SOC6707: Intermediate Data Analysis

Monica Alexander

Week 1: Introduction

Overview of today

- Overview of course
- Why learn statistics?
- Review concepts
- Introduction to R

Overview of course

Instructor and TA

Instructor: Monica Alexander (she/her)

- Email: monica.alexander@utoronto.ca
- Office hours: TBA

TA: Julia Igenfeld (she/her)

- Email: julia.ingenfeld@mail.utoronto.ca
- Office hours: TBA

A bit about me

Me:

- statistics $\cap$ chemistry $\rightarrow$ social science $\cap$ statistics
- 50/50 Statistical Sciences and Sociology departments
- Not Canadian (Australia $\rightarrow$ USA $\rightarrow$ Canada)

What I work on: a mix of demography, applied stats, epidemiology and computational social science

Mode of delivery

For the most part, this course will be delivered in **online synchronous** format.

- Lectures, tutorials, and office hours online through BBCollaborate (accessed through Quercus)
- Lectures will be recorded such that they can be accessed at a later date
- TBD: maybe some short pre-recorded materials

Objectives

This course introduces statistical techniques and methods to analyze data to draw inferences about social processes. You will learn

- How to read in, describe, plot and analyze data in a statistical software that uses a programming language
- Some important methods of statistical analysis to explore relationships between social phenomena
- How to assess and evaluate the suitability and performance of statistical methods in different contexts

Methods covered include generalized linear models, Bayesian inference and multilevel models.

General philosophies

- Understanding your data
- Reproducibility
- Knowing when and when not to use methods

This course will be very hands-on with coding and data munging.
The learning curve for R is steep but will (hopefully?!) pay off.

Textbooks

There is no required textbook for this class. Some resources that might be useful:

- R for Data Science: https://r4ds.had.co.nz/ (free)
- Gelman, Andrew; Hill, Jennifer, and Vehtari, Aki. 2020. 'Regression and Other Stories' (This is around $70 on Book Depository).

Software

We will be using the programming language R in this course, through RStudio.

- Free to download and install
- More info on how to install these on Quercus
- More introduction later on in lecture

Assessment

Overview

- Three assignments (3x15% = 45%)
- Mid-term (20%)
- Research project (35%)

Assignments

- Data analysis with R
- Interpretation
- Hand in code, instructor/TA should be able to run without errors

Mid-term exam

- To be taken online through Quercus
- Multiple choice and short answer questions
- Will need to be able to interpret R output

Research Project

Choose a data set, research question, and analysis approach

In the research project you will

- Develop a research question based on data set of choice
- Analyze data using methods learned in class
- Present, interpret and summarize findings

Research Project

Worth a total of 35%, but will be graded in four parts:

1. Research question, variables to be used (5%) due with A1
2. EDA (5%) due with A2
3. Analysis (5%) due with A3
4. Final report, which incorporates 1-3 (20%), due at end of semester.

More detail in course outline, and as we go along.

Lecture + Lab

- Each week I will lecture for about 1-1.5 hours, then we will have a lab with hands-on practice in R.

Course Policies

- **Communication**: First, see if you can answer your question by checking the syllabus. Second, try to ask questions during class, tutorials, or office hours. Third, there will be a discussion board on Quercus. Fourth, email myself or your TA (please include the course number in the subject line)
- **Accessibility**: visit http://studentlife.utoronto.ca/accessibility as soon as possible.

We're all out here doing our best

- The current situation makes both learning and teaching challenging
- Try to be understanding of everyone's sub-optimal situation
- Communication is key
- There may be guest appearances from my toddler

Doing okay?

Whether it is the chaos of 2020 getting you down, or something else, it is always okay to reach out for support!

- My Student Support Program – My SSP – mental health support for all U of T students. Free, confidential, immediate support. Available 24/7 in multiple languages. Download the My SSP App or call 1-844-451-9700. uoft.me/myssp
- Call Good2Talk. Free, confidential helpline with professional counseling, information and referrals for mental health, addictions and well-being, 24/7/365 1-866-925-5454

Why learn statistics?

Why learn statistics?

As sociologists, we are trying to understand different aspects of society.

Statistical techniques give us a means to investigate and test research questions and policy impacts across different areas of people's lives.

Example research questions could include

- How is population mobility changing in the era of Covid-19?
- How do people cope with financial hardship?
- How does paid maternity leave affect women's workforce participation?
- Does volunteering increase your sense of wellbeing?

Why learn statistics

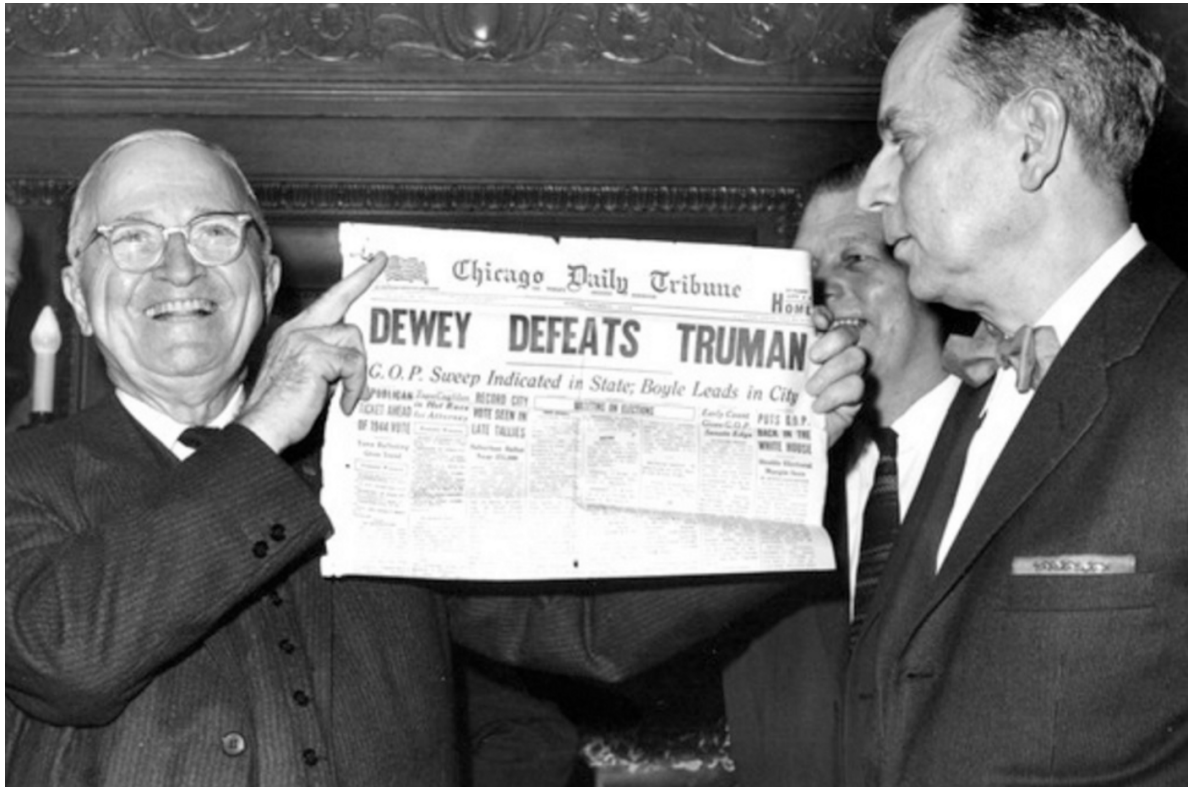
It's not just learning what you could do with data, it's learning what not to do with data

- How biases and selection can give misleading conclusions
- When is it inappropriate to use certain techniques

It's not just to support your own arguments, it's learning how to assess other people's arguments

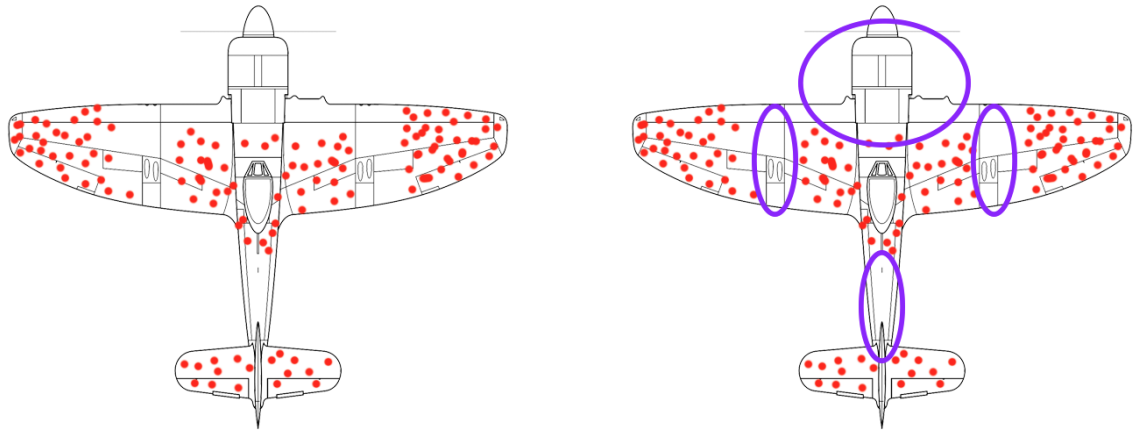
- Statistics, data analysis and visualization is an art form
- Cutting through the lies, damned lies and (misused) statistics

Misleading statistics I



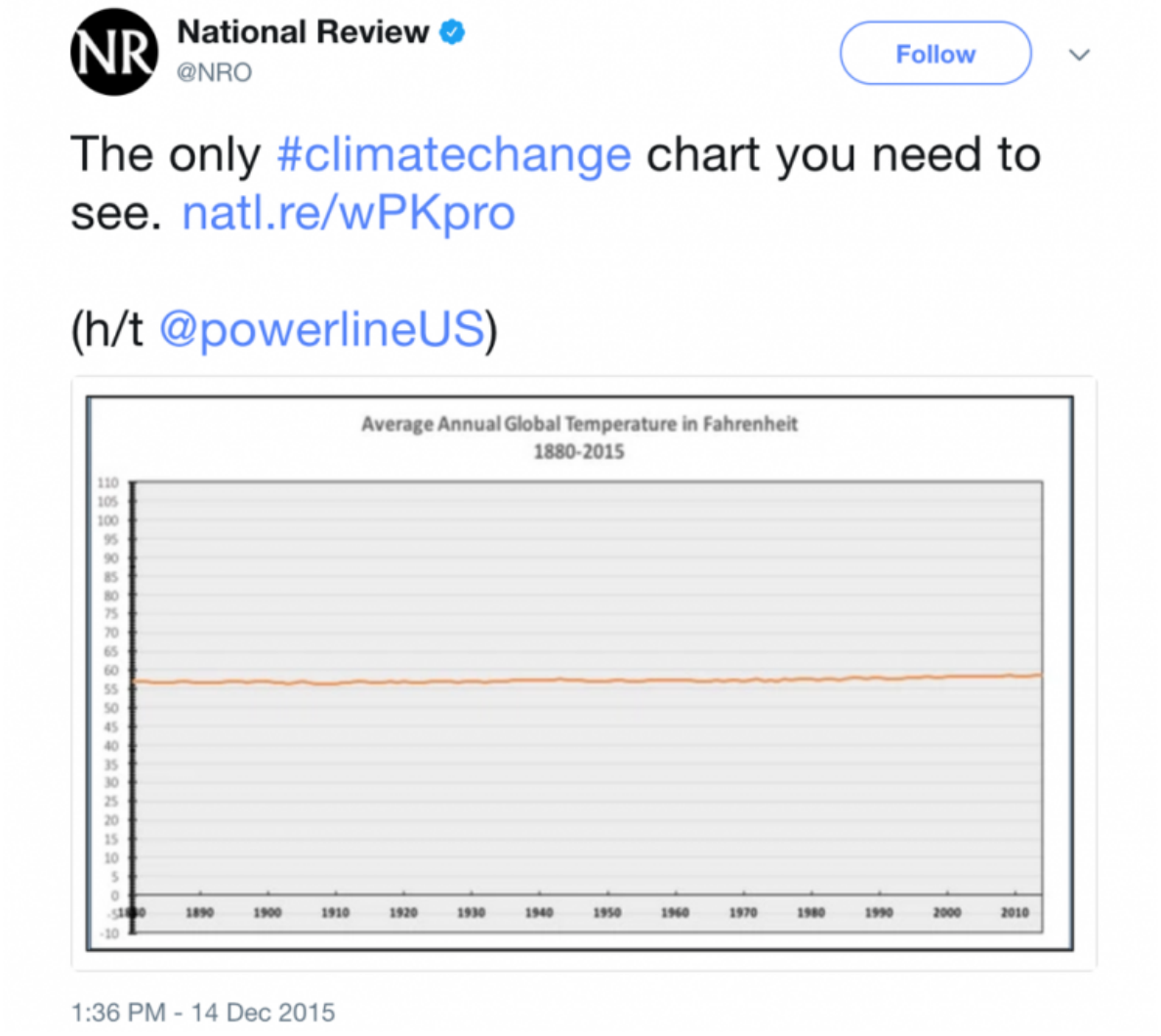
- Truman won the Presidential election in 1948
- This is a photo of Truman holding an up an erroneous headline
- Based on phone survey which predicted overwhelming win for Dewey
- What went wrong?

Misleading statistics II

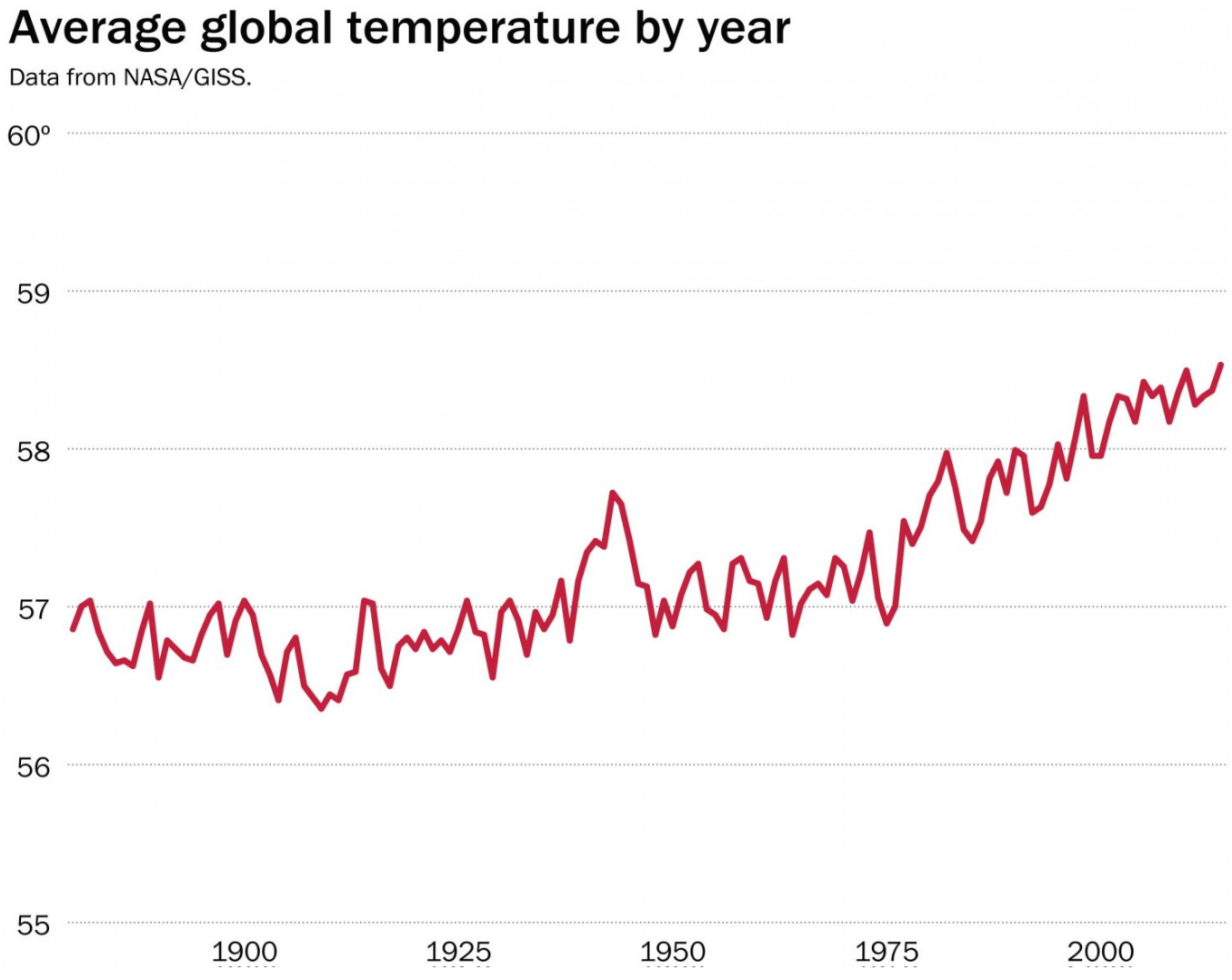


- Abraham Wald in WWII
- Want to place armor on planes in most effective place
- Gathered data from planes returning from battle and observed bullet holes
- Most holes in the fuselage, not so many in the engines
- Where should armor go?

Misleading statistics III



Alternative



Review

Populations

At the core of statistical methods is wanting to say something about a **population** of interest.

What is a population? Depends on the context of study

- Everyone enrolled in university in Canada
- Everyone at UofT
- Everyone studying graduate-level sociology at UofT
- Everyone in this class

Samples

Say we want to study the relationship between hours studied and overall achievement for all university students in Canada.

- Not really plausible to get data on this for the whole of Canada.
- In reality, we would collect data on a **subset** or **sample** of the population and try and generalize to the whole of Canada.
- With statistics, we are going to make conclusions based on what we see in the sample that we hope will be true for the population.

Samples

Our example: the relationship between hours studied and overall achievement for all university students in Canada.

- We could plausibly measure the hours studied and overall achievement in SOC6707
- This class would be a sample of the population of interest, because you are all university students in Canada.

Is it a good sample? What do I mean by good?

Sampling techniques

Terminology:

- **Element**: An element is an object or case, the unit on which a measurement is made.
- **Population**: The population is a collection of elements about which we wish to make an inference
- **Sampling units**: The sampling units are non-overlapping collections of elements in the population.
- **Sampling frame**: The sampling frame is a list of the elements or, for more complex samples, a list of the sampling units.
- **Sample**: A sample is a subset of the elements drawn from the population using one of several sampling methods.

What are each of these in our example?

Sampling techniques

Include:

- **Simple Random Sampling (SRS)**: A random sample is sometimes defined as a sample in which all possible elements have an equal chance of occurring.
- **Stratified Sampling**: based on variable of interest
- **Cluster Sampling**: SRS within clusters (e.g. districts within a province, schools within districts)
- **Convenience Sampling**

What is the sampling method in our example?

Two main domains of statistics

- **Descriptive statistics**: uses the data to provide descriptions of the population, either through numerical calculations or graphs or tables.
- **Inferential statistics**: makes inferences and predictions about a population based on a sample of data taken from the population in question.

We will cover both types in this course. Understanding patterns in descriptive statistics is essential to doing good inferential statistics

Variables

Traits, characteristics, outcomes that we are interested in. e.g.

- hours of study
- course grade
- province of residence
- age
- self-reported health

Variables

Often we are interested in studying the relationship between two or more variables.

- The **outcome** of interest is the **dependent variable**
- Variables **used to explain the outcome** can be called
  - independent variables
  - explanatory variables
  - covariates
  - predictors

I will use these terms interchangeably.

What is the independent and dependent variable in our example?

Types of measurement of variables

- **Quantitative**: has a numeric meaning
  - **Continuous**: any possible number
  - **Discrete**: possible values can assume only certain values, usually the counting numbers
- **Qualitative**: categorical, no numeric meaning

What are the types of variables in our example?

Random variables

- A **random variable** is a variable whose values depend on the outcomes of a random process.
- For our purposes, the "random process" is taking a random sample of a population
- For example, consider the variable annual income:
  - We randomly select someone from the population and note their income.
  - The value of this depends on the person who was selected
  - If we randomly selected someone else again, it's likely that the income value would be different

RVs are the basis of probability and statistical inference! Much more on their properties later.

Some common symbols and notation

We will see a fair bit of math notation in this class, but there are some common notation that will come up over and over. To start with

- Population size $= N$
- Sample size $= n$
- A particular individual in a sample denoted by index i
- Random variables (note these are captials!):
  - Dependent variable: Y
  - Independent variables X
- A set of random variables for individuals $i = 1, 2, \ldots, n$: $X_1, X_2, \ldots, X_n$
- Specific values or outcomes of the corresponding random variables:
  - Dependent variable: y
  - Independent variables x

Summary measures of quantitative data

Summary measures of quantitative data

Pretend you have a set of observations of a quantitative variable, e.g. everyone's height in this class. We often want to **summarize** our set of observations with one or more numbers. Often interested in:

- Measures of central tendency, i.e. what would we expect someone's height to be, what's the most common height
- Measures of spread, i.e. what are the ranges of heights observed, what is the deviation of heights away from the expected height?

Measures of central tendency

- **Mean**: the average
  - Population mean usually denoted as μ
  - Sample mean denoted with a bar e.g. $\bar{x}$

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

- **Median**: the value for which 50% of the sample is below and 50% of the sample is above. It is the 50% percentile. To calculate
  - Order set of values from smallest to largest
  - find the middle number
- **Mode**: the value that occurs the most frequently

Example

```
x <- c(4,6,2,1,6,2,76,3,2,56,10,1,4,5,2,15,32)
sort(x)
```

```
##  [1]  1  1  2  2  2  2  3  4  4  5  6  6 10 15 32 56 76
```

- Mean = ?
- Median = ?
- Mode = ?

Example

```r
mean(x)
```

```
## [1] 13.35294
```

```r
median(x)
```

```
## [1] 4
```

```r
# the mode is a bit more tricky
table(x)
```

```
## x
##  1  2  3  4  5  6 10 15 32 56 76
##  2  4  1  2  1  2  1  1  1  1  1
```

```r
names(sort(table(x), decreasing = TRUE)[1])
```

```
## [1] "2"
```

Measures of variability

- **Range**: The difference between the minimum and maximum value
- **Interquartile range**: The difference between the 25% and 75% percentiles. To calculate
  - Order set of values from smallest to largest
  - Separate into quarters
  - Find the first quarter (Q1) and third quarter (Q3)
  - IQR = Q3 - Q1

Measures of variability

- **Variance**: average of the squares of the deviations
  - Population variance: σ^2
  - Sample variance: s^2

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$$

- **Standard deviation**: average of the deviations
  - Population standard deviation: σ
  - Sample standard deviation: s

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{s^2}$$

Example R code

```r
sort(x)
```

```
##  [1]  1  1  2  2  2  2  3  4  4  5  6  6 10 15 32 56 76
```

```r
max(x) - min(x)
```

```
## [1] 75
```

```r
IQR(x)
```

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

```r
var(x)
```

```
## [1] 461.6176
```

```r
sd(x)
```

```
## [1] 21.48529
```

```r
#check this is the same
sqrt(var(x))
```

```
## [1] 21.48529
```

Introduction to R

R and RStudio

- You will need to download and install both R and RStudio
- More info on Quercus
- Try and do this before the lab

What is R?

- R is a programming language for statistical computing and graphics
- Using R is like speaking another language (but you type it)

You may have used other programs to do statistical calculations before (Excel, SPSS)

- With R you have to give the computer typed commands in order for it to do stats (rather than clicking buttons)
- Much more powerful methods

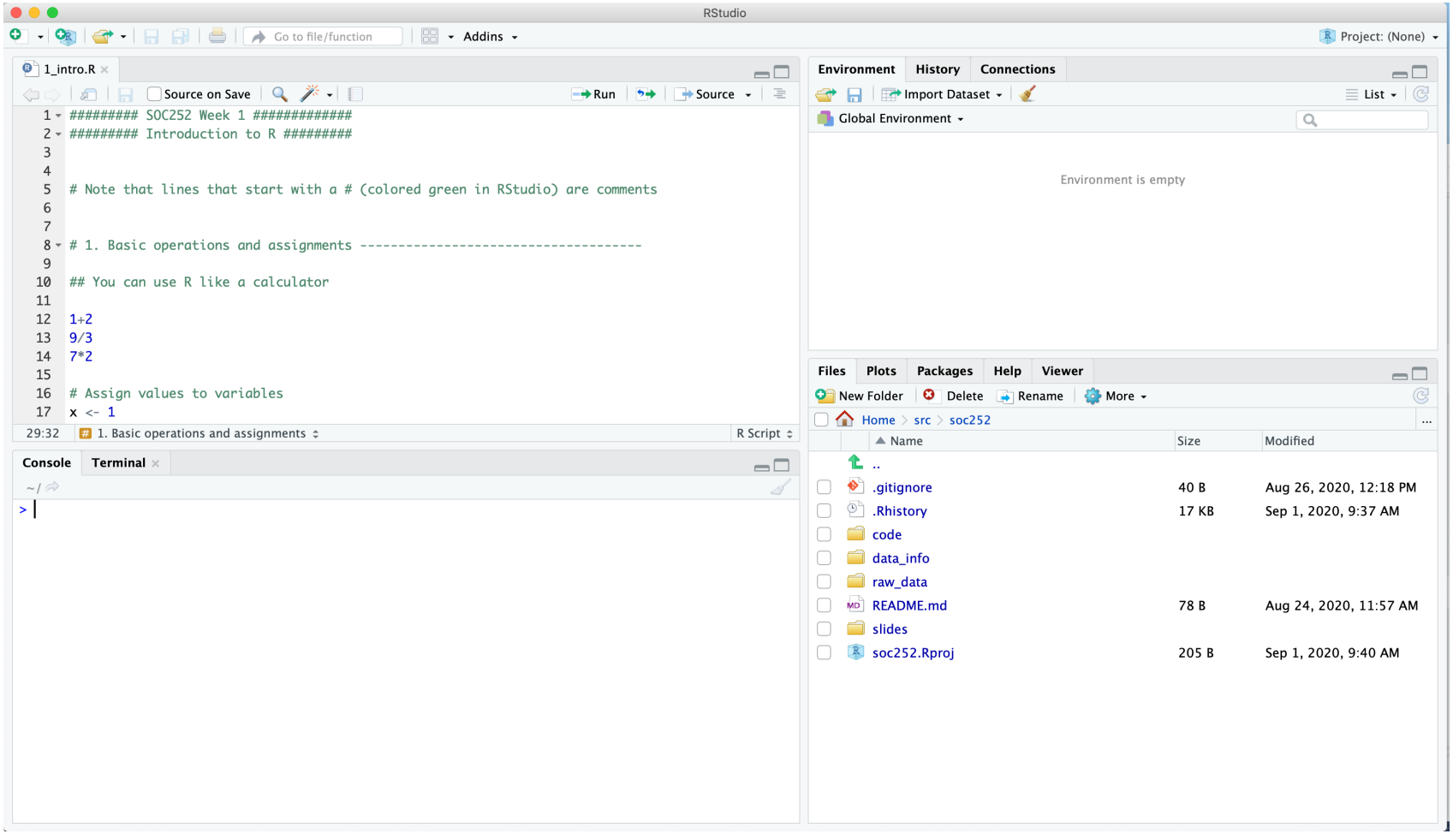
What is RStudio?

- RStudio is an integrated development environment for R.
- It makes it easier to write R code and visualize inputs and outputs.
- You will need to download and install both R and RStudio, details in course outline (also in lab 1)

Think of a car analogy:

- R is the engine
- RStudio is the car dashboard (steering wheel, controls etc)

RStudio


RStudio
Go to file/function
Addins
Project: (None)
1_intro.R
Source on Save
Run
Source
1 ######### SOC252 Week 1 #############
2 ######### Introduction to R #########
3
4
5 # Note that lines that start with a # (colored green in RStudio) are comments
6
7
8 # 1. Basic operations and assignments ----------------------------------
9
10 ## You can use R like a calculator
11
12 1+2
13 9/3
14 7*2
15
16 # Assign values to variables
17 x <- 1
29:32
1. Basic operations and assignments
R Script
Console
Terminal
~/
>
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
src
soc252
Name
Size
Modified
..
.gitignore
40 B
Aug 26, 2020, 12:18 PM
.Rhistory
17 KB
Sep 1, 2020, 9:37 AM
code
data_info
raw_data
README.md
78 B
Aug 24, 2020, 11:57 AM
slides
soc252.Rproj
205 B
Sep 1, 2020, 9:40 AM

Writing code in R

1. R Console: Executes each line of code as you go; does not save code for later use
2. R Script: Saves code and comments in a file so you can select some or all of the code in a script file to run; does not include output
3. R Markdown: A file which combines text and chunks of R code (which can be executed independently). This allows you to see output without "knitting" the whole file.

We will focus on number 3.

R Markdown documents

```
---
title: "Example R Markdown document"
author: "Monica Alexander"
date: "06/09/2020"
output: pdf_document
---
```

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

Heading text

This is an R Markdown document. The main text goes here. Below is a chunk of R code. You can excute this by clicking the green play arrow button. The output is shown below.

```{r cars}
7+8
```

```
[1] 15
```

Knitted to a PDF

Example R Markdown document

Monica Alexander

06/09/2020

Heading text

This is an R Markdown document. The main text goes here. Below is a chunk of R code. You can excute this by clicking the green play arrow button. The output is shown below.

```
7+8
```

```
## [1] 15
```

R Packages

- A lot of people have written **R packages**, which are add ons to base R that increase the functionality

Think of a phone analogy:

- R/RStudio is a phone
- R packages are apps

We will be using a few different R packages quite a lot during the course, e.g.

- `dplyr` (data manipulation)
- `ggplot2` (graphing)

These and other packages can be downloaded through downloading the `tidyverse` package (you will do this in the lab)

R code

Create a vector of numbers:

```
x <- c(1,3,5,7,9)
```

Calculate summary statistics:

```
mean(x)
```

```
## [1] 5
```

```
max(x)
```

```
## [1] 9
```

```
IQR(x)
```

```
## [1] 4
```

Reading in and manipulating real data

```
library(tidyverse)
gss <- read_csv(file = "data/gss.csv")
```

- `read_csv` is a **function** from the `tidyverse` package
- We are assigning the contents of the file to an object called `gss`
- The `gss` object is a data frame or **tibble** that contains all the GSS data
- We can now use other functions to manipulate and analyze the GSS data in R

Selecting a column with `select()`

```
select(gss, age)
```

```
## # A tibble: 20,602 x 1
##       age
##     <dbl>
##  1  52.7
##  2  51.1
##  3  63.6
##  4  80
##  5  28
##  6  63
##  7  58.8
##  8  80
##  9  63.8
## 10  25.2
## # ... with 20,592 more rows
```

The pipe %>%

```
gss %>%
  select(age)
```

```
## # A tibble: 20,602 x 1
##       age
##     <dbl>
##  1  52.7
##  2  51.1
##  3  63.6
##  4  80
##  5  28
##  6  63
##  7  58.8
##  8  80
##  9  63.8
## 10  25.2
## # ... with 20,592 more rows
```

- Read as "and then"
- So above we are taking the gss data **and then** selecting the age column

More than one pipe / Important functions

Arrange

```
gss %>%
  select(age) %>%
  arrange(age)
```

```
## # A tibble: 20,602 x 1
##       age
##     <dbl>
##  1   15
##  2   15
##  3   15
##  4   15
##  5   15
##  6   15
##  7   15
##  8   15.1
##  9   15.1
## 10   15.1
## # ... with 20,592 more rows
```

Important functions for manipulating data

Arrange by descending order

```
gss %>%
  select(age) %>%
  arrange(-age)
```

```
## # A tibble: 20,602 x 1
##       age
##     <dbl>
##  1    80
##  2    80
##  3    80
##  4    80
##  5    80
##  6    80
##  7    80
##  8    80
##  9    80
## 10    80
## # ... with 20,592 more rows
```

Important functions for manipulating data

Filter

```
gss %>%
  select(age) %>%
  filter(age<17)
```

```
## # A tibble: 269 x 1
##       age
##     <dbl>
##  1  15.7
##  2  16.3
##  3  16.8
##  4  15.4
##  5  16.4
##  6  16.7
##  7  16.8
##  8  16.1
##  9  15.9
## 10  16.4
## # ... with 259 more rows
```

Important functions for manipulating data

Mutate = add a new column

```
gss %>%
  select(age) %>%
  mutate(age_plus_1 = age+1)
```

```
## # A tibble: 20,602 x 2
##       age age_plus_1
##     <dbl>      <dbl>
##  1  52.7        53.7
##  2  51.1        52.1
##  3  63.6        64.6
##  4  80          81
##  5  28          29
##  6  63          64
##  7  58.8        59.8
##  8  80          81
##  9  63.8        64.8
## 10  25.2        26.2
## # ... with 20,592 more rows
```

Important functions for manipulating data

```
gss %>%
  select(age) %>%
  summarize(mean_age = mean(age))
```

```
## # A tibble: 1 x 1
##   mean_age
##      <dbl>
## 1     52.2
```

Important functions for manipulating data

- The pipe %>%
- `select` (columns)
- `filter` (rows)
- `arrange`
- `mutate`
- `summarize`

Example: lego

Make your own tibble

```
lego <- tibble(color = c('brown', 'pink', 'red',
                         'orange', 'yellow', 'light yellow
                         'light green', 'green', 'blue',
                         'light blue', 'white'),
               number = c(1,1,6,3,7,4,1,7,4,2,7)
               )
```

Make your own tibble

```
lego
```

```
## # A tibble: 11 x 2
##    color        number
##    <chr>         <dbl>
##  1 brown             1
##  2 pink              1
##  3 red               6
##  4 orange            3
##  5 yellow            7
##  6 light yellow      4
##  7 light green       1
##  8 green             7
##  9 blue              4
## 10 light blue        2
## 11 white             7
```

Summary statistics

Find the mean number of blocks

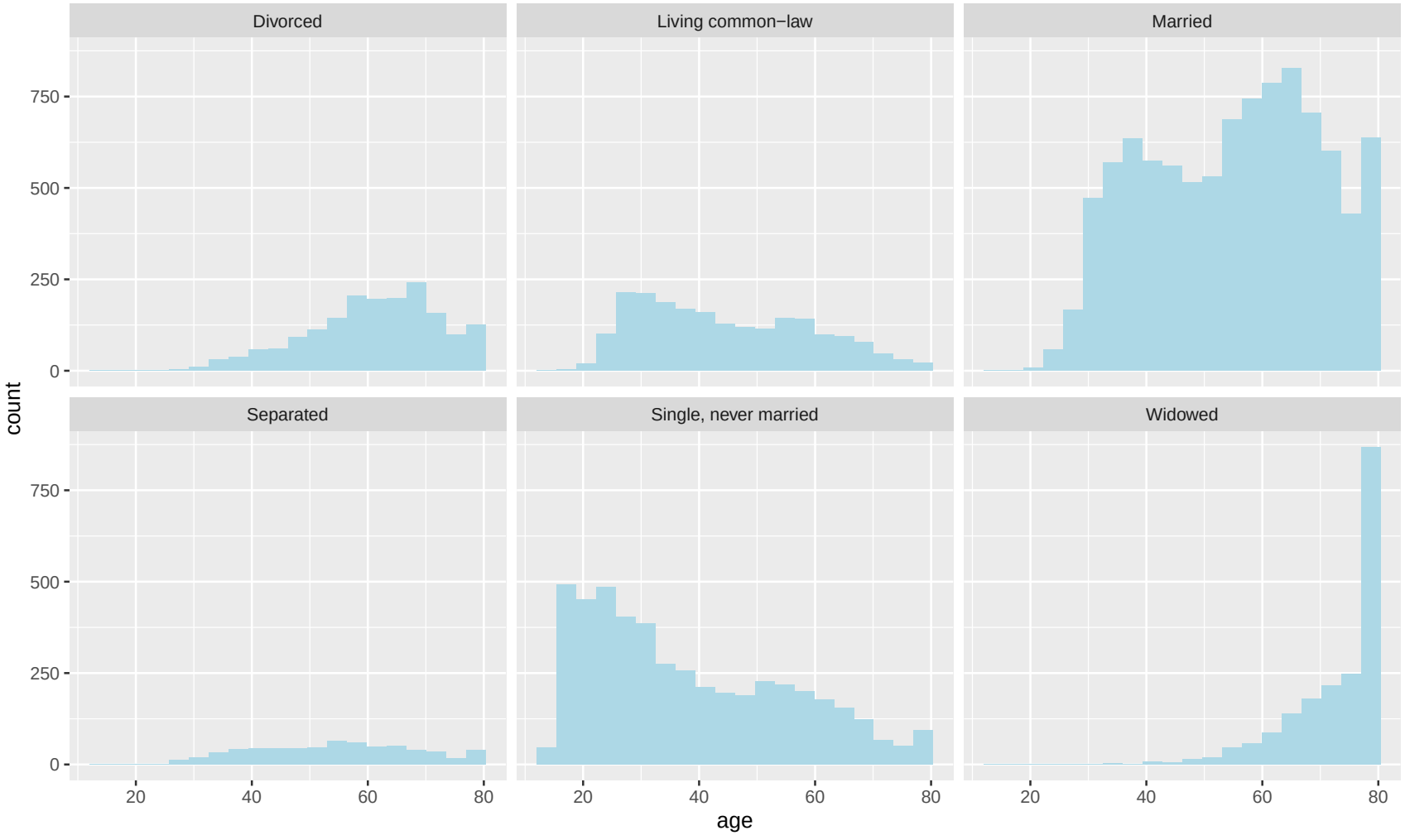
```
lego %>%
  summarize(mean_blocks = mean(number))
```

```
## # A tibble: 1 x 1
##   mean_blocks
##         <dbl>
## 1        3.91
```

Will leave it as an exercise to find median, mode, standard deviation, minimum and maximum. (Hint: for the mode, use `arrange`)

Preview: graphing

```
ggplot(data = gss %>% filter(marital_status!= "NA"), aes(age)) +
  geom_histogram(position = "dodge", fill = "lightblue", bins = 20) +
  facet_wrap(~marital_status)
```



Why you should learn R

- Free, open, reproducible, large community
- Used in industry
- Relatively easier to implement powerful stat methods
- Once you get the hang of it, can use to make
  - slides
  - websites (e.g. mine)
  - interactive applications (e.g. here)

This week's lab

You should hopefully have R and RStudio already installed. If not, this is the first step!

- Learn how to make an RMarkdown file and execute code
- Install and load `tidyverse` package
- Read in GSS
- Practice important functions

Where to get help

- Intro to R:
  - R4DS is the most relevant textbook for learning "tidyverse" R
  - Telling stories with data: https://www.tellingstorieswithdata.com/01-03-r_essentials.html
- Lab time and office hours