GOVT PhD Math Camp Coding Lab Day 2

Theodore Landsman

Georgetown University

August 17 2021

Why Do You Need to Learn R?

- Why do you need to learn programming?

Why Do You Need to Learn R?

- Why do you need to learn programming?
  - Because it makes your life easier!

Why Do You Need to Learn R?

- Why do you need to learn programming?
  - Because it makes your life easier!
  - It is helpful in all the steps of data analysis.

Why Do You Need to Learn R?

- Why do you need to learn programming?
  - Because it makes your life easier!
  - It is helpful in all the steps of data analysis.
- Why R?

Why Do You Need to Learn R?

- Why do you need to learn programming?
  - Because it makes your life easier!
  - It is helpful in all the steps of data analysis.
- Why R?
  - Free

Why Do You Need to Learn R?

- Why do you need to learn programming?
  - Because it makes your life easier!
  - It is helpful in all the steps of data analysis.
- Why R?
  - Free
  - Relatively easy to learn compared to other languages

Why Do You Need to Learn R?

- Why do you need to learn programming?
    - Because it makes your life easier!
    - It is helpful in all the steps of data analysis.
- Why R?
    - Free
    - Relatively easy to learn compared to other languages
    - Many people are using it → there are lots of helpful resources out there!

Why Do You Need to Learn R?

- Why do you need to learn programming?
  - Because it makes your life easier!
  - It is helpful in all the steps of data analysis.
- Why R?
  - Free
  - Relatively easy to learn compared to other languages
  - Many people are using it → there are lots of helpful resources out there!
  - It's the standard software for quantitative political science (and other social science) research.

Basic Steps of Data Analysis

1. Specify research question

Basic Steps of Data Analysis

1. Specify research question

2. Collect data

Basic Steps of Data Analysis

1. Specify research question
2. Collect data
3. Preprocess data
    - The data you have can contain errors, be incomplete etc.
    - Clean and tidy the dataset so that it can be used for analysis

Basic Steps of Data Analysis

1. Specify research question
2. Collect data
3. Preprocess data
    - The data you have can contain errors, be incomplete etc.
    - Clean and tidy the dataset so that it can be used for analysis
4. Summarizing/Visualizing data

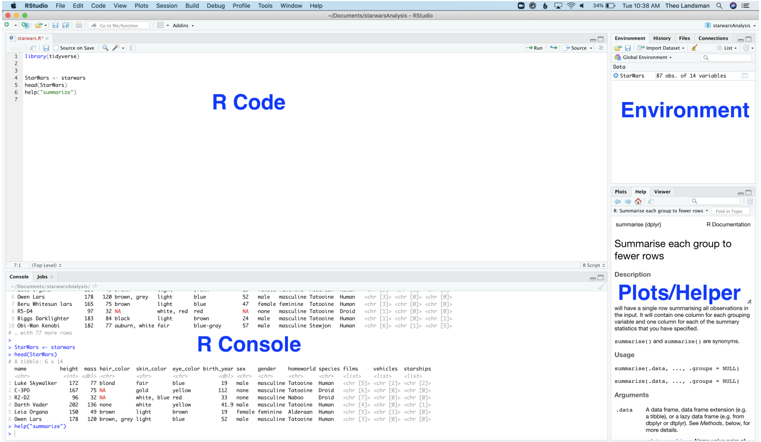
Basic Steps of Data Analysis

1. Specify research question
2. Collect data
3. Preprocess data
    - The data you have can contain errors, be incomplete etc.
    - Clean and tidy the dataset so that it can be used for analysis
4. Summarizing/Visualizing data
5. Statistical analysis

Basic Steps of Data Analysis

1. Specify research question
2. Collect data
3. Preprocess data
    - The data you have can contain errors, be incomplete etc.
    - Clean and tidy the dataset so that it can be used for analysis
4. Summarizing/Visualizing data
5. Statistical analysis
6. Summarizing/Visualizing the results

Understanding RStudio



```r
library(tidyverse)


StarWars <- starwars
head(StarWars)
help("summarize")
```

Data

StarWars 87 obs. of 14 variables

```
 6 Owen Lars             178   120 brown, grey   light       blue             52 male   masculine Tatooine  Human   <chr [3]> <chr [0]> <chr [0]>
 7 Beru Whitesun lars    165    75 brown         light       blue             47 female feminine  Tatooine  Human   <chr [3]> <chr [0]> <chr [0]>
 8 R5-D4                  97    32 NA            white, red  red              NA none   masculine Tatooine  Droid   <chr [1]> <chr [0]> <chr [0]>
 9 Biggs Darklighter     183    84 black         light       brown            24 male   masculine Tatooine  Human   <chr [1]> <chr [0]> <chr [1]>
10 Obi-Wan Kenobi        182    77 auburn, white fair        blue-gray        57 male   masculine Stewjon   Human   <chr [6]> <chr [1]> <chr [5]>
# … with 77 more rows
>
> StarWars <- starwars
> head(StarWars)
# A tibble: 6 x 14
  name           height  mass hair_color  skin_color  eye_color birth_year sex    gender    homeworld species films     vehicles  starships
  <chr>           <int> <dbl> <chr>       <chr>       <chr>          <dbl> <chr>  <chr>     <chr>     <chr>   <list>    <list>    <list>
1 Luke Skywalker    172    77 blond       fair        blue            19   male   masculine Tatooine  Human   <chr [5]> <chr [2]> <chr [2]>
2 C-3PO             167    75 NA          gold        yellow         112   none   masculine Tatooine  Droid   <chr [6]> <chr [0]> <chr [0]>
3 R2-D2              96    32 NA          white, blue red             33   none   masculine Naboo     Droid   <chr [7]> <chr [0]> <chr [0]>
4 Darth Vader       202   136 none        white       yellow          41.9 male   masculine Tatooine  Human   <chr [4]> <chr [0]> <chr [1]>
5 Leia Organa       150    49 brown       light       brown           19   female feminine  Alderaan  Human   <chr [5]> <chr [1]> <chr [0]>
6 Owen Lars         178   120 brown, grey light       blue            52   male   masculine Tatooine  Human   <chr [3]> <chr [0]> <chr [0]>
> help("summarize")
>
```

summarise {dplyr} R Documentation

Summarise each group to fewer rows

Description

…ut will have a single row summarising all observations in the input. It will contain one column for each grouping variable and one column for each of the summary statistics that you have specified.

`summarise()` and `summarize()` are synonyms.

Usage

```r
summarise(.data, ..., .groups = NULL)

summarize(.data, ..., .groups = NULL)
```

Arguments

`.data` A data frame, data frame extension (e.g. a tibble), or a lazy data frame (e.g. from dbplyr or dtplyr). See *Methods*, below, for more details.

Figure: 1: RStudio

Tips for Learning/Programming R

1. Keep Records

    - Don't directly type your code into the R console. Store your R code as .R, .Rmd, or .Rnw files.

    - Save your workspace to projects, go to Tools - Global Options and uncheck "Restore most recently opened project at startup" to prevent yourself from accidentally overwriting old projects.

Tips for Learning/Programming R

1. Keep Records
    - Don't directly type your code into the R console. Store your R code as .R, .Rmd, or .Rnw files.
    - Save your workspace to projects, go to Tools - Global Options and uncheck "Restore most recently opened project at startup" to prevent yourself from accidentally overwriting old projects.
2. Make your Code Easy to Understand
    - Add comments to your code: R ignores everything written after # in implementing the codes
    - Follow coding guidelines: e.g., Google's R Style Guide: https://google.github.io/styleguide/Rguide.html

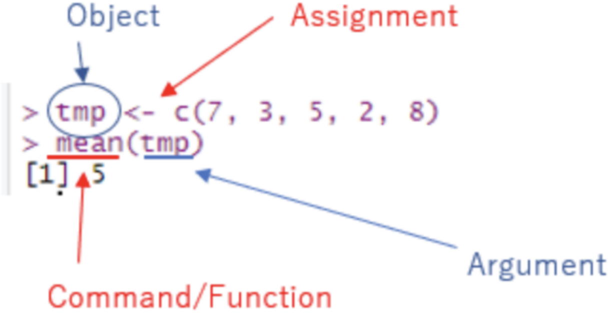
Tips for Learning/Programming R

1. Keep Records
    - Don't directly type your code into the R console. Store your R code as .R, .Rmd, or .Rnw files.
    - Save your workspace to projects, go to Tools - Global Options and uncheck "Restore most recently opened project at startup" to prevent yourself from accidentally overwriting old projects.
2. Make your Code Easy to Understand
    - Add comments to your code: R ignores everything written after # in implementing the codes
    - Follow coding guidelines: e.g., Google's R Style Guide: https://google.github.io/styleguide/Rguide.html
3. Practice Makes Perfect!

Glossary

- Object: Named “box”/“container” that we store values/data etc. in R
- Assignment: The process of creating/modifying objects
- Command/Function: We use a command/function to perform some tasks on an object/objects
- Argument: The definitions, directions, or objects that are passed to a command/function
- Package: A collection of functions, data, and documentations which is publicly shared to enhance the functionality of R.

Example



```
> tmp <- c(7, 3, 5, 2, 8)
> mean(tmp)
[1] 5
```

Figure: 2: Example

Today's Exercise

- Using R as a calculator.
- Object Classes/Types

Using R as a Calculator

```r
5 + 3 # summation
```

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

```r
4 - 7 # subtration
```

```
## [1] -3
```

```r
5 * 6 # product
```

```
## [1] 30
```

```r
7 / 3 # division
```

```
## [1] 2.333333
```

Using R as a Calculator Continued

```r
7 %% 3 # modular (residual)
```

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

```r
2 ^ 6 # power
```

```
## [1] 64
```

```r
(2 + 5) * 4 + 2 ^ 3 # note the order of calculation
```

```
## [1] 36
```

Objects

- In R, we store information as an object. Once we create an object, we can refer to it by its name.
- We assign values to an object using the assignment operator <-
    - Some R packages (tidyverse) also use the = symbol as an assignment operator, do not do this for now.
- Object class/type
    - what kind of information is stored in the object and how it is stored
    - typeof() or class() command to see the object type

Object Class/Type

- Data Types:
  - **Character**: character strings
  - **Numeric**: numbers
  - **Logical**: Boolean data (`TRUE/FALSE`)
  - *Factor*

Object Class/Type Continued

- Data Structures:
  - **Vector**: a single-dimension sequence of data of the same type
  - **Matrix**: a two-dimension sequence of data of the same type
  - **Data Frame:**: a two-dimension structure of data of varying data types
  - *List*
- In R, **Data Frames** and **Matrices** are composed of **Vectors** denoted by a $ in front of their names, thus for the data frame `starwars`, a dataset on star wars characters, the species is denoted by `starwars$species`.

Objects Example

```
# Numeric vector
num.1 <- c(4, -2, -7, 6, 8, 5, -3, 6, -4)
num.2 <- c(4, 6, 5, 2, 3)
# Character vector
program.lang <- c("R", "Python", "C", "Java")
# Logical vector
comparison <- (num.1 >= 5)
comparison
```

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

Objects Example Continued

```r
# Object class/type
class(num.1)
```

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

```r
class(num.2)
```

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

```r
class(program.lang)
```

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

```r
class(comparison)
```

```
## [1] "logical"
```

Logical Operators

Operator	Meaning
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
==	equal to
!=	not equal to
|	or
&	and
is.na()	TRUE if missing
!is.na()	FALSE if missing

Logical Operators Example

```r
7 < 5
```

```
## [1] FALSE
```

```r
(6 > 4) | (8 < 5)
```

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

```r
(7 > 3) & (9 <= 11)
```

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

Exercises!

- Perform the following calculations:
  - 0.0098 * 0.005
  - 9 * (log(3) - $\sqrt{2}) + 7$
  - Create a vector of Star Wars characters with missing homeworlds in the star wars dataset.
  - Assign a new vector to the `starwars` data frame `'density'` composed of character `height` divided by character `mass`.