

Workshop on R Part One

R Workshop 2019



Introduction

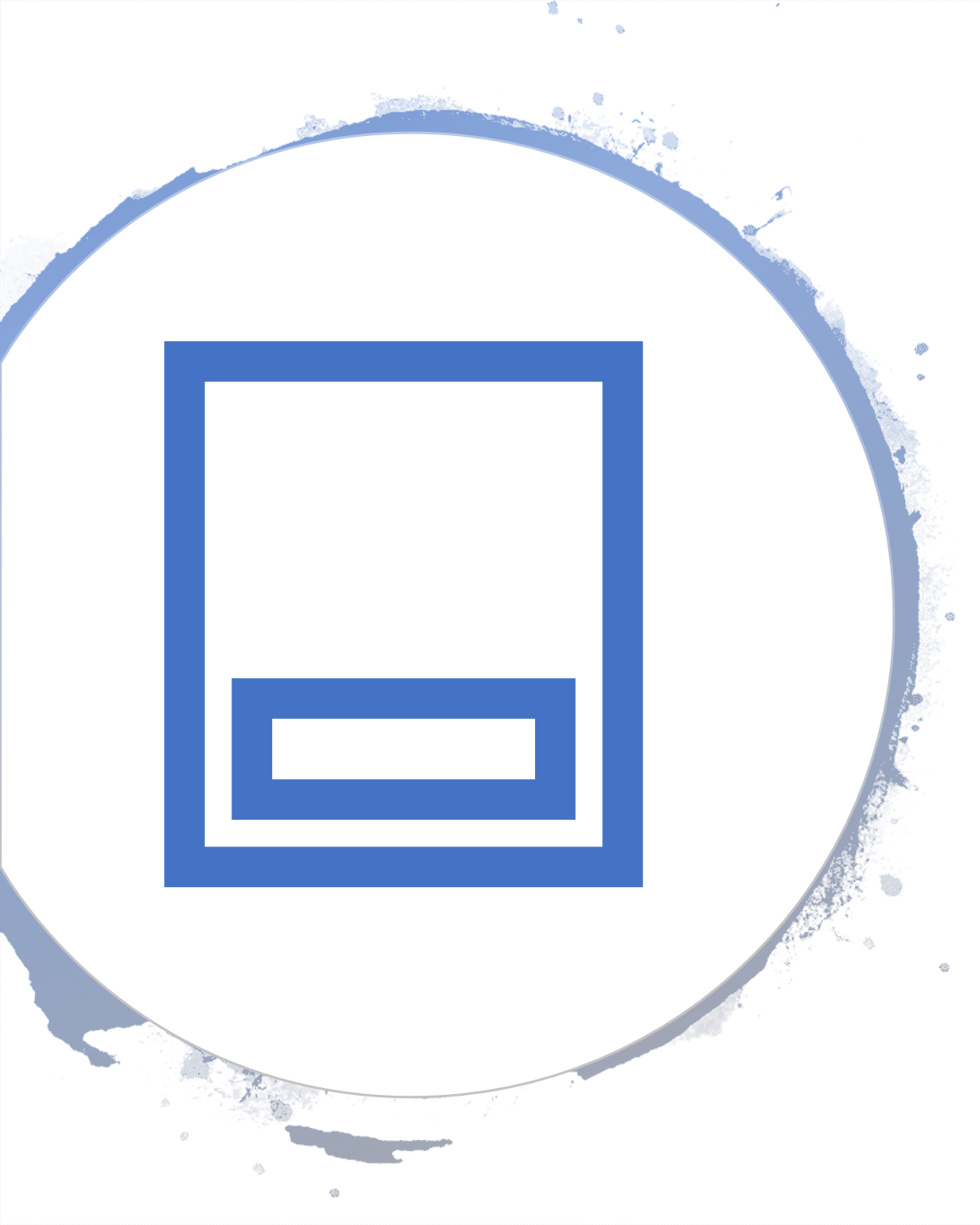
- This is a hands-on workshop on R
- The hands-on learning could be done in a computer lab
- The reference material for this workshop will be a free textbook on basic R
- The free book along with the accompanying data sets can be found at <https://learningstatisticswithr.com/>
- Material has been chosen from the above free book that is most relevant for this workshop



Workshop Outline

Part One

- Part 1: Getting Started With R
 - This is chapter 3 of the free book (FB)
- Part 2: Installing R
 - This is section 3.1 of the free book (FB)
- Part 3: The R console
 - This is section 3.2 of the free book (FB)
- Note: this workshop is for internal training purposes only. No content in this workshop is intended for re-distribution in any format.

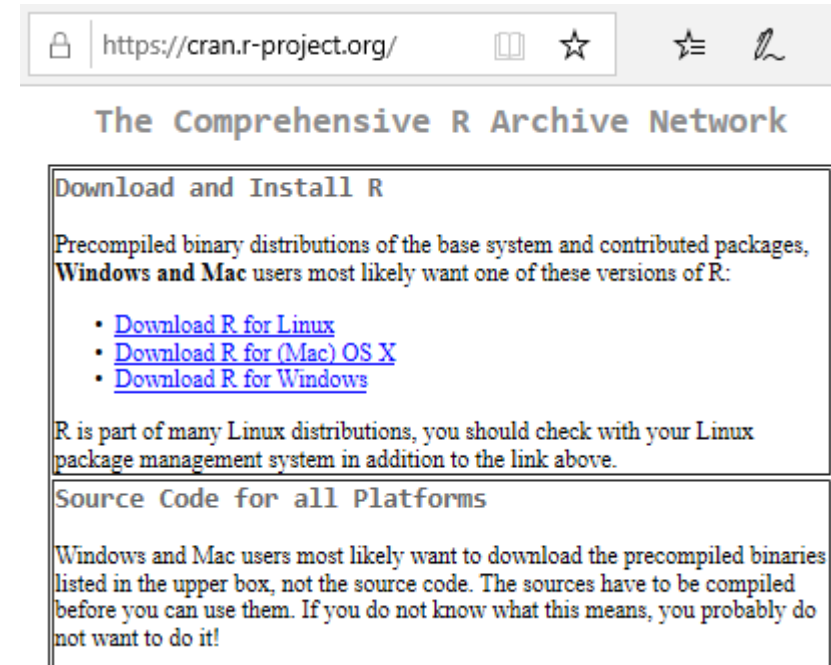


Part 1: Getting Started with R

- Why R? (Chapter 3 of free book (FB))
- R is highly extensible. When you download and install R, you get all the basic “packages”, which are very powerful on their own. However, because R is very open and very widely used, it is becoming a standard tool in statistics, and thus many people
- write their own packages that extend the system. And these packages are often freely available as well. One of the consequences of this is that many recent advanced textbooks use R. So, if you learn how
- to do basic statistics in R, then this will help you with the more advanced statistics that also come with R

Part 2: Installing R

- Relatively straight forward
- See free book
- Also see home of R:
- <https://cran.r-project.org/>



The screenshot shows a web browser window with the address bar containing <https://cran.r-project.org/>. The page title is "The Comprehensive R Archive Network". The main content is titled "Download and Install R" and contains the following text:

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

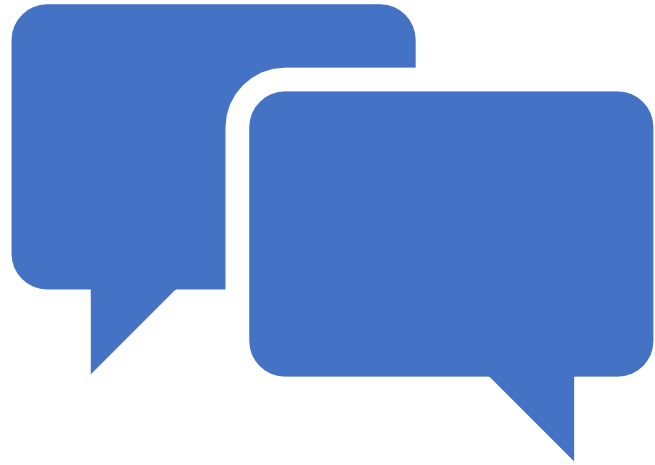
- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

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!

Part 3: The R console



```
R Console (64-bit)
File Edit Misc Packages Windows Help
Copyright (C) 2019 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

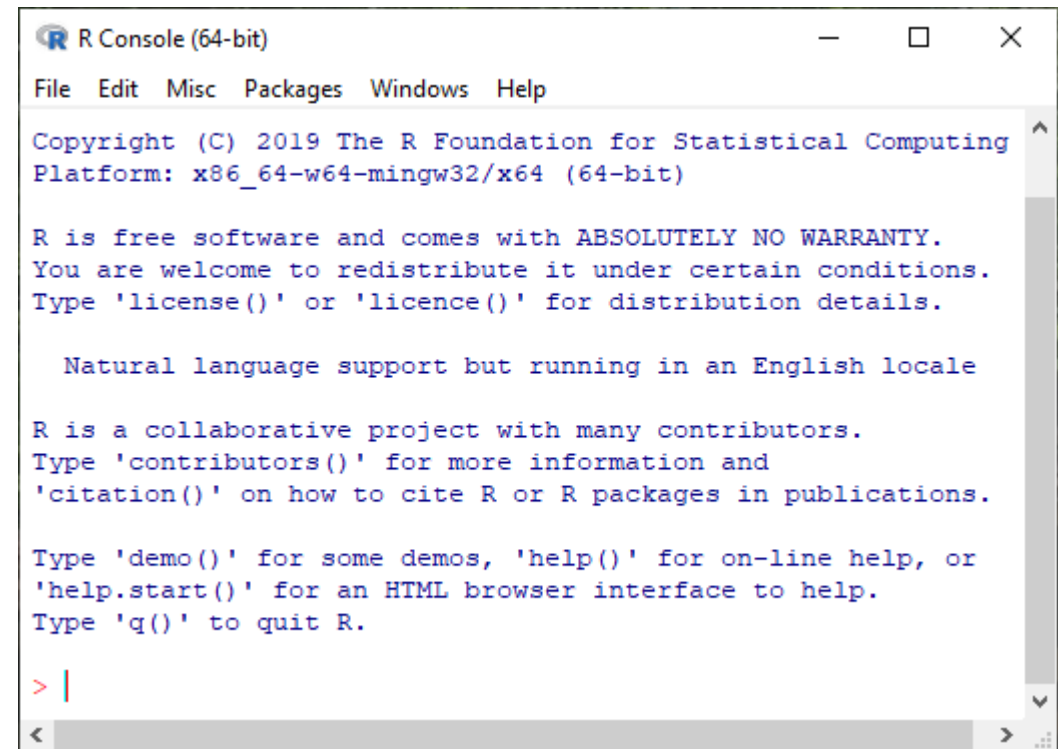
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

> |
```

Part 3: The R console

- Yes there is a command prompt!



```
R Console (64-bit)
File Edit Misc Packages Windows Help

Copyright (C) 2019 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

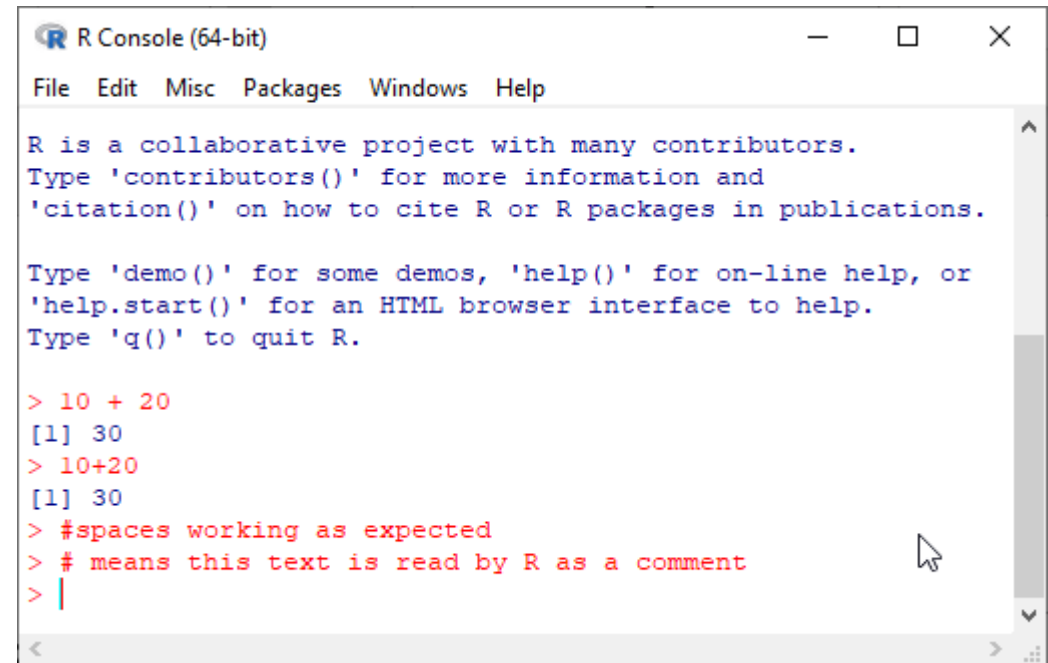
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

> |
```

Part 3: The R console

- See section 3.2.1 of free book (FB)
- Be careful to avoid typos when entering commands at prompt!
- R is case sensitive!
- Case sensitive means r does not equal R. When you type in R commands make sure you use the right upper- or lower-case letters!
- Copy and paste can be easier than just typing in commands!



```
R Console (64-bit)
File Edit Misc Packages Windows Help

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.

> 10 + 20
[1] 30
> 10+20
[1] 30
> #spaces working as expected
> # means this text is read by R as a comment
> |
```



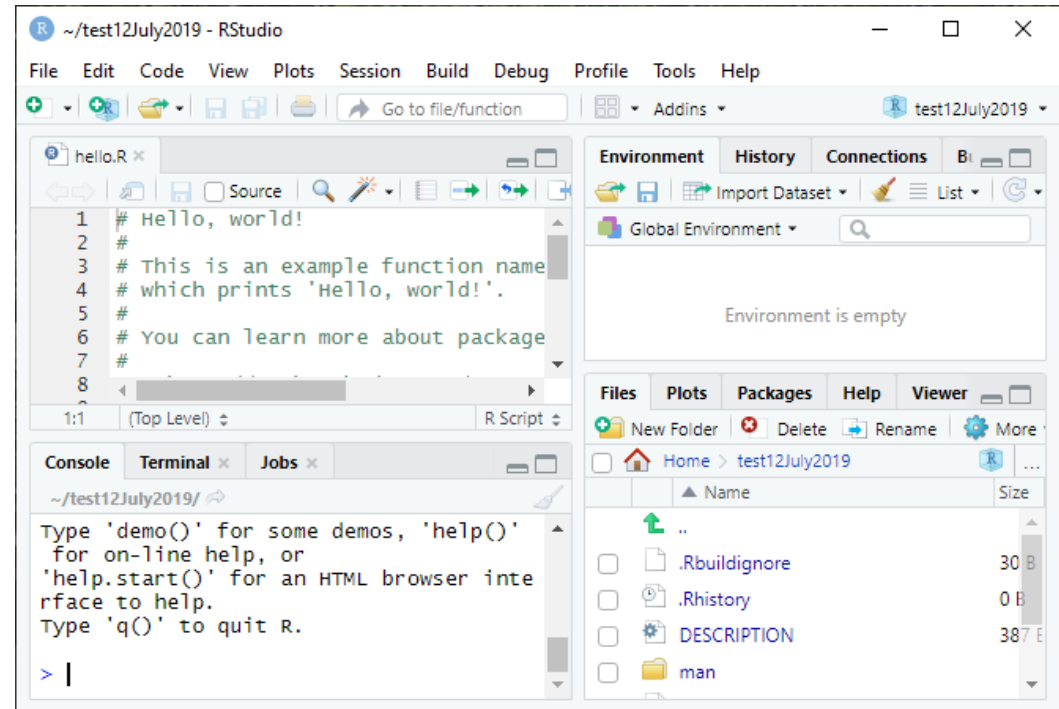

Storing numbers in a variable – 3.4.1 FB

```
R Console (64-bit)
File Edit Misc Packages Windows Help

> sales <- 350
> sales
[1] 350
> royalty<-7
> 350*7
[1] 2450
> sales*royalty
[1] 2450
> revenue <- sales*royalty
> revenue
[1] 2450
> revenue <- revenue + 550
> revenue
[1] 3000
> |
```



Side Note: RStudio is useful and recommended but will not be used in this workshop. See Rstudio.com and [Free Book](#) for more details.



Part 3: More R Console

Functions in R

```
R Console (64-bit)
File Edit Misc Packages Windows Help
> # Functions are often used (command line, scripts, R programs)
> # Example of a function: citation()
> # Functions can return a value that can be stored in a variable
> # Variable names can be a letter like X or a word like food but a
> # variable name cannot be an R keyword like TRUE or NA. Some examples:
> 10^2
[1] 100
> X <- sqrt(100) # X is a variable and sqrt is an R function
> X
[1] 10
> X^2
[1] 100
> abs(13)
[1] 13
> abs(-13)
[1] 13
> sqrt(1+(-8))
[1] NaN
Warning message:
In sqrt(1 + (-8)) : NaNs produced
> sqrt(1+abs(-8))
[1] 3
> round(3.1416, 2) # Functions can take optional arguments like 2 here
[1] 3.14
> # Hit the up arrow on keyboard to get history of previous commands
```

R Vectors

Note: Use copy and paste, scripts, and R programs rather than typing in at command prompt.

These lines of R could be part of a script.

```
R Console (64-bit)
File Edit Misc Packages Windows Help

> # Often results are returned as a vector
> # Note, I usually don't type in commands (see previous image to spot the reason why)
> # This looks like command line entry but I really just copied and pasted from notepad (PC app)
> # For reproduction ease: use scripts, programs, and cutting and pasting
>
> # R vectors, note variable name can have periods in them...
> # c() creates a vector of data like this... ex. below puts data in a variable
> sales.by.month <- c(0, 100, 200, 50, 0, 0, 0, 0, 0, 0, 0, 0)
> sales.by.month
[1] 0 100 200 50 0 0 0 0 0 0 0 0
>
> sales.by.month[2]
[1] 100
>
> february.sales <- sales.by.month[2]
> february.sales
[1] 100
>
> sales.by.month[5] <- 25
> sales.by.month
[1] 0 100 200 50 25 0 0 0 0 0 0 0
>
> length( x = sales.by.month )
[1] 12
>
> sales.by.month * 7
[1] 0 700 1400 350 175 0 0 0 0 0 0 0
>
> days.per.month <- c(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
> profit <- sales.by.month * 7
>
> profit / days.per.month
[1] 0.000000 25.000000 45.161290 11.666667 5.645161 0.000000 0.000000 0.000000 0.000000
[10] 0.000000 0.000000 0.000000
>
> |
```

R text data and logic data

Note...

More truth
on the next
slide...

```
R Console (64-bit)
File Edit Misc Packages Windows Help

> #Storing text data
> greeting <- "hello"
> greeting
[1] "hello"
> #R stores the entire word "hello" as a single element, it is not a vector of five different letters
> #The following is a character vector containing 12 elements
> months <- c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")
> months
[1] "January" "February" "March" "April" "May" "June" "July" "August" "September" "October" "November" "December"
> #Note the character + is seen when extending a line of R code at the R prompt, so the following is the same as above
> months <- c("January", "February", "March", "April", "May", "June",
+ "July", "August", "September", "October", "November", "December")
>
> months
[1] "January" "February" "March" "April" "May" "June" "July" "August" "September" "October" "November" "December"
> # months is a vector (don't type Months, because it's not the same as months)
> Months[4]
Error: object 'Months' not found
> months[4]
[1] "April"
> #Storing logical data (true/false data)
> # The following is a logical operator in R: ==
> 2 + 2 == 4
[1] TRUE
> 2 + 2 == 5
[1] FALSE
> # the R keyword FALSE denotes a logically false value
> var.false <- 2 + 2 == 5
> # thus the variable var.false contains a logically false value
> var.false.too <- FALSE
>
> var.false
[1] FALSE
> var.false.too
[1] FALSE
>
> # logical values in vectors
> x <- c(TRUE, TRUE, FALSE)
> x
[1] TRUE TRUE FALSE
> sales.by.month <- c(1,2,3,4,5,6,7,8,9,8,0,0)
> annual.sales <- sales.by.month > 0
> annual.sales
[1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE FALSE FALSE
>
```

More R logic

```
R Console (64-bit)
File Edit Misc Packages Windows Help

> #Operation,operator,example,input,answer
> #less than,<,2 < 3,TRUE
> #less than or equal to,<=,2 <= 2,TRUE
> #greater than,>,2 > 3,FALSE
> #greater than or equal to,>=,2 >= 2,TRUE
> #equal to,==,2 == 3,FALSE
> #not equal to,!=,2 != 3,TRUE
>
> 99 < 100
[1] TRUE
> 100 > 100
[1] FALSE
> 100 <= 99
[1] FALSE
> 100 >= 100
[1] TRUE
> 100 == 99
[1] FALSE
> 2 + 2 != 5
[1] TRUE
>
> #operation,operator,example input,answer
> #not,!,!(1==1),FALSE
> #or,|(1==1) | (2==3),TRUE
> #and,&,(1==1) & (2==3),FALSE
>
>
> (2+2 == 4) | (2+2 == 5)
[1] TRUE
> (2+2 == 4) & (2+2 == 5)
[1] FALSE
> !(2+2 == 5)
[1] TRUE
> ! FALSE
[1] TRUE
> 2+2 != 5
[1] TRUE
>
> # Logic with text data
> "cat" == "dog"
[1] FALSE
> "cat" == "cat"
[1] TRUE
> " cat" == "cat"
[1] FALSE
> "cat" == "CAT"
[1] FALSE
> "cat" == "c a t"
[1] FALSE
```

Next
Slide:
Vector
Indexing

R Vector indexing

Yet more logic

Wrap up...

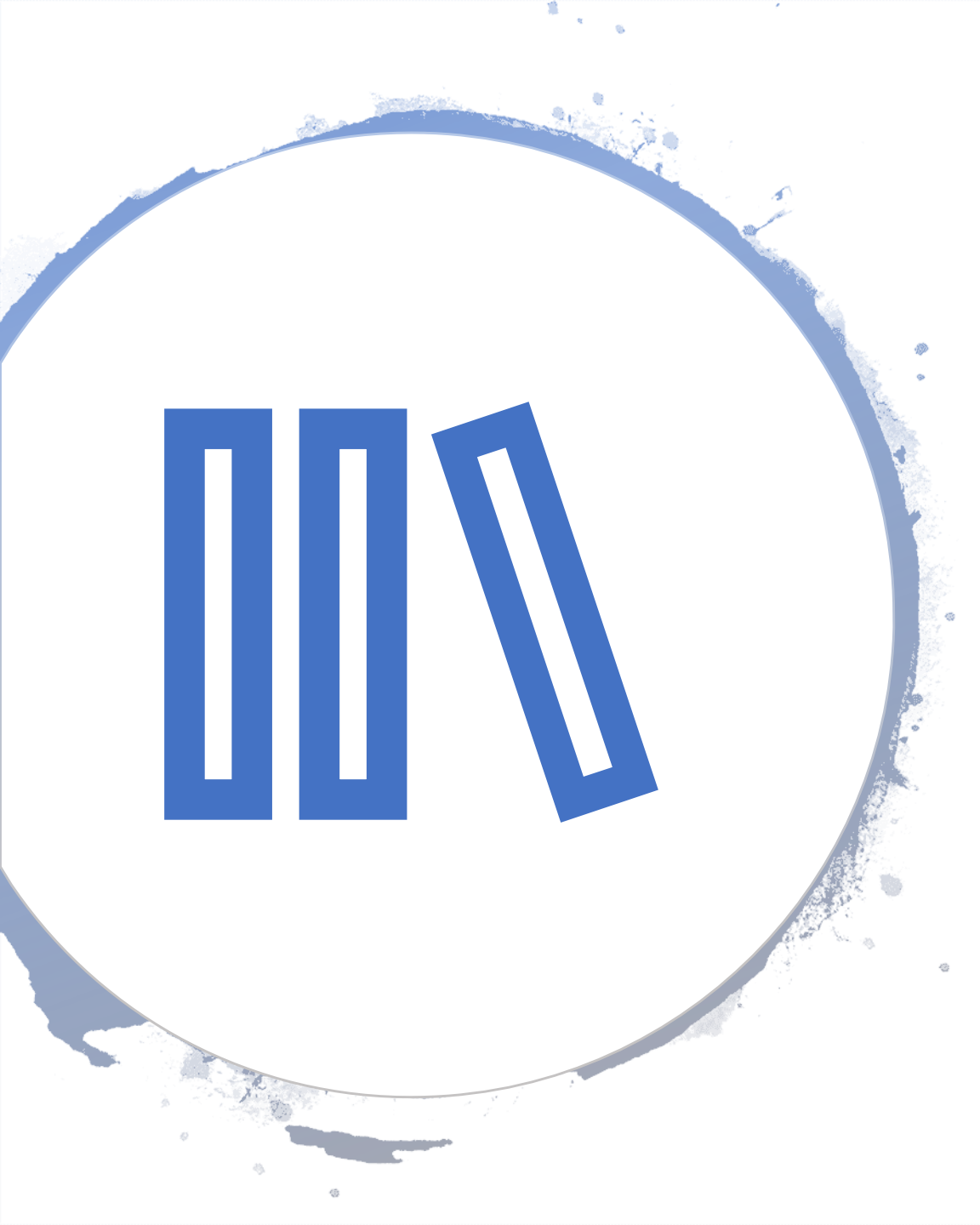
```
R R Console (64-bit)
File Edit Misc Packages Windows Help

> # Vectors and Extracting multiple elements
>
> # The following illustrates why programs in R can be shorter than in a programming language like Java
> # Plus side is shorter code, minus side is that the code can be harder to read
>
> sales.by.month <- c(0, 100, 200, 50, 0, 0, 0, 0, 0, 0, 0, 0)
> sales.by.month[5] <- 25
> sales.by.month
 [1]  0 100 200  50 25  0  0  0  0  0  0  0
>
> months <- c("January", "February", "March", "April", "May", "June",
+ "July", "August", "September", "October", "November", "December")
>
> sales.by.month[ c(2,3,4) ]
 [1] 100 200  50
> sales.by.month[ c(4,3,2) ]
 [1]  50 200 100
> sales.by.month[ c(2,3,4,5,6,7,8) ]
 [1] 100 200  50 25  0  0  0
>
> 2:8
 [1] 2 3 4 5 6 7 8
>
> sales.by.month[2:8]
 [1] 100 200  50 25  0  0  0
>
> months[ sales.by.month > 0 ]
 [1] "February" "March"  "April"  "May"
>
> #create a variable
> any.sales.this.month <- sales.by.month > 0
> any.sales.this.month
 [1] FALSE TRUE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE
>
> months[ sales.by.month > 0 ]
 [1] "February" "March"  "April"  "May"
> months[ any.sales.this.month ]
 [1] "February" "March"  "April"  "May"
> sales.by.month [ sales.by.month > 0 ]
 [1] 100 200  50 25
>
> stock.levels <- c("high" ,"high", "low", "out", "out", "high", "high", "high" ,"high" ,"high", "high","high")
> stock.levels
 [1] "high" "high" "low" "out" "out" "high" "high" "high" "high" "high" "high" "high"
> months[stock.levels == "out"]
 [1] "April" "May"
>
> # months low or out
> months[stock.levels == "out" | stock.levels == "low"]
 [1] "March" "April" "May"
> # months not "high"
> months[stock.levels != "high" ]
 [1] "March" "April" "May"
```



R Workshop Part One: Wrap-up

- Best to learn in front of a computer
- Covered so far: highlights through chapter 3 of free book



Questions?

- Questions about the slides?
- Questions about the free book?
- Questions about implementing what has been covered on a computer?

Next... R Workshop Part Two

- Chapter 4 of Free Book
- More Basic R concepts