Introduction to R and RStudio

EPID 799B

Nathaniel MacNell
Mike Fliss

Outline

- R Syntax (Nat)
  - Expressions
  - Operators
  - Functions
  - Assignment
- RStudio Features (Mike)
  - IDEs
  - Navigation
  - Keystrokes

General notes on R syntax (vs. SAS)

- R is case SeNSiTiVe
- # for comments
- No ; needed
- R *can* execute things in any order
  - PLEASE put your code in order!
- Datasets in R are called **data frames**

Using R: writing expressions

- R evaluates complete statements called expressions and returns the results to the console.

```
3-1
 2
```

```
(13+15)/2
 14
```

- You can use numbers and any of these operators:

```
+ - * / ^ ( )
```

Logical Operators

- If you ask R to evaluate an equation, inequality, or Boolean expression, it will return TRUE or FALSE:

1 == 2+3	FALSE
3 < 4	TRUE
12 >= 13-1	TRUE
TRUE & FALSE	FALSE
TRUE | FALSE	TRUE
(3<4) & !(FALSE)	TRUE

Assignment

- Like the STO-> button on a calculator, you can use **<-** to save results for later (you can also use **=**)

students <- 20	[no output]
students	20
students+1	21
people = students - 5	[no output]
people	15

- R doesn't "keep" anything unless you assign it

Functions

- Functions take input and produce output
- Can be nested within a larger expression

sin(pi)	0
2*sin(pi/2)	2
print(“hello”)	“hello”
print(“hello”)+1	ERROR

Example: Importing data using a function and assignment

read.csv(“births.csv”)	[prints a spreadsheet to the console]
births <- read.csv(“births.csv”)	[no output]
births	[prints spreadsheet]

Getting inside a dataset

- You can use the **names()** function to get the variable names of a dataset
- You can use $ to pull out a variable in a dataset

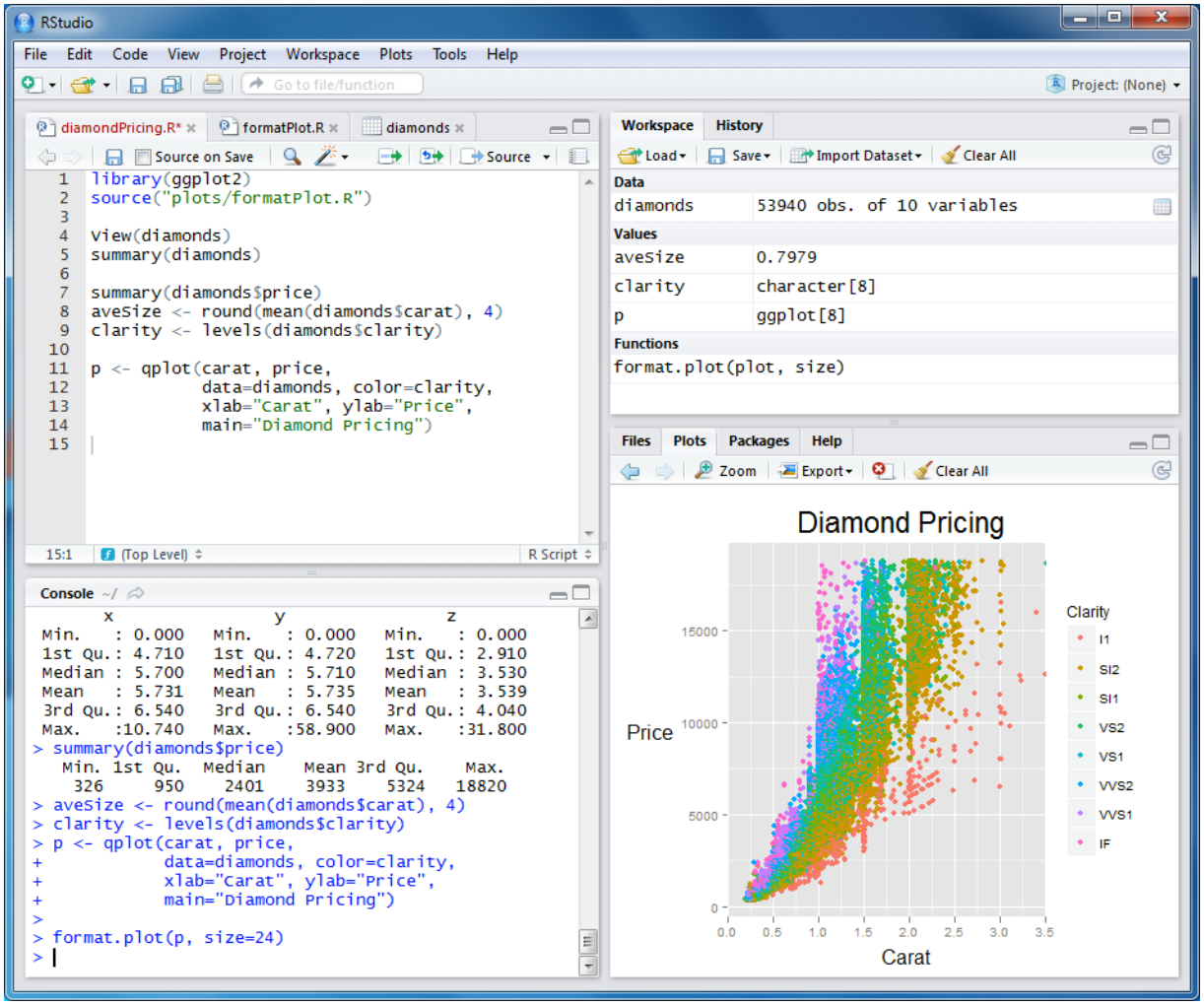
names(births)	[variable names for births]
births$GEST	0 0 1 1 1 0
mean(births$GEST)	38.56
table(births$GEST)	[prints table]

Getting inside a variable

- You can use [] to get inside of a variable or a dataset (you put the *index* of what you want inside)

letters	print letters variable
letters[2]	prints “b”
births$GEST[3]	prints 3rd value
births[2,3]	prints the 2nd value of the 3rd variable in births

Rstudio…


RStudio
File Edit Code View Project Workspace Plots Tools Help
diamondPricing.R* formatPlot.R diamonds
Source on Save Source
1 library(ggplot2)
2 source("plots/formatPlot.R")
3
4 View(diamonds)
5 summary(diamonds)
6
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
10
11 p <- qplot(carat, price,
12 data=diamonds, color=clarity,
13 xlab="Carat", ylab="Price",
14 main="Diamond Pricing")
15
15:1 (Top Level) R Script
Console ~/
x y z
Min. : 0.000 Min. : 0.000 Min. : 0.000
1st Qu.: 4.710 1st Qu.: 4.720 1st Qu.: 2.910
Median : 5.700 Median : 5.710 Median : 3.530
Mean : 5.731 Mean : 5.735 Mean : 3.539
3rd Qu.: 6.540 3rd Qu.: 6.540 3rd Qu.: 4.040
Max. :10.740 Max. :58.900 Max. :31.800
> summary(diamonds$price)
Min. 1st Qu. Median Mean 3rd Qu. Max.
326 950 2401 3933 5324 18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+ data=diamonds, color=clarity,
+ xlab="Carat", ylab="Price",
+ main="Diamond Pricing")
>
> format.plot(p, size=24)
>
Workspace History
Load Save Import Dataset Clear All
Data
diamonds 53940 obs. of 10 variables
Values
aveSize 0.7979
clarity character[8]
p ggplot[8]
Functions
format.plot(plot, size)
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
15000
10000
5000
0
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5
Carat
Clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

Rstudio…

- …is an IDE!
  (an “Integrated Development Environment”)

- A good IDE “… allows you to work at full speed.”

- References to check out later:
  - https://www.rstudio.com/resources/webinars/rstudio-essentials-webinar-series-part-1/
  - https://www.rstudio.com/online-learning/#R

http://programmers.stackexchange.com/questions/102018/what-features-of-an-ide-would-make-it-more-useful-than-a-general-purpose-editor

https://channel9.msdn.com/Forums/Coffeehouse/106446-What-makes-a-good-IDE

RStudio

- IDE Layout
  - Panes & their use
  - Help→cheatsheets→RStudio
- Running code
  - Console, script, blocks, comments, inline
- Comments
  - #, post-#, code blocks, comment blocks
- Key keyboard shortcuts
  - https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts
  - Alt-Shift-K
- Style
  - R: http://adv-r.had.co.nz/Style.html
  - Google: https://google.github.io/styleguide/Rguide.xml

RStudio

- Let’s dive in!

- Recognize the elements of syntax we covered
- Feel the “flow” – no strict “data” blocks
- What code looks (can look!) like
- Cool, frequently used options in Rstudio