1. R essentials

Transfer exploration seminar: Statistics and Data Science

Dr. Uma Ravat

PSTAT 194TR

Lecture 0 Summary

- Core elements of Data Science project life-cycle
  - Programming
  - Statistics
  - Probability
- Accessing Rstudio server instance for the course
- Created a Data Science project report for UN votes.
- Rmarkdown essentials.(Complete it but do not turn it in.)

Post Lecture 0 to-do for you

- Read syllabus carefully
- Note down important dates,
- Get familiar with Course site on Canvas
- Visit Office hours
  - Get help with lecture material if you struggled in lecture today.
  - Practice will make it perfect for you!

Have a great start to the course! See you next lecture!

Next we will see...

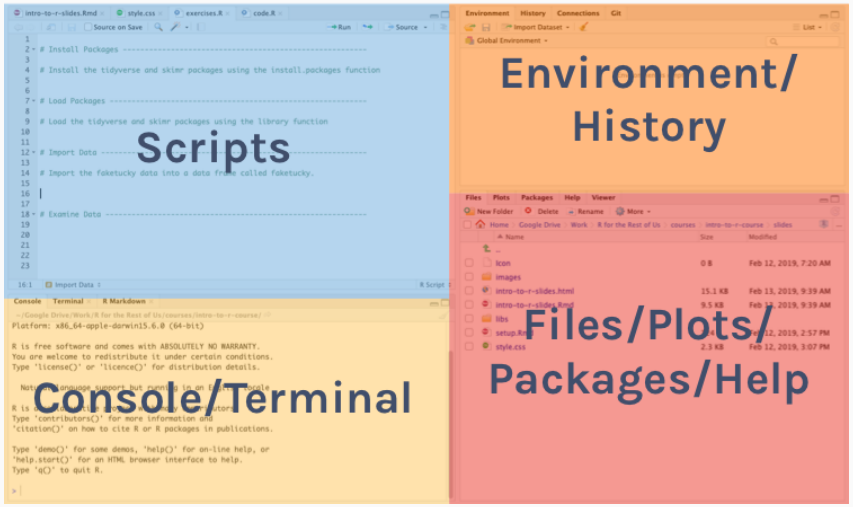
R essentials

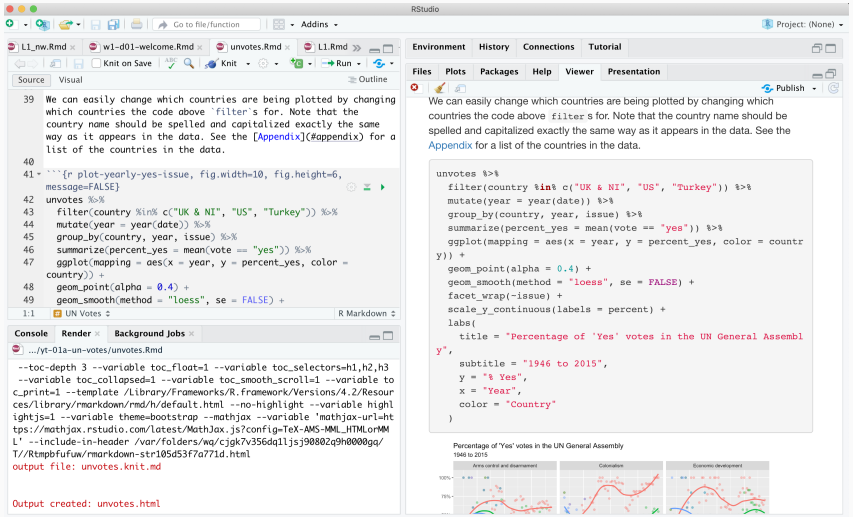
- Objects
- Data types, Data structures
- Variables
- Comments
- Functions
- Packages
- Help
- Style Guide
- File Organization

Work Along

Your Turn

1.1 Last time: Rstudio and Rmd





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

Wait, but what is data?

Data is all around us! Amount of data generated each day is incomprehensible!

Data comes in numerous types and formats that impact how we prepare, analyze it as well as the accuracy of insights and decisions that can be made using it.

Why R?

R is a programming language designed for statistical analysis

- open-source statistical **programming language**
- a great environment for statistical computing and **graphics**
- large and active community of developers and users
- 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

Why RStudio?

RStudio is an integrated development environment (IDE) designed to make your life easier.

- Organizes scripts, files, plots, code console, ...
- Highlights syntax
- Helpful interactive graphical interface
- Will make an efficient, reproducible workflow much easier
- **R Markdown integration**

R and Rstudio

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 an Rmarkdown document in the RStudio IDE"*
- just like *"I write an English essay in my notebook or in a Word document in MS Word software or .."*

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

Objects in R

To understand computations in R, two slogans are helpful:

Everything that exists is an object.

Everything that happens is a function call.

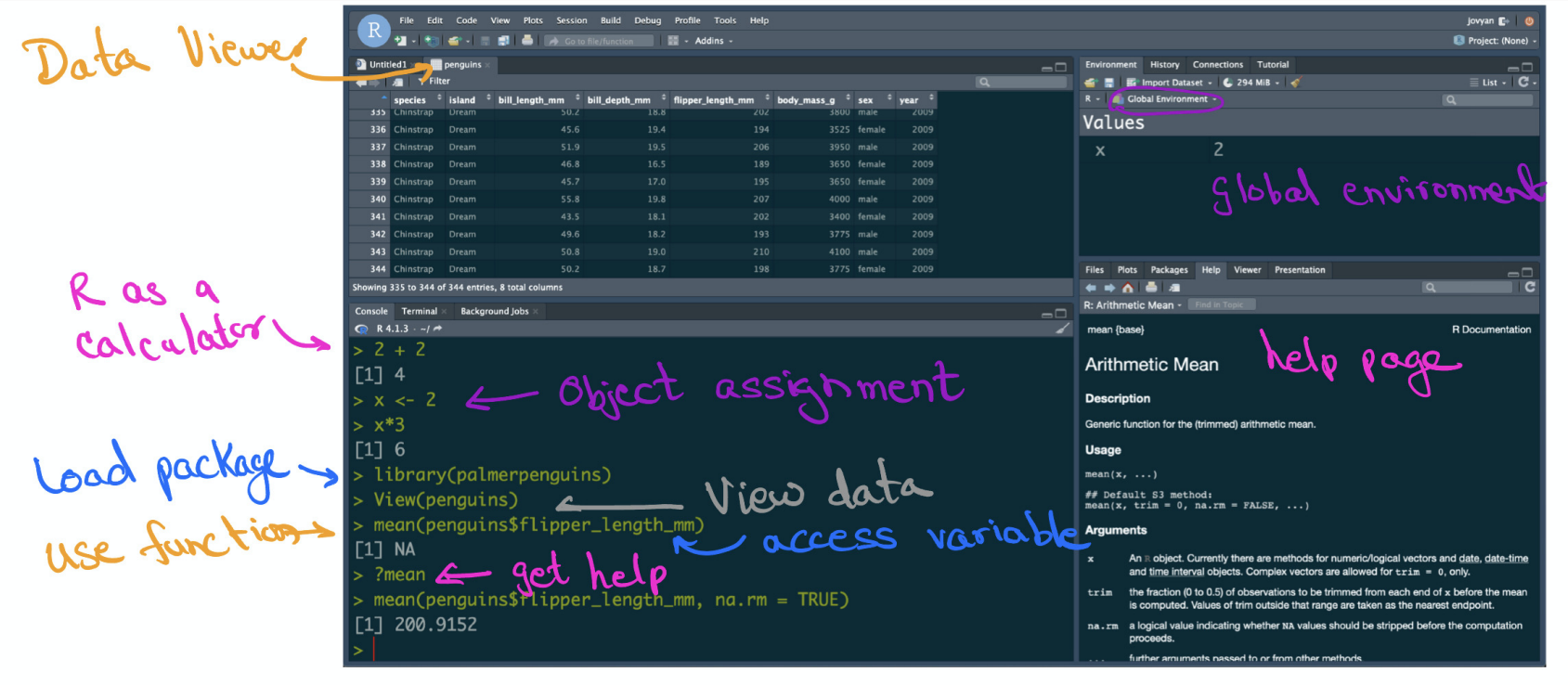
— John Chambers

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

Objects in R:

- data - numbers, letters, words and more
- functions
- packages

Working from the R console aka at command prompt.



Work Along Activity1: R essentials

Disclaimer! Many New Terms coming!

Don't worry about memorizing and remembering everything right now.

Instead, focus on recognizing the way `R` has things broken down

R essentials: 1. Types of Objects aka Data types

R allows us to create many objects: numbers, letters, words and more.

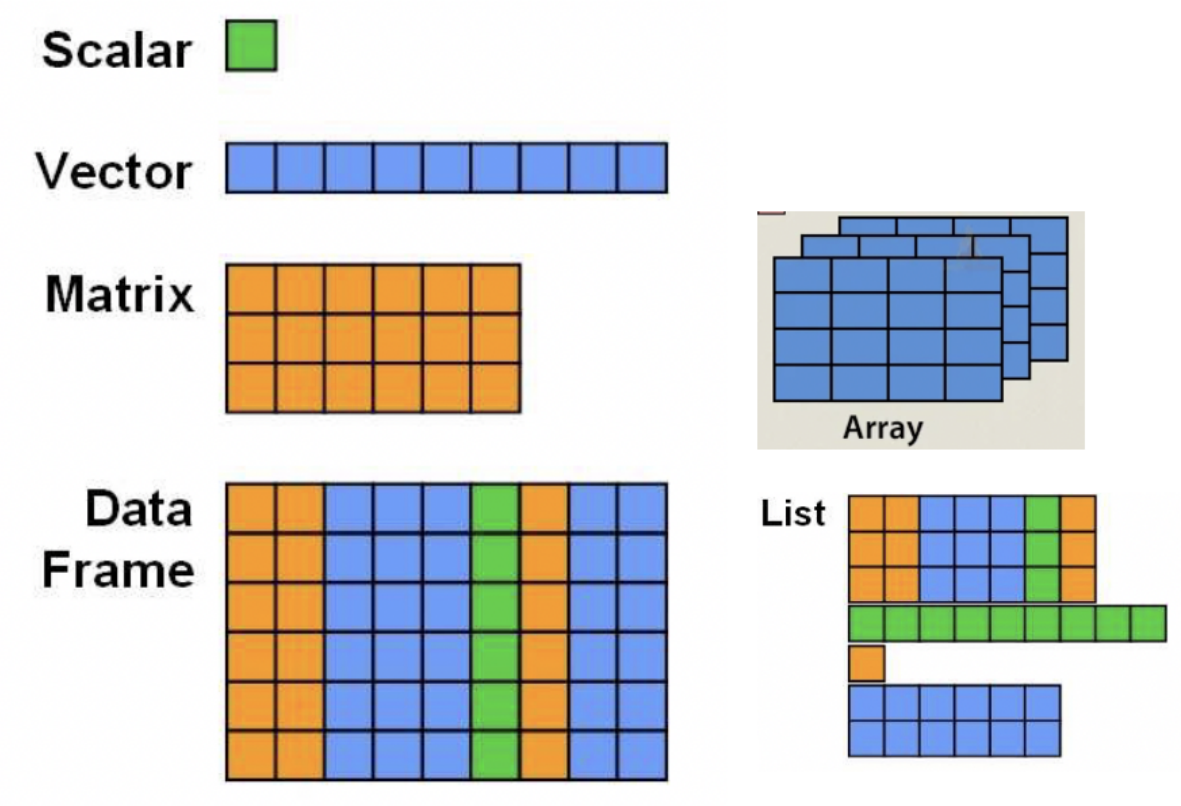
R objects are categorized into **types**, also known as data types.

A data type describes the type, or category, of the data and not the data itself.

- **integer**: integer (1, 2, 3)
- **double**: floating decimal (10.5, 55.0, 78.6)
- **numeric**: integer or double
- **character**: takes string values (e.g. a person's name, address) and must be surrounded by quotes (ex: "PSTAT 194", "SH 5500").
- **logical**: TRUE (1), FALSE (0)
- **factor**: categorical variable with different levels(e.g. eye color can be black, brown, blue, etc)

R essentials: 2. Data structures

Data structures are tools for holding multiple data types in one object.



Data structures : dimensionality and data type

Dimensions →

Homogeneity ↑

	1 D	2 D	Multi-D
Homogeneous	Vector	Matrix	Array
Heterogeneous	List	Dataframe	

We will primarily use dataframes here.

You'll cover other data structures in PSTAT 10.

R essentials : 3. Variables, comments

Variables are containers for storing data values.

```
x <- 2  # note the change in the environment
# <- is called an assignment operator
```

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

```
x # accessing the value stored in the variable/object x
```

```
## [1] 2
```

Variables store the value you assign to them until you assign them a new value. In above example, x will always have the value 2 until x is set again.

In a code chunk, any part of the code starting with '#' is a comment

```
# accessing variables(column) in a data frame using `$`
penguins$flipper_length_mm
```

R essentials : 4. Operators

```r
x <- 7 # recognize the assignment operator!
x + 3 # using + operator on a variable x
```

```
## [1] 10
```

```r
x
```

```
## [1] 7
```

```r
x == 7 # using comparision operator on a variable x
```

```
## [1] TRUE
```

```r
y <- 3
x + y # using + operator on two variables
```

```
## [1] 10
```

```r
## Predict and then try it: Does as.character(x) change the data type of `x`?
class(x)
```

```
## [1] "numeric"
```

```r
as.character(x) # built-in function, changing variable type or data type of x
```

```
## [1] "7"
```

```r
class(x)
```

R essentials: 5. Comments and Commenting Code

What is a comment?

- Computers completely ignore comments
- In a code chunk, any part of the code starting with '#' is a comment. *What is '#' used for outside a code chunk?*
- Comments do not impact functionality of your code at all

Why do them?

- Commenting a code allows you to write notes for readers of your code only
- Usually, that reader is you!
- Comment your code early and often and appropriately.

R essentials : 6. functions

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

```
do_this(to_this)
```

- `do_this` is the **name** of the function
- `to_this` is the **argument** to the function

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

- `do_that` is the name of the function with three **arguments**
- `to_this` is the first argument of `do_that` function
- `to_that` is the second argument of `do_that` function etc.

Functions in R are either

- built-in (free for you to use!)
- user-defined (you need to code them up. Do this later.)

R essentials : 7. working with packages (aka libraries)

- install package once on the computer
  - use `install.packages` function call
- load package each time you need to use package functionality
  - use `library` function call
  - 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
```

R essentials : 8. Help

- To get help, use ? followed by function (or object) name

```
?mean
```

- Check **this stackover flow page** for a write up of more ways to get help in R
- Chatgpt

R essentials : 9. Style Guide

```
readingthisisnoteasyisntit?
```

- Object names: Use CamelCase or snake_case
- Put a space after a comma, not before. `candy_num <- c(6, 9, 15)`
- Leave sapce before and after operators (e.g. `a<-2, b=3`) vs `a <- 2, b = 3`
- No space after function names (eg `mean (age)`) vs ‘`mean(age)`

R essentials: 10. File Organization Matters and Environment

Easier to start with best practice rather than fix things later!

1. You should have created the folder `netid_workingdirectory`
   - always copy files from the main content folder to your working directory before editing.
2. Within that folder, create the subfolders `lecture`, `homework`,`projects` etc
3. Within your `lecture` folder, create a subfolder `L00` and subfolders as necessary.
4. Put your files from `L00` and `L01` into correct subfolder.
5. Create sub-subfolders as necessary to keep things organized but not too diffiuclt to find.

Your Turn 01: Practice these R essentials

Go to `your_workingdirectory -> Lecture01 -> YT01 -> Activity1_R-essentials.Rmd section ## 3. R essentials :`

Summary of our Tools

R is a programming language used mainly for statistical computing.

R Markdown is a file format (`.Rmd`) that can handle `R` code as well as text

RStudio is an integrated development environment (IDE).

R essentials: summary

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

Some extras [OPTIONAL]

Debugging

Debugging is the process of getting rid of errors in your code.

3 types of errors:

1. Syntax Errors: code does not follow `R`'s rules
2. Runtime Errors: errors that occur during knitting
3. Logic Errors: code runs but produces unexpected results.

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`

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/

Next we will see...

Types of Statistical Data

- Numerical
- Categorical

EDA - Simple Techniques

- Data wrangling
- Quantative data summary
- Visual data summary

Disclaimer: Lot's of new terminology. Focus on how `R` handles things

Review after lecture

Maintain a glossary of functions used.