

# Social Sciences Intro to Statistics

## Week 1.1 Introduction to R and RStudio

Week 1: Learning goal - Understand what and how to access R and R studio.

### Instructor Introductions

Professor Ozan Jaquette

Preferred pronouns: he, him, his

Contact: [ozanj@ucla.edu](mailto:ozanj@ucla.edu)

Student Office Hours: via Zoom and by appointment.

### Student Introductions

What we want to know about you:

1. Preferred name
2. Pronouns
3. Department and academic year (i.e. 2nd year in Sociology)
4. And answer one of these two questions:
  - What's something you would eventually like to learn how to do in R?
  - What's something that you have observed or think is important that people aren't paying attention to?

## What is this course about?

### Who is this class for?

This class is for anyone who wants to work with data, that is people who want to be:

- Researchers working with survey data and doing traditional statistical analyses
- Researchers who want to do “data science” oriented research
- Analysts working at think tanks/non-profits or as institutional researchers
- Journalists who create interactive data visualizations

### What is data management?

- All the stuff you have to do to create analysis datasets that are ready to analyze, e.g.:
  - Collect data
  - Read/import data into statistical programming language
  - Clean data
  - Integrate data from multiple sources (e.g, join/merge, append)
  - Change organizational structure of data so it is suitable for analysis
  - Create “analysis variables” from “input variables”
  - Make sure that you have created analysis variables correctly

## Syllabus/logistics

### Syllabus and course logistics

Course links:

- Link to [syllabus](#) #change later to a pdf file that's on github
- Link to resources page of course [github website](#) #Change later if needed

### Course Structure

Course structure consists of weekly asynchronous course materials and weekly synchronous meetings. Each week we will focus on a particular topic (e.g., creating variables; writing functions). For each unit, students will complete bi-weekly problem sets. Problem sets will be completed in groups and focus on practical application of concepts/skills from the topic of the week.

## Synchronous meetings

Synchronous class meetings will be on Zoom. Attendance during the entire period is required, but students may ask instructor/TAs for exceptions due to scheduling conflicts.

## Assessment and Grading

Course grade will be based on the following components:

- Weekly problem sets (4 bi-weekly problem sets, 60 percent of total grade)
- Participation (15 percent of total grade)
- Take-home final (25 percent of total grade)

## Learning Goals

At the end of this course, students will be able to:

1. Learn how to describe data using summary statistics and graphs.
2. Learn fundamental concepts of inferential statistics.
3. Use R to investigate data, to summarize and visualize data, and perform hypothesis tests
4. Perform and interpret regression analysis with R and by hand.

## What is R?

R is a programming language for data visualization and statistical computing. You will often see R used for data mining, bioinformatics, and data analysis. R language has a large number of extension packages, containing reusable code, documentation, and sample data.

According to the Inter-university consortium for political and social research ([ICPSR](#)): > R is “an alternative to traditional statistical packages such as SPSS, SAS, and Stata such that it is an extensible, open-source language and computing environment for Windows, Macintosh, UNIX, and Linux platforms. Such software allows for the user to freely distribute, study, change, and improve the software under the [Free Software Foundation’s GNU General Public License](#).”

- For more info visit [R-project.org](https://www.r-project.org)

## Base R vs. R packages

There are “default” packages that come with [R](#). Some of these include:

- `as.character`
- `print`
- `setwd`

And there are [R packages](#) developed and shared by others. Some R packages include:

- `tidyverse`
- `stargazer`
- `foreign`

more about these in later weeks...

## Installing and Loading R packages

You only need to install a package once. To install an R package use `install.package()` function.

```
#install.packages("tidyverse")
```

However, you need to load a package everytime you plan to use it. To load a package use the `library()` function.

```
library(tidyverse)
```

## RStudio

“[RStudio](#) is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.”

## R Markdown

[R Markdown](#) produces dynamic output formats in html, pdf, MS Word, dashboards, Beamer presentations, etc.

- We will be using R Markdown for lectures and homeworks.

## Why learn R? R can do a lot of stuff!

How we have used R+RStudio+RMarkdown in our research team

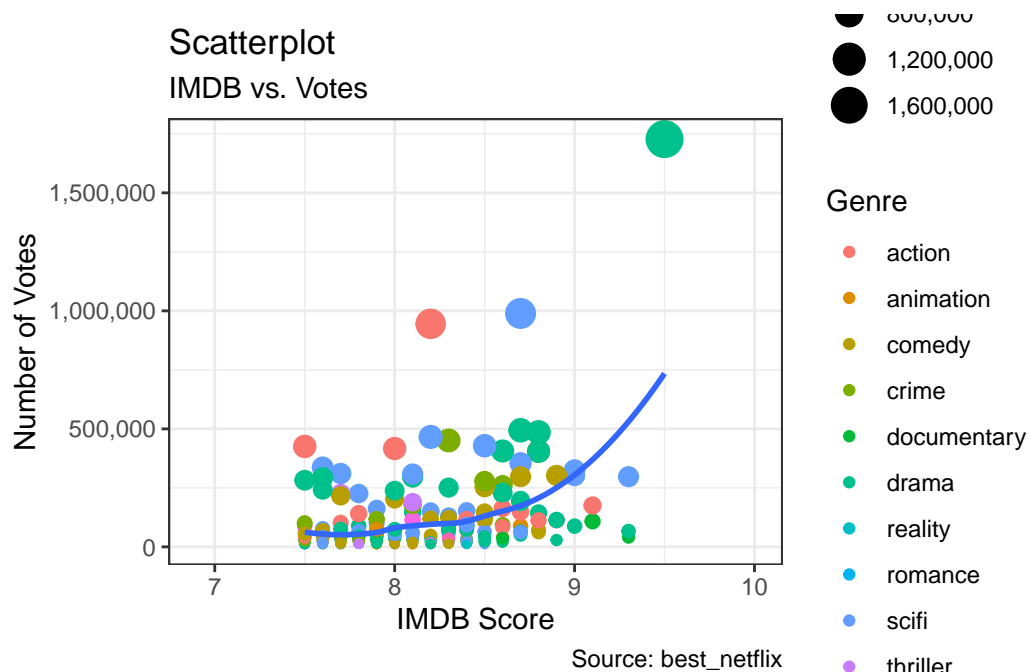
- Stuff traditional statistical software (e.g., SPSS, Stata) can do
  - Data manipulation, creating analysis datasets
  - [Descriptive statistics and statistical models](#)
  - Graphs
- Stuff traditional statistical software cannot do
  - [Static policy reports](#)
  - Static presentations
    - \* All lectures for this class written in RMarkdown
  - [Interactive presentations](#)
  - [Interactive maps](#)
  - [Interactive dashboards](#)
  - Interactive graphs

Some of the other stuff R can create/do:

- [Websites](#); [journals](#); [books](#); [web- scraping](#); network analysis; machine learning/artificial intelligence

## Graphs

- We can create graphs with [ggplot2](#) package



## Directories and filepaths

### Directories and filepaths

- Give you a very brief overview of “directories” (i.e., folders) and “filepaths” (tells you where folder is located) in R
- Why? After this overview we will ask you to create the directory structure you will use for all files related to this class

### Working directory

#### (Current) Working directory

- the folder/directory in which you are currently working
- this is where R looks for files
- Files located in your current working directory can be accessed without specifying a filepath because R automatically looks in this folder

Function `getwd()` shows current working directory

```
getwd()
```

```
[1] "/Users/bellelee/Downloads/SSS Lectures/Week 1 Lectures"
```

Command `list.files()` lists all files located in working directory

```
list.files()
```

### Working directory, “Code chunks” vs. “console” and “R scripts”

When you run **code chunks** in RMarkdown files (.Rmd), the working directory is set to the filepath where the .Rmd file is stored

```
getwd()
```

```
[1] "/Users/bellelee/Downloads/SSS Lectures/Week 1 Lectures"
```

```
list.files()
```

When you run code from the **R Console** or an **R Script**, the working directory is:

- if you are working on an R “Project”, the working director is the main directory for the project

```
getwd()
```

### Absolute vs. relative filepath

**Absolute file path:** The absolute file path is the complete list of directories needed to locate a file or folder.

```
setwd("/Users/pm/Desktop/SSSclass/lectures/lecture2")
```

**Relative file path:** The relative file path is the path relative to your current location/directory. Assuming your current working directory is in the “lecture2” folder and you want to change your directory to the data folder, your relative file path would look something like this:

```
setwd(".././data")
```

File path shortcuts

Key	Description
~	tilde is a shortcut for user's home directory (mine is my name pm)
../	moves up a level
../../	moves up two level

## Create “R project” and directory structure

### What is an R project? Why are you doing this?

What is an “R project”?

- helps you keep all files for a project in one place
- When you open an R project, the file-path of your current working directory is automatically set to the file-path of your R-project

Why are we asking you to create R project and download a specific directory structure?

- We want you to be able to run the .Rmd files for each lecture on your own computer
- Sometimes these .Rmd files point to certain sub-folders
- If you create R project and create directory structure we recommend, you will be able to run .Rmd files from your own computer without making any changes to file-paths!

### Follow these steps to create “R project” and directory structure

1. Download this zip folder: [LINK HERE] #need to update
  - this zip file contains the shell file directory you should use for this class
  - Unzip the folder
    - contains folder named “SSS class”; this is the folder that will contain all materials for this course
    - “rclass” contains two folders: “data” and “lectures”
  - Move “rclass” folder to your preferred location (e.g, documents, desktop, dropbox, etc)
2. In RStudio, click on “File” » “New Project” » “Existing Directory” » “New Project”
  - “Browse” to find “SSS class” folder you just saved
  - Then click on Create Project
3. Save the following files in “SSS class/lectures/lecture1”



- lecture1.1.Rmd
- lecture1.1.pdf
- lecture1.2.Rmd
- lecture1.2.Pdf
- lecture1.2.R

**Next, you follow these steps**

- you can add any additional sub-folders you want to the “SSS class” folder
  - e.g., “syllabus”, “resources”
- You can add any additional files you want to the sub-directory folders you unzipped
  - e.g., in “SSS class/lectures/lecture1” you might add an additional document of notes you took