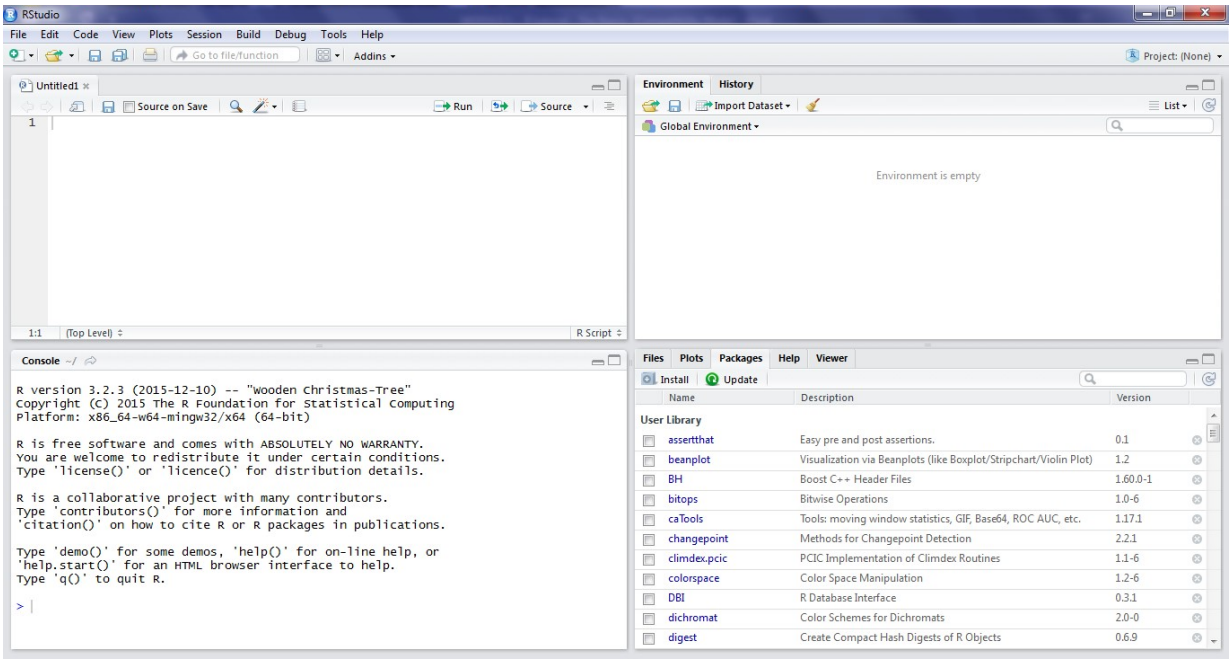
Instructions to Download R, RStudio, and Packages

A) You must download base R first, since it must be installed on your computer for RStudio to work.
   1) Go to this webpage: https://cloud.r-project.org/
   2) Under "Download and Install R," click the link that corresponds to your computer's operating system.
   3) FOR WINDOWS:
      a) In the first line (after "base"), click the link "install R for the first time".
      b) Click "Download R 4 …" at the top of the page.
   4) FOR MAC:
      a) Under "Latest Release", click on the first file name that ends in ".pkg".
   5) After downloading, proceed as you normally would to complete installation of a program.

B) Next, install RStudio – this requires base R to already be installed on your computer.
   1) Go to this webpage: https://www.rstudio.com/products/rstudio/download/#download
   2) Under "All Installers," click the link that corresponds to your computer's operating system.
   3) After downloading, proceed as you normally would to complete installation of a program.

C) I recommend opening RStudio after installation is complete to ensure it runs correctly. You should see a screen that looks something like this:



D) Install four packages in RStudio that we will use during the workshop.
   1) Open RStudio.
   2) Click the "Tools" toolbar at the top of the screen.

Computer Science in Modern Biology

Introduction to R

Instructor:
Rachel Pilla

TAs:
Andrew Cannizzarro

MIAMI

Nicole Berry

1

Workshop Schedule

Day 1 (Monday):

- Introduction to R
- Coding basics
- Working with data in R
- Troubleshooting R scripts

Day 2 (Tuesday):

- Formatting and manipulating data
- Basic analyses and plotting
- Practice group exercises

2

Workshop Logistics

Zoom
- Feel free to unmute yourself to ask questions at any point

Chat
- Post any questions or comments here
- The TAs will be checking the chat to help answer questions

Break-Out Rooms
- If you are really stuck, you can move to a break-out room with a TA to share your screen and get you up to speed
- Break time is a great opportunity for this if needed!

3

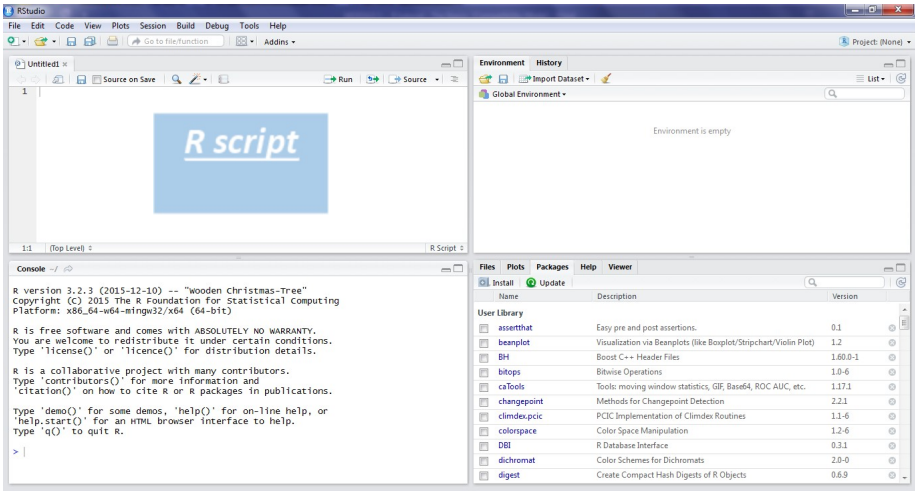
What is R?

- FREE and OPEN SOURCE statistical and computational program
- Simple Google searches can give you effective solutions to questions and troubleshooting issues in R
- Widely used in sciences and rapidly growing in popularity
- Can handle more advanced computations, statistical analyses, and bigger data files than Excel
- Lots of styles to code and get the exact same output or result

4

Introduction & Basics in R

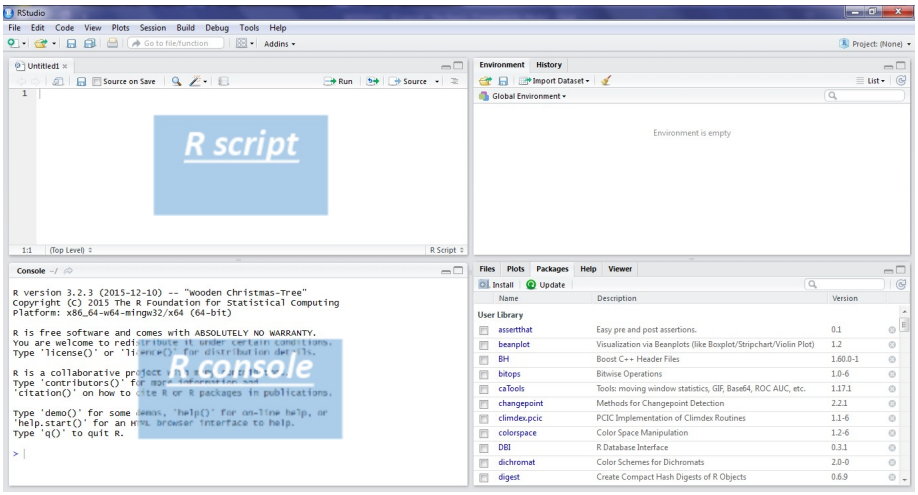
5



R script:

- This is where you write and edit your code and comments
- RStudio color codes your scripts to make it easier to identify names of functions, numbers, and comments
- It also automatically fills in parentheses & quotations

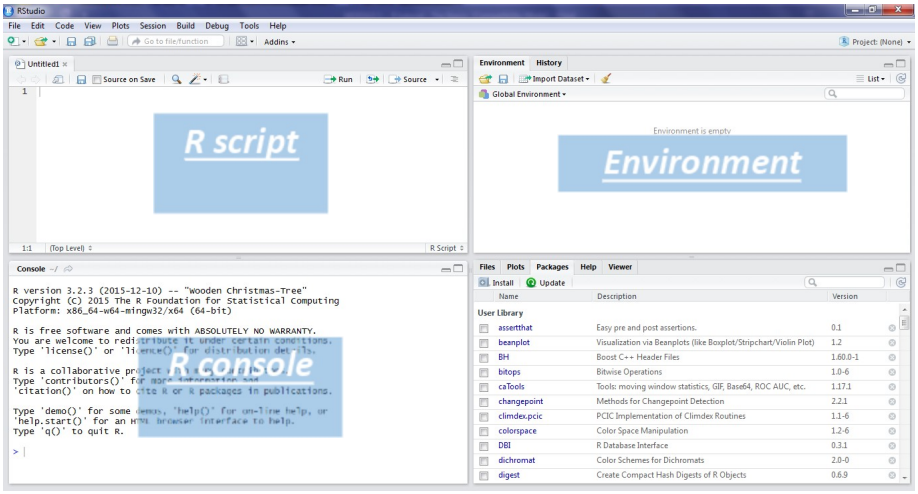
6



R console:

- This is where your commands are run, and where results appear
- The "**>**" symbol in blue means R is ready to work.
- If you see a "**+**" after running code, then *something did not finish running*, and/or you have an error or forgot a parenthesis

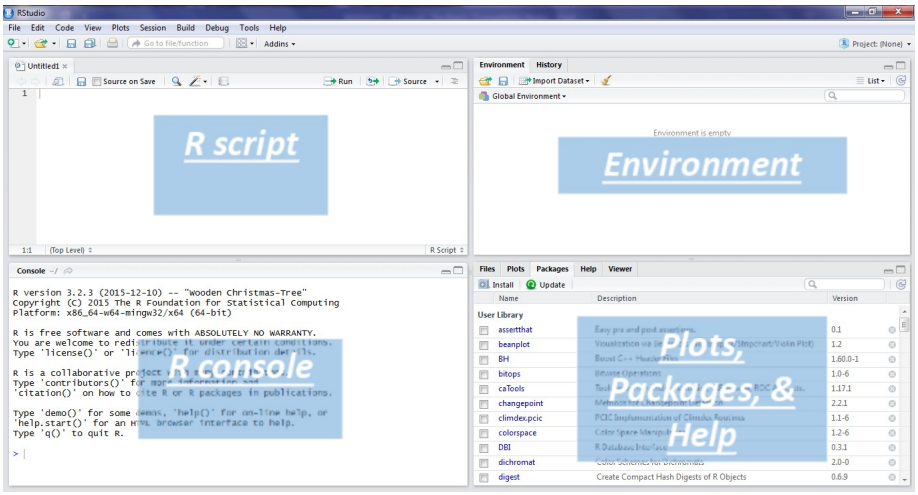
7



Environment:

- This is where you can see which objects you have created
- It also tells you what *type* of object it is, and it's size and dimensions

8



Plots/Packages/Help:

- This is where your plots will appear
- You can also see what packages are installed and available
- *Most importantly*, this is where you find help files that tell you how to use functions/arguments

9

Packages in R

- ***Packages*** are bundles of tools and functions that others have developed to be used in R
- They are often grouped to specific types of functions, analyses, or datasets
- `rLakeAnalyzer`, for example, has lots of functions developed by limnologists to help you analyze common types of data collected from lakes

10

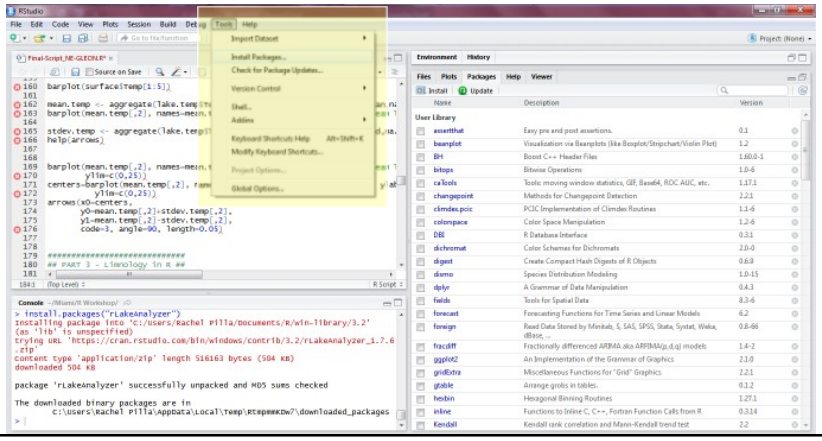
Packages in R

- Currently, there are over 16,000 packages!
- We are going to install 2 packages to use today:
  1) `dplyr`
  2) `tidyr`

11

Installing a Package

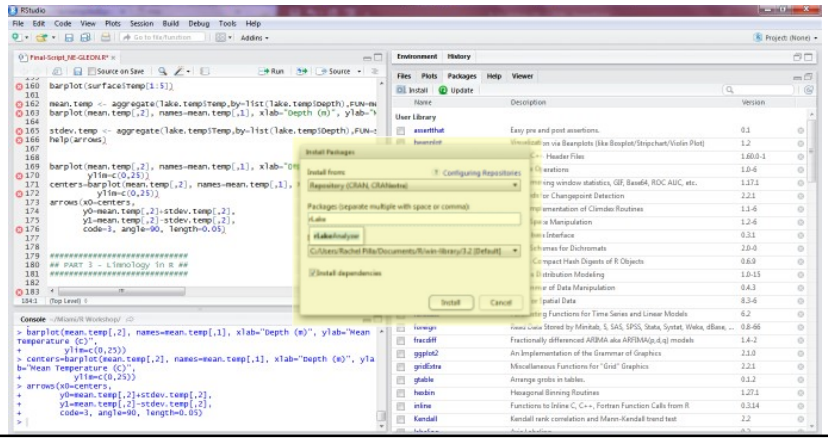
- Tools → Install Packages…



12

Installing a Package

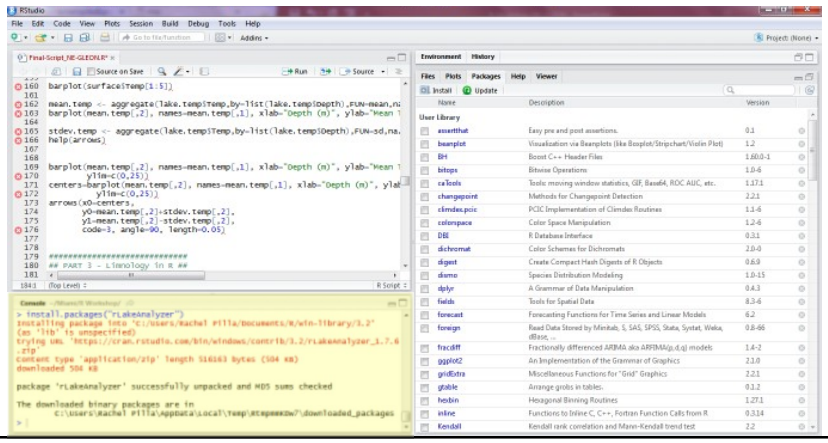
- Type the packages you want to install:
  `dpylr, tidyr`



13

Installing a Package

- Console will tell you when they have been installed

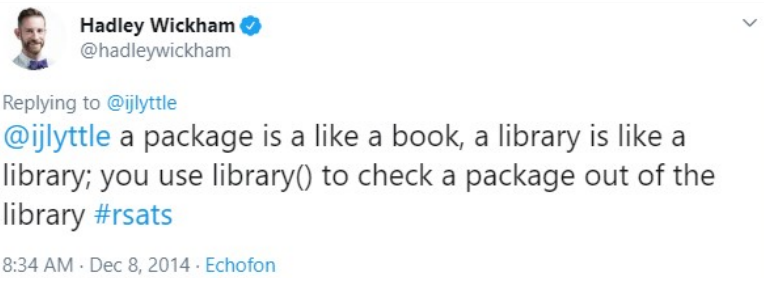


14

Installing a Package

- Packages only need to be installed ONCE
- But, each time you re-open R, you need to load the package(s) you want to use so R can make all the functions available:

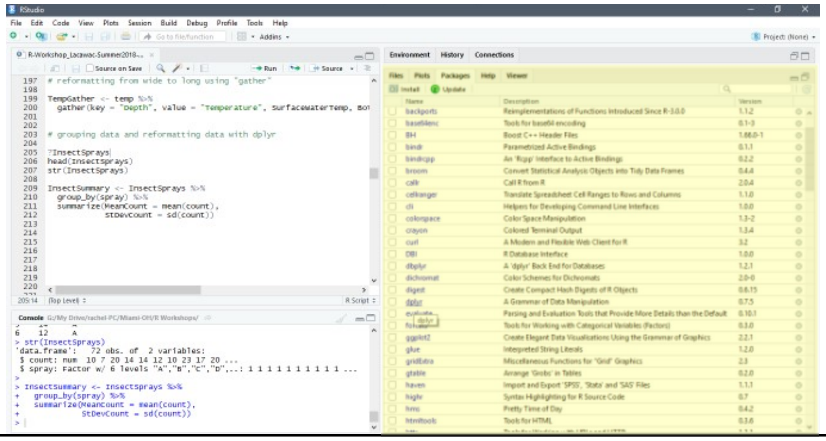
```
library(package name)
```



15

Package Information

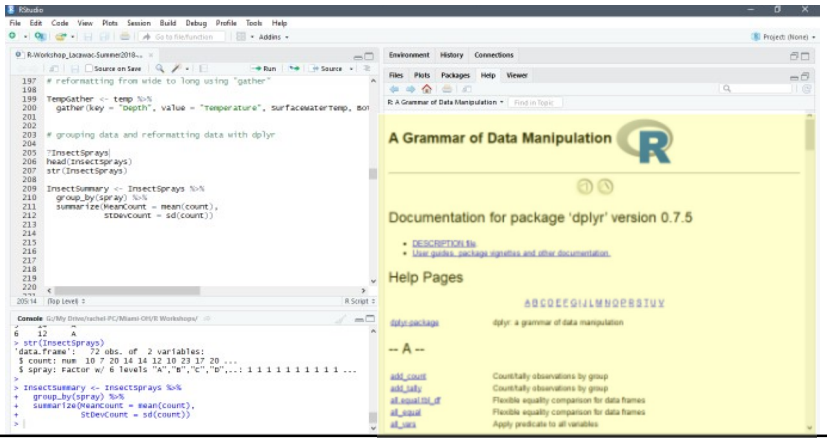
- Under the "Packages" tab, click on **`dplyr`**



16

Package Information

- This will list all the available functions
- Clicking on a function takes you to the help file



17

Using an R Script

- Write commands in the script (top left pane)
  - Save it, edit it, revisit it later, etc.
- Code NOT automatically run when you hit Enter
- To run code:
  - "Run" button in upper right corner of R Script
  - "CTRL + Enter" shortcut on PC
  - "Command + Return" shortcut on Mac

18

Using an R Script

- Add in ***comments*** using "#"
  - Comments can be any information in an R script that will NOT be run, but can be used for notes/explanations/etc.
  - Anything behind "#" will be ignored by R
- R is case sensitive
  - "Mean" ≠ "mean"
- R doesn't care too much about spaces (or lack thereof)

19

Using an R Script

- Be careful to close all parentheses & quotations (and in the correct spots!)
  - A majority of your errors will come from missing or misplaced parentheses or quotations – but they're usually easy fixes!
- ">" at the beginning of Console means it's ready to go!
  - "+" means code is still running or it is not complete
  - Often missing parenthesis, bracket, or quotation

20

Key Components of Code

- ***Functions*** allow you to manipulate data, apply calculations, run statistical analysis, and more
- ***Arguments*** are the defining information for functions, and allow you to "customize" the function to have it do what you want it to do
- ***Objects*** are pieces of data saved in R that can be called up, reused, or manipulated

21

Key Components of Code

```
x <- seq(1,10)
```

name of the ***function*** to create a sequence

22

Key Components of Code

```
x <- seq(1,10)
```

the ***arguments*** to define the function, to create a sequence from 1 through 10

23

Key Components of Code

```
x <- seq(1,10)
```

name of the ***object*** that saves the results of the function in R

24

Key Components of Code

```
x <- seq(1,10)
```

names of ***objects*** must begin with a letter, but can include uppercase or lowercase letters, numbers, underscores, and periods – make your object names clear and useful!

25

Key Components of Code

```
x <- seq(1,10)
```

assignment operator tells R to save the result of the function as the named object

26

Types of Data

An object can hold data that is:

- Numeric
- Integer
- Characters (strings)
- Logical (TRUE/FALSE)
- Complex (1+4*i*)

27

Types of Data Structures

- An object can be in the form of:
  - Vector
    - 1-D object with the same data type
  - Matrix
    - 2-D object with the same data type
  - Data frame
    - 2-D object with different data types for each column if desired
    - Generally the most useful for biological and ecological data

28

Live Coding

open RStudio and code along!

29

Computer Science in Modern Biology

Introduction to R

Instructor:
Rachel Pilla

TAs:
Andrew Cannizzarro

MIAMI

Nicole Berry

30

Review of Day 1

31

Workshop Logistics

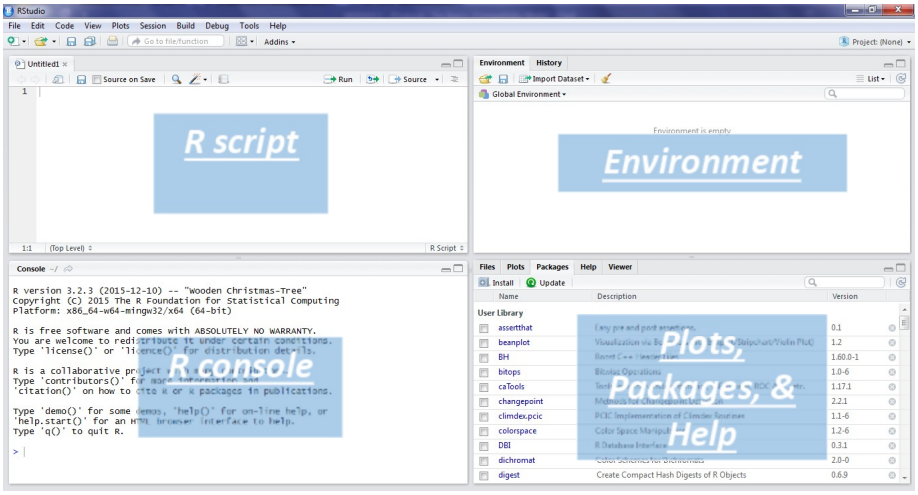
Zoom
- Feel free to unmute yourself to ask questions at any point

Chat
- Post any questions or comments here
- The TAs will be checking the chat to help answer questions

Break-Out Rooms
- If you are really stuck, you can move to a break-out room with a TA to share your screen and get you up to speed
- Break time is a great opportunity for this if needed!

32

Layout of RStudio

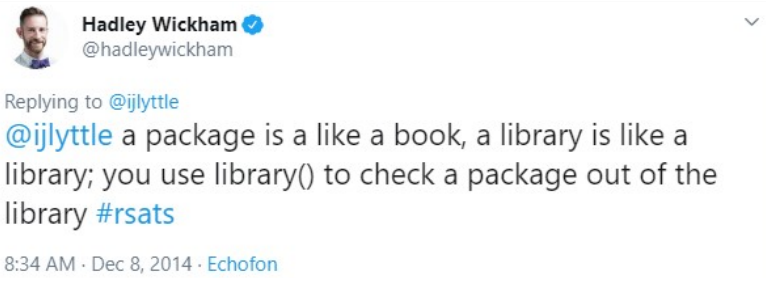


33

Packages

- Packages only need to be installed ONCE
- But, each time you re-open R (***today!***), you need to load the packages you want to use so R can make all the functions available:

```
library(package name)
```



34

Using an R Script

- To run code:
  - "Run" button in upper right corner of R Script
  - "CTRL + Enter" shortcut on PC
  - "Command + Return" shortcut on Mac
- ">" at the beginning of Console means it's ready to go!
  - "+" means code is still running or it is not complete
  - Often missing parenthesis, bracket, or quotation

35

Key Components of Code

- ***Functions*** allow you to manipulate data, apply calculations, run statistical analysis, and more
- ***Arguments*** are the defining information for functions, and allow you to "customize" the function to have it do what you want it to do
- ***Objects*** are pieces of data saved in R that can be called up, reused, or manipulated

36

Types of Data

An object can hold data that is:

- Numeric
- Integer
- Characters (strings)
- Logical (TRUE/FALSE)
- Complex (1+4*i*)

37

Types of Data Structures

- An object can be in the form of:
  - Vector
    - 1-D object with the same data type
  - Matrix
    - 2-D object with the same data type
  - Data frame
    - 2-D object with different data types for each column if desired
    - Generally the most useful for biological and ecological data

38

Live Coding

open RStudio and code along!

39

Additional Tips & Useful Resources

40

"NA" Values

- R recognizes "NA" or blank cells from Excel as missing values
- If you have another format for indicating "NA" values, you can add an argument when importing your data file:

```
read.csv( … , na.strings = "n/a")
```

- Many functions will automatically ignore/skip over "NA" values
  - Be sure you read the associated help files to ensure the function is treating "NA" values as you are expecting, or adjust the argument accordingly!

41

Assorted Useful Functions

- `ncol(…)`
- `nrow(…)`
- `colnames(…)`
- `rownames(…)`
- `length(…)`
- `dim(…)`
- `unique(…)`
- `paste(…)`
- `sort(…)`
- `order(…)`
- `t(…)`
- `cbind(…)`
- `rbind(…)`
- `rep(…)`
- `seq(…)`
- `range(…)`
- `sum(…)`
- `sd(…)`
- `var(…)`
- `diff(…)`
- `round(…)`
- `ifelse(…)`

42

Swirl

- Package for self-guided learning of R in the R environment
- Learn at your own pace and select topics most useful to you
  - Introductory material
  - Regressions
  - Data wrangling
  - Statistical inference
  - Advanced programming
- https://swirlstats.com/

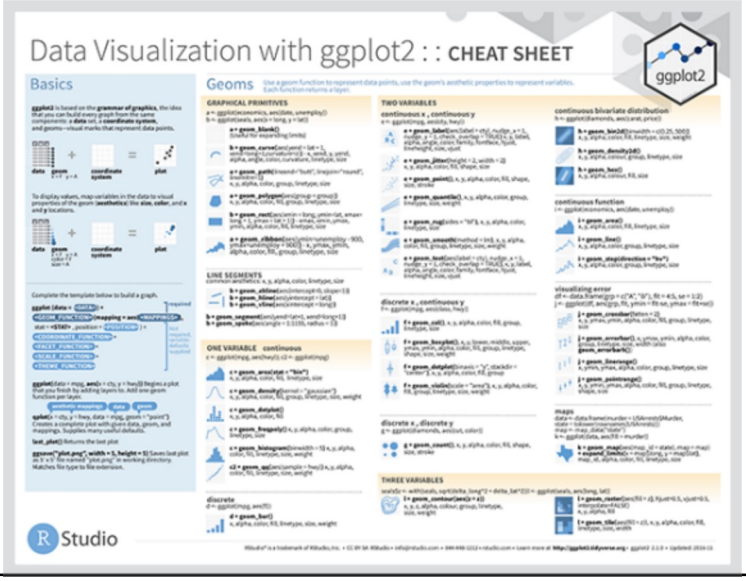
43

RStudio Cheat Sheets

- Short, handy guides to help with many packages and programming needs in RStudio
- Available free for download in multiple language options
- https://rstudio.com/resources/cheatsheets/

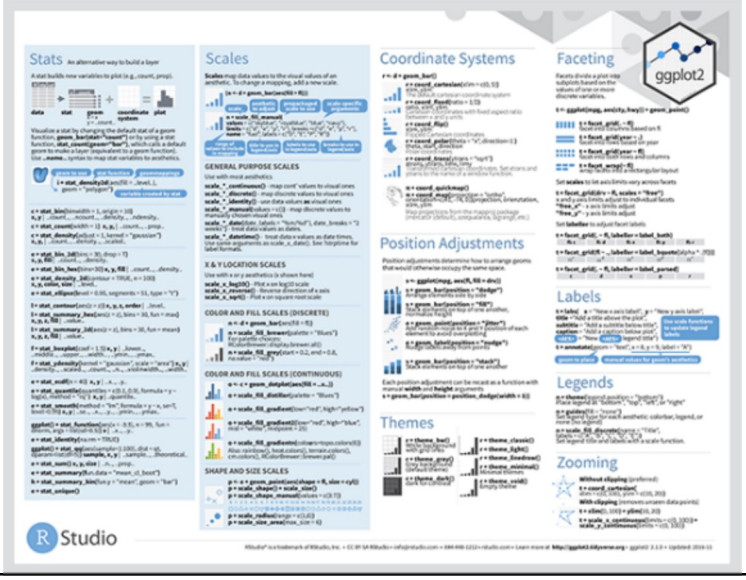
44

ggplot2 Cheat Sheet



45

ggplot2 Cheat Sheet



46

Tidyverse

- Packages to help organize data and streamline R coding, specifically for data science
- Functions in the packages share similar syntax and coding structure
- “dplyr”, “tidyr”, and “ggplot2” are part of tidyverse
- https://www.tidyverse.org/

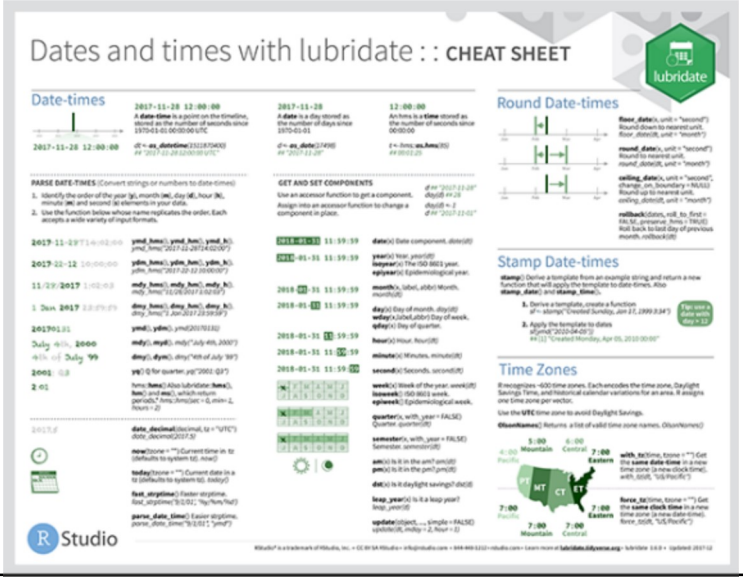
47

Lubridate

- Package for more effectively working with dates between Excel and R
- Can recognize date/times with minimal formatting issues
- Also allows for date/time differencing, among other useful functions
- Included in tidyverse
- https://lubridate.tidyverse.org/

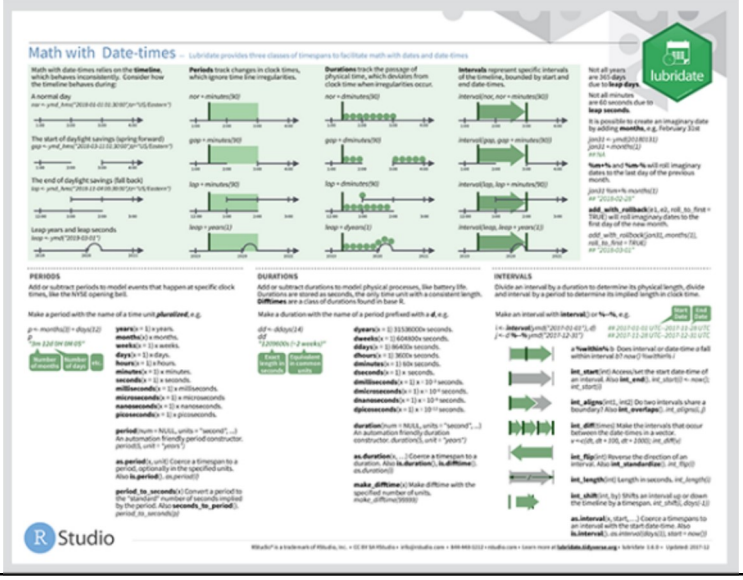
48

Lubridate



49

Lubridate



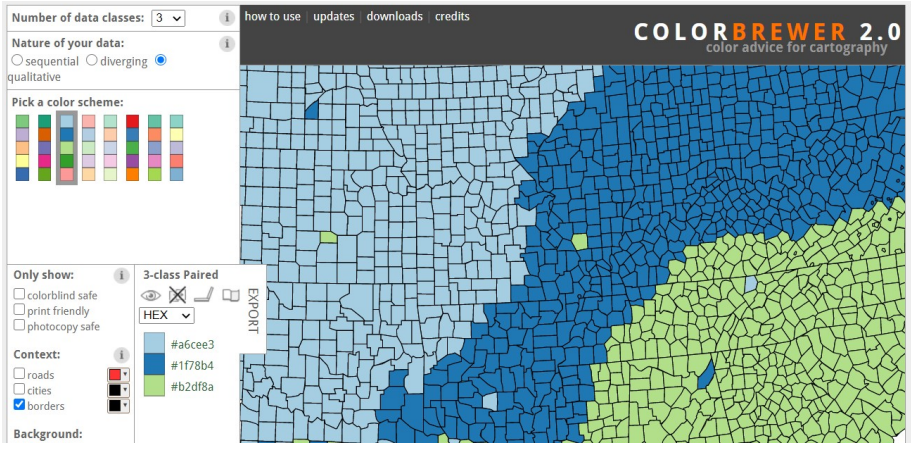
50

ColorBrewer

- Pre-made color schemes and palettes
- Effective graphic color selections for maximum perception
  - Qualitative, sequential, and diverging
- Includes options for photo-copy friendly, color-blind friendly, etc.
- http://colorbrewer2.org/

51

ColorBrewer



52

Computer Science in Modern Biology

Thank you for joining!

to share feedback or ask questions,
please feel free to contact me:

pillarm@miamioh.edu
@rmpilla

MIAMI

53

3) Click “Install Packages…”.
4) Make sure “Repository (CRAN)” is selected in the “Install from:” box.
5) Type the names of packages to install in the “Packages” box:
   `dplyr, tidyr`
6) Make sure “Install dependencies” is checked.
7) Click “Install”.
8) If successfully installed, you should see something like this in the “Console” panel:

If you have any problems during the installation,
please email me (pillarm@miamioh.edu) and I’ll do my best to help you out!

Group 1: Mammals (load "ggplot2", and use built-in dataset "msleep")

1) How many different species are included in this data set? How many orders do they represent? Provide numerical results.
2) Is there a correlation between total sleep and REM sleep? Provide statistical results and a visual.
3) Is there a difference in time spend awake based on eating habit? Provide statistical results and a visual.
4) Can body weight explain brain weight of these species? Is this relationship improved when removing outliers? Provide statistical results and visuals.

Group 2: Diamonds (load "ggplot2", and use built-in dataset "diamonds")

1) What is the average length and width of each cut of diamond? Provide numerical results and a visual(s).
2) Is the carat of these diamonds significantly different than 1 (average for center stone)? Provide statistical results.
3) Is there a difference in price based on color? If so, which color(s) have the highest prices? Provide statistical results and a visual.
4) Can price be explained by depth? Provide statistical results and a visual.

Group 3: Housing (load "ggplot2", and use built-in dataset "txhousing")

1) What is the total number of housing sales for each city from 2000-2015 combined? Provide numerical results and a visual.
2) Is median sale price across all cities significantly different across months? Provide statistical results and a visual.
3) Is there a change in average yearly inventory for Dallas or for Houston since 2000? Provide statistical results and a visual(s).
4) Does total number of listings explain total number of sales by city, summed across all years? Which cities are the two highest? Provide statistical results and a visual.

Notes