1. R essentials

Principles of Data Science with R

Dr. Uma Ravat

PSTAT 10

Announcement: Regarding Worksheet00

- Make sure to look at the **.html** output of your `Knit` command :
  - Include narrative in your own words where asked to provide narrative
  - Exercise 5: you should print out the code chunk and it's output(the plot) for pressure2 code chunk.
  - Reflect on your work in the worksheet and include learning gains.
- For **Worksheet00 only**, I have extended the deadline till this Friday Sept 30 at 8am.
  - Update your Workheet00 if necessary by the extended deadline.
  - All other worksheets will be due 30 minutes after your lab section ends.

Announcement:

Post on **Ed** https://edstem.org/ questions as lecture is progressing

- Tag it with correct Week and Lecture
- Today is Week 1, Lecture 1
- A TA will monitor during lecture and alert me to answer important questions during lecture
  - You can also answer questions from peers
  - Or alert me to an important question

See: https://tinyurl.com/AskingLectureQuestionOnEd

Lecture 0 Summary

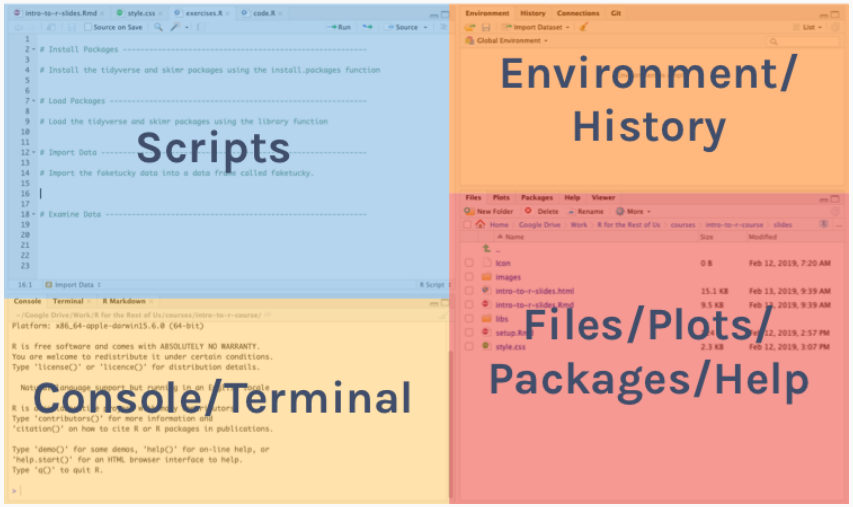
- Core elements of Data Science project life-cylcle
  - Programming
  - Statistics and Probability
  - Databases
- Accessing Rstudio instance for the course
- created a Data Science project report for UN votes.
- Course overview and Brief Syllabus walk through
- Rmarkdown essentials.(Complete it in section 1)

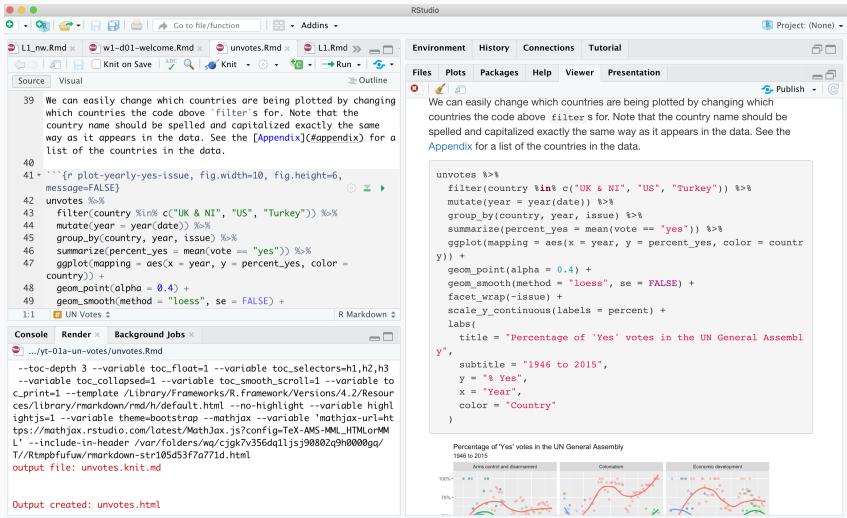
Post Lecture 1 to-do for you

- Read syllabus carefully
- Note down important dates, midterm and final exams
- Get familiar with Course site on Canvas
- Register for Ed with your UCSB email
- Go to Section 1 on Thursday/Monday
- Try and finish Worksheet 1 during section and submit within 30 minutes after your section
- Visit PSTAT 10 Welcome Clinic on Friday (check hours and location on Canvas)
  - Get help with lecture material if you struggled in lecture today.
  - Practice will make it perfect for you!

Have a great start to the quarter! See you Tuesday!

1.1 Last time: Rstudio and Rmd





Today: Get started with R : Console, Environment panes, R essentials

What the heck *is* R?

- R is an open-source statistical **programming language**
- R is also an environment for statistical computing and **graphics**
- It's easily extensible with *packages* (more on this later)
- R is based on the S language, which was developed by Bell laboratories in the 90's
- Home page: http://www.r-project.org

R and Rtudio

R: Engine	RStudio: Dashboard
	

- R is a programming language.
- RStudio is a convenient interface for R called an **IDE** (integrated development environment), e.g. *"I write R code in the RStudio IDE"*

R packages

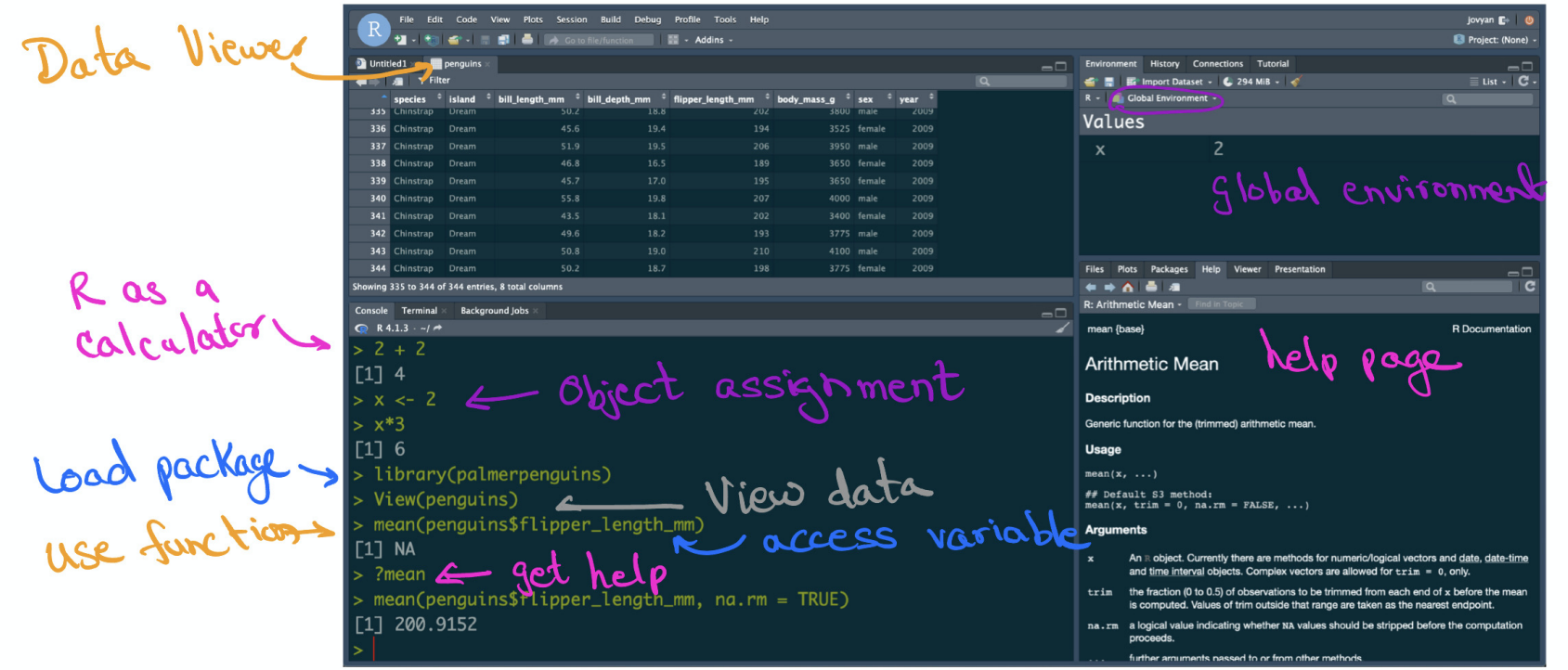
- **Packages** are the fundamental units of reproducible R code. They include reusable R functions, the documentation that describes how to use them, and sample data
- There are over 18,000 R packages available on **CRAN** (the Comprehensive R Archive Network)1

1 Community contributed packages are stored at CRAN Comprehensive R Archive Network

Your Turn 1

Go to RStudio's **console pane:** http://bit.ly/f2210 and do the following at **the command prompt**

1. **R as a calculator:** use any mathematical operators (+, -, /, and *) to create an expression and make sure it works as expected. *What is `2+2`, `100*3`, `100/10`, `sqrt(25)`?*
2. **Object Assignment:** Assign the value of your mathematical operation to the variable `y`. Note the change in the environment pane.
3. **Load** the `palmerpenguins` **package** so that we have access to all functions and data in this package.
4. **View** the `penguins` **dataset**
5. Take a look at the `flipper_length` variable
6. **Find the average** `flipper_length`
7. **Get help** on the mean **function**



Review Your Turn 1

Use Rmd for reproducible code!

Easier to reproduce this code from an .Rmd document than the console

Always remember !

1. to copy over files to **your workingfiles directory, in sub directories where you can locate it**
2. Do not work in **content** directory.

Now: Copy over `L01 -> YT01` to `your_workingfiles -> Lecture01->YT01`

Forgot?: Take another look here

OYO: On your own! Check and practice by completing `your_workingfiles-> Lecture01 -> YT01 -> R-essentials.Rmd section # 1. Console vs .Rmd` later

Recap: A short list of R essentials

To understand computations in R, two slogans are helpful:
Everything that exists is an object.
Everything that happens is a function call.
— John Chambers

Even a **function** is an **object**

R essentials : functions

- **Functions** are (most often) verbs, followed by what they will be applied to in parentheses:

```
do_this(to_this)
```

Here `do_this` is the function and `to_this` is the **argument** to the function

```
do_that(to_this, to_that, with_those)
```

Here `do_that` is the function and `to_this, to_that, with_those` are the three **arguments** to the `do_that` function

Functions in R are either

- built-in (free for you to use!)
- user-defined (you need to code them up.. you'll do this later in the course.)

R essentials : working with packages (aka libraries)

- If the package has never been installed on your computer before, **install** it with the `install.packages` function
  - done once on the computer
  - you'll see it in the Packages pane/tab, **unchecked**
- **load** a package with the `library` function
  - once per session when you need to use the functions or data from the package.
  - you'll see it in the Packages pane/tab, **checked**
  - some built-in packages are loaded and ready to use when you start an R session.

```
install.packages("package_name") # don't forget quotes
library(package_name) # no need for quotes
```

The packages we will use in this class

- base R
- datasets
- graphics
- stats
- and a few others that are already loaded for you in our PSTAT10 Rstudio instance. *You may need to download these special packages if you are working on an Rstudio instance you downloaded on your personal machine*

OYO: On your own!

What command will load these packages into your `R` session?

Are these packages loaded already?

R essentials : Accessing variables

- Columns (variables) in data frames are accessed with $:

```
dataframe$var_name
```

R essentials : Help

- Object documentation(aka help files) can be accessed with ?

```
?function_name
```

```
?mean
```

FTC: For the curious: Check this stackover flow page for a write up of more ways to get help in R

R essentials : Assignment operator and comments

<-

is used to assign values to objects.

```
OBJECT <- VALUE
```

```
x <- 2  # note the change in the environment
```

"Say: Create an object x and assign it the value 2"

R essentials : Comments

- Comment your code early and often and appropriately.
- any text after a '#' will be ignored by `R`
- short comments can be written at the end of the line of code, with a double space, followed by #, then another space (then comments) (see previous slide)
- longer comments should be written on their own lines, and begin with #, followed by a space (then comments)

R essentials: Naming conventions for Objects in R

R is case sensitive

- Use lower case letters and numbers
- Use underscores (_) so called snake case to separate words within a name
- Use names that are concise and meaningful (this is not easy!)
- Generally, variable names should be nouns and function names should be verbs
- Must not start with a digit.
- Avoid names that start with a period. They have a special meaning in `R`
- Names are case-sensitive.
- Use names that convey information about the object. Descriptive names are best.

E.g. welcome_msg <- "Welcome to PSTAT10"

R essentials : More flexibility in printing objects

print() SYNTAX: print(x, . . .)

x is an OBJECT (the string to be printed)

. . . denotes additional arguments.

quote = FALSE do not include quotation marks

Examples of Print statement usage

Go to `your_workingdirectory -> Lecture01 -> YT01 -> R-essentials.Rmd section ## 4. R essentials : More flexibility in printing objects`

Caution! : Environments

The environment of your R Markdown document is separate from the Console!

Remember this, and expect it to bite you a few times as you begin learning to work with R Markdown!

Environments!

1. Clear your Environment

2. run the following in the console

```
x <- 2
x * 3
```

Observe the object `x` appear in the Global Environment

Then, add the following in an R chunk in your R Markdown document and **knit**

```
x * 3
```

What happens? Why the error when you knit?

Run the code chunk individually in the Rmd document.

What happens? Why don't you get an error for `x*3`?

R essentials: summary

- Console and Environment Panes, Command Prompt
- Objects
  - Variables: nouns
  - Functions: verbs
  - Naming conventions
- Packages: ready made functions and datasets from others
  - Install once
  - Load every time you need it
- Help : ?
- Assignment Operator : <-
  - printing objects
- Comments: #
  - **use them!** for yourself, the grader
- Coding style : **have one** and be consistent
  - See chapters 1-3 of the tidyverse style guide
- Environment

Next we will see...

- Data Structures
- Data Types
- Scalars
- Vectors

Some extras FTC!

Downloading R

Go to: https://cran.r-project.org/

Chose from:

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

Mac users choose Mac download

Windows users choose Windowns download

Downloading RStudio

1. Download and install R first.
2. Go to https://rstudio.com/products/rstudio/download/

Know Your RStudio Environment

There are a *lot* of keyboard shortcuts in RStudio. To view all the options, you must engage the keyboard shortcut that rules them all:

- Windows: `Alt` + `Shift` + `K`
- macOS: `Option` + `Shift` + `K`

Some favorites

1. Autocomplete command.
    - Both: `Tab`
2. Run the current line, selection from the editor.
    - Windows: `Ctrl` + `Enter`
    - macOS: `Cmd` + `Enter`
3. Run the current code chunk from the editor.
    - Windows: `Ctrl` + `Shift` + `Enter`
    - macOS: `Cmd` + `Shift` + `Enter`