

Computational Bootcamp 2: Intro to R

Henry Watson

Georgetown University

8/16/22

What We'll Be Covering Overall

- ① Software installation, file management
- ② Basics of R: writing code, creating objects, thinking in matrices
- ③ More R: working with datasets
- ④ Stata: pros & cons vs. R, working with datasets
- ⑤ LaTeX: producing documents with Markdown and Overleaf

What We'll Be Covering Today

① Basic Steps of Data Analysis

What We'll Be Covering Today

- 1 Basic Steps of Data Analysis
- 2 The R Studio Interface

What We'll Be Covering Today

- ① Basic Steps of Data Analysis
- ② The R Studio Interface
- ③ Working with Objects in R

What We'll Be Covering Today

- ① Basic Steps of Data Analysis
- ② The R Studio Interface
- ③ Working with Objects in R
- ④ Matrices in R

Why Learn Statistical Programming?

- Excel is powerful and useful...but won't get you all the way

Why Learn Statistical Programming?

- Excel is powerful and useful...but won't get you all the way
- Replicability & Collaboration

Why Learn Statistical Programming?

- Excel is powerful and useful...but won't get you all the way
- Replicability & Collaboration
- Traceable errors

Why Learn Statistical Programming?

- Excel is powerful and useful...but won't get you all the way
- Replicability & Collaboration
- Traceable errors
- Large datasets

Why Learn Statistical Programming?

- Excel is powerful and useful...but won't get you all the way
- Replicability & Collaboration
- Traceable errors
- Large datasets
- Advanced data analysis

Why Learn Statistical Programming?

- Excel is powerful and useful...but won't get you all the way
- Replicability & Collaboration
- Traceable errors
- Large datasets
- Advanced data analysis
- Data visualization

Advantages of R

- Free!
- Strong online community (lots of helpful resources)
- Standard software for quantitative political science research
- Integrates well with GIS
- Great for data visualization
- Great for simulation

Basic Steps of Data Analysis

- Specify research question

Basic Steps of Data Analysis

- Specify research question
- Collect data

Basic Steps of Data Analysis

- Specify research question
- Collect data
- Preprocess data

Basic Steps of Data Analysis

- Specify research question
- Collect data
- Preprocess data
 - The data you have can contain errors, be incomplete etc.

Basic Steps of Data Analysis

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

Basic Steps of Data Analysis

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

Basic Steps of Data Analysis

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

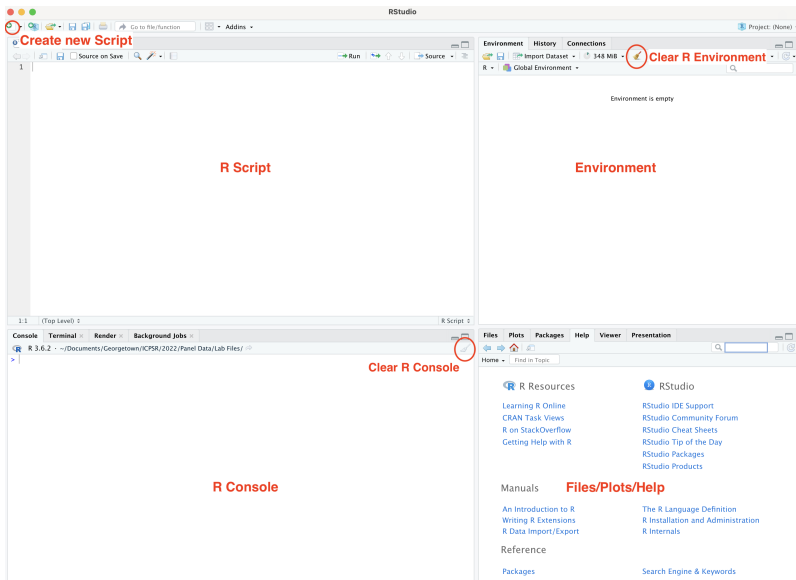
Basic Steps of Data Analysis

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

Basic Steps of Data Analysis

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

R Studio Interface



R Studio Interface

- R Script
 - Write your code here

R Studio Interface

- R Script
 - Write your code here
 - Each line is a separate command

R Studio Interface

- R Script
 - Write your code here
 - Each line is a separate command
 - "Run" a line by highlighting it and clicking "Run" or pressing `Cmd+Return` (Mac) or `Ctrl+Return` (Windows)

R Studio Interface

- R Script
 - Write your code here
 - Each line is a separate command
 - "Run" a line by highlighting it and clicking "Run" or pressing Cmd+Return (Mac) or Ctrl+Return (Windows)
 - Annotate/Comment your code by prefacing a line with #

R Studio Interface

- R Script
 - Write your code here
 - Each line is a separate command
 - "Run" a line by highlighting it and clicking "Run" or pressing Cmd+Return (Mac) or Ctrl+Return (Windows)
 - Annotate/Comment your code by prefacing a line with #
 - The script can be saved as a .R file

R Studio Interface

- R Script
 - Write your code here
 - Each line is a separate command
 - "Run" a line by highlighting it and clicking "Run" or pressing Cmd+Return (Mac) or Ctrl+Return (Windows)
 - Annotate/Comment your code by prefacing a line with #
 - The script can be saved as a .R file
- Console
 - Where results are displayed

R Studio Interface

- R Script
 - Write your code here
 - Each line is a separate command
 - "Run" a line by highlighting it and clicking "Run" or pressing Cmd+Return (Mac) or Ctrl+Return (Windows)
 - Annotate/Comment your code by prefacing a line with #
 - The script can be saved as a .R file
- Console
 - Where results are displayed
 - You can type code directly into the console **but this is bad practice since the Console input cannot be saved**

R Studio Interface

- R Script
 - Write your code here
 - Each line is a separate command
 - "Run" a line by highlighting it and clicking "Run" or pressing Cmd+Return (Mac) or Ctrl+Return (Windows)
 - Annotate/Comment your code by prefacing a line with #
 - The script can be saved as a .R file
- Console
 - Where results are displayed
 - You can type code directly into the console **but this is bad practice since the Console input cannot be saved**
- Recall: major advantage of R is saving a script of commands for replicability, collaboration, and error tracing

Objects and Assignment

- In R, we *assign* values/data to *objects*

Objects and Assignment

- In R, we *assign* values/data to *objects*
 - Object: named "box" or "container" to store values/data

Objects and Assignment

- In R, we *assign* values/data to *objects*
 - Object: named "box" or "container" to store values/data
 - Assignment Operator: `<-`

Objects and Assignment

- In R, we *assign* values/data to *objects*
 - Object: named "box" or "container" to store values/data
 - Assignment Operator: `<-`
 - "Less than" (Shift+comma) Hyphen

Objects and Assignment

- In R, we *assign* values/data to *objects*
 - Object: named "box" or "container" to store values/data
 - Assignment Operator: `<-`
 - "Less than" (Shift+comma) Hyphen
 - `object <- values/data`

Objects and Assignment

- In R, we *assign* values/data to *objects*
 - Object: named "box" or "container" to store values/data
 - Assignment Operator: `<-`
 - "Less than" (Shift+comma) Hyphen
 - `object <- values/data`
- In R, we use *commands* (or *functions*) to perform tasks on *objects* and other *arguments*

Objects and Assignment

- In R, we *assign* values/data to *objects*
 - Object: named "box" or "container" to store values/data
 - Assignment Operator: <-
 - "Less than" (Shift+comma) Hyphen
 - object <- values/data
- In R, we use *commands* (or *functions*) to perform tasks on *objects* and other *arguments*
 - Your object is usually one argument, other arguments can be thought of as "command options"

Objects and Assignment

- In R, we *assign* values/data to *objects*
 - Object: named "box" or "container" to store values/data
 - Assignment Operator: <-
 - "Less than" (Shift+comma) Hyphen
 - object <- values/data
- In R, we use *commands* (or *functions*) to perform tasks on *objects* and other *arguments*
 - Your object is usually one argument, other arguments can be thought of as "command options"
 - Type your command, then your object and other arguments in parentheses

Objects and Assignment

- In R, we *assign* values/data to *objects*
 - Object: named "box" or "container" to store values/data
 - Assignment Operator: <-
 - "Less than" (Shift+comma) Hyphen
 - object <- values/data
- In R, we use *commands* (or *functions*) to perform tasks on *objects* and other *arguments*
 - Your object is usually one argument, other arguments can be thought of as "command options"
 - Type your command, then your object and other arguments in parentheses
 - command(object, other arguments)

Relational Syntax

$<$	less than
$<=$	less than or equal to
$>$	greater than
$>=$	greater than or equal to
$==$	equal to
$!=$	not equal to
$\&$	and
$ $	or
NA	missing
$is.na$	is missing
$!is.na$	is not missing

Object Class/Type/Mode

- character: non-numeric (text)

Object Class/Type/Mode

- character: non-numeric (text)
- numeric: numbers

Object Class/Type/Mode

- character: non-numeric (text)
- numeric: numbers
 - Includes "integer" and "double" — concerns level of precision; R works this out behind the scenes

Object Class/Type/Mode

- character: non-numeric (text)
- numeric: numbers
 - Includes "integer" and "double" — concerns level of precision; R works this out behind the scenes
- logical: TRUE and FALSE

Object Class/Type/Mode

- character: non-numeric (text)
- numeric: numbers
 - Includes "integer" and "double" — concerns level of precision; R works this out behind the scenes
- logical: TRUE and FALSE
- factor: a numeric which R has categorized into "levels"

Object Class/Type/Mode

- character: non-numeric (text)
- numeric: numbers
 - Includes "integer" and "double" — concerns level of precision; R works this out behind the scenes
- logical: TRUE and FALSE
- factor: a numeric which R has categorized into "levels"

R Thinks in Matrices

- Definitions

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector
 - List: Special type of vector; multiple data modes/types

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector
 - List: Special type of vector; multiple data modes/types
 - Matrix: a two-dimensional sequence of data of the same type

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector
 - List: Special type of vector; multiple data modes/types
 - Matrix: a two-dimensional sequence of data of the same type
 - Data Frame: a two-dimensional structure of data of varying data types

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector
 - List: Special type of vector; multiple data modes/types
 - Matrix: a two-dimensional sequence of data of the same type
 - Data Frame: a two-dimensional structure of data of varying data types
- You can refer to elements of a vector, matrix, or dataframe using "Indexing"

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector
 - List: Special type of vector; multiple data modes/types
 - Matrix: a two-dimensional sequence of data of the same type
 - Data Frame: a two-dimensional structure of data of varying data types
- You can refer to elements of a vector, matrix, or dataframe using "Indexing"
 - `vector[position]`

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector
 - List: Special type of vector; multiple data modes/types
 - Matrix: a two-dimensional sequence of data of the same type
 - Data Frame: a two-dimensional structure of data of varying data types
- You can refer to elements of a vector, matrix, or dataframe using "Indexing"
 - `vector[position]`
 - `matrix[row, column]`

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector
 - List: Special type of vector; multiple data modes/types
 - Matrix: a two-dimensional sequence of data of the same type
 - Data Frame: a two-dimensional structure of data of varying data types
- You can refer to elements of a vector, matrix, or dataframe using "Indexing"
 - `vector[position]`
 - `matrix[row, column]`
 - `dataframe[row, column]`

R Thinks in Matrices

- Definitions
 - Scalar: A single value
 - Single-element vector
 - Vector: A one-dimensional sequence of data of the same type
 - Very common R command: `c()` combines Values into a Vector
 - List: Special type of vector; multiple data modes/types
 - Matrix: a two-dimensional sequence of data of the same type
 - Data Frame: a two-dimensional structure of data of varying data types
- You can refer to elements of a vector, matrix, or dataframe using "Indexing"
 - `vector[position]`
 - `matrix[row, column]`
 - `dataframe[row, column]`
 - `dataframe$column_name`