Lecture 2 : Describing Data with Pictures and Numbers

Check-In (Loading Data)

Hi class, please work on the check-in. We will get started soon.

Agenda and Goals

This week, we'll learn how to describe data using R and statistics and human language.

- **Check-In, Announcements, and Week 1 Review (20 Minutes)**
  - Presentations:
  - Begin today (professor)
  - Sign-Up Sheet Updated w/ another presentation.
  - Comments / Questions from Week 1.
    - **Shifting Mini Exam to 2/28**
      - More time to practice
      - Can cover (and review) Linear Models [foundational!]
      - No class "after" the exam. Just the exam. That's enough.
      - *2/28 presentation moved to 2/21*
    - **Do we want a class discord (community / troubleshooting?)**
- **PART 1 : Describing Data with R, Statistics, and Human Language (50 Minutes)**
  - Describing Categorical Data
  - Describing Numeric Data
  - Using Quarto to Make and Share Nice Reports
- **BREAK TIME (10 Minutes) + Article Presentation (20 Minutes)**
- **PART 2 : The Mean (50 Minutes)**
  - The Mean as Prediction
  - The Mean Minimizes Residual Error
  - The Mean is Sensitive to Outliers
  - The Mean is a Social Construction

Week 1 Recap and Review

The Monty-Hall Problem [100 Door Version]

```
## CREATING 100 DOORS
door100 <- array() # this defines an empty array; a place to "store" my values
for(i in c(1:100)){ # this starts the for-loop, and tells R I want to repeat some process 10
  door100[i] <- paste("door", i, sep = "") # the paste() function sticks the string "door" a
                                           # since i updates every time we iterate through t
                                           # the assign <- sticks this value to the variable
} # this ends the loop

door100 # testing to see if the loop works. it did.
```

```
  [1] "door1"   "door2"   "door3"   "door4"   "door5"   "door6"   "door7"  
  [8] "door8"   "door9"   "door10"  "door11"  "door12"  "door13"  "door14" 
 [15] "door15"  "door16"  "door17"  "door18"  "door19"  "door20"  "door21" 
 [22] "door22"  "door23"  "door24"  "door25"  "door26"  "door27"  "door28" 
 [29] "door29"  "door30"  "door31"  "door32"  "door33"  "door34"  "door35" 
 [36] "door36"  "door37"  "door38"  "door39"  "door40"  "door41"  "door42" 
 [43] "door43"  "door44"  "door45"  "door46"  "door47"  "door48"  "door49" 
 [50] "door50"  "door51"  "door52"  "door53"  "door54"  "door55"  "door56" 
 [57] "door57"  "door58"  "door59"  "door60"  "door61"  "door62"  "door63" 
 [64] "door64"  "door65"  "door66"  "door67"  "door68"  "door69"  "door70" 
 [71] "door71"  "door72"  "door73"  "door74"  "door75"  "door76"  "door77" 
 [78] "door78"  "door79"  "door80"  "door81"  "door82"  "door83"  "door84" 
 [85] "door85"  "door86"  "door87"  "door88"  "door89"  "door90"  "door91" 
 [92] "door92"  "door93"  "door94"  "door95"  "door96"  "door97"  "door98" 
 [99] "door99"  "door100"
```

```
## WHAT HAPPENS IF THERE ARE 100 DOORS?
win.stay100 <- array() # defining a new place to save values when we have 100 doors. I could
win.switch100 <- array() # defining a new place to save values when we have 100 doors. I cou
for(i in c(1:10000)){ # running 10000 simulations.
  choice <- sample(door100, 1) # the new set of 100 doors
  treasure <- sample(door100, 1)
  can.open <- setdiff(door100, c(choice, treasure))
  monty.open <- sample(can.open, 98) # monty will open 98 other doors, leaving my choice + t
  choice.switch <- sample(setdiff(door100, c(monty.open, choice)), 1) # if switch, I can sti
  win.stay100[i] <- ifelse(choice == treasure, "WIN", "LOSE") # this is the result if we sta
  win.switch100[i] <- ifelse(choice.switch == treasure, "WIN", "LOSE") # this is the result
}

sum(win.stay100 == "WIN")/length(win.stay100) # probability if I stay. Note that rather than
```

```
[1] 0.0078
```

```
sum(win.switch100 == "WIN")/length(win.switch100) # YEAH!!!!
```

```
[1] 0.9922
```

Data Organization and Loading

Below are the twelve (12) criteria Broman & Woo (2018) articulated for thinking about data. Review the list. What practices below have you encountered or struggled with in working with data so far? What terms or ideas do you have questions about? Is there anything else that should be added to this list?

1. Consistency
2. Good Names
3. Write Dates as YYYY-MMM-DD
4. No Empty Cells
5. Put Just One Thing in a Cell
6. Make it a rectangle.
7. Data Dictionary
8. No calculations in the raw data file.
9. No font color or highlighting as data.
10. Make backups
11. Data validation to avoid errors.
12. Saving data in plaintext.

Check-In : Loading and Navigating Data

PART 1 : Describing Data with R, Statistics, and Human Language

Categorical Data

- `summary()` or `table()`: to count frequencies of categorical data
- `as.factor()` and `as.numeric()` : to translate data
- `levels()` or `factor()` : to relevel or change factor names

- `plot` : graphing categorical data

Numeric Data

- `summary()` or `psych::describe() # this comes from the psych package which you must install` : to summarize data
  - `mean()`
  - `median()`
  - `sd()`
  - `range()`
  - note : if there are missing data (often!) you must manually tell R to remove the missing data : `mean(d$variable, na.rm = T)`
- **`hist()`** or `boxplot()` **:** graphing numeric data

Using Quarto to Make and Share Nice Reports

Quarto is a version of R Markdown, which is a version of Markdown, which is a powerful way to author code that is meant for both humans and computers to read.

- Advantages :
  - can create a document that works for R code, can create a presentation, or a website
  - much faster to get your code from R to something that humans can read
    - no more copy-paste graphs or output.
    - can update graphs and output as your needs / datset changes
  - lots of features - open-source heritage and culture, but supported financially my Micro$oft.
    - ability to format your code; render it as a website, pdf, book, etc.
    - interactive documents (Shiny; html-live; etc.)
- Disadvantages :
  - code must be "perfect" in order to correctly render.
  - can go down formatting and feature rabbit holes that are not necessarily condusive to good science.
  - another dialect of the language you are trying to learn
    - in R : code is the default; human comments added with #s

    - in Quarto : human text is the default; you insert a code block when you want R to do something (and can then comment in that code)

```
1+1 # like this
```

```
[1] 2
```

- In this class, we will work with both .R scripts and .qmd Quarto Markdown Files
  - .R Scripts for tinkering with data (in-class tutorials; initial analyses)
  - .qmd files for "final" products (Lecture notes, lab documents, your project)
- There are many thorough guides on how to use Quarto, but honestly the official Quarto reference book is almost constantly open on my computer (in multiple tabs....sigh.) But let me know if you find another cool resource!
- **Things to do in Quarto :**
  - write in human text
  - insert a code block
  - insert inline code
  - render everything with no pain and drama as a .pdf or .html file and share this with others.

BREAK TIME & ARTICLE PRESENTATION

PART 2 : Mean Thoughts [R Demo]

We will see how far we get in exploring these ideas!

The Mean is Prediction

- Illustrating the mean as a "line of best fit" [our first linear model!]
- Illustrating the standard deviation as the "average" amount of residual error.

The Mean Minimizes Residual Error

- Illustrating that the mean is the value that reduces the sum of our residual error.

The Mean is Sensitive to Outliers

- Illustrating how the mean is more sensitive to outliers than the median

The Mean Is a Social Construction

- Testing whether anyone is actually described by “the mean”?
- What is average, anyway?