

# 1. R essentials

Principles of Data Science with R

---

Dr. Uma Ravat

PSTAT10, University of California at Santa Barbara

## Announcement: Regarding Homework 01

### 1. Introduction to R



## Summary:

- Core elements of Data Science project life-cycle
  - 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 0 to-do for you

- Read syllabus carefully
- Note down important dates, final exam
- Get familiar with Course site on Canvas
- Go to both Sections each week and ask questions when you are stuck.
- Complete Homework 1 and submit on time.
- 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 quarter! See you next lecture!

# **Announcement: Regarding Homework 01**

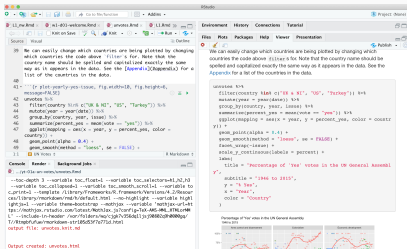
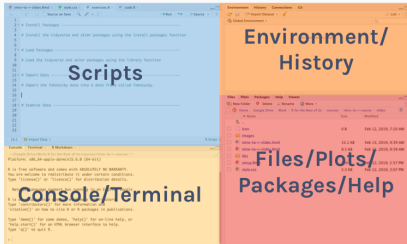
---

## Announcement: Regarding Homework 01

Make sure to look at the **.html** output of your Knit command :

- Make sure the text **Solution x:** appears at the top of each of your solutions
- Include narrative in your own words.
- Reflect on your work in the worksheet and include learning gains in the last exercise.
- Use office hours and HW clinics for help each day - If you don't use them they will go away!

# Last time: RStudio panes 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.



# 1. Introduction to R

---

# Why *is* R?



R is a programming language designed for statistical analysis.

- R is an open-source(free) 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>

# 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 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)<sup>1</sup>
- We will use various packages in this course such as `base`, `R,graphics`, `igraph`, `igraphdata` etc.

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 in R console.

1. Working in the `console` pane, 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. Load the `palmerpenguins` package so that we have access to all functions and data in this package.
3. View the `penguins` dataset
4. Take a look at the `flipper_length` variable
5. Find the average `flipper_length`
6. Get help on the `mean` function

**data viewer**

**environment**

**Arithmetic Mean**

**Object assignment**

**load package**

**view data**

**get help**

**access variable**

**use function**

Species | Island | bill\_length\_mm | bill\_depth\_mm | flipper\_length\_mm | body\_mass\_g

Species	Island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
Adelie	Torgersen	39.1	18.7	181	
Adelie	Torgersen	39.5	17.4	186	
Adelie	Torgersen	40.3	18.0	195	
Adelie	Torgersen	NA	NA	NA	
Adelie	Torgersen	36.7	19.3	193	
Adelie	Torgersen	39.3	20.6	190	
Adelie	Torgersen	38.9	17.8	181	
Adelie	Torgersen	39.2	19.6	195	
Adelie	Torgersen	34.1	18.1	193	
Adelie	Torgersen	42.0	20.2	190	
Adelie	Torgersen	37.8	17.1	186	

Showing 1 to 11 of 344 entries, 7 total columns

```

> 2 + 2
[1] 4
> x <- 2
> x + 3
[1] 5
> library(palmerpenguins)
> View(penguins)
> penguins$flipper_length_mm
[1] 181 186 195 NA 193 190 181 195 193 190 186 180 182 191

[327] 206 189 195 207 202 193 210 198
> mean(penguins$flipper_length_mm)
[1] NA
> ?mean
> mean(penguins$flipper_length_mm, na.rm = TRUE)
[1] 200.9152

```

Environment: Global Environment

Files | Plots | Packages | Help | Viewer

R Arithmetic Mean

mean (base)

R Documentation

Arithmetic Mean help

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

## Default S3 method:

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

x: An atomic (numeric) vector, an unsorted list, or a matrix (numeric) or data frame.

Examples

```

x <- c(0:10, 50)
m1 <- mean(x)
c(x, mean(x, trim = 0.10))

```

[Package base version 4.0.2 | index]

# The packages we will use in this class

- base R
- datasets
- graphics
- stats

What command will load these packages into your R session?

## 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

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

```
dataframe$var_name
```

# Objects and Functions in R

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

?mean

# Assignment operator and comments

<-

is used to assign values to objects.

```
OBJECT <- VALUE
```

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

*“Creates an object x and assign it the value 2”*

## Defining objects

```
xsq <- 2^3 # note the change in the environment
```

```
xsq
```

```
## [1] 8
```

**R is case sensitive.**

```
XSQ
```

```
## Error in eval(expr, envir, enclos): object 'XSQ' not found
```



## Your Turn: Tour of R essentials

Go to RStudio and do this now L01/YT01/Ressentials.Rmd

->

# Summary:

## Introduction to R

- Get comfortable with Rstudio
- R
- R essentials.