WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Center for Limnology
University of Wisconsin–Madison

A Short Introduction into Data Science with R

Robert Ladwig
University of Wisconsin-Madison

rladwig2@wisc.edu

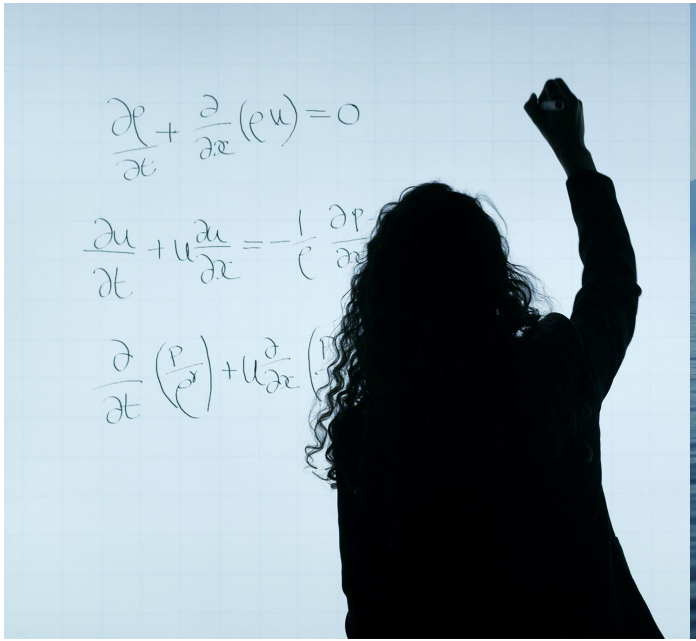
@hydrobert

Disclaimer

- material is based on the AWESOME workshop by Rachel Pilla (now postdoc at Oak Ridge National Lab, @rmpilla) “Introduction to R” (*Computer Science in Modern Biology* at Miami University, Ohio)
- with help by Andrew Cannizzarro, Alva Strand and Nicole Berry

About myself

$$\frac{\partial DO}{\partial t} = \frac{1}{A}\frac{\partial}{\partial z}(AK_z^{DO}\frac{\partial DO}{\partial z}) + S_{DO}$$

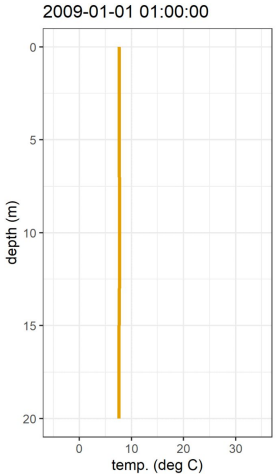


About myself

- **Physical limnologist**
  - use mathematical models to explore mixing dynamics and water quality in lakes

Aquatic ecosystem modeling

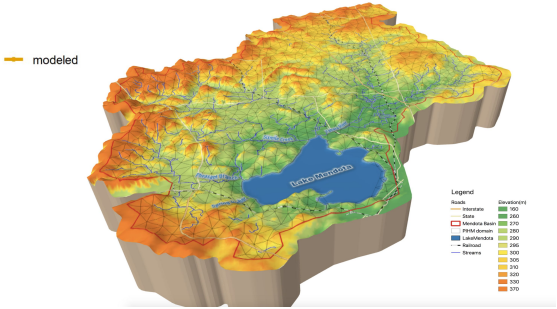
mechanistic understanding of cause and effects



$$\frac{\partial T}{\partial t} = \frac{1}{A}\frac{\partial}{\partial z}\left(A(\upsilon_t^{\cdot} + \upsilon^{\cdot})\frac{\partial T}{\partial z}\right) + \frac{1}{\rho_0 c_p}\frac{\partial H_{sol}}{\partial z} + \frac{dA}{dz}\frac{H_{geo}}{A\rho_0 c_p}$$

$$\frac{\partial [O_2]}{\partial t} = NEP = GPP - ER \pm D$$

$$\frac{\partial P}{\partial t} = D\frac{\partial^2 P}{\partial z^2} + \lambda P\left(1 - \frac{P}{K}\right)$$



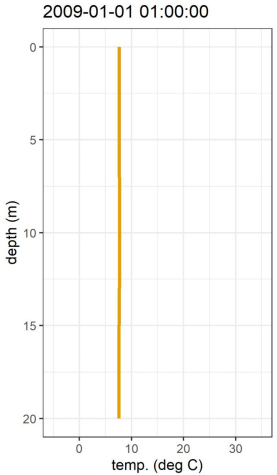
- developing and applying open-source and open-access software in **R, Python and Matlab**
- background in civil engineering and geology

About myself

- **Physical limnologist**
  - use mathematical models to explore mixing dynamics and water quality in lakes

Aquatic ecosystem modeling

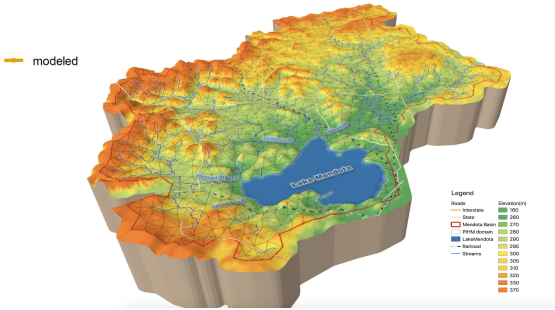
mechanistic understanding of cause and effects



$$\frac{\partial T}{\partial t} = \frac{1}{A}\frac{\partial}{\partial z}\left(A(\upsilon_t^{\cdot} + \upsilon^{\cdot})\frac{\partial T}{\partial z}\right) + \frac{1}{\rho_0 c_p}\frac{\partial H_{sol}}{\partial z} + \frac{dA}{dz}\frac{H_{geo}}{A\rho_0 c_p}$$

$$\frac{\partial [O_2]}{\partial t} = NEP = GPP - ER \pm D$$

$$\frac{\partial P}{\partial t} = D\frac{\partial^2 P}{\partial z^2} + \lambda P(1 - \frac{P}{K})$$

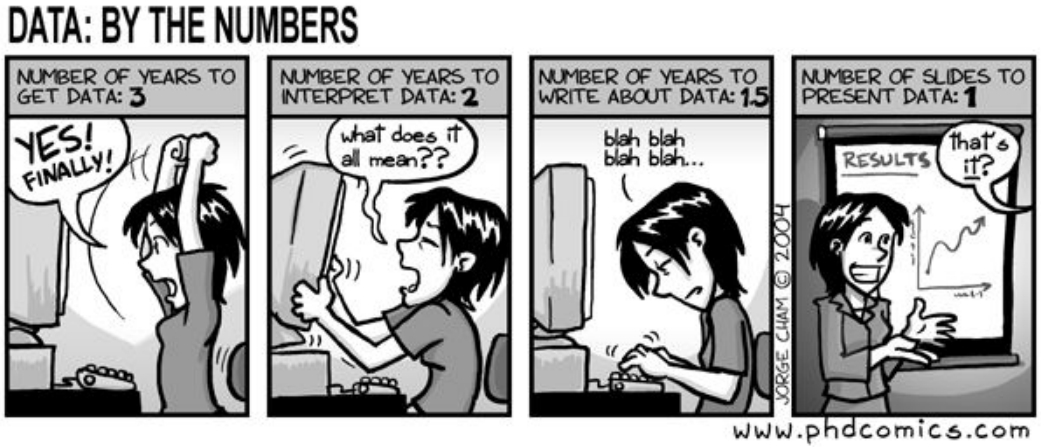


- developing and applying open-source and open-access software in **R, Python and Matlab**
- background in civil engineering and geology

What's your name & data science (or R) experience?

What's data science?

- combining math & statistics, programming and machine learning to explore any kind of data
- data is often noisy or unstructured → cleaning is big part of the job
- buzzwords like big data

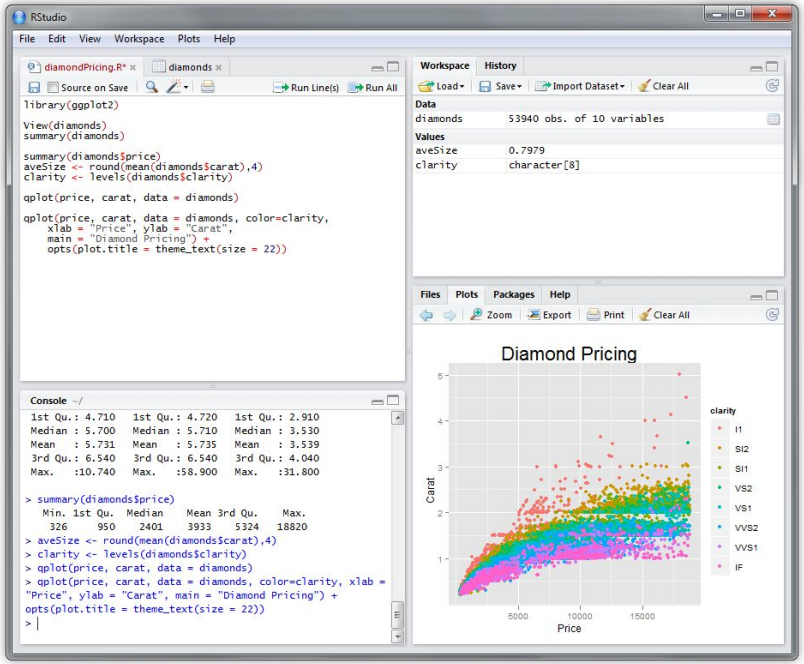


What's the plan for today?

- short introduction into the R scripting language
  - all material on github: https://github.com/robertladwig/introdatasciencer
  - GitHub: uses Git (version control), fantastic for coding and collaborations
- hands-on coding to introduce you to data analysis
- **at the end**: able to know basic R commands, load data, analyze data, plot data

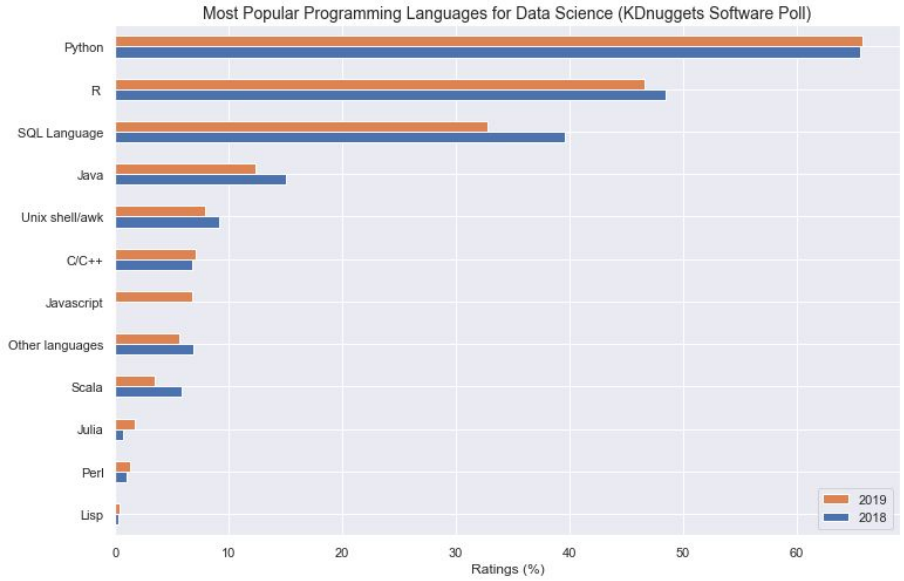
Install R and RStudio

- Install R from here (language): https://www.r-project.org/
- Install RStudio (editor and GUI): https://rstudio.com/products/rstudio/download/#download



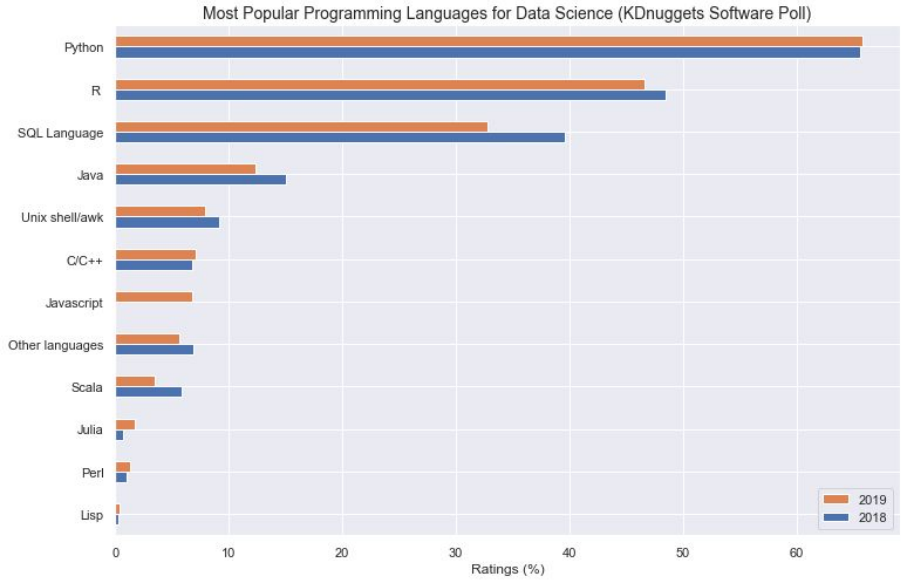
What's R?

- FREE and OPEN SOURCE statistical and computational software



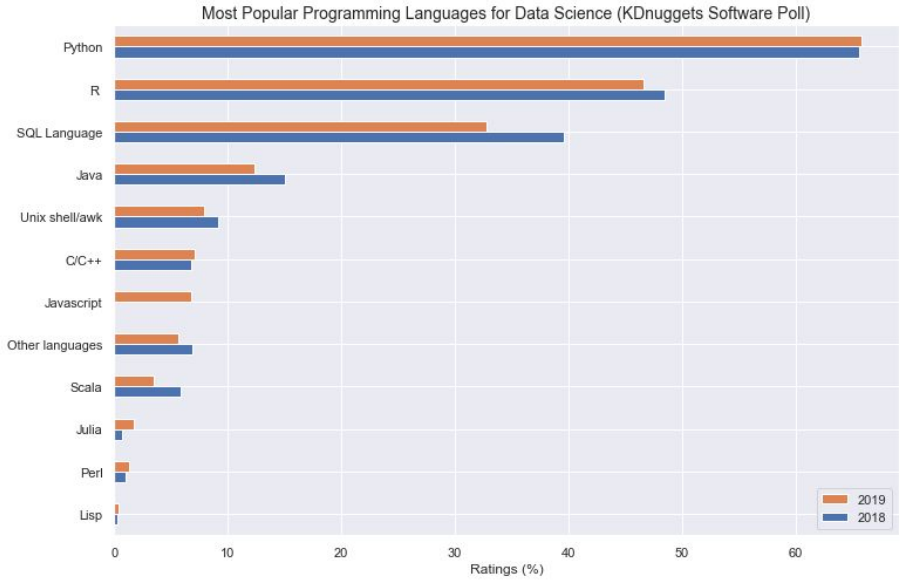
What's R?

- FREE and OPEN SOURCE statistical and computational software
- big community: easy to find solutions and troubleshooting online



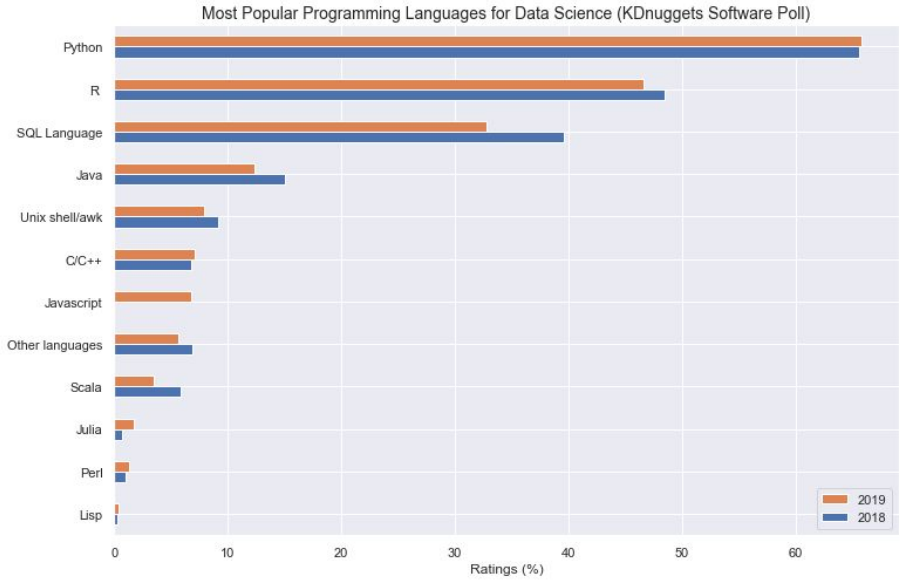
What's R?

- FREE and OPEN SOURCE statistical and computational software
- big community: easy to find solutions and troubleshooting online
- widely used in sciences (esp. statistics and visualization) and rapidly growing in popularity



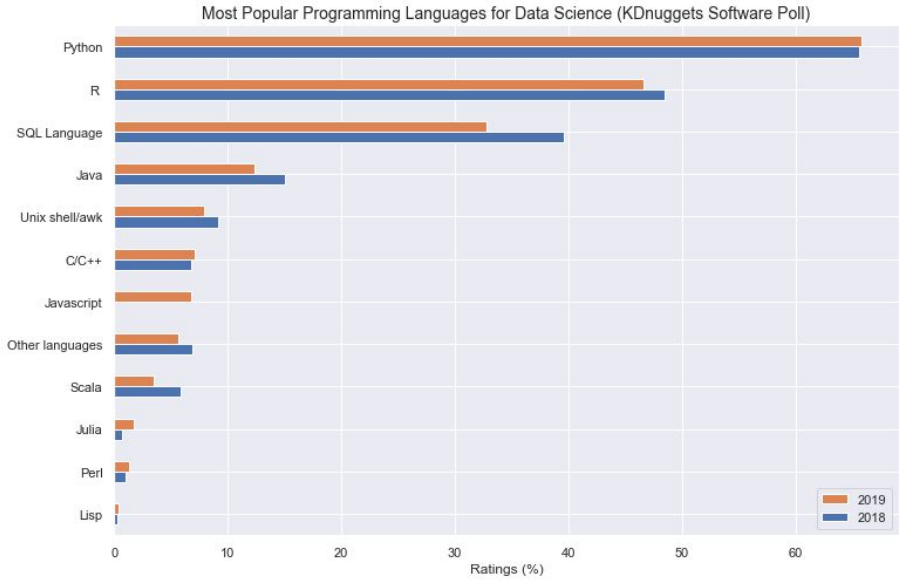
What's R?

- FREE and OPEN SOURCE statistical and computational software
- big community: easy to find solutions and troubleshooting online
- widely used in sciences (esp. statistics and visualization) and rapidly growing in popularity
- can handle more advanced computations, statistical analyses and bigger data files than Excel

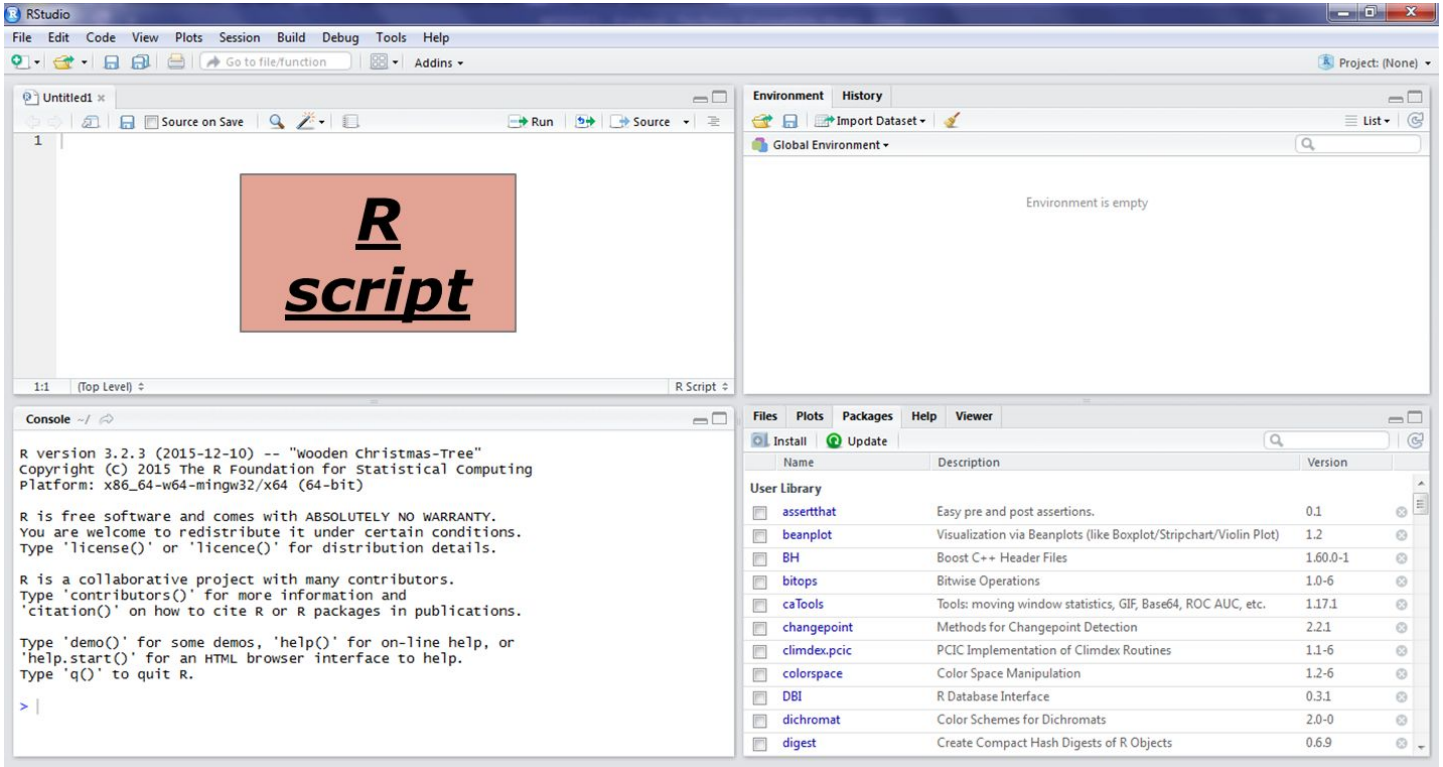


What's R?

- FREE and OPEN SOURCE statistical and computational software
- big community: easy to find solutions and troubleshooting online
- widely used in sciences (esp. statistics and visualization) and rapidly growing in popularity
- can handle more advanced computations, statistical analyses and bigger data files than Excel
- lots of styles for coding



Open RStudio on your laptop now



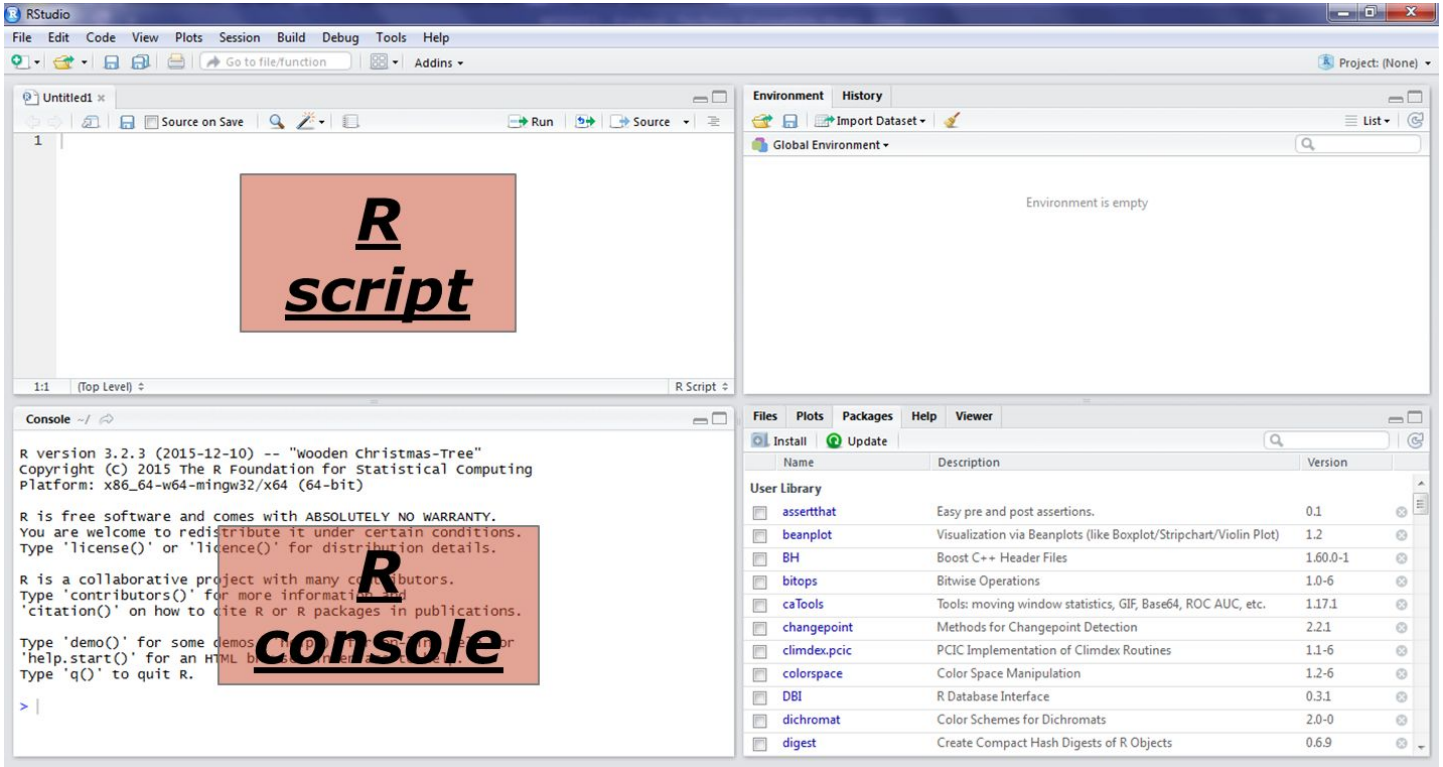
R script:

write and edit code

color-coding for easier identification

automatically fills in parentheses and quotations

Open RStudio on your laptop now

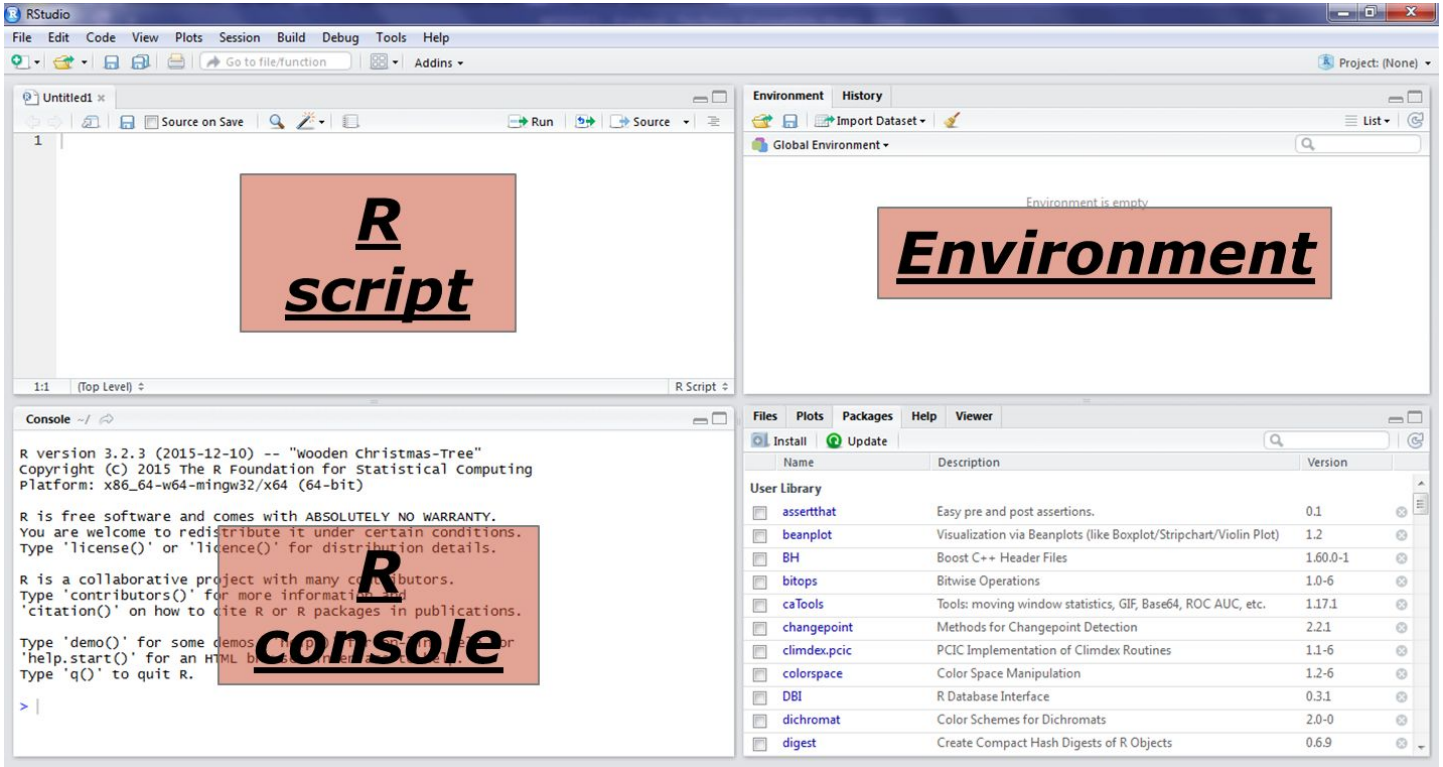


R console:

commands are run and results appear

> means ‘ready to work’

Open RStudio on your laptop now

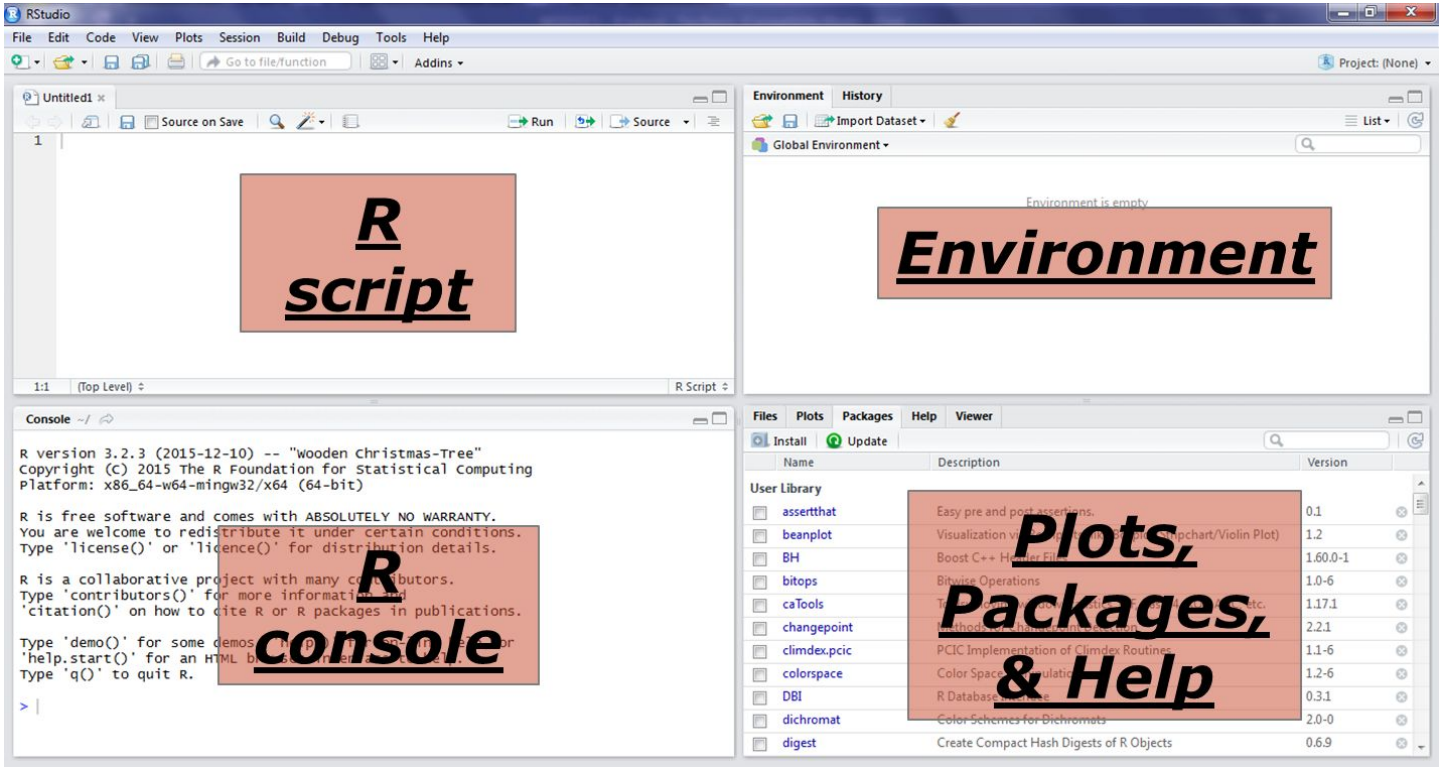


Environment:

see which objects were created

type of objects (double, integers, etc.), size and dimensions

Open RStudio on your laptop now



Plots/Packages/ Help:

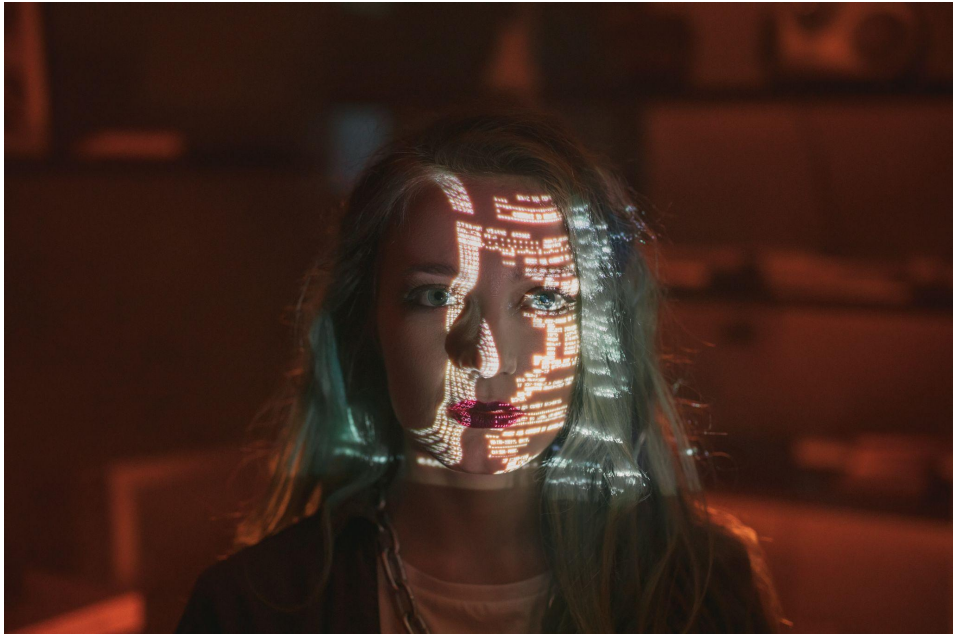
plots will appear

see which packages are installed

help functions

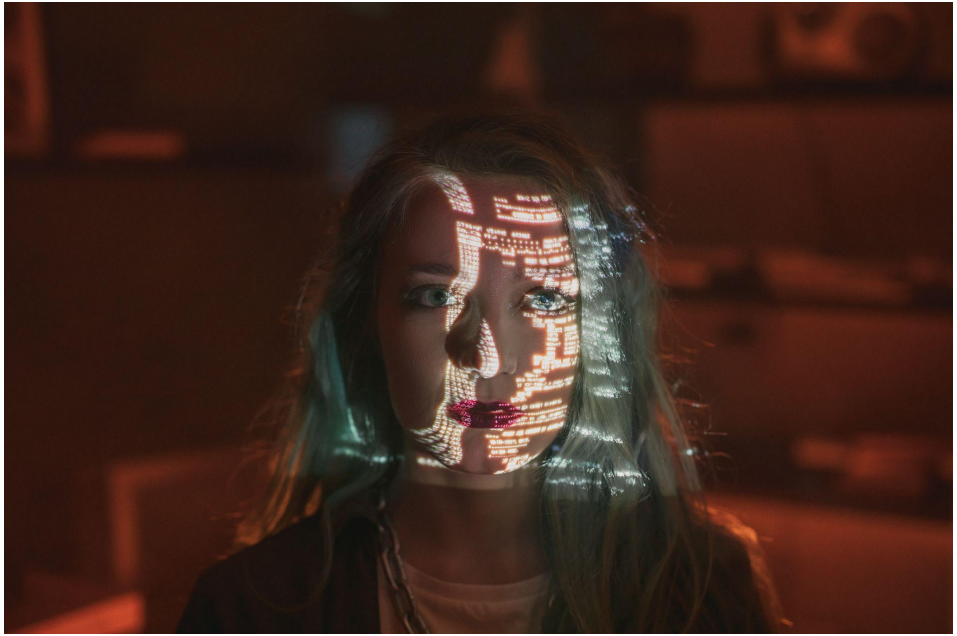
What are packages?

- bundles of tools and functions that others have developed to be used in R



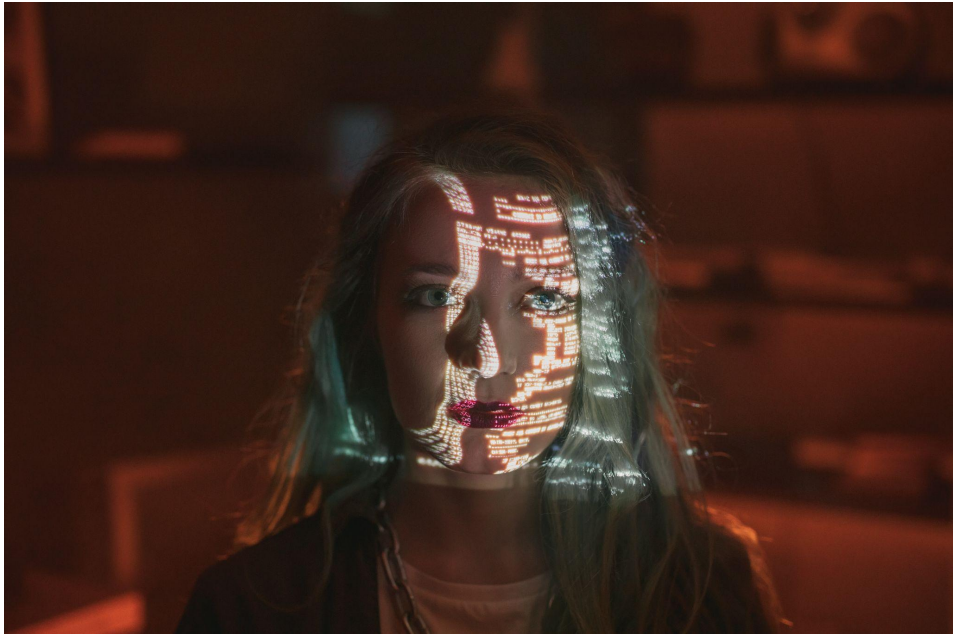
What are packages?

- bundles of tools and functions that others have developed to be used in R
- grouped to specific types of functions, analyses or datasets



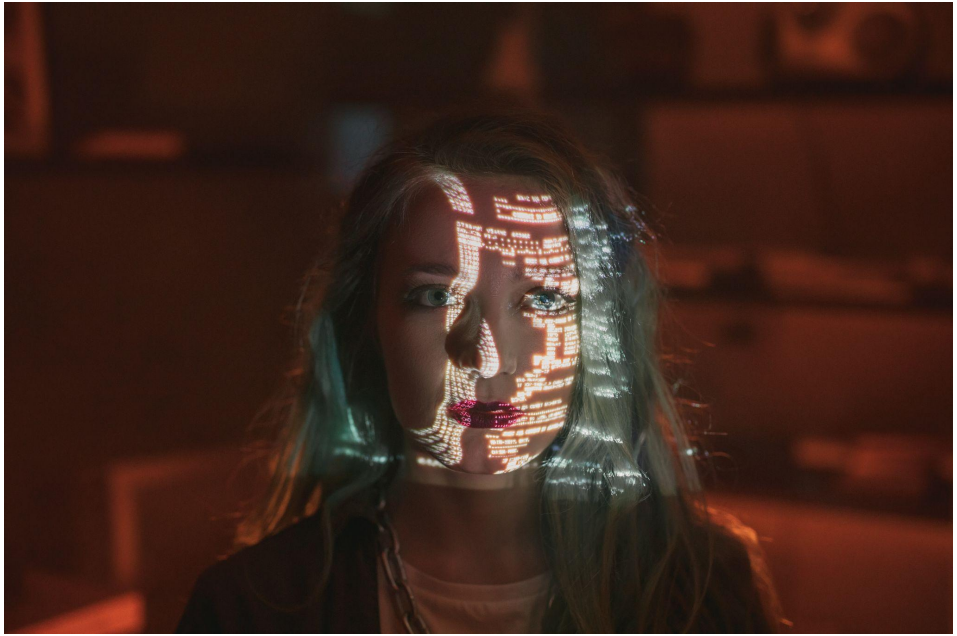
What are packages?

- bundles of tools and functions that others have developed to be used in R
- grouped to specific types of functions, analyses or datasets
- you can also create your own packages for specific tasks!

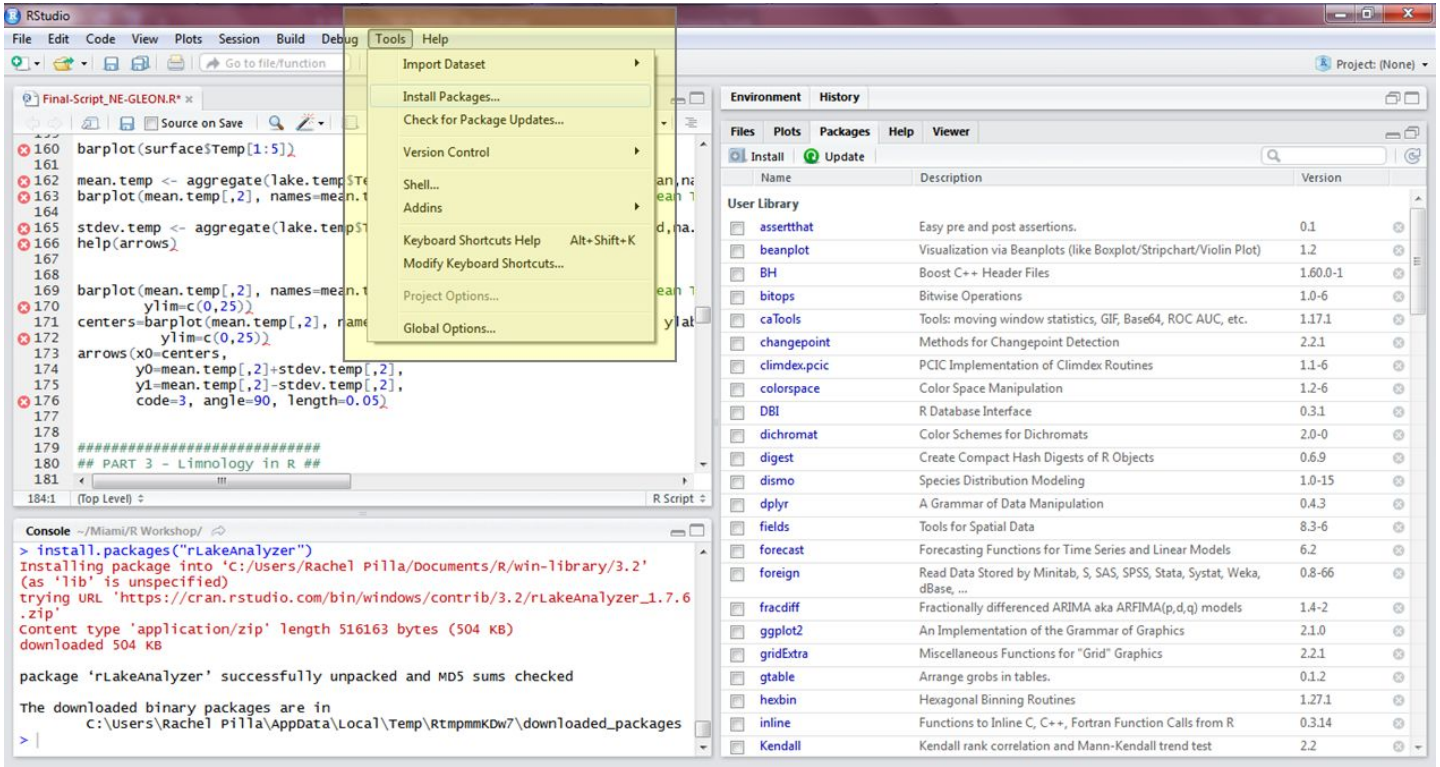


What are packages?

- bundles of tools and functions that others have developed to be used in R
- grouped to specific types of functions, analyses or datasets
- you can also create your own packages for specific tasks!
- currently, above 16,000 packages are available



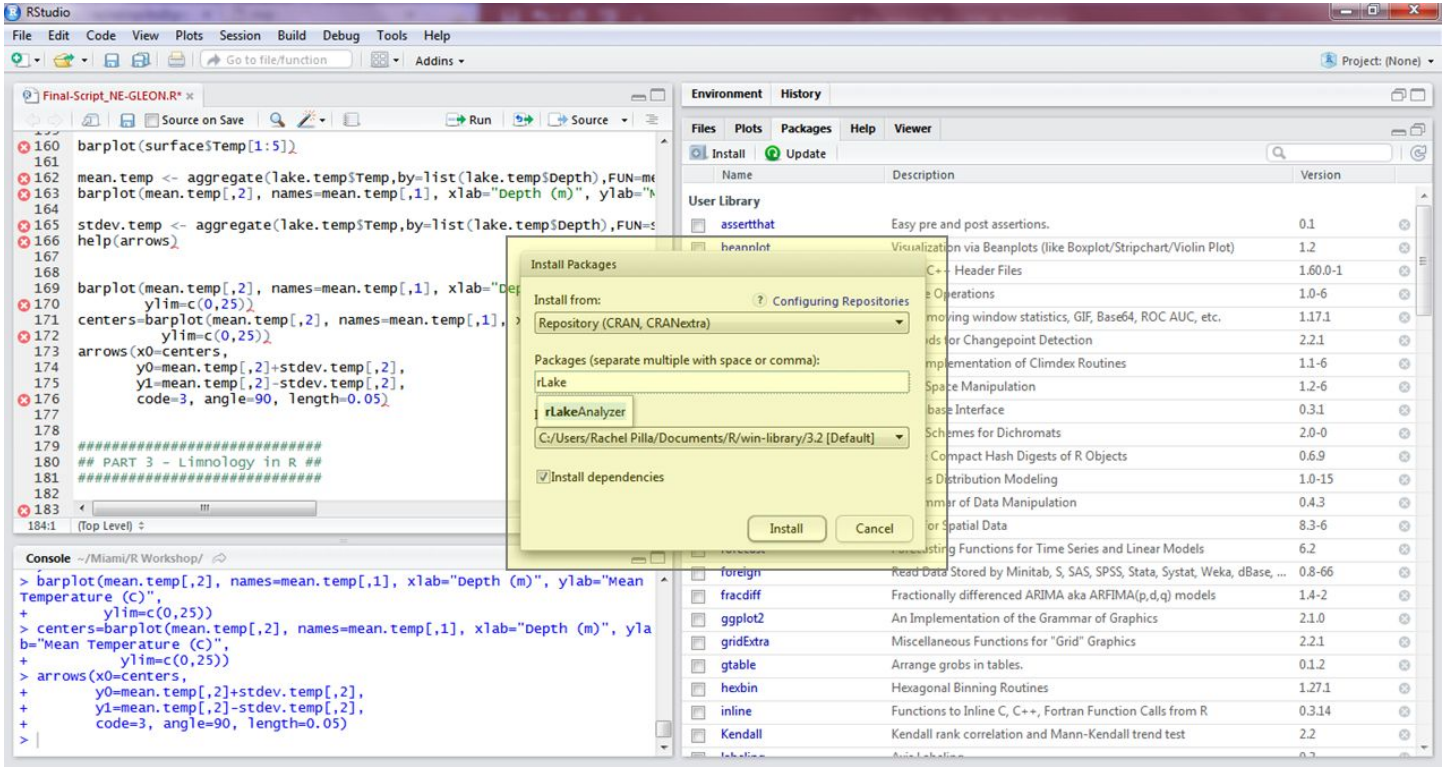
Install one package



we need:

tidyverse

Install one package



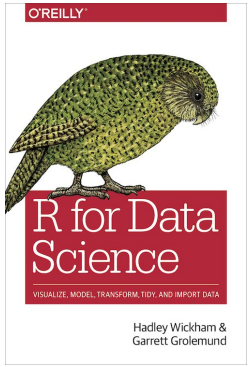
we need:

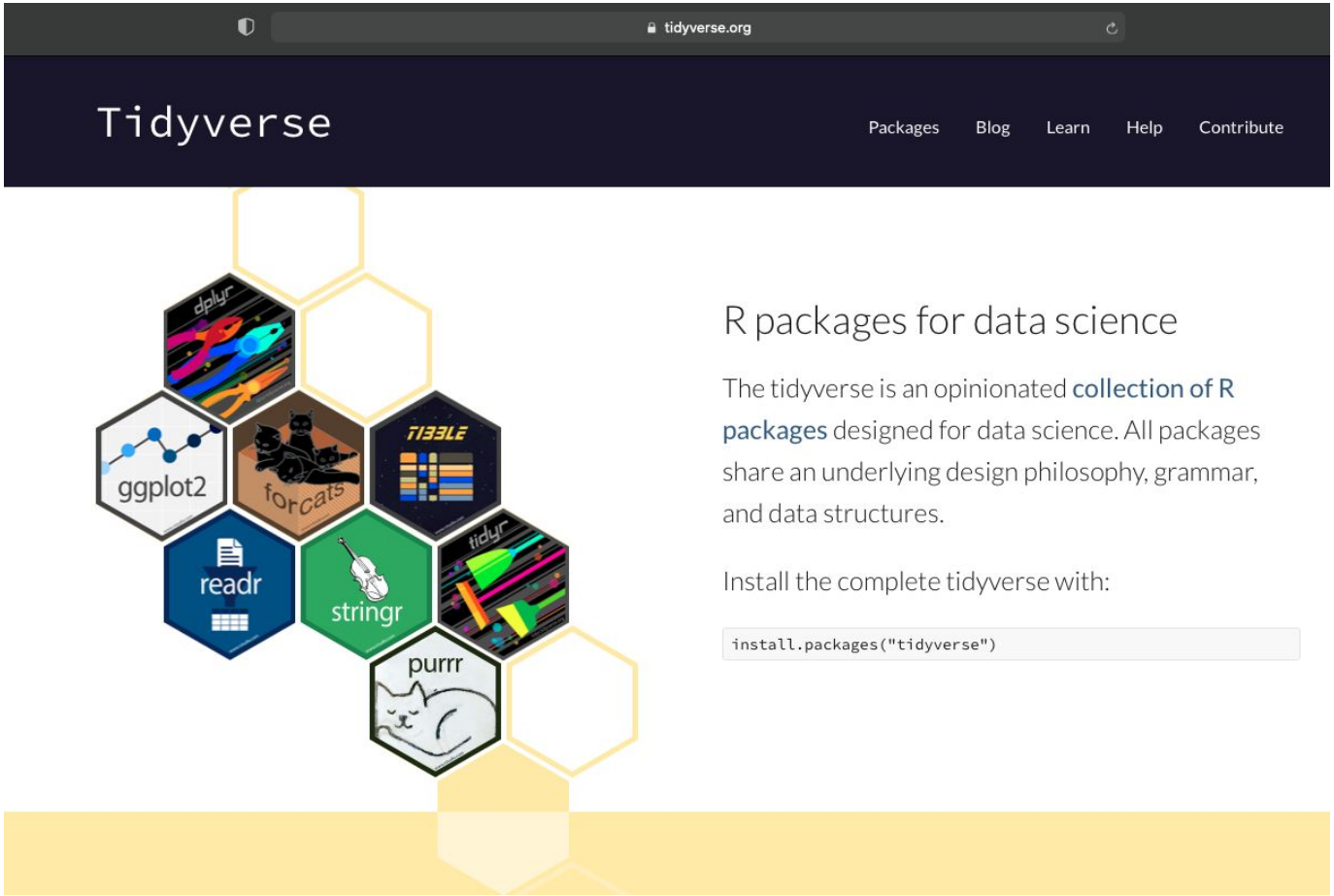
tidyverse

Tidyverse

R for Data Science (Wickham & Grolemund):

https://r4ds.had.co.nz

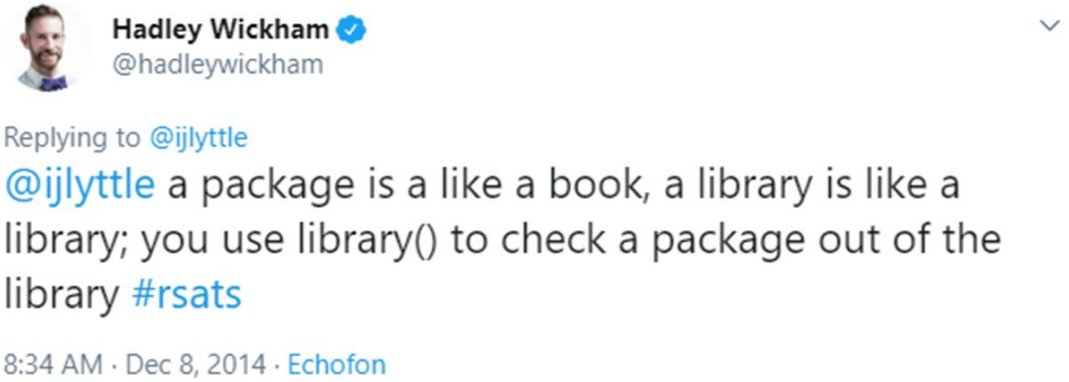




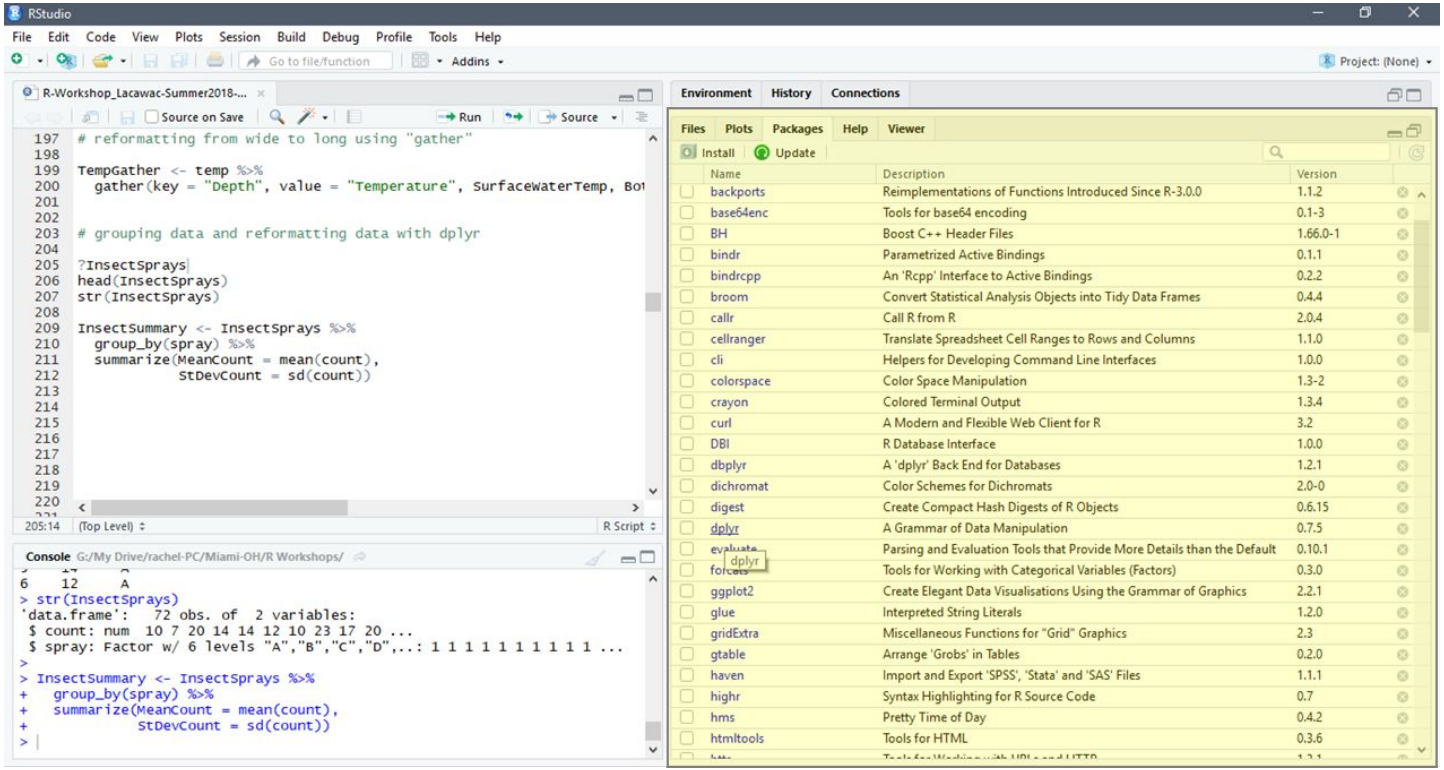
Installing a package

- packages only need to be installed ONCE
- but, if you want to use it you'll need to load it:

```
library(package name)
```

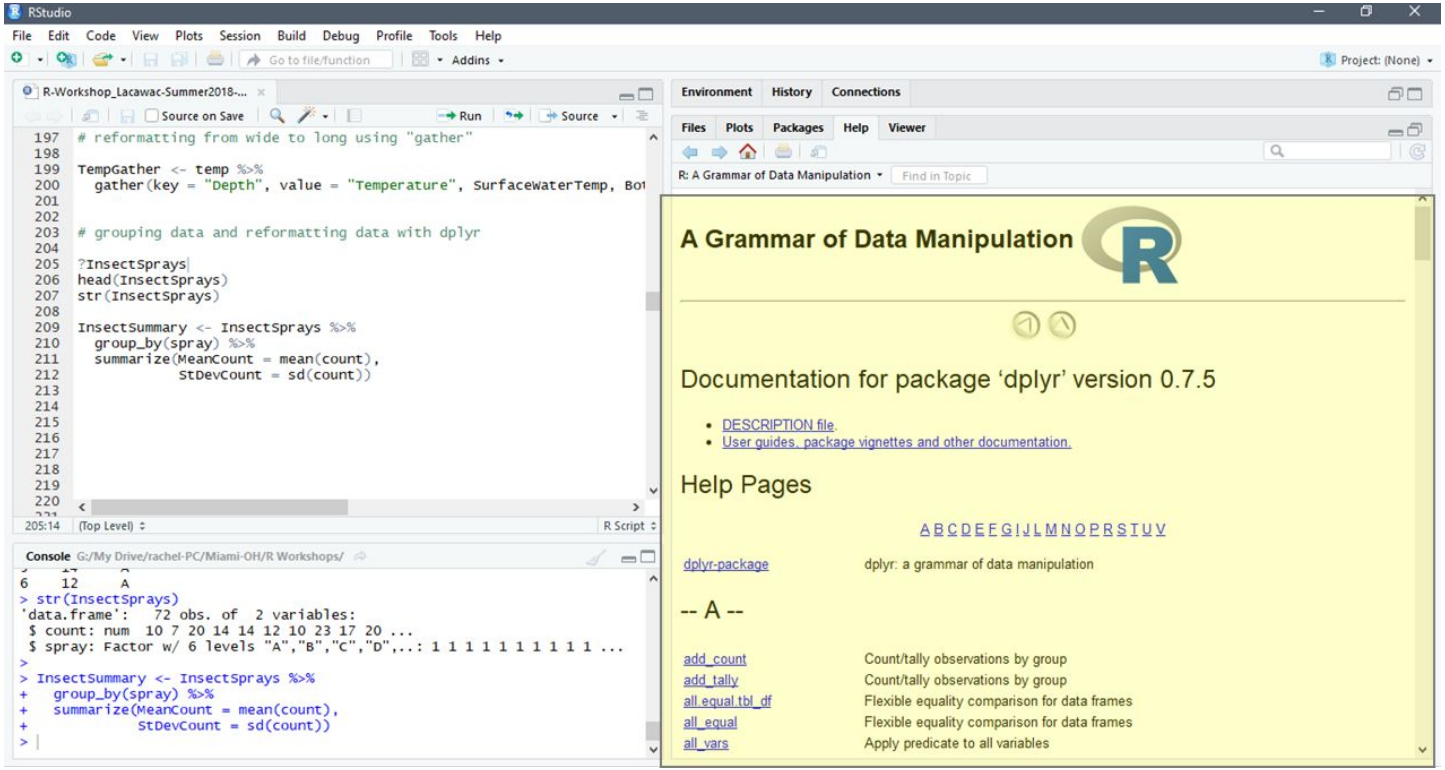


Package information



under packages, click on **dplyr**

Package information



this lists all available functions

click on any function to see the help file

Coding in R

- write commands in the script (upper left panel)
  - save it, edit it, revisit it later, etc.

Coding in R

- write commands in the script (upper left panel)
  - save it, edit it, revisit it later, etc.
- code NOT automatically run when you hit Enter

Coding in R

- write commands in the script (upper left panel)
  - save it, edit it, revisit it later, etc.
- code NOT automatically run when you hit Enter
- to run it:
  - “Run” button in upper right corner
  - “CTRL + Enter” (Windows)
  - “Command + Enter” (Mac)

Coding in R

- add comments using #
  - additional information
  - will be ignored

Coding in R

- add comments using #
  - additional information
  - will be ignored
- R is case sensitive
  - “Mean” ≠ “mean”

Coding in R

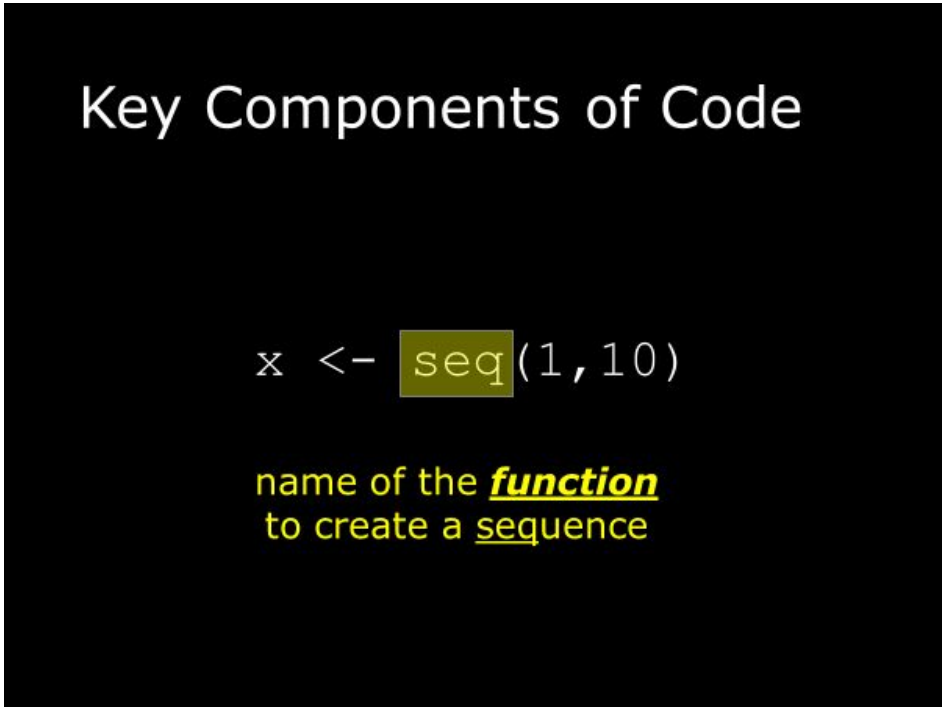
- add comments using #
  - additional information
  - will be ignored
- R is case sensitive
  - “Mean” ≠ “mean”
- R doesn’t care about spaces and tabs

Coding in R

- add comments using #
  - additional information
  - will be ignored
- R is case sensitive
  - “Mean” ≠ “mean”
- R doesn’t care about spaces and tabs
- need to close all parentheses and quotations

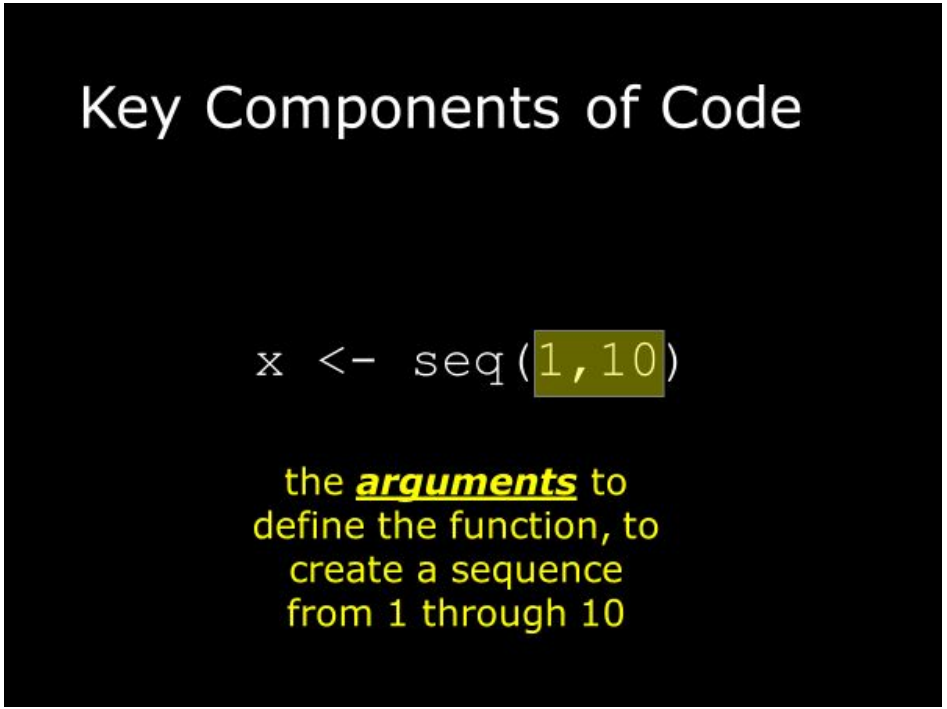
Key components

- **functions**: allow you to manipulate data, apply calculations, run statistical analysis, much more!



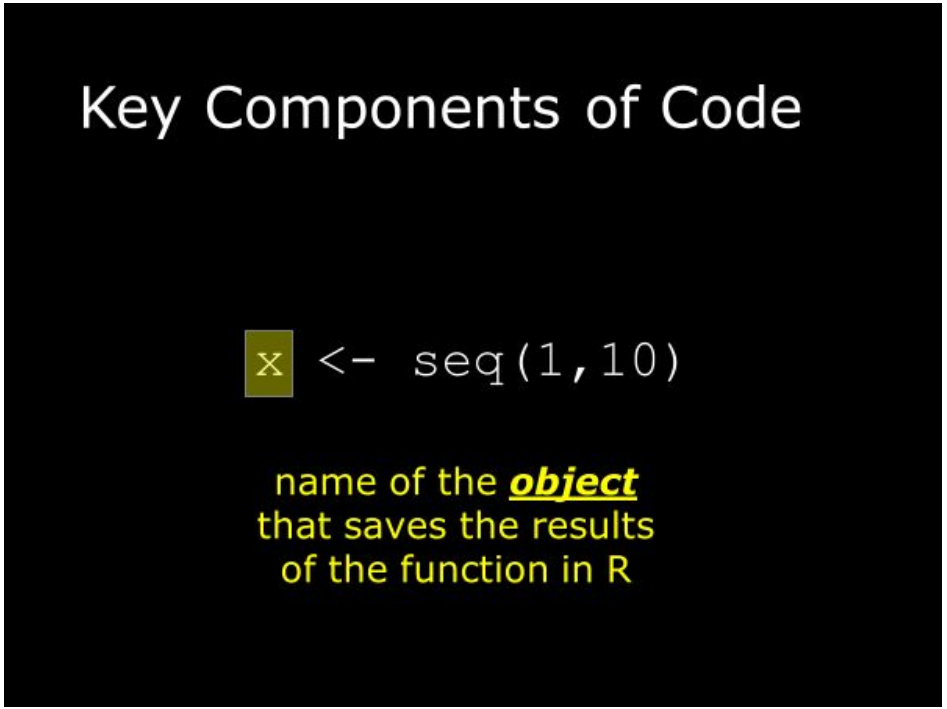
Key components

- **functions**: allow you to manipulate data, apply calculations, run statistical analysis, much more!
- **arguments**: defining information for functions, “customize” it



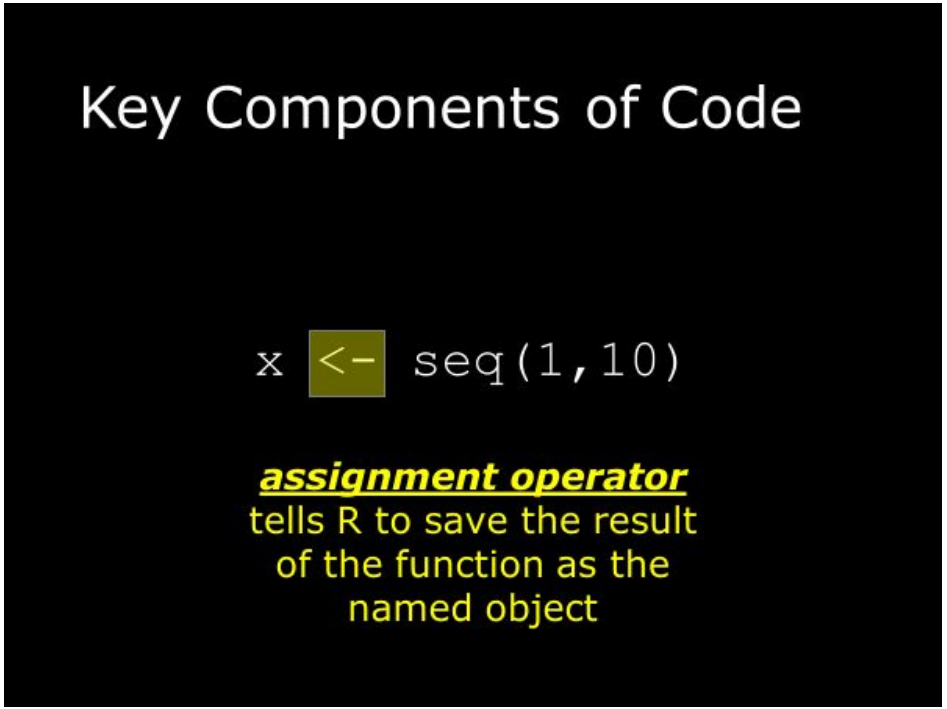
Key components

- **functions**: allow you to manipulate data, apply calculations, run statistical analysis, much more!
- **arguments**: defining information for functions, “customize” it
- **objects**: pieces of data saved in R, can be called up, reused and manipulated



Key components

- **functions**: allow you to manipulate data, apply calculations, run statistical analysis, much more!
- **arguments**: defining information for functions, “customize” it
- **objects**: pieces of data saved in R, can be called up, reused and manipulated



Types of data

- **objects (data types):**
  - numeric (2.334, 3.14159)

Types of data

- **objects (data types):**
  - numeric (2.334, 3.14159)
  - integer (2, 2405, 54)

Types of data

- **objects (data types):**
  - numeric (2.334, 3.14159)
  - integer (2, 2405, 54)
  - characters (“hello world”)

Types of data

- **objects (data types):**
  - numeric (2.334, 3.14159)
  - integer (2, 2405, 54)
  - characters (“hello world”)
  - logical (TRUE/FALSE)

Types of data

- **objects (data types):**
  - numeric (2.334, 3.14159)
  - integer (2, 2405, 54)
  - characters (“hello world”)
  - logical (TRUE/FALSE)
  - complex (1+4*i*)

Types of data

- **objects (data types):**
  - numeric (2.334, 3.14159)
  - integer (2, 2405, 54)
  - characters (“hello world”)
  - logical (TRUE/FALSE)
  - complex (1+4*i*)

- **data structures:**
  - vector (1-D object with same data type)

```
> c(3, 4, 5, 1, 2, 3)
[1] 3 4 5 1 2 3
```

Types of data

- **objects (data types):**
  - numeric (2.334, 3.14159)
  - integer (2, 2405, 54)
  - characters (“hello world”)
  - logical (TRUE/FALSE)
  - complex (1+4*i*)

- **data structures:**
  - vector (1-D object with same data type)
  - matrix (2-D object with same data type

```
> matrix(data = c(3, 4, 5, 1, 2,
3), nrow = 3, ncol = 2)
     [,1] [,2]
[1,]    3    1
[2,]    4    2
[3,]    5    3
```

Types of data

- **objects (data types):**
  - numeric (2.334, 3.14159)
  - integer (2, 2405, 54)
  - characters (“hello world”)
  - logical (TRUE/FALSE)
  - complex (1+4*i*)

- **data structures:**
  - vector (1-D object with same data type)
  - matrix (2-D object with same data type
  - data frame (2-D object with different data types per column if needed, very useful!)

```
> data.frame('Numbers' = c(3, 4, 5, 1, 2, 3), "Strings" =
c("Hi", "I", "am", "collecting", "all", "characters"))
  Numbers    Strings
1       3         Hi
2       4          I
3       5         am
4       1 collecting
5       2        all
6       3 characters
```

Types of brackets

round brackets:

- are used for functions
- include arguments

```
seq(...)
```

Types of brackets

round brackets:

- are used for functions
- include arguments

```
seq(...)
```

squared brackets:

- are used to subset data

```
> x <- data.frame("name" = c("Thing 1",
"Thing 2"), "Chaos" = c(10, 9))
> x[1, 1]
[1] "Thing 1"
> x[2, 2]
[1] 9
> x$name
[1] "Thing 1" "Thing 2"
> x$name[2]
[1] "Thing 2"
```

Types of brackets

round brackets:

- are used for functions
- include arguments

```
seq(...)
```

squared brackets:

- are used to subset data

data frame column 1 stores strings

```
> x <- data.frame("name" = c("Thing 1",
"Thing 2"), "Chaos" = c(10, 9))
> x[1, 1]
[1] "Thing 1"
> x[2, 2]
[1] 9
> x$name
[1] "Thing 1" "Thing 2"
> x$name[2]
[1] "Thing 2"
```

data frame column 2 stores doubles

Types of brackets

round brackets:

- are used for functions
- include arguments

```
seq(...)
```

squared brackets:

- are used to subset data

data frame column 1 stores strings

data frame column 2 stores doubles

```
> x <- data.frame("name" = c("Thing 1",
"Thing 2"), "Chaos" = c(10, 9))
> x[1, 1]
[1] "Thing 1"
> x[2, 2]
[1] 9
> x$name
[1] "Thing 1" "Thing 2"
> x$name[2]
[1] "Thing 2"
```

row 1

column 1

Types of brackets

round brackets:

- are used for functions
- include arguments

```
seq(...)
```

squared brackets:

- are used to subset data

```
> x <- data.frame("name" = c("Thing 1",
"Thing 2"), "Chaos" = c(10, 9))
> x[1, 1]
[1] "Thing 1"
> x[2, 2]
[1] 9
> x$name
[1] "Thing 1" "Thing 2"
> x$name[2]
[1] "Thing 2"
```

data frame column 1 stores strings

data frame column 2 stores doubles

row 1

column 1

dollar sign for indexing columns

The wonder of pipes

pipes %>% are powerful tools for making data manipulation clearer

- originate from `magrittr` package (included in tidyverse)

```
> x
     name Chaos
1 Thing 1    10
2 Thing 2     9
> filter(x, name == "Thing 1")
     name Chaos
1 Thing 1    10
> x %>%
    filter(name == "Thing 1")
     name Chaos
1 Thing 1    10
```

The wonder of pipes

pipes %>% are powerful tools for making data manipulation clearer

- originate from `magrittr` package (included in tidyverse)

```
> x
     name Chaos
1 Thing 1    10
2 Thing 2     9
> filter(x, name == "Thing 1")
     name Chaos
1 Thing 1    10
> x %>%
    filter(name == "Thing 1")
     name Chaos
1 Thing 1    10
```

same code but easier to read

Live Coding time!

Bird banding data provided by Dr. Dave Russell at Miami University in collaboration with Hueston Woods State Park

RStudio®