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

2. Introduction to R essentials

Summary: Introduction

- 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 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!

Next we will see...

R essentials

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

Work Along

Your Turn

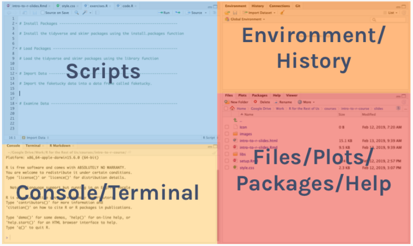
Announcement: Regarding Homework 01

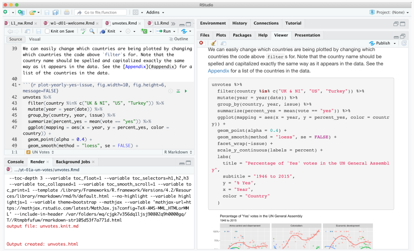
Announcement: Regarding Homework 01

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





~~~
We can easily change which countries are being plotted by changing
which countries the code above `filter`s for. Note that the
country name should be spelled and capitalized exactly the same
way as it appears in the data. See the [Appendix](#appendix) for a
list of the countries in the data.

```{r plot-yearly-yes-issue, fig.width=10, fig.height=6,
message=FALSE}
unvotes %>%
  filter(country %in% c("UK & NI", "US", "Turkey")) %>%
  mutate(year = year(date)) %>%
  group_by(country, year, issue) %>%
  summarize(percent_yes = mean(vote == "yes")) %>%
  ggplot(mapping = aes(x = year, y = percent_yes, color =
country)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
~~~

```
.../yt-01a-un-votes/unvotes.Rmd
 --toc-depth 3 --variable toc_float=1 --variable toc_selectors=h1,h2,h3
 --variable toc_collapsed=1 --variable toc_smooth_scroll=1 --variable to
c_print=1 --template /Library/Frameworks/R.framework/Versions/4.2/Resour
ces/library/rmarkdown/rmd/h/default.html --no-highlight --variable highl
ightjs=1 --variable theme=bootstrap --mathjax --variable 'mathjax-url=ht
tps://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMM
L' --include-in-header /var/folders/wq/cjgk7v356dq1ljsj90802q9h0000gq/
T//Rtmpbfufuw/rmarkdown-str105d53f7a771d.html
output file: unvotes.knit.md


Output created: unvotes.html
```

We can easily change which countries are being plotted by changing which countries the code above `filter`s for. Note that the country name should be spelled and capitalized exactly the same way as it appears in the data. See the Appendix for a list of the countries in the data.

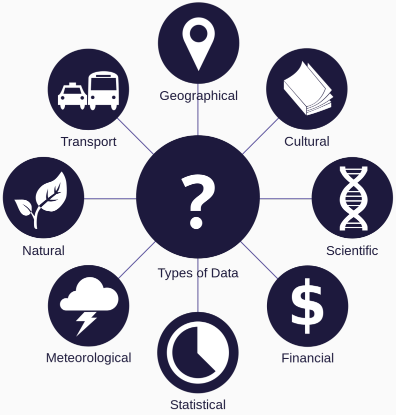
```
unvotes %>%
  filter(country %in% c("UK & NI", "US", "Turkey")) %>%
  mutate(year = year(date)) %>%
  group_by(country, year, issue) %>%
  summarize(percent_yes = mean(vote == "yes")) %>%
  ggplot(mapping = aes(x = year, y = percent_yes, color = countr
y)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
  facet_wrap(~issue) +
  scale_y_continuous(labels = percent) +
  labs(
    title = "Percentage of 'Yes' votes in the UN General Assembl
y",
    subtitle = "1946 to 2015",
    y = "% Yes",
    x = "Year",
    color = "Country"
  )
```

Percentage of 'Yes' votes in the UN General Assembly
1946 to 2015
Arms control and disarmament
Colonialism
Economic development
100%
75%

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

1. Introduction to R

Wait, but what is data and what has R got to do with it?



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.

What *is* R?

R is a programming language designed for statistical analysis

- open-source(free) 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.

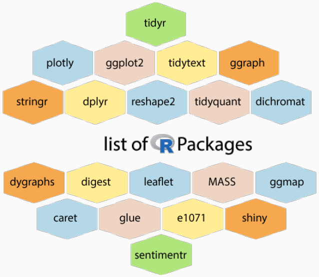
- Organizes scripts, files, plots, code console, . . .
- Highlights syntax
- Helpful interactive graphical interface
- Will make an efficient, reproducible workflow much easier ▪ R Markdown integration

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 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
- We will use various packages in this course such as `base R`,`graphics`, `datasets`, `stats` etc.

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*

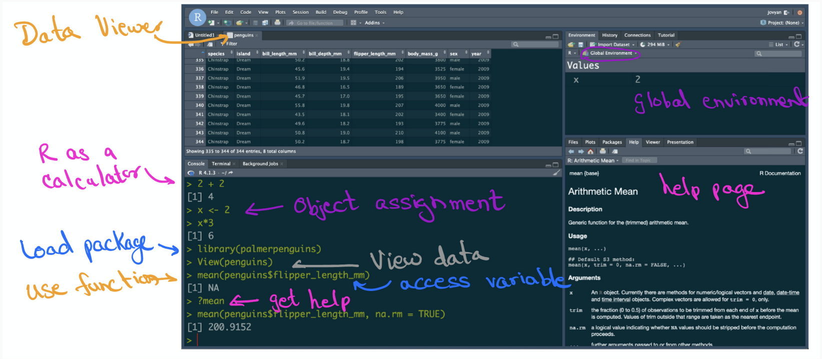
Even a *function* is an *object*.

Objects in R:

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

2. Introduction to R essentials

Working in R console aka at command prompt



	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
335	Chinstrap	Dream	50.2	18.8	202	3800	male	2009
336	Chinstrap	Dream	45.6	19.4	194	3525	female	2009
337	Chinstrap	Dream	51.9	19.5	206	3950	male	2009
338	Chinstrap	Dream	46.8	16.5	189	3650	female	2009
339	Chinstrap	Dream	45.7	17.0	195	3650	female	2009
340	Chinstrap	Dream	55.8	19.8	207	4000	male	2009
341	Chinstrap	Dream	43.5	18.1	202	3400	female	2009
342	Chinstrap	Dream	49.6	18.2	193	3775	male	2009
343	Chinstrap	Dream	50.8	19.0	210	4100	male	2009
344	Chinstrap	Dream	50.2	18.7	198	3775	female	2009

Showing 335 to 344 of 344 entries, 8 total columns

```
> 2 + 2
[1] 4
> x <- 2
> x*3
[1] 6
> library(palmerpenguins)
> View(penguins)
> mean(penguins$flipper_length_mm)
[1] NA
> ?mean
> mean(penguins$flipper_length_mm, na.rm = TRUE)
[1] 200.9152
>
```

Values

x	2

mean {base} R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

Arguments

`x`	An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for `trim = 0`, only.
`trim`	the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
`na.rm`	a logical value indicating whether `NA` values should be stripped before the computation proceeds.
`...`	further arguments passed to or from other methods

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.

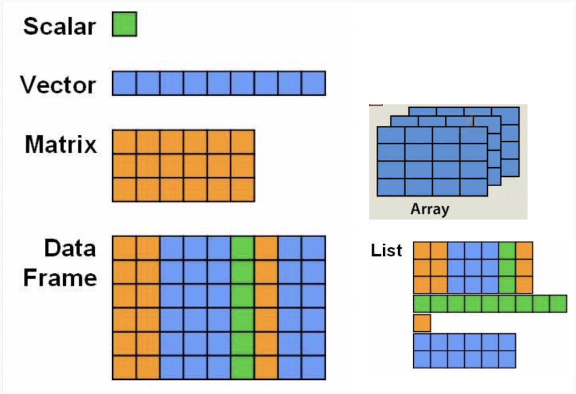
`R` objects are categorized into **types**, also known as data types.

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

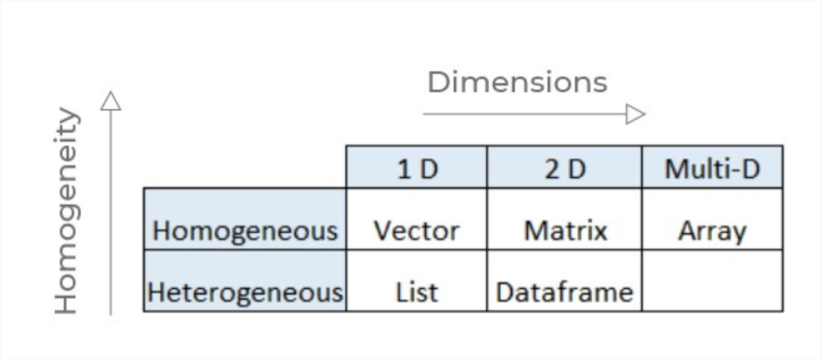
- **integer**: integer (1, 2, 3)
- **double**: floating decimal (10.5, 55.0, 78.6)
- **numeric**: integer or double
- **character**: takes string values (e.g. a person's name, address) and must be surrounded by quotes (ex: "PSTAT 194", "SH 5500").
- **logical**: TRUE (1), FALSE (0)
- **factor**: categorical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

R essentials: 2. Data structures

A data structure is a way of organizing data for efficient use.



Data structures : dimensionality and data type


Dimensions
Homogeneity

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

R essentials : 3. Variables, comments

Data is stored inside of named variables.

```
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)
```

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

```r
class(as.character(x))
```

```
## [1] "character"
```

R essentials: Rules for Naming Variables

- Letters, digits and dot (period) can all be used.
- Must not start with a digit or a digit followed by a digit.
- Must not start with a digit followed by a period.
- Names that start with a period are special and should be avoided.
- Names are case-sensitive.
- Descriptive names are best. E.g. temp, grade

```
1_grade #try it!
1.grade
grade.1
```

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.

```r
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 -> R-essentials.Rmd`

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

Post-Lecture To DO

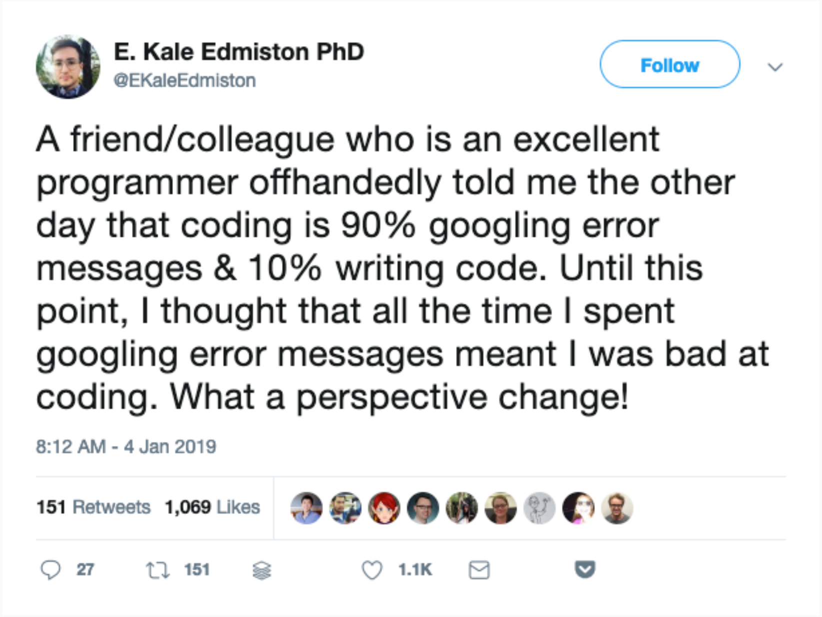
1. Review the lecture again
2. Write down a summary of today's lecture. Include all functions we went over and a short description of what each function does.

You will be asked to do this to your homework.

Next we will see. . .

- Data types (character, double, integer, logical)
- Data structures
    - Scalars
    - Vectors
    - more next time
- What are the different data types and data structures in `R`?
- What are differences in each of these
- How do I create these, access, update data within the various data structures?

Learning Programming is HARD!



E. Kale Edmiston PhD
@EKaleEdmiston

A friend/colleague who is an excellent programmer offhandedly told me the other day that coding is 90% googling error messages & 10% writing code. Until this point, I thought that all the time I spent googling error messages meant I was bad at coding. What a perspective change!

8:12 AM - 4 Jan 2019

151 Retweets 1,069 Likes

27 151 1.1K