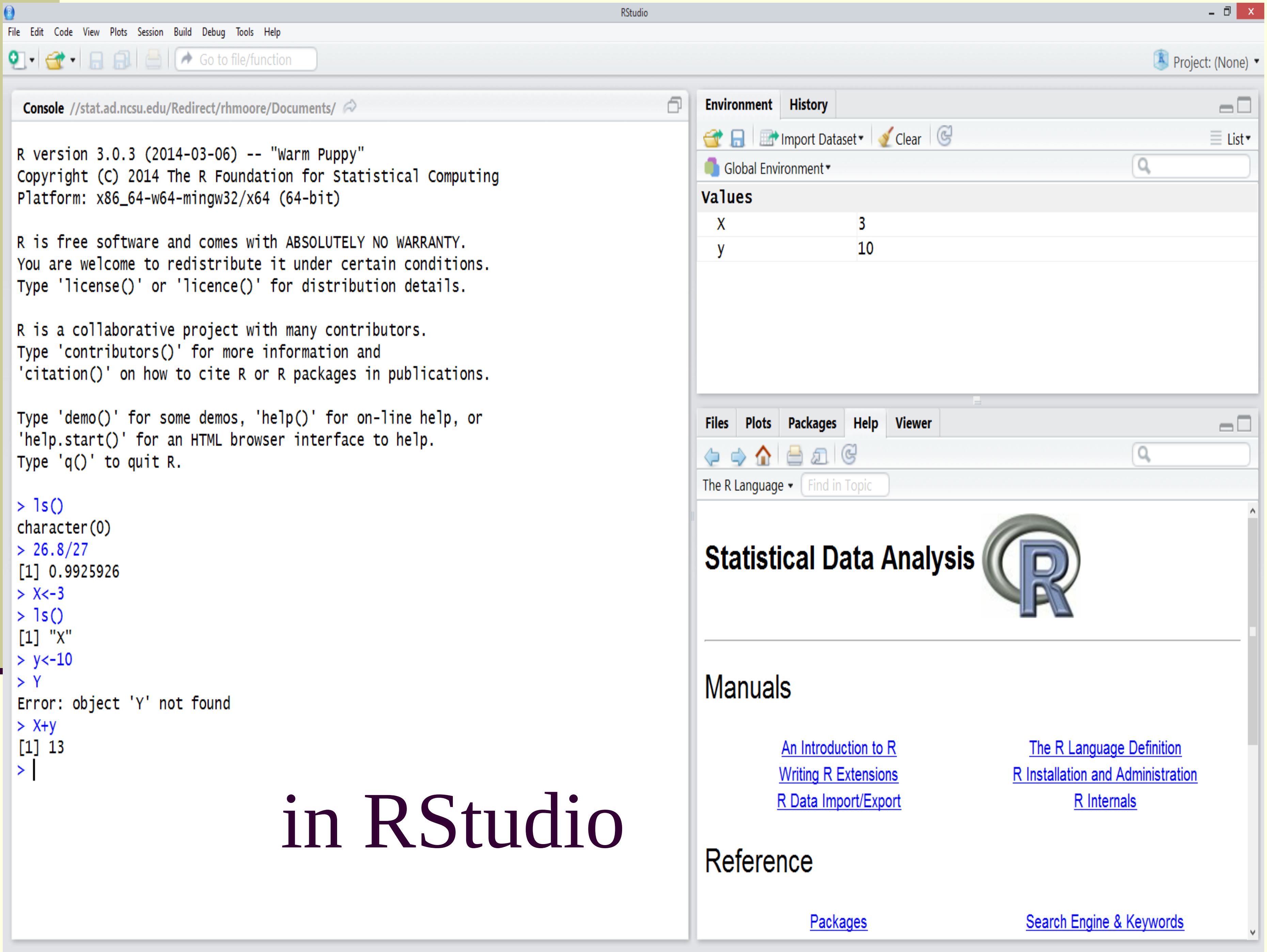
```
>   sqrt((18+2)*(45/3^2))           # order of operations maintained
[1] 10
```



RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help

```
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library()
> 25.8/27
[1] 0.9555556
> 16.25/27
[1] 0.6018519
> ls()
character(0)
> X<-3
> X
[1] 3
> ls()
[1] "X"
> y<-10
> Y
Error: object 'Y' not found
> ##oh yea I used lowercase y
> y
[1] 10
> X+y
[1] 13
> ls()
[1] "X" "y"
> sqrt((18+2)*(45/3^2))           # order of operations maintained
[1] 10
> 
```

at R Prompt


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Project: (None)
Console //stat.ad.ncsu.edu/Redirect/rhmoore/Documents/
Environment History
Import Dataset
Clear
List
Global Environment
Files Plots Packages Help Viewer
The R Language
Find in Topic

```
R version 3.0.3 (2014-03-06) -- "Warm Puppy"
Copyright (C) 2014 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.

> ls()
character(0)
> 26.8/27
[1] 0.9925926
> X<-3
> ls()
[1] "X"
> y<-10
> Y
Error: object 'Y' not found
> X+y
[1] 13
>
```

Values

```
X     3
y     10
```

Statistical Data Analysis

Manuals

An Introduction to R
The R Language Definition
Writing R Extensions
R Installation and Administration
R Data Import/Export
R Internals

Reference

Packages
Search Engine & Keywords

# in RStudio

# Help and Documentation

- http://CRAN.R-project.org/manuals.html
  - An Introduction to R
  - R Data Import/Export
- Within R, can type
  - help.start()
    - Manuals, Reference, Packages, Search Engine, FAQs
  - help() opens up window w tabs for Contents, Index, Search
  - help(name of a topic), e.g. help(mean)

# ST 555: Statistical Programming I

RStudio part 1

Bo “Paul” Ning

Dr. Reneé H. Moore

# Outline

- Introduce to RStudio
- Install RStudio
- Change RStudio settings

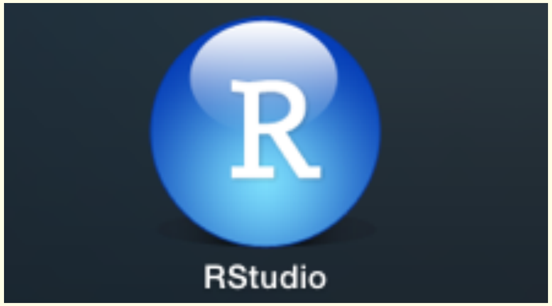
# What is RStudio?

- RStudio is the premier integrated development environment (IDE) for R
- It is FREE!
- User-friendly, easy to learn
- Open source and free to write R packages
- Available in both open source and commercial editions on the desktop (Windows, Mac, and Linux)
- Includes powerful coding tools designed to enhance your productivity
- Supports R Markdown, R Sweave, R Presentation

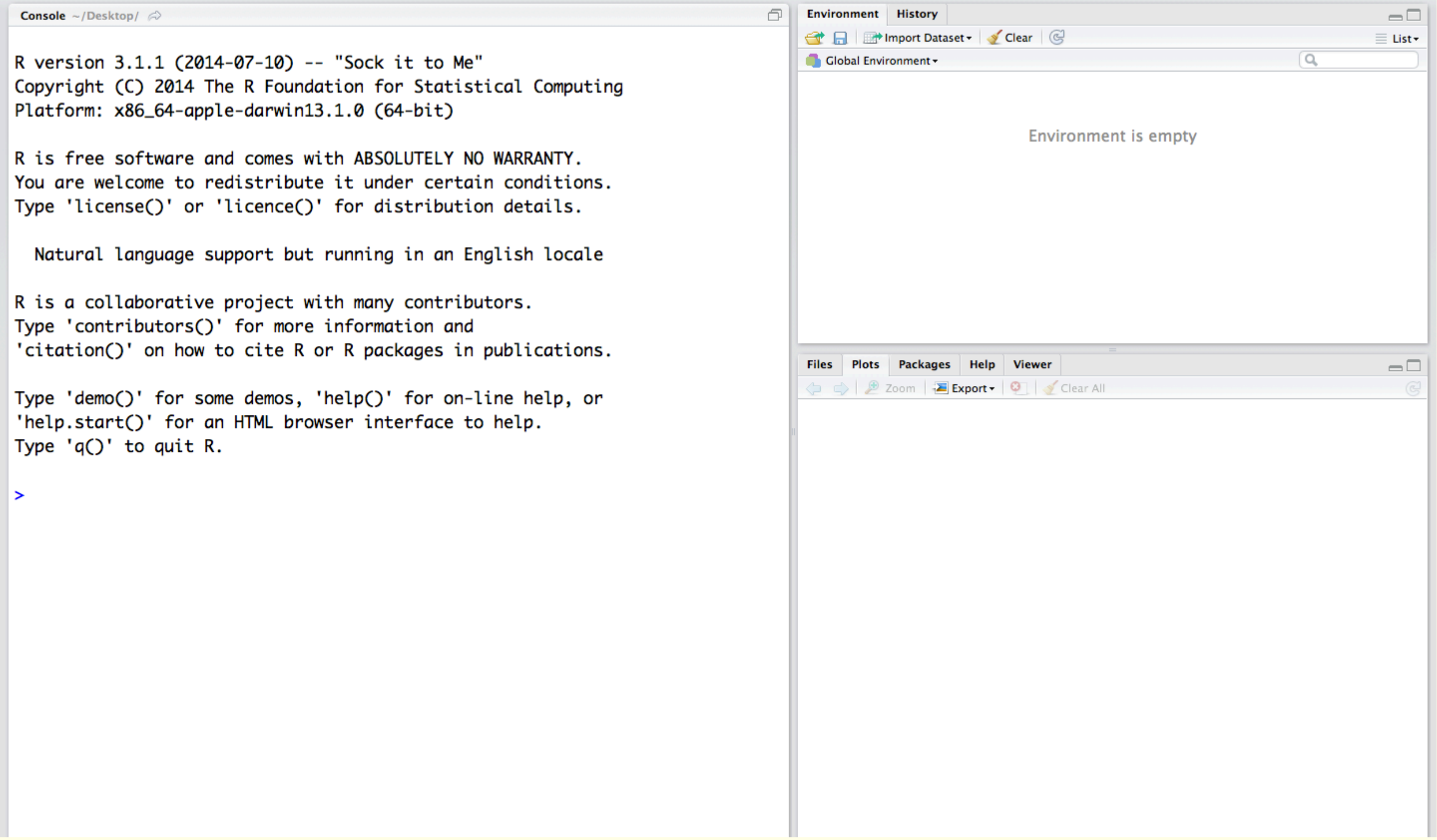
# Installing RStudio

- Download regular release or preview version
- Regular release: http://www.rstudio.com/products/rstudio/
- Preview version: http://www.rstudio.com/products/rstudio/download/preview/

After installing, you could see this sign in your desktop

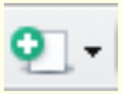
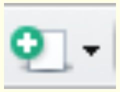
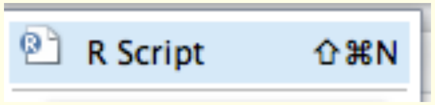


# Installing RStudio



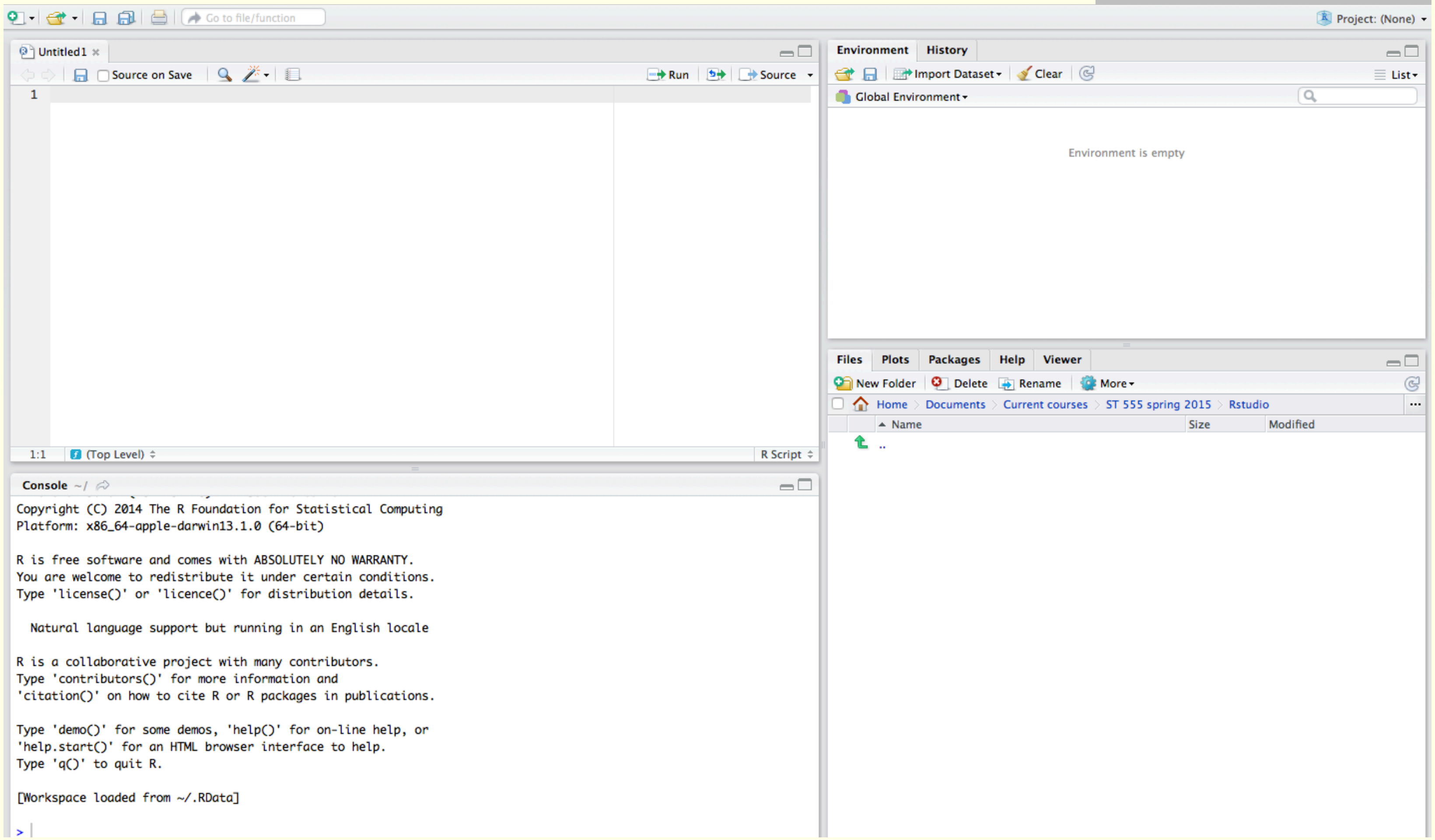
```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.1.0 (64-bit)

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.

>
```

# Add a new R Script

- Find out  button in the left upper side of RStudio.
- Click  , then click 

R Script ⇧⌘N

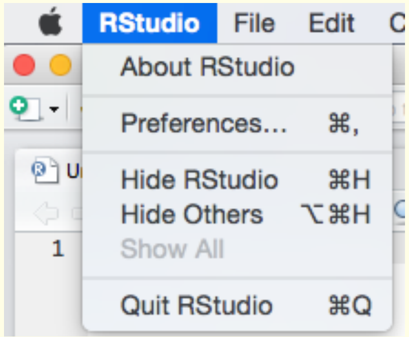
# Add a new R Script



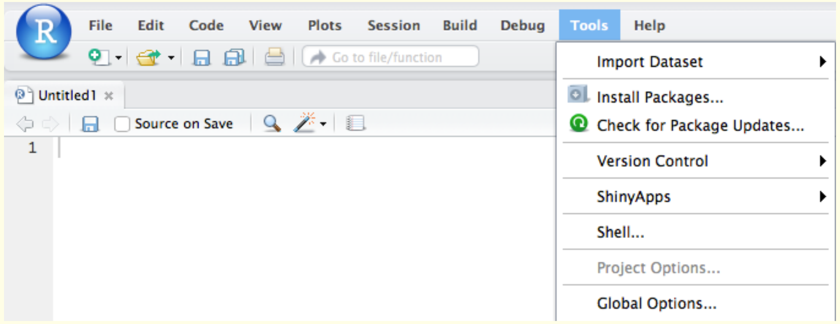
```
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.1.0 (64-bit)

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.

[Workspace loaded from ~/.RData]

>
```

# Change preferences

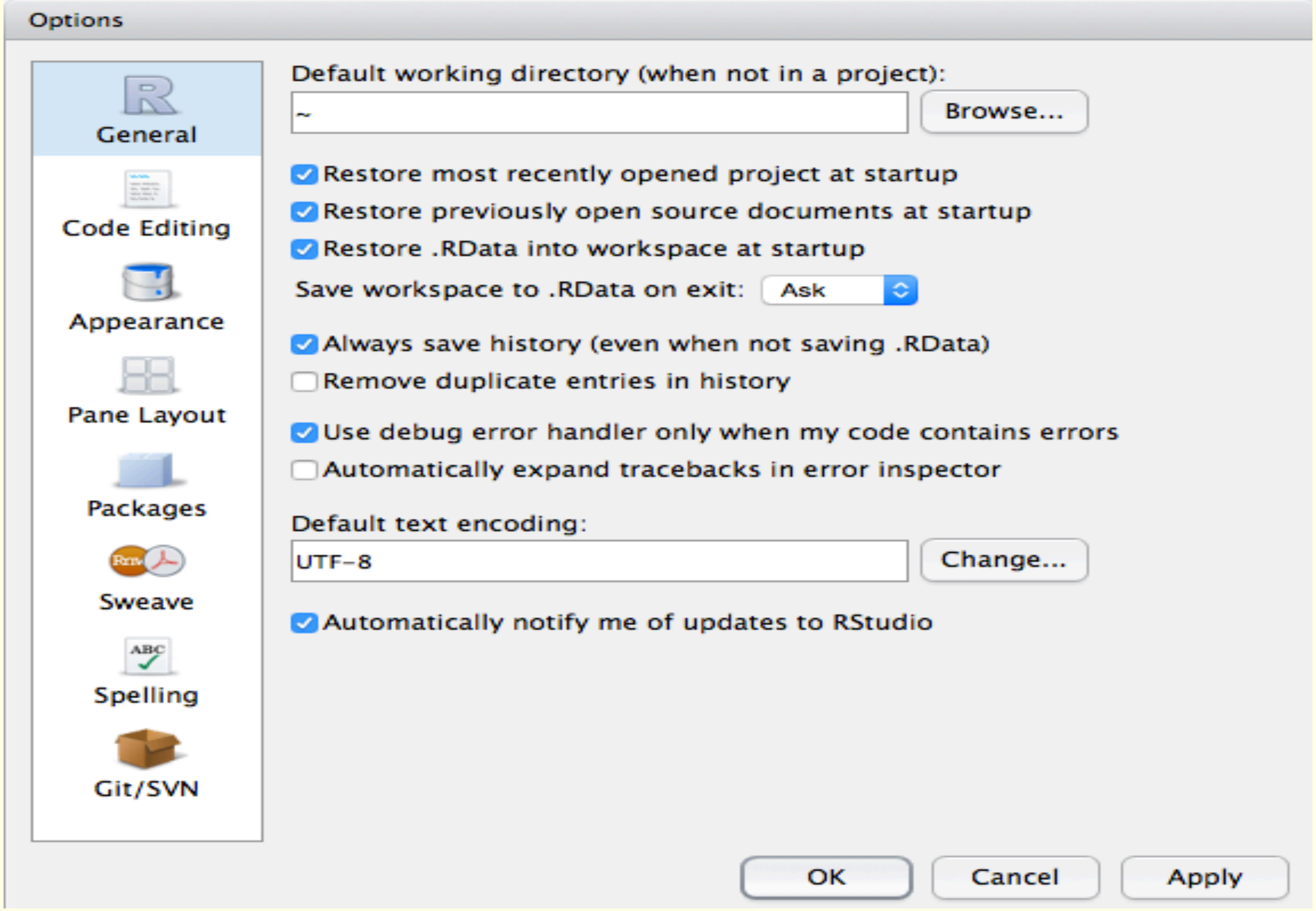
For mac user, click RStudio button, then click "Preferences"



For windows user, click Tools button, then click "Global Options"



# Change preferences



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: Ask

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

□ Remove duplicate entries in history

☑ Use debug error handler only when my code contains errors

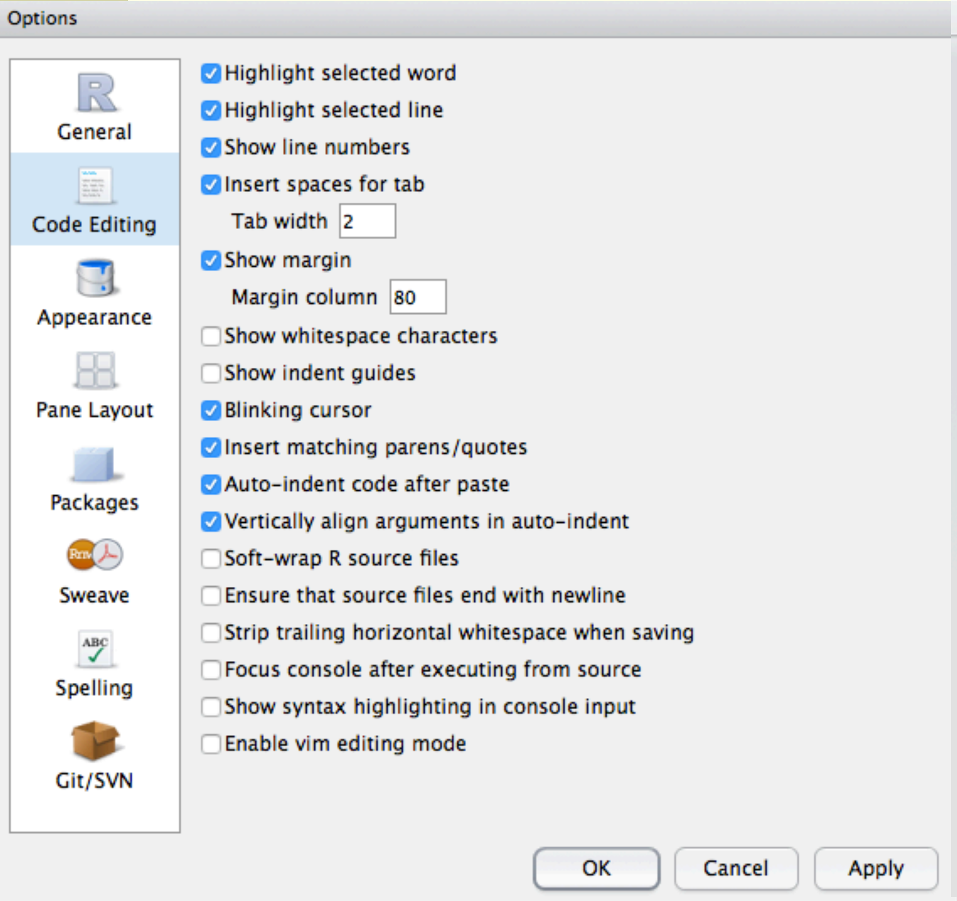
□ Automatically expand tracebacks in error inspector

Default text encoding:

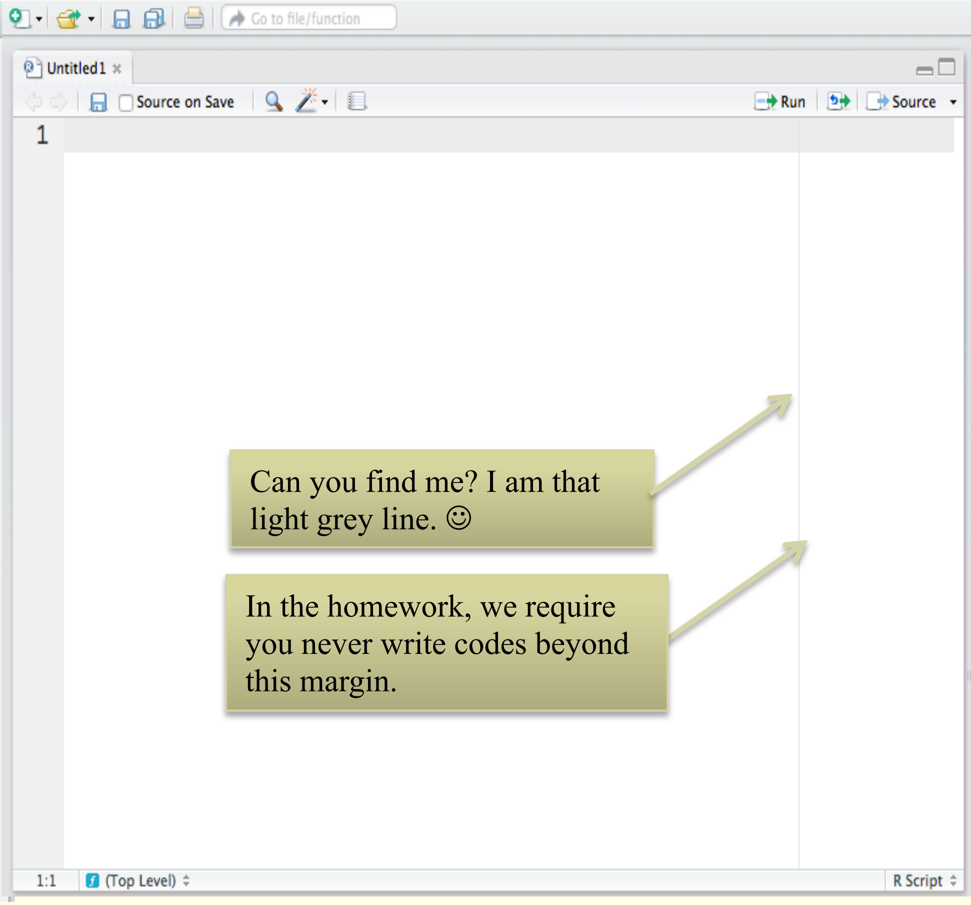
UTF-8

☑ Automatically notify me of updates to RStudio

# Change preferences



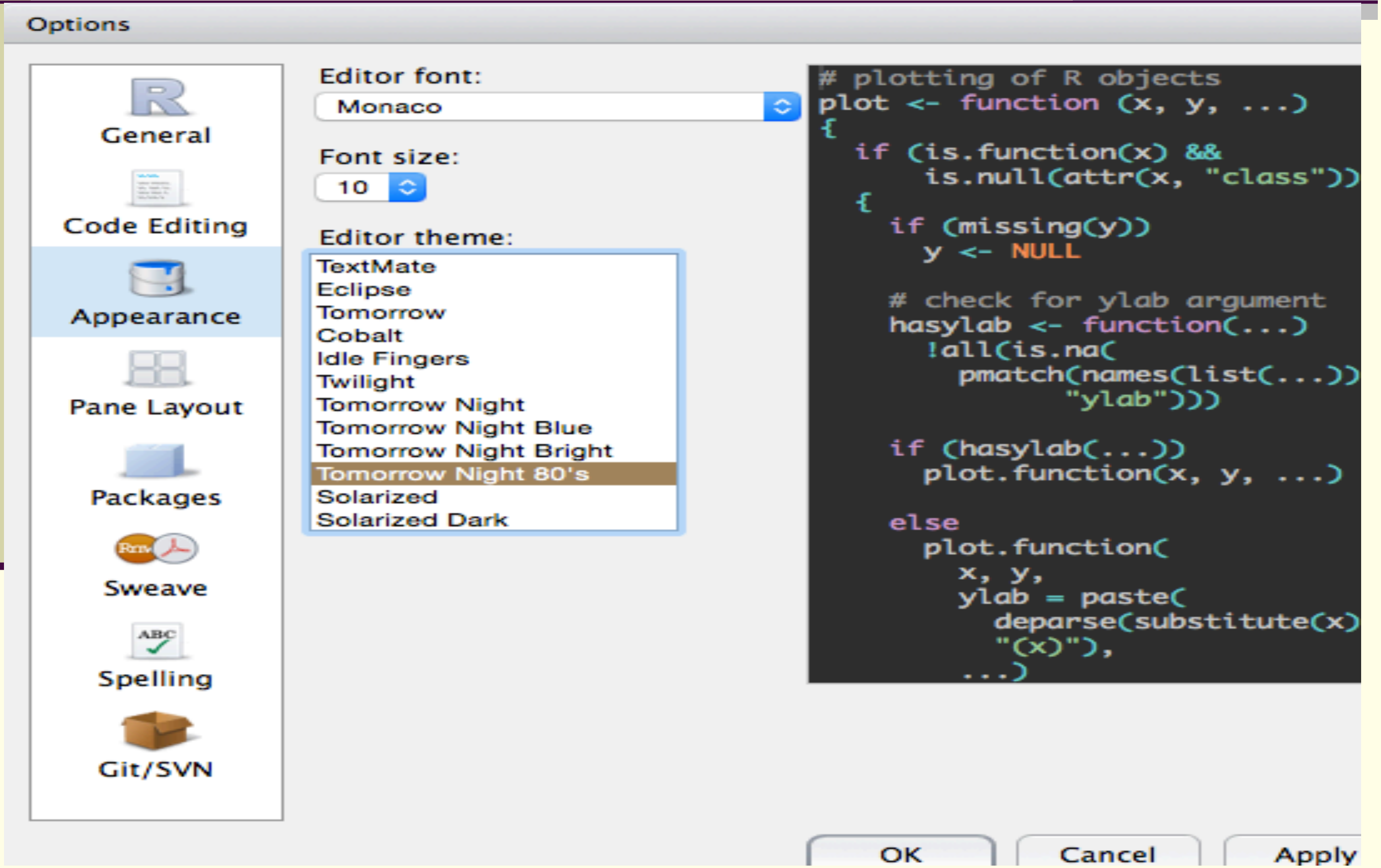
☑ Highlight selected word
☑ Highlight selected line
☑ Show line numbers
☑ Insert spaces for tab
Tab width 2
☑ Show margin
Margin column 80
□ Show whitespace characters
□ Show indent guides
☑ Blinking cursor
☑ Insert matching parens/quotes
☑ Auto-indent code after paste
☑ Vertically align arguments in auto-indent
□ Soft-wrap R source files
□ Ensure that source files end with newline
□ Strip trailing horizontal whitespace when saving
□ Focus console after executing from source
□ Show syntax highlighting in console input
□ Enable vim editing mode



Can you find me? I am that light grey line. ☺

In the homework, we require you never write codes beyond this margin.

# Change Rstudio appearance



Editor font: Monaco

Font size: 10

Editor theme:

- TextMate
- Eclipse
- Tomorrow
- Cobalt
- Idle Fingers
- Twilight
- Tomorrow Night
- Tomorrow Night Blue
- Tomorrow Night Bright
- [✓ Tomorrow Night 80's]
- Solarized
- Solarized Dark

```
# plotting of R objects
plot <- function (x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class"))
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...))
               "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...)

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(x)
          "(x)"),
        ...)
```

# Exercises: makes changes in RStudio

- 1. Create a new R script
- 2. In the “Preferences” or “Global options” do the following:

  i. Find out the following options and click them

  “Highlight selected word”

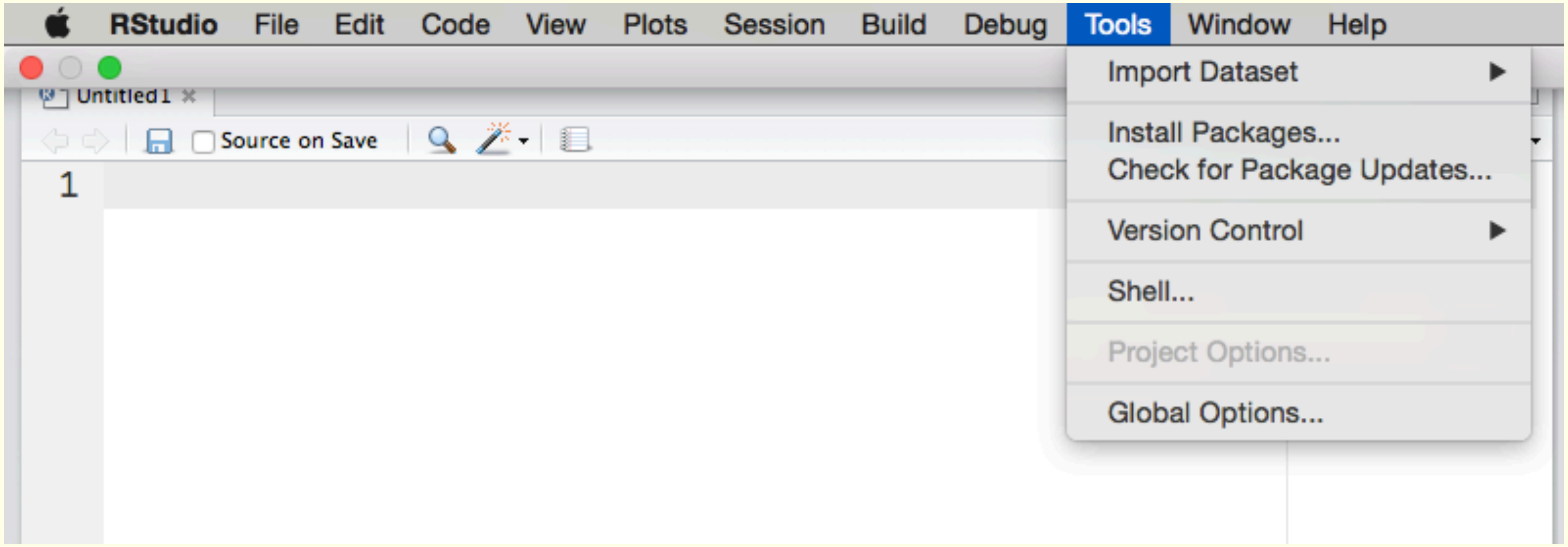
  “Highlight selected line”

  “Show line numbers”

  ii. Change your RStudio`s appearance

# Install packages

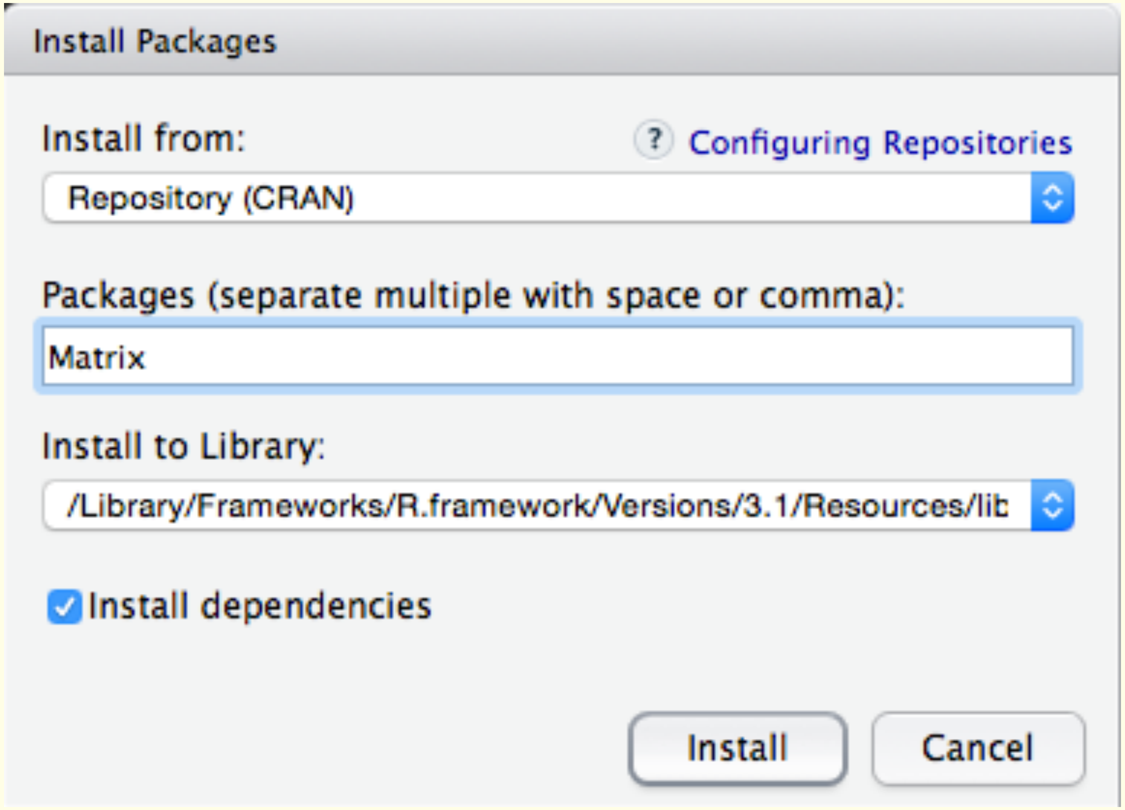
Click “Tools”, then click “Install Packages…”



# Install packages

For example, to install “Matrix” packages.

Choose Install from to Repository (CRAN), in the Packages, type package`s name “Matrix”, click Install.



Install Packages

Install from: Repository (CRAN)

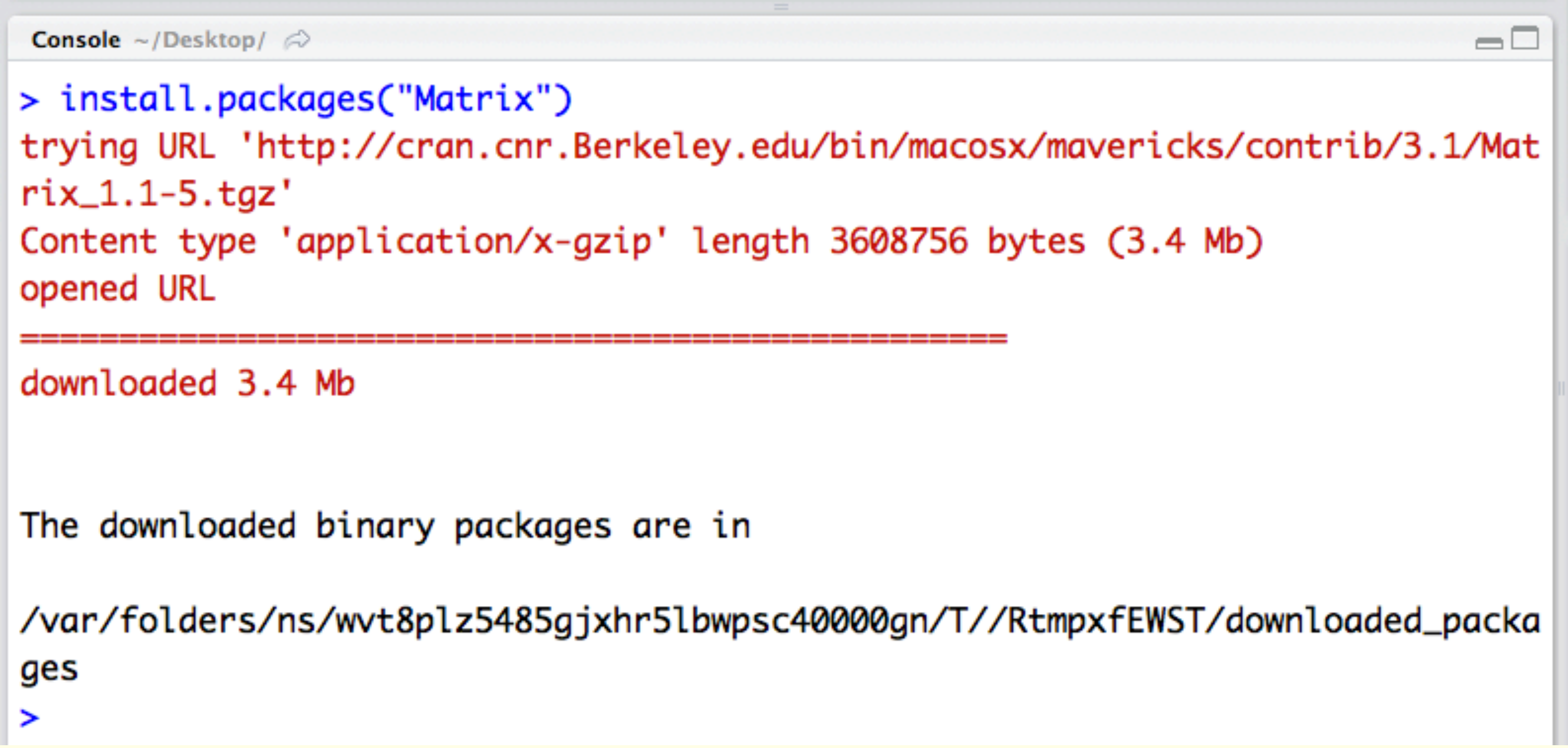
Packages (separate multiple with space or comma): Matrix

Install to Library: /Library/Frameworks/R.framework/Versions/3.1/Resources/lib

☑ Install dependencies

# Install packages

- In Console, you can find out if the package was installed or not.



```
> install.packages("Matrix")
trying URL 'http://cran.cnr.Berkeley.edu/bin/macosx/mavericks/contrib/3.1/Mat
rix_1.1-5.tgz'
Content type 'application/x-gzip' length 3608756 bytes (3.4 Mb)
opened URL
==================================================
downloaded 3.4 Mb


The downloaded binary packages are in

/var/folders/ns/wvt8plz5485gjxhr5lbwpsc40000gn/T//RtmpxfEWST/downloaded_packa
ges
>
```

# Install packages

Another way to install packages in RStudio is to type

```
> install.packages("Matrix")
```

in the Script or Console.

Note the package`s name is case sensitive.

# Exercises: Install packages

1. Try to install the following packages in R studio

   Matrix

   base

   ggplot2

2. Now, let`s try to install R packages in another way. Some R packages are not in “Repository (CRAN)”, we need to download from website and install from “Package Archive File”

   Download the “glmnet” package from

   http://cran.r-project.org/web/packages/glmnet/index.html

   Try to install from Package Archive File.

# ST 555:
# Statistical Programming I

## RStudio part 2

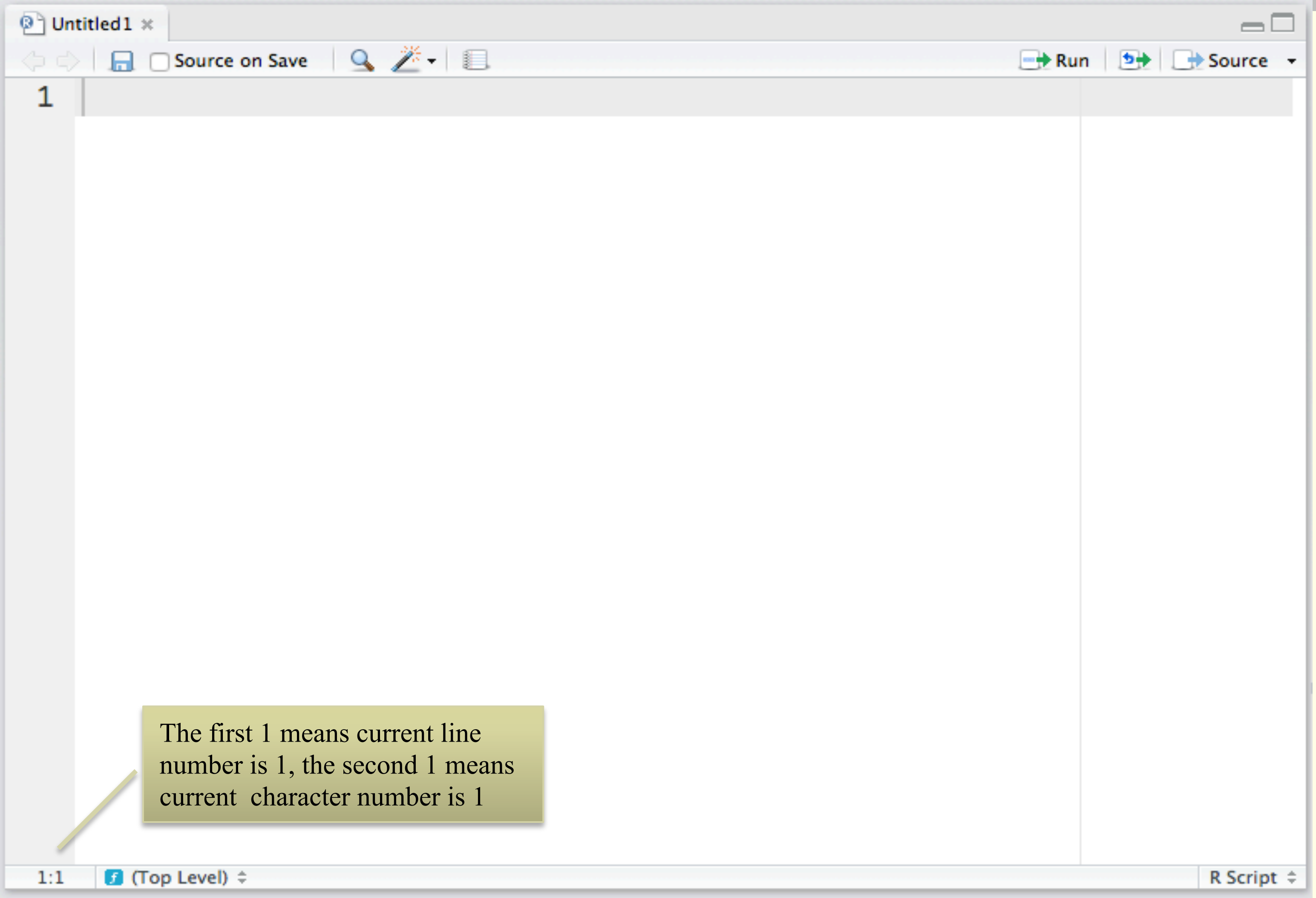
Bo “Paul” Ning

Dr. Reneé H. Moore

# Outline

- Running R programs
- R programming standards
- Useful R programming resources

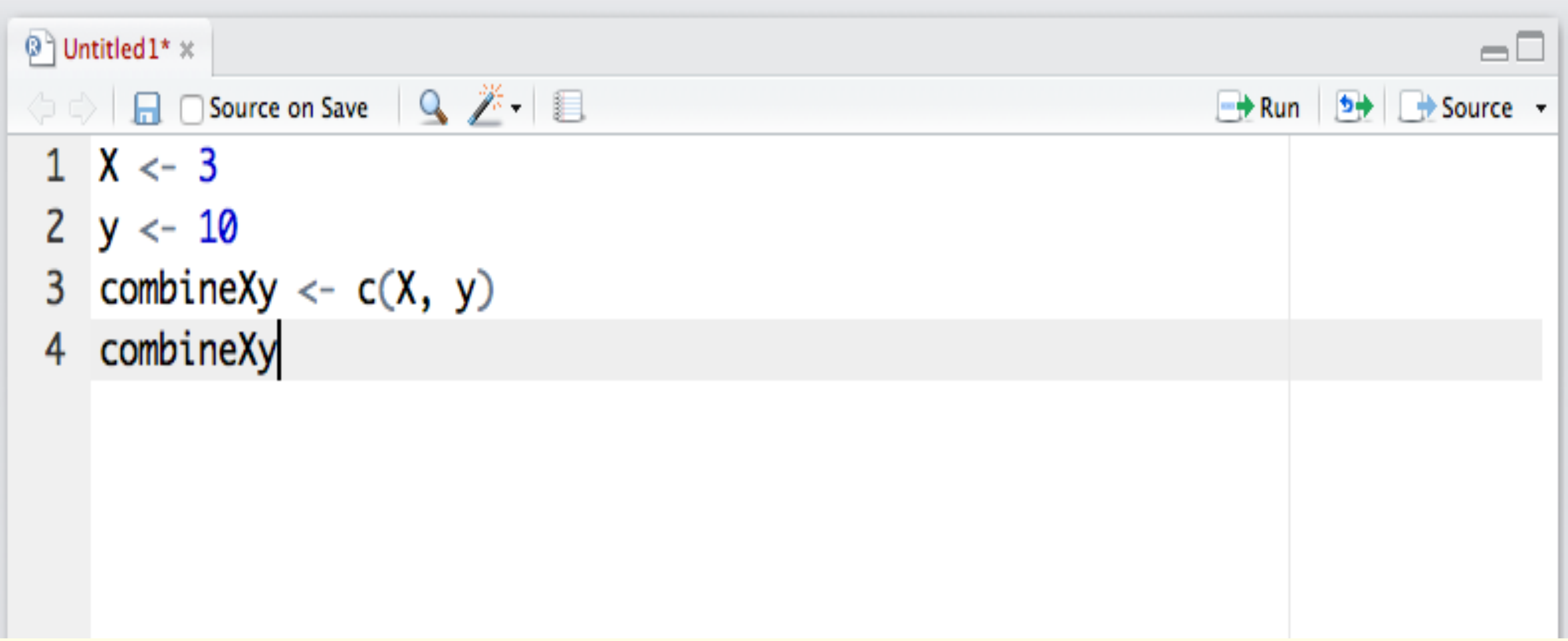
# Running R Programs using RStudio



The first 1 means current line number is 1, the second 1 means current  character number is 1

# Running R Programs using RStudio

## Writing in the Script

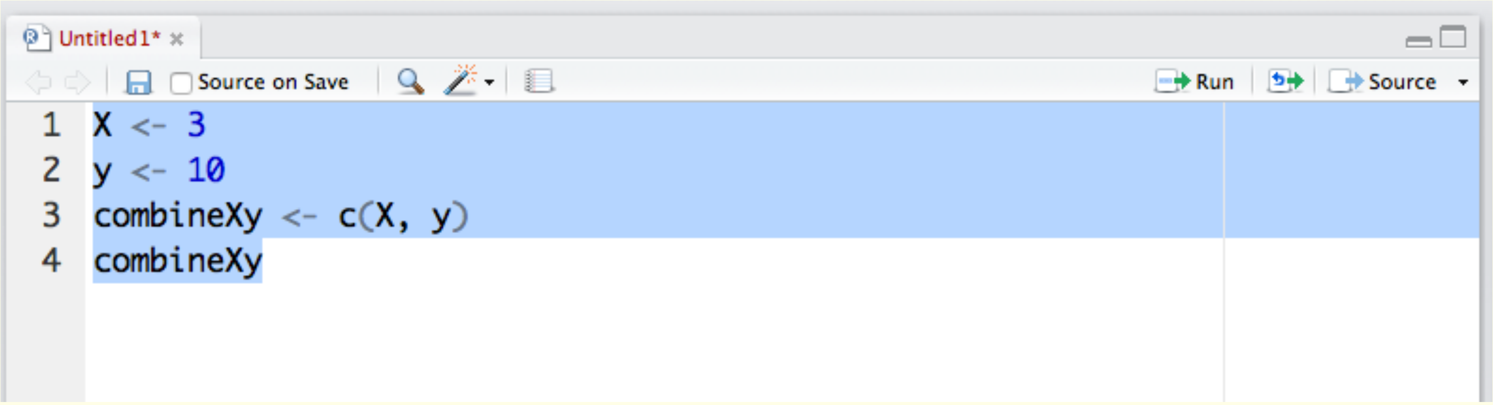


```r
X <- 3
y <- 10
combineXy <- c(X, y)
combineXy
```

# Running R Programs using RStudio

- Two other ways to run the code:
- 1. Copy the code and paste it in the console;
- 2. Use keyboard shortcuts.

  First, highlight the code, then for mac user, use “Command + Return”; for windows user, use “Ctrl + Enter”



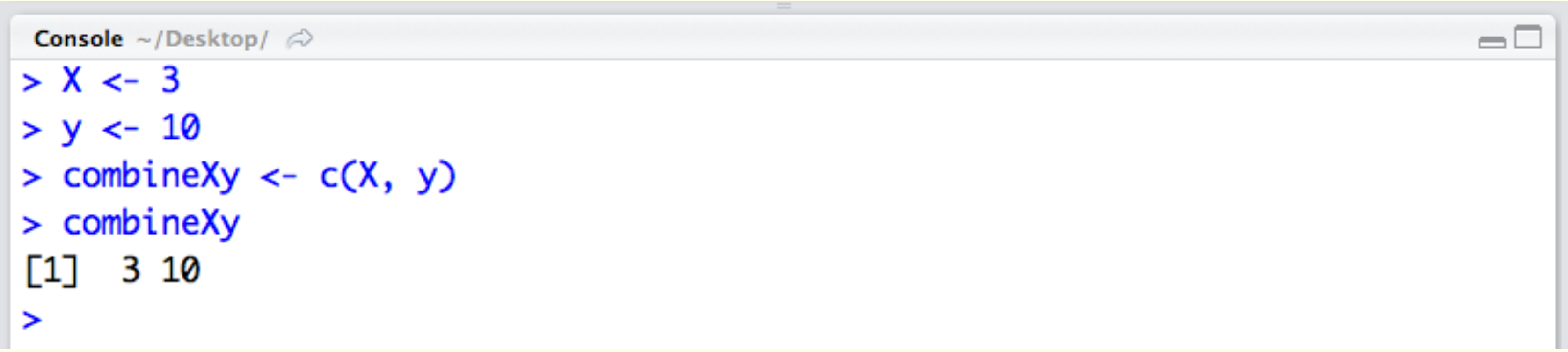
```r
X <- 3
y <- 10
combineXy <- c(X, y)
combineXy
```

For more shortcuts check out this webpage:

https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts

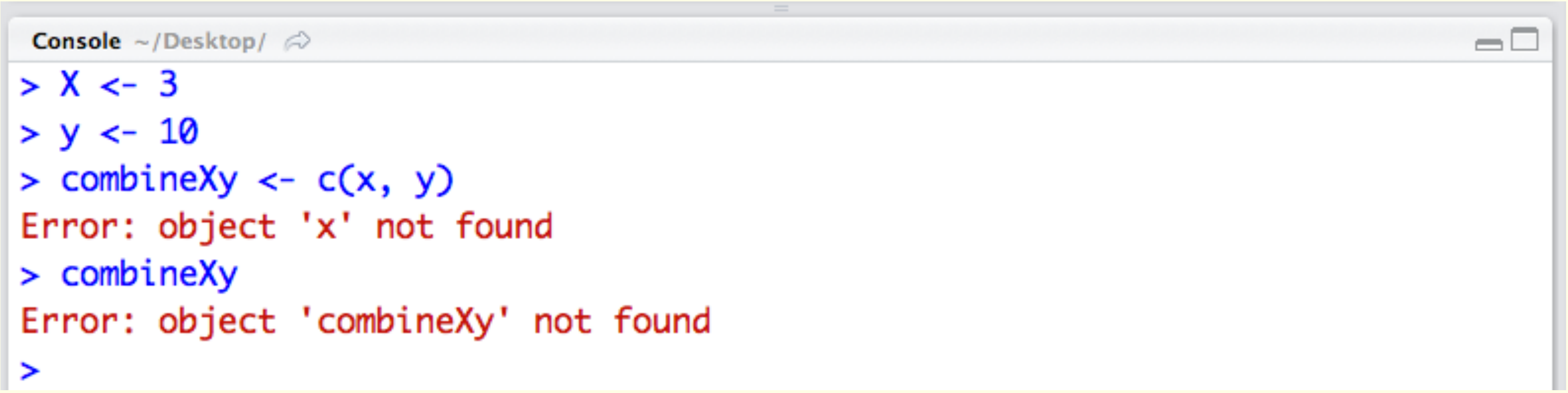
# Running R Programs using RStudio

- In the RStudio console, you could see the result.



```
> X <- 3
> y <- 10
> combineXy <- c(X, y)
> combineXy
[1]  3 10
>
```

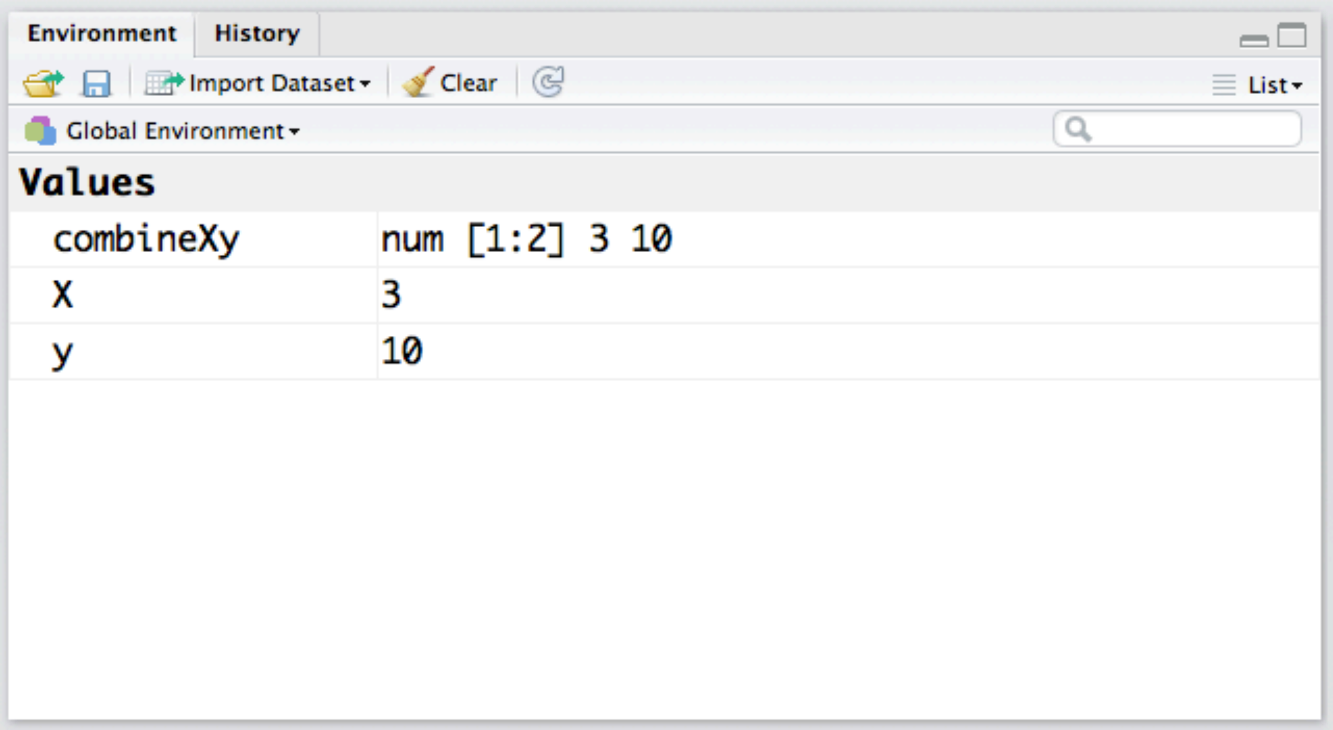
- If your code is wrong, RStudio will give you warning massages.
- For example, input “X” as “x”, then



```
> X <- 3
> y <- 10
> combineXy <- c(x, y)
Error: object 'x' not found
> combineXy
Error: object 'combineXy' not found
>
```

# Running R Programs using RStudio

- Furthermore, RStudio saves variables name in "Environment" dialogue box, which is in the upper right corner of RStudio.



Values

```
combineXy    num [1:2] 3 10
X            3
y            10
```

# R programming standards

1. All programs should be well organized and easy to follow.
2. There should be no errors or warnings in the console when the code is run.
3. Student should follow all specifications in the assignment.
4. Program and output should be correct.
5. Student should complete all the tasks in the assignment.

# R programming standards, continued

6. Programs should have a complete header comment, including students name, date, assignment name, goal of program, and data files used.

7. Comments should be used throughout the program to identify and explain the rational for each section of code

8. Variables assignment should use "<-" not "="

9. Always add "rm(list = ls())" at the top and at the bottom of your program. [Be sure to save first]

# R programming standards, more

10. Each line, the character length should not exceed 80

11. Always add space before and after math operations such as “+”, “-”, “*”, “/”, always add space after “,”.

12. Name variables consistently.

    Here are some options for naming variables (by Yiwen Zhang)

    all lower case: searchpaths …

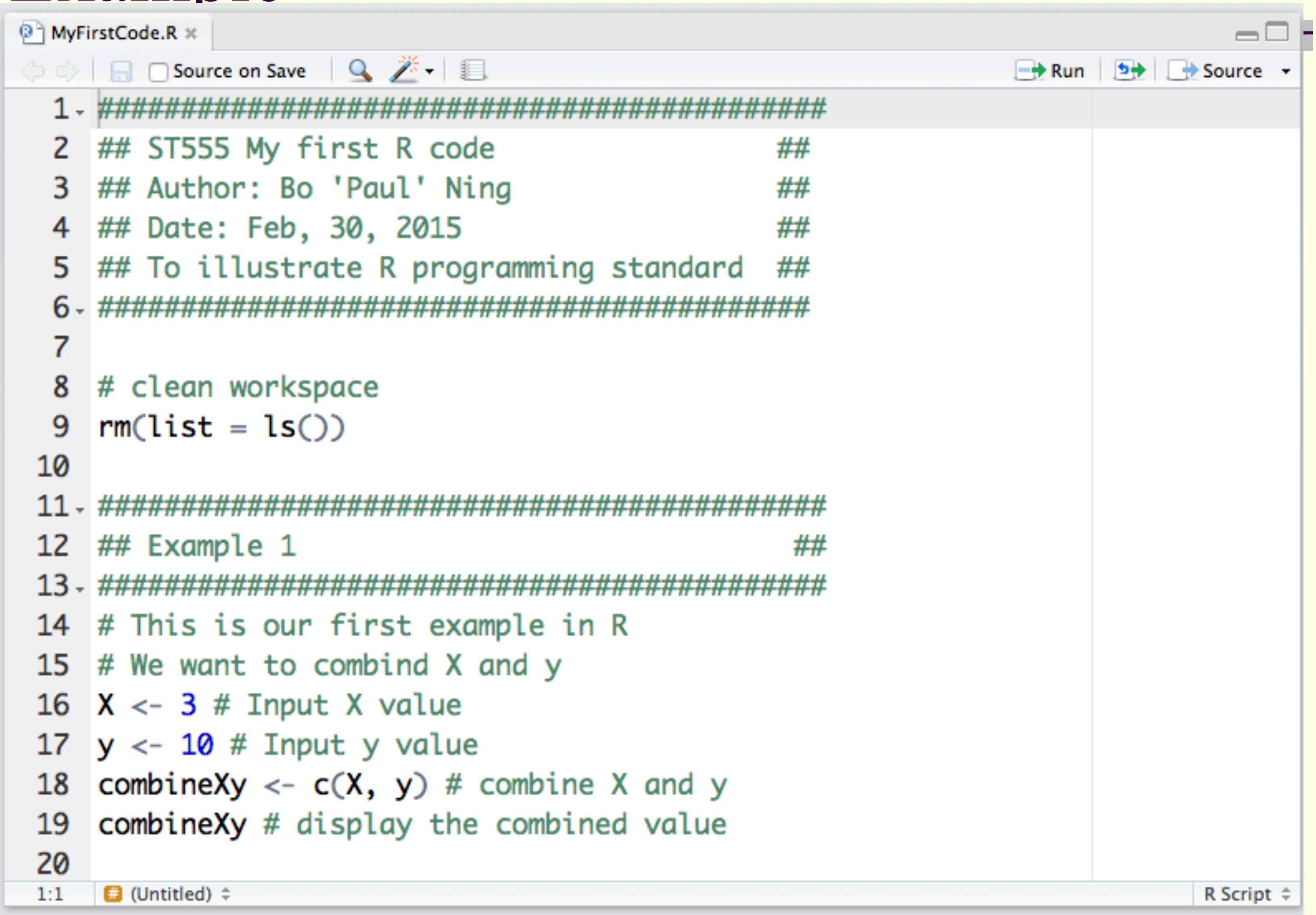
    period separated: as.numeric, read.table …

    underscore separated: package_version …

    lower camel case (suggested): colSums, sessionInfo …

    upper camel case: Vectorize, NextMethod …

# R proramming standards Example



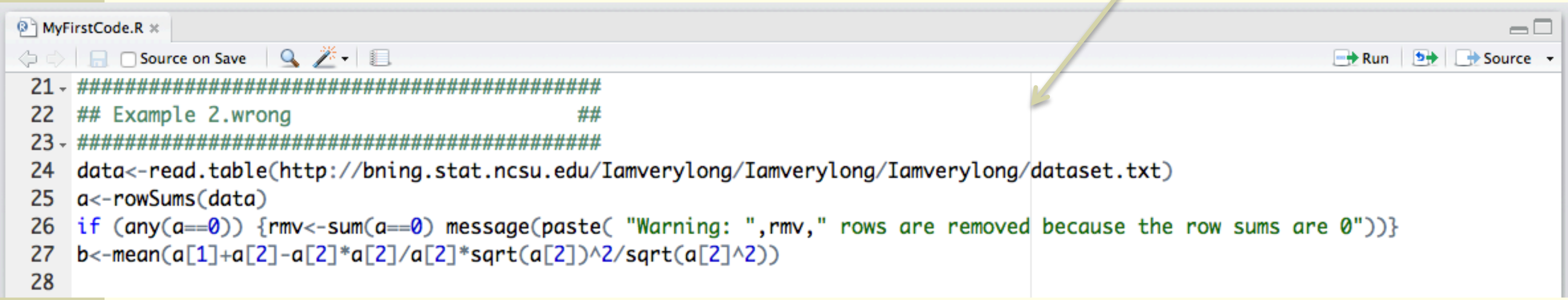
```
############################################
## ST555 My first R code                  ##
## Author: Bo 'Paul' Ning                 ##
## Date: Feb, 30, 2015                    ##
## To illustrate R programming standard  ##
###########################################

# clean workspace
rm(list = ls())

############################################
## Example 1                               ##
############################################
# This is our first example in R
# We want to combind X and y
X <- 3 # Input X value
y <- 10 # Input y value
combineXy <- c(X, y) # combine X and y
combineXy # display the combined value
```

# R programming standards

Look at the following two examples, which one do your prefer?

Example A:

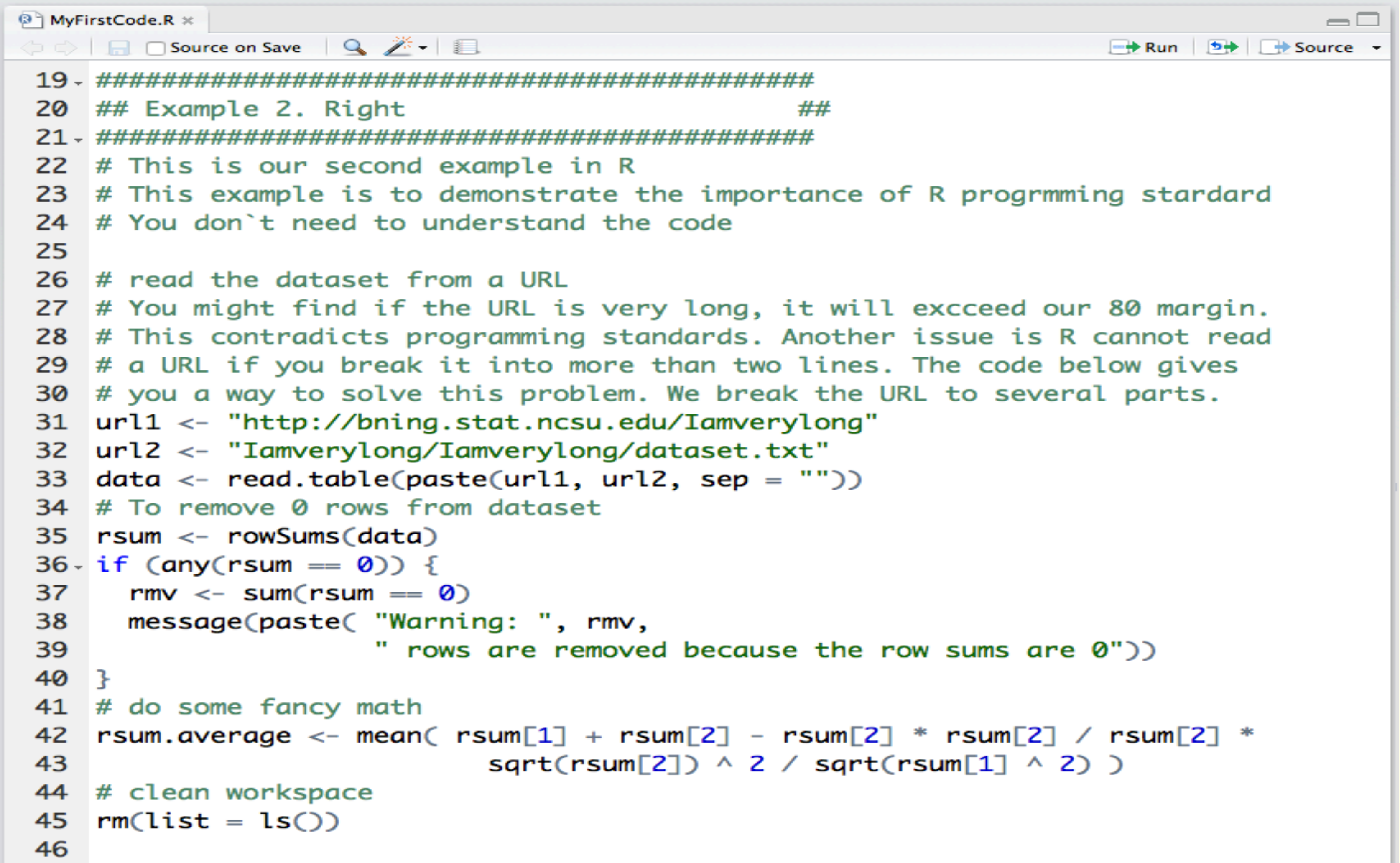
I am that light grey line again!



```
############################################
## Example 2.wrong                        ##
############################################
data<-read.table(http://bning.stat.ncsu.edu/Iamverylong/Iamverylong/Iamverylong/dataset.txt)
a<-rowSums(data)
if (any(a==0)) {rmv<-sum(a==0) message(paste( "Warning: ",rmv," rows are removed because the row sums are 0"))}
b<-mean(a[1]+a[2]-a[2]*a[2]/a[2]*sqrt(a[2])^2/sqrt(a[2]^2))

```

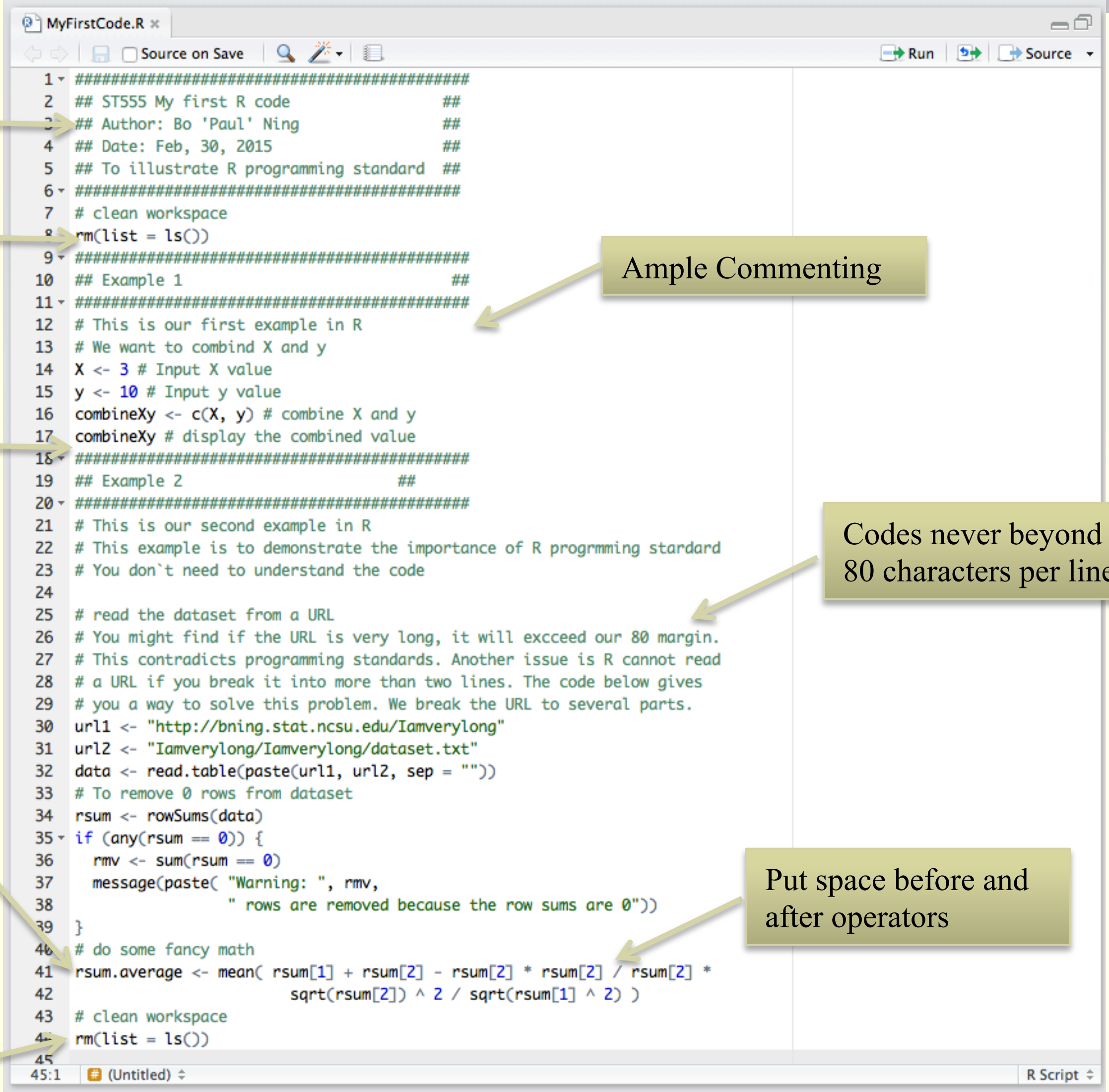
# R programming standards

Example B:



```r
############################################
## Example 2. Right                        ##
############################################
# This is our second example in R
# This example is to demonstrate the importance of R progrmming stardard
# You don`t need to understand the code

# read the dataset from a URL
# You might find if the URL is very long, it will excceed our 80 margin.
# This contradicts programming standards. Another issue is R cannot read
# a URL if you break it into more than two lines. The code below gives
# you a way to solve this problem. We break the URL to several parts.
url1 <- "http://bning.stat.ncsu.edu/Iamverylong"
url2 <- "Iamverylong/Iamverylong/dataset.txt"
data <- read.table(paste(url1, url2, sep = ""))
# To remove 0 rows from dataset
rsum <- rowSums(data)
if (any(rsum == 0)) {
  rmv <- sum(rsum == 0)
  message(paste( "Warning: ", rmv,
                 " rows are removed because the row sums are 0"))
}
# do some fancy math
rsum.average <- mean( rsum[1] + rsum[2] - rsum[2] * rsum[2] / rsum[2] *
                        sqrt(rsum[2]) ^ 2 / sqrt(rsum[1] ^ 2) )
# clean workspace
rm(list = ls())

```

# R programming standards



```r
############################################
## ST555 My first R code                 ##
## Author: Bo 'Paul' Ning                ##
## Date: Feb, 30, 2015                   ##
## To illustrate R programming standard  ##
###########################################
# clean workspace
rm(list = ls())
############################################
## Example 1                              ##
############################################
# This is our first example in R
# We want to combind X and y
X <- 3 # Input X value
y <- 10 # Input y value
combineXy <- c(X, y) # combine X and y
combineXy # display the combined value
############################################
## Example 2                        ##
############################################
# This is our second example in R
# This example is to demonstrate the importance of R progrmming stardard
# You don`t need to understand the code

# read the dataset from a URL
# You might find if the URL is very long, it will excceed our 80 margin.
# This contradicts programming standards. Another issue is R cannot read
# a URL if you break it into more than two lines. The code below gives
# you a way to solve this problem. We break the URL to several parts.
url1 <- "http://bning.stat.ncsu.edu/Iamverylong"
url2 <- "Iamverylong/Iamverylong/dataset.txt"
data <- read.table(paste(url1, url2, sep = ""))
# To remove 0 rows from dataset
rsum <- rowSums(data)
if (any(rsum == 0)) {
  rmv <- sum(rsum == 0)
  message(paste( "Warning: ", rmv,
                 " rows are removed because the row sums are 0"))
}
# do some fancy math
rsum.average <- mean( rsum[1] + rsum[2] - rsum[2] * rsum[2] / rsum[2] *
                        sqrt(rsum[2]) ^ 2 / sqrt(rsum[1] ^ 2) )
# clean workspace
rm(list = ls())

```

Complete header

Add rm(list = ls()) at the beginning

Use “<-” to assign values to variables

Variable name is consistent and understandable

Add rm(list = ls()) at the end

Ample Commenting

Codes never beyond 80 characters per line

Put space before and after operators

# R programming standards

- Good references for R programming Standards (Compliments of Dr. Hua Zhou)

http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml

https://sites.google.com/site/matlabstyleguidelines/home

# Useful R programming resources

Here are resources for R programming

1. Advanced R by Hadley Wickham: http://adv-r.had.co.nz/
2. Dr. John Monahan's class (2013 fall) on R: http://www.stat.ncsu.edu/people/monahan/courses/ST610/
3. Online tutorial: http://tryr.codeschool.com/levels/1/challenges/1

Taken from Dr. Hua Zhou`s ST 758: computation for statistical research lecture notes.

# ST 555: Statistical Programming I

## R Markdown

Bo “Paul” Ning

Dr. Reneé H. Moore

# Outline

- What is R Markdown?
- Why use R Markdown?
- Use R Markdown to generate report

# What is R Markdown? Why Use?

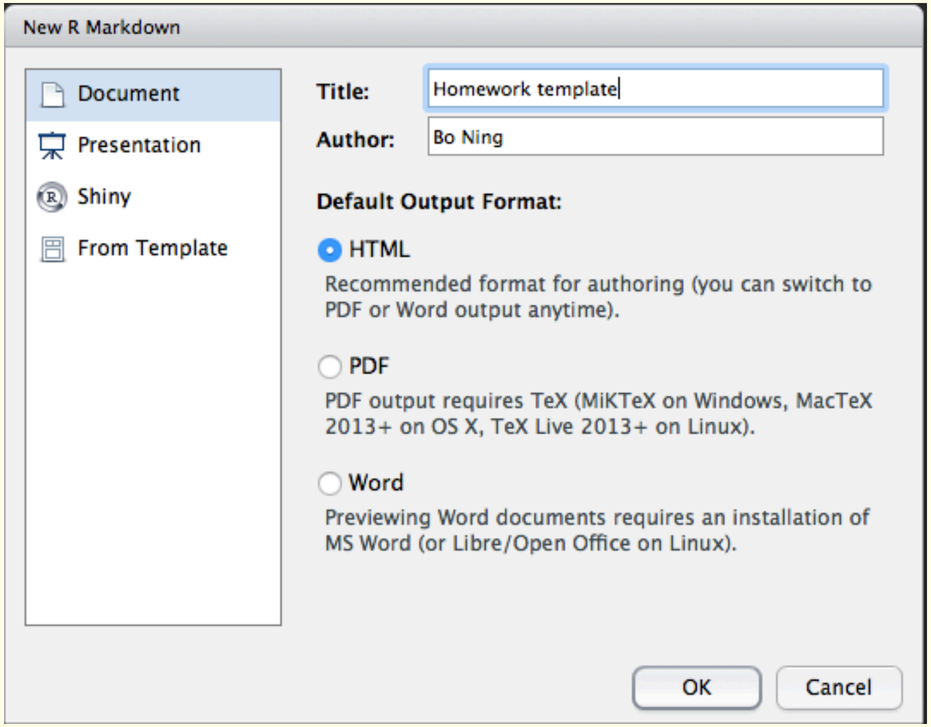
- R Markdown is a dynamic document for R
- It combines the core syntax of markdown (an easy-to-write plain text format) with embedded R code chunks that are run so their output can be included in the final document.

(from http://rmarkdown.rstudio.com)

- New technology, widely used
- Integrate texts, R code and output together in one document in a nice looking way
- Automatically generate dynamic report for R programming

# Open R Markdown

- Find out button in the left upper side of RStudio.
- Click, then click R Markdown...
- Choose “Document”, “Title”, “Author”, “Format Type”, Then click “OK”.



New R Markdown

- [✓ Document]
- Presentation
- Shiny
- From Template

Title: Homework template

Author: Bo Ning

Default Output Format:

● HTML

Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF

PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

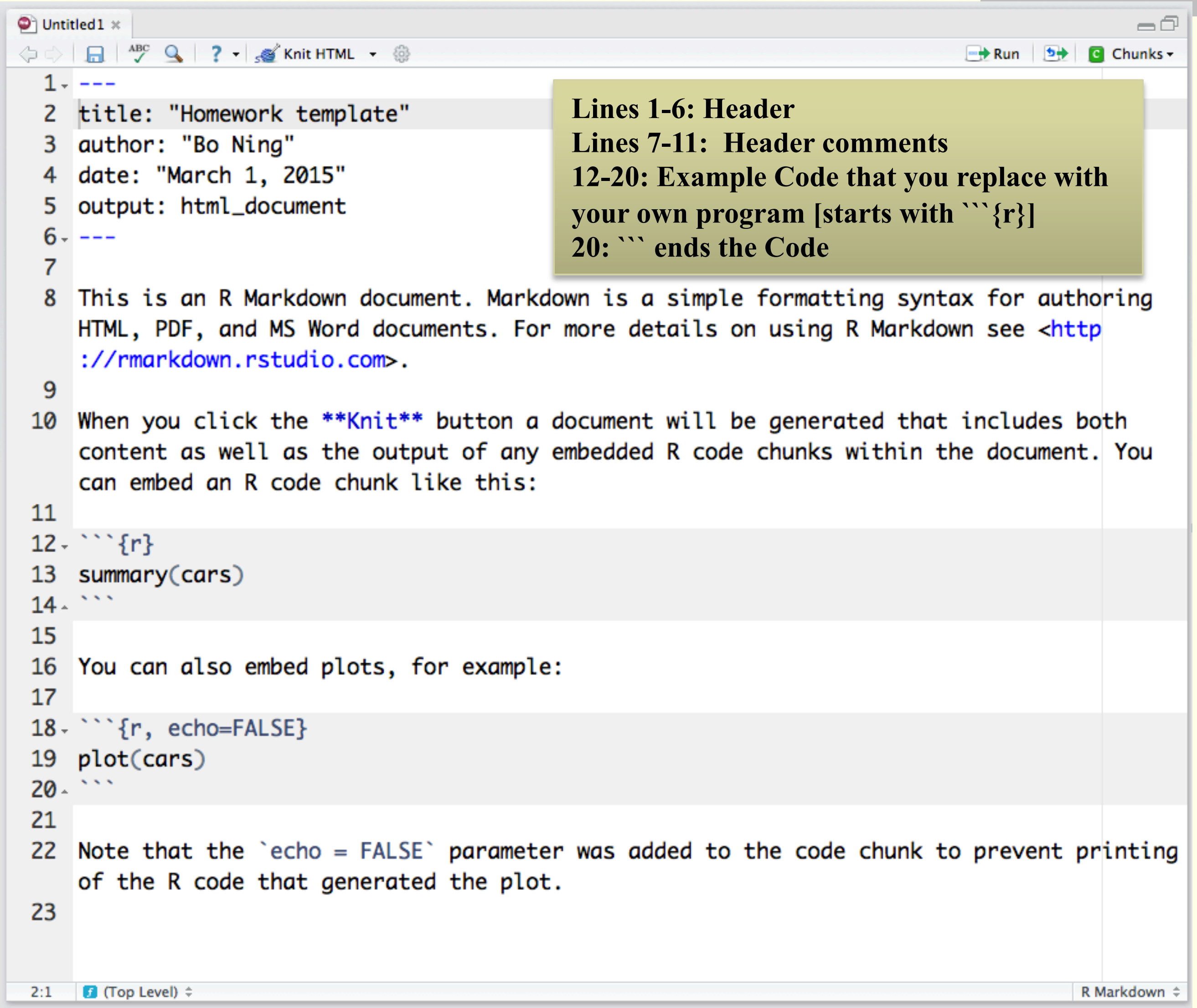
# A Few Comments before Generating RMarkdown Report

- Make sure you have the lastest version of RStudio
- Yes to install 3 packages

- Where are you files saving?
- `>getwd()`

- Ways to change directory
- `>setwd(“C:/My Documents”)`
- From File Menu Save your Script in preferred directory
- Bottom Right Window, choose Files and then find preferred directory

# Open R Markdown

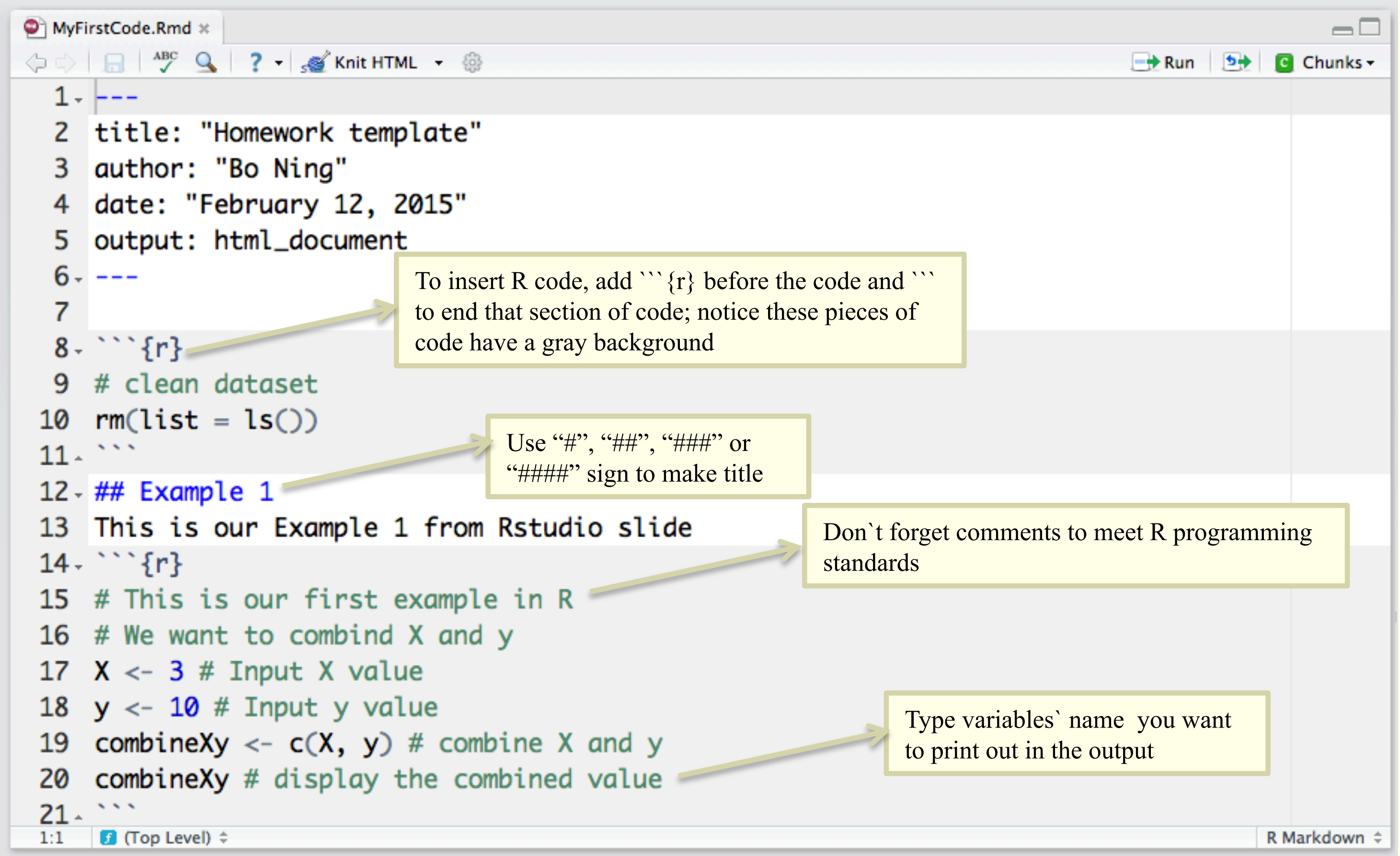


```
1  ---
2  title: "Homework template"
3  author: "Bo Ning"
4  date: "March 1, 2015"
5  output: html_document
6  ---
7
8  This is an R Markdown document. Markdown is a simple formatting syntax for authoring
   HTML, PDF, and MS Word documents. For more details on using R Markdown see <http
   ://rmarkdown.rstudio.com>.
9
10 When you click the **Knit** button a document will be generated that includes both
   content as well as the output of any embedded R code chunks within the document. You
   can embed an R code chunk like this:
11
12 ```{r}
13 summary(cars)
14 ```
15
16 You can also embed plots, for example:
17
18 ```{r, echo=FALSE}
19 plot(cars)
20 ```
21
22 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing
   of the R code that generated the plot.
23
```

**Lines 1-6: Header**
**Lines 7-11: Header comments**
**12-20: Example Code that you replace with your own program [starts with ```{r}]**
**20: ``` ends the Code**

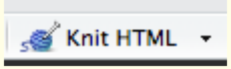
# Write code in R Markdown

- Let`s start to write our first R Markdown file
- Includes title (question number) and “rm(list = ls())”



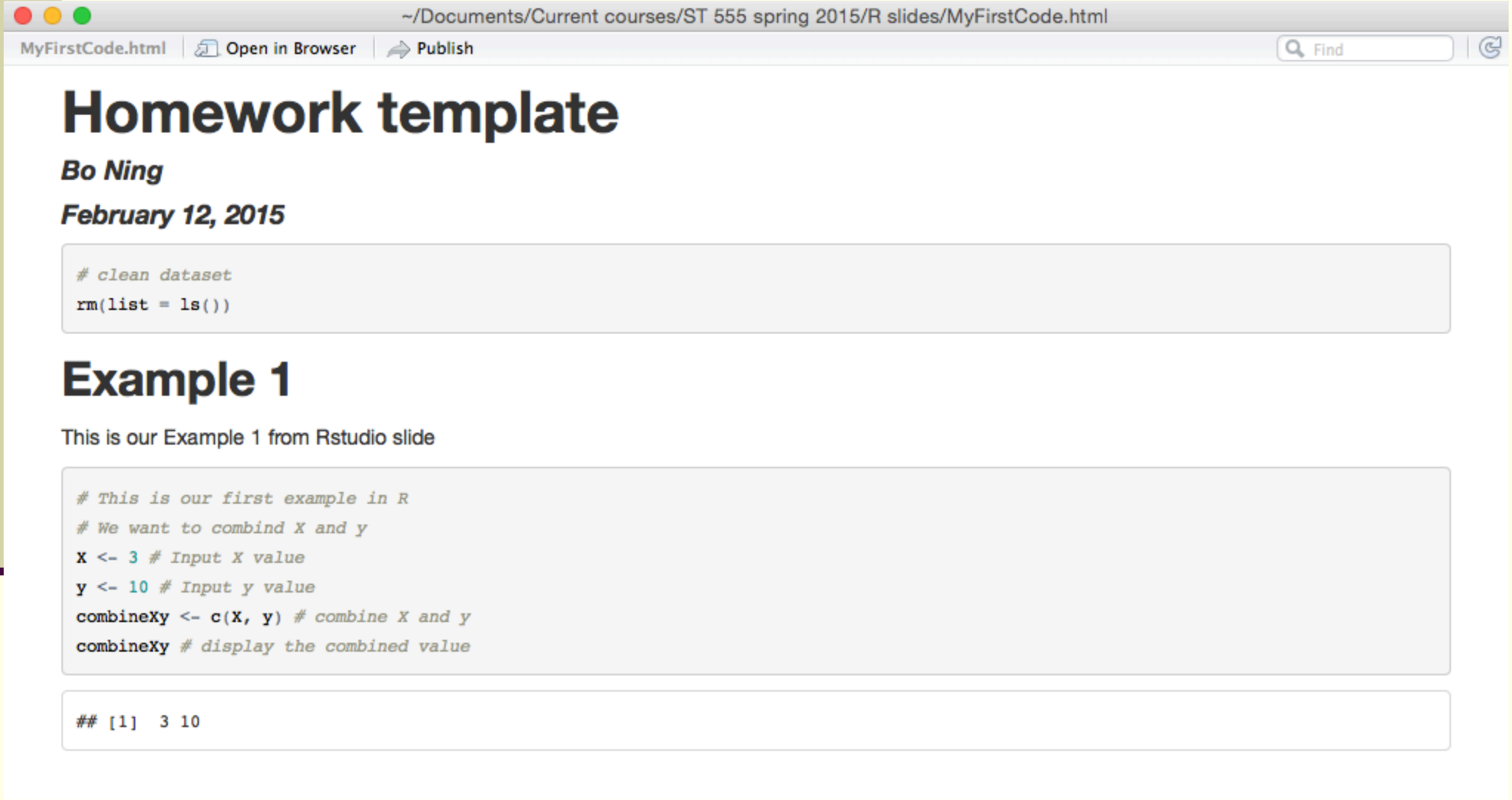
~~~
---
title: "Homework template"
author: "Bo Ning"
date: "February 12, 2015"
output: html_document
---

```{r}
# clean dataset
rm(list = ls())
~~~
## Example 1
This is our Example 1 from Rstudio slide
```{r}
# This is our first example in R
# We want to combind X and y
X <- 3 # Input X value
y <- 10 # Input y value
combineXy <- c(X, y) # combine X and y
combineXy # display the combined value
```
```

To insert R code, add ```{r} before the code and ``` to end that section of code; notice these pieces of code have a gray background

Use “#”, “##”, “###” or “####” sign to make title

Don`t forget comments to meet R programming standards

Type variables` name you want to print out in the output
```

# Generate report

- To generate a report, in the editor window, find out  .

- There are three formats for reports.
- Knit HTML gives you a .html format report;
- Knit PDF, gives you a .pdf format report;
- Knit Word (.doc/.docx)
- Knit PDF may require you to install LaTeX, which you could download from the website: http://latex-project.org
- If you don`t wish to install LaTeX, Knit HTML and Knit Word are the options for you.

# Generate report



~/Documents/Current courses/ST 555 spring 2015/R slides/MyFirstCode.html
MyFirstCode.html
Open in Browser
Publish
Find

# Homework template

***Bo Ning***

***February 12, 2015***

```
# clean dataset
rm(list = ls())
```

## Example 1

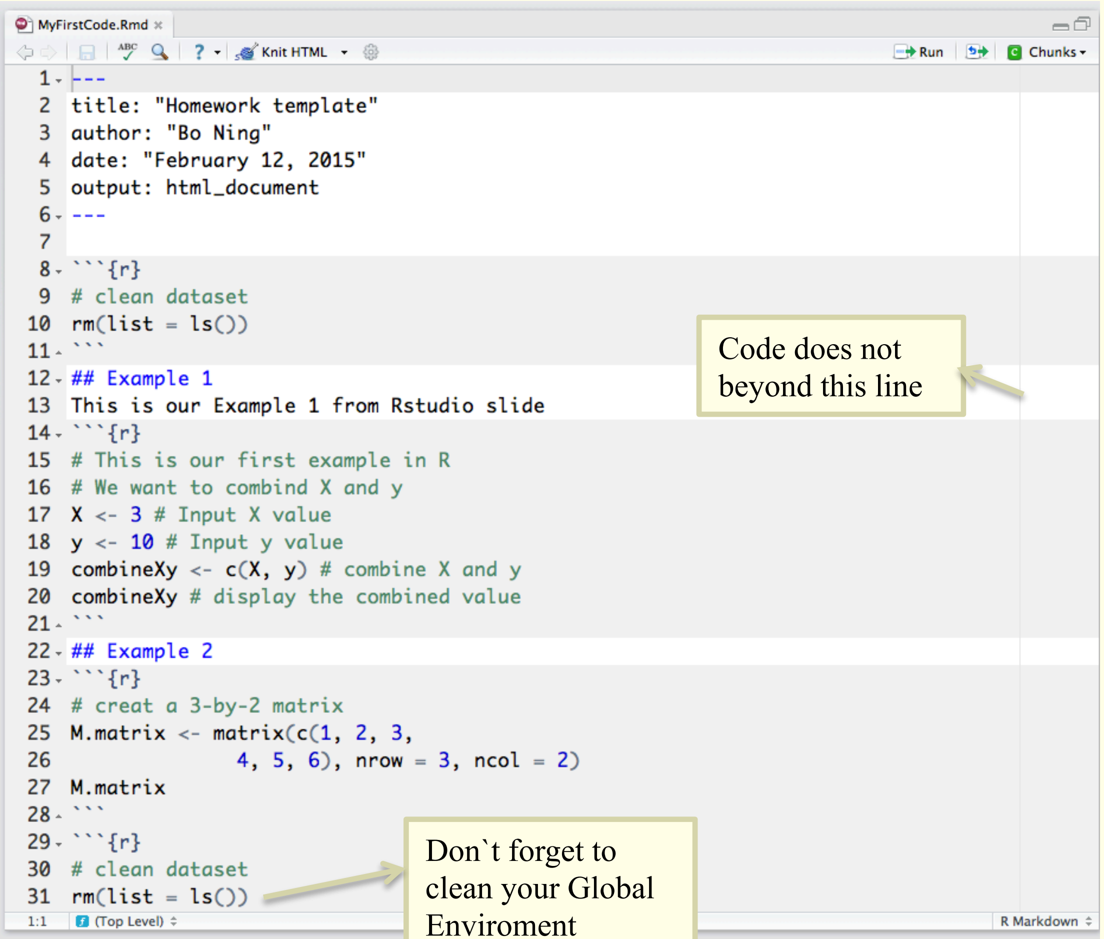
This is our Example 1 from Rstudio slide

```
# This is our first example in R
# We want to combind X and y
X <- 3 # Input X value
y <- 10 # Input y value
combineXy <- c(X, y) # combine X and y
combineXy # display the combined value
```

```
## [1]  3 10
```

# Generate report

- Suppose our homework template has 2 examples, here is the final version of the code.



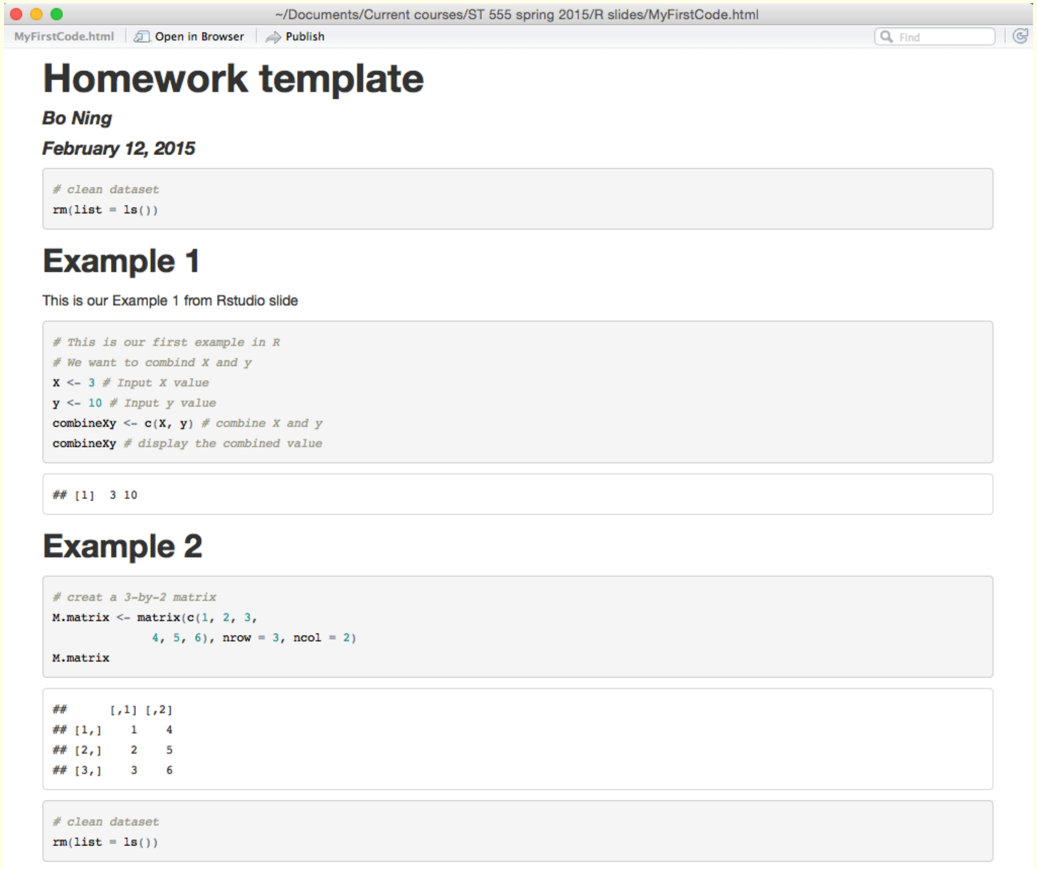
~~~
---
title: "Homework template"
author: "Bo Ning"
date: "February 12, 2015"
output: html_document
---

```{r}
# clean dataset
rm(list = ls())
~~~
## Example 1
This is our Example 1 from Rstudio slide
```{r}
# This is our first example in R
# We want to combind X and y
X <- 3 # Input X value
y <- 10 # Input y value
combineXy <- c(X, y) # combine X and y
combineXy # display the combined value
```
## Example 2
```{r}
# creat a 3-by-2 matrix
M.matrix <- matrix(c(1, 2, 3,
              4, 5, 6), nrow = 3, ncol = 2)
M.matrix
```
```{r}
# clean dataset
rm(list = ls())
```

Code does not beyond this line

Don`t forget to clean your Global Enviroment

# Generate report

- Let`s Knit HTML again.


~/Documents/Current courses/ST 555 spring 2015/R slides/MyFirstCode.html
MyFirstCode.html
Open in Browser
Publish
Find

# Homework template

***Bo Ning***

***February 12, 2015***

```
# clean dataset
rm(list = ls())
```

## Example 1

This is our Example 1 from Rstudio slide

```
# This is our first example in R
# We want to combind X and y
X <- 3 # Input X value
y <- 10 # Input y value
combineXy <- c(X, y) # combine X and y
combineXy # display the combined value
```

```
## [1]  3 10
```

## Example 2

```
# creat a 3-by-2 matrix
M.matrix <- matrix(c(1, 2, 3,
              4, 5, 6), nrow = 3, ncol = 2)
M.matrix
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

```
# clean dataset
rm(list = ls())
```

# R Markdown supplements

- An advantage for R Markdown is that it incorporates LaTeX.

- If you want to know more about how to incorporates LaTeX code in R Markdown, please google it, or go to Yihui Xie`s blog ( http://yihui.name)

- For each homework, please submit a R Markdown file (.Rmd file) and the corresponding output file (.html file, .pdf file or a word file).