



Data Science for Social Scientists:
An applied course using IPUMS data

Developed by:

Daniel E. Ehrlich, IPUMS, University of Minnesota
Anna Tremblay, Dept of Soc, Anth, & CJ, Clemson University
Current Version compiled: 2022-11-17



Contents

Hello Anthropologists!!	7
Preface	9
What is IPUMS	9
Why make this course	10
Course Description	11
Syllabus - Overview	13
Detailed Syllabus	14
DEV NOTES	19
1 Unit 1: The Basics	21
2 Lesson 1:	23
2.1 WHAT IS DATA	23
2.2 COLLECTING DATA	24
2.3 POPULATIONS AND SAMPLING	24
AAA “In-Class” Results	27
Access the Data	27
Summarize Data	28
Visualizing Data	30

3 Lesson 2: Intro to R, data types, data structures	33
3.1 INTRO TO R	33
4 Lesson 3: Distributions	35
4.1 Data Distributions	35
4.2 Week 3: Statisitcal testing of simple data sets	40
4.3 Week 4: Relationships between variables in simple data sets . . .	40
4.4 Week 5:	40
4.5 Intro to R/RStudio	41
4.6 Reading Data / Distributions	41
5 Week 2: Collecting Data	43
Load the Data	43
Access the Data	45
Summarize Data	47
Continuous Data	47
All in summary()	47
Mode	48
Visualizing Data	49
Continuous Data - Boxplots	49
Continuous Data - Histograms	50
Categorical Data	51
6 Intro/Setup	53
7 Engage	55
8 Explore	57
9 Explain	59
10 Elaborate	61

<i>CONTENTS</i>	5
11 Evaluate/Exercises	63
12 Intro/Setup	65
13 Engage	67
14 Explore	69
15 Explain	71
16 Elaborate	73
17 Evaluate/Exercises	75
18 IPUMS	77
18.1 Week 6 Introduction to IPUMS	77
18.2 Week 7 Exploratory analysis	77
18.3 Week 8 Hypothesis Testing	85
18.4 Week 9 Statistical Inference	85
18.5 Week 10 (TBD)	85
18.6 Intro to IPUMS website	85
18.7 math example	86
19 Unit 3: Independent Research	87
19.1 Week 11: Students develop research Question	88
19.2 Week 12: Students find relevant variables from IPUMS	88
19.3 Week 13: Students test and evaluate results	88
19.4 Week 14: Students prepare presentations of results	88
19.5 Week 15: Students present work (slides, poster, podium, etc)	88
19.6 Example one	89
19.7 Example two	89
20 Example RMD code	91
20.1 Core	91
20.2 Tips	92
20.3 Syntax	92

21 Intro/Setup	95
22 Engage	97
23 Explore	99
24 Explain	101
25 Elaborate	103
26 Evaluate/Exercises	105

Hello Anthropologists!!

Getting Started

In order to use this textbook, you will need to:

* set up an IPUMS * download and install RStudio + This link also contains instructions and links to download R from CRAN + Be sure you download the appropriate file for your Mac or PC

Preface

An applied methods class for social scientists that uses real-world IPUMS data.
This course is:

Open-source and customizable -
All materials available on Github

Made with open-source tools -
R, RStudio, bookdown

Driven by ^(nearly) open-source data -
Harmonized across time and space: IPUMS

What is IPUMS

IPUMS started as a project to digitize the historical records of the US census. It has expanded to include 9 data collections, which are united in their methods and principles of making social science research easier. IPUMS data consists of individual-level census and survey data from more than 100 countries around the world. Notably:

- IPUMS **harmonizes** these data - ensuring consistently coded values across time and space.
- IPUMS provides harmonized **GIS Shapefiles** for most census and survey data.
- IPUMS provides extensive **metadata**, including:
 - Original questionnaire text.
 - Universe definition and comparability statements.
 - Alerts about notable changes in variable definition, universe, or coding schema

IPUMS data is free to use for education and research purposes. Researchers need only to register with an email address and brief project description. Nothing too formal - we're just trying to understand what kinds of questions researchers are interested in. For educators, we have additional resources to facilitate set up of classroom accounts - making it easy to get your students registered and share IPUMS data with them.

Why make this course

In a world where information and data are increasingly accessible, it is of utmost importance for individuals to understand data science and the interpretation of data. We believe that education should be easily accessible and teaching resources should be freely available to aid in this endeavor. While we (DEE) may be slightly biased, we think IPUMS is a fantastic resource for both **Education** and **Research**. Real-world example datasets provide the bulk of the content for this course, providing an applied context we hope students (and instructors) will find engaging. We also know many instructors may be teaching across multiple disciplines, in large departments, or be the only “data person” at their institution. We think IPUMS data is useful to virtually any social science field. We provide some example lessons, and encourage instructors to develop their own, using our `lesson_template.Rmd` to tailor this course to their subject or interest.

Course Description

This course is broken down into 3, 5-week units. Unit 1 focuses on familiarizing yourself with R and the IPUMS dataset. In Unit 2, each week will showcase a method/analysis using preselected variables. In class, students will walk through a given problem set and produce a lab report by the end of class. In Unit 3, students will work towards answering a research question that they pose, creating a research paper with literature review, data analysis, conclusion, and data outputs.

Course Aims

Provide students with relevant, hands on, methodological training in data literacy and visualization.

Learning Outcomes

After this course, students will be able to:

- Understand the depth of the IPUMS database and the variables it has to offer
- Compose R code to analyze the IPUMS data
- Produce visually pleasing data outputs in R
- Synthesize the information in a written report
- Present the analysis in a poster format for other students

Guiding Principles

- Phenomenon-based learning
 - try to start the class with a **question** or **problem**
 - *why* does the data look the way it does
 - structure class so students work towards solving the problem

- Relevant examples
 - Drawn from multiple disciplines (eg, economics, demography)
 - Can be added as modular examples/exercises

Syllabus - Overview

This syllabus is initially envisioned as 3 5-week sections. However, compilation and content are intended to be modular with templates for instructors to include their own specialties.

The basic structure of this course is:

Unit 1 (Weeks 1-5): Understanding and Testing Data

- Students use simple datasets bundled with the course or provided by the instructor.
- Simplified data to illustrate trends.
 - EG: plotting continuous variable (AGE); Table of categorical variable (SEX); Crosstabs

Unit 2 (Weeks 6-10): Finding Data and Asking Questions

- Students begin to analyze real world, IPUMS, datasets, provided by course/instructor.
- Students begin to model real world phenomena
 - EG: $SEX \sim EDUATTAIN$; $SEX \sim EDATTAIN + EMPSTAT$
- Students learn to perform exploratory analysis, hypothesis testing, and statistical inference.
- Students learn to navigate IPUMS website, and find relevant data to their research interest.

Unit 3 (Weeks 11-15): Discussing Data and Student Research

- Students develop a research question to be answered with IPUMS data.
 - Students are encouraged to fit it to their interests/major/discipline.
- Course time should be devoted to individual/small-group research.
- Instructor/class present on recent research.
 - Instructor models constructive / scholarly criticism.
 - Encourage students to critique published work - responsibly.

Detailed Syllabus

Unit 1 Understanding and Testing Data

Students become gain familiarity and comfortability navigating RStudio, coding in R and performing simple data manipulation and visualization exercises. Datasets in this section consist of real-world (or synthetic) data, but the focus is on understanding data types (EG: using Age as a continuous variable; sex, education, employment as categorical; etc). Instructors should acknowledge these as **educational** datasets and make explicit trends found within these data are devoid of context, and must be taken with a (rather large) grain of salt, if at all.

By the end of Unit 1, students will be able to:

- Download R and RStudio
- Read data into R and
- Write (save) data out of R
- Summarize data visually
 - Using **base R**
 - Using **ggplot** (tidyverse)
- Summarize data in tables
 - Using base R
 - Using **gttable** / **tidyverse**
- Formally state and test assumptions of data
 - *EG*: t-test, anova, correlations, regression

By the end of Unit 1, students will understand

- Main types of data
 - *EG*: logical, numeric, character, etc
 - R specic vs general terms
- How to create and describe various data distributions
 - *EG*: normal, poisson, normal-skewed, etc
- Know which types statistical tests are appropriate for a given set of data.

Week 1: Intro to R, data types, data structures

Week 2: Plotting Data, Distributions

Week 3: Statistcal testing of simple data sets

Week 4: Correlation and Relationships of simple data sets

Week 5: (TBD)

Unit 2 Finding Data and Asking Questions (Using IPUMS Data)

Here we demonstrate two **different** approaches to conducting research. Students become familiar writing up short lab reports detailing their findings. For Section 18.2, we/instructor provides students with simple datasets from IPUMS (or other real-world data). Students will learn exploratory data analysis techniques and how to create lab reports to summarize key findings.

For unit 18.3, students will learn to develop their own simple research questions or social-science hypotheses. They will seek out data to answer these questions, learning to navigate ipums.org, and create **data extracts**, as well as hypothesis-testing statistical methods. Again, lab reports to summarize findings.

0.0.0.1 Week 6: Intro to IPUMS

Week 7: Exploratory analysis

If you've just collected a survey, or other raw data, you may not know what you're looking for. This is perfectly ok but goes against *the scientific method* most people learned in grade school.

This unit begins by presenting data/distributions and asking students to begin interpreting the data . visual exploration is encouraged and basic of data manipulation are taught * *EG*: how to subset data, how to reshape data, how to re-code data, how to convert from one **data type** to another.

Example lab exercise:

Students given a data set (xls, csv, etc) * load data, perform manipulations, basic summaries + cross tabs + group means by a covariate * inspect data visually + *DESCRIBE* the distribution - is it normal? significant? * *FIND* aquestion in the spread of the data + how can you test this (maybe small group work) * write up/ present results + think on confounding factors / biases

Week 8: Hypothesis Testing

If, on the other hand you have an a pre-existing idea you want to test. We can follow the traditional *scientific method*. With a question in mind, the first question is: where to look. What better place than IPUMS!

Begin introducing navigation of web resources - mainly IPUMS international

Students should become comfortable working through lab exercises: * Define a question (or be presented with one) * Download variables from IPUMS (course downloads possible) * Perform a basic analysis (discussed in Unit 1) * Generate a **visual argument** for your analysis + Include explanation/interpretation/reflection on the question at hand, and the data used + Any obvious biases + Any obvious confounding factors

Week 9: Statistical Inference**Week 10: (TBD)****Unit 3 Discussing Data and Student Research**

Students will select their own research question that can be answered with the IPUMS data set and will spend five weeks conducting a research project complete with data analysis, visualization, and interpretation.

In this section we encourage the instructor to provide ample time for independent student/small-group research. Some class time should be devoted to modeling healthy discussion and critique of methods. Students should learn to discuss not just *how* to answer a research question but *why* they are asking/answering it. What impact does the question/answers have. Is the question relevant/meaningful, and importantly, Is this research question perpetuating racist ideas.

We provide some examples here but encourage instructors (or students) to bring in recent journal/popular articles that do (or do not) apply data science methods well.

Week 11: Students develop research Question**Week 12: Students find relevant variables from IPUMS****Week 13: Students test and evaluate results****Week 14: Students prepare presentations of results****Week 15: Students present work (slides, poster, podium, etc)**

DEV NOTES

TO DO

- Make chapter 1 chapter 2
- Anna Adds chapter con data science intro exclusive of R/IPUMS
- discuss style
 - key terms section for each chapter?
 - key terms in **bold**
 - italics for *emphasis*
 - are we pro-hyphens, or are they pedantic?

MISC IDEAS

- Application forward
- Present research/ analysis/results FIRST, then explain the mathematical principals behind it
- daily/weekly “i’m stuck on...”
 - Students send in questions (night before class) and instructor spends 10-15 mins talking through (or collaboratively working through with class) solutions
 - Alternatively, once a month maybe a longer class covering “common problems asked this month” daily/weekly “recent research”
- pick out a recent article with good visualization (or bad) and spend 5-10 mins discussing what makes it good (or bad)
 - Encourage students to find articles for extra credit

Documentation This function grabs any packages in your project and adds them to a local list that can be referenced using `R-pacakgename * NOTE` in practice, that needs to be wrapped in markdown syntax, eg: `[@R-bookdown] *` See help files for more info - might be able to create/add a `citation` file

Chapter 1

Unit 1: The Basics

Chapter 2

Lesson 1:

2.1 WHAT IS DATA

2.1.1 POV:

In a social science class, your teacher tells you that the CDC reports average male height in the United States to be 69 inches or 5ft 9in. You have been using online dating sites and noticed that the men you match with all report that they are 6ft or over. You become curious if this is a bias in reporting, or if the area where you live and attend college has significantly taller men. To test your theory, you want to collect data on height from individuals in your data science class to test if males are truly taller on campus than the country average.

Source: <https://www.cdc.gov/nchs/fastats/body-measurements.htm>**

Data is defined as “facts and statistics collected together for reference or analysis.”¹ As seen in Figure 1.1, there are two types of data: quantitative and qualitative. **Quantitative data** are able to be expressed in numerical format and are countable. These data are either discrete or continuous where **discrete data** uses numeric bins. For example, we use our age as discrete quantitative data, we round our age to the previous year (eg., 20, 21, 22). **Continuous data** does not use bins, but rather includes all of the fractions between two whole numbers. An example could be most physical measures like height, weight, the speed at which an individual runs.

¹This is from the internet and needs to be our words

Qualitative data describes characteristics or categories and can be broken down into two categories, nominal or ordinal. **Nominal data** has no inherent ordering but it can be categorized. Examples include country or origin, gender, hair color, race, etc. **Ordinal data** can both be categorized and orders (e.g., first, second, and third place in a race).

Going back to our hypothesis of male height on campus, heights are continuous, qualitative data. It is difficult for people to report their specific height and you assume that most individuals will report it rounded to the closest inch. This makes the data you will actually use, discrete quantitative data.

2.2 COLLECTING DATA

The first step to answering a research question is to collect your data. Broadly, data comes in two forms, primary and secondary. (Fig 1.2) **Primary data** is data that is collected directly by the researcher. Surveys, observations, experimentation, questionnaires, and interviews are all examples of primary data. **Secondary data** is collected from published or unpublished literature. It is collected by different researchers and compiled for use by a second scientist. This type of data includes data found in published articles, books, journals, biographies, and government records like the US Census.

Once compiled, you now have a data set which is comprised of observations and variables. An **observation** is all of the measures taken for one person or item. A **variable** is what is being measured.

The US CDC data is secondary, but you are collecting height data yourself in class as a comparison. The survey or questionnaire you use on your classmates is primary data. Each individual is an observation and the variable of interest is height.

2.3 POPULATIONS AND SAMPLING

Random Sampling: It is a sampling method in which all the items have an equal chance of being selected and the individuals who are selected are just like the ones who are not selected

Stratified Random Sampling: It is a process to gather data by separating the actual population into the distinct subset or strata, and then choosing simple random samples from each stratum. Your research question is about the height of all males at your college, but recording height data for each individual would be very difficult and time consuming. You instead decide to use a sample of males

in your data science class. This is a random sample as each male individual has an equally likely chance of being samples (that is, unless a prerequisite exists).

Sampling strategy can lead to **bias**

If you had chosen a different sample, like the men's basketball team, your results would have been biased.

AAA “In-Class” Results

Load the Data

Thanks for filling out our simple survey! Below, we'll tabulate results. Make sure to update the `data_template.xlsx` file located in: `inst/unit1_data`

```
dir_path <- file.path("inst","unit1_data")
survey_path <- file.path(dir_path, "data_template.xlsx")

data <- readxl::read_excel(survey_path)
```

Access the Data

What is `data`?? Below we call the `class()` function on `data` and see that it has 3 classes: `tbl_df`, `tbl`, `data.frame`

The first two classes, `tbl_df`, `tbl` indicate it is a special kind of table, in the `tibble` format. In general, you can interact with these like a `matrix` or `data.frame` but they have additional features.

```
class(data)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

We can call `colnames()` on `data`, like a regular `data.frame` or `matrix`. Or we can take advantage of the `tibble` structure and use the `glimpse()` function which provides a succinct summary of your data.

```
colnames(data)
```

```
## [1] "individual"    "Birth_Month"   "Height_inches"
```

```
tibble::glimpse(data)
```

```
## Rows: 52
## Columns: 3
## $ individual    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
## $ Birth_Month   <chr> "January", "September", "march", "april", NA, NA, NA, NA~
## $ Height_inches <dbl> 70, 64, 72, 61, NA, NA, NA, NA, NA, NA, NA, NA, ~
```

Summarize Data

Continuous Data

For continuous data, we often want to summarize our data by describing the **mean**, **median**, and/or **range**. **Mean** and **median** describe the *central tendency* of the data, while **range** describes the full extent of the data, as seen below.

NOTE: if NA are present in the data, be sure to use the `na.rm=TRUE` flag for these operations.

```
mean(data$Height_inches, na.rm = T)
```

```
## [1] 66.75
```

```
median(data$Height_inches, na.rm = T)
```

```
## [1] 67
```

```
range(data$Height_inches, na.rm = T)
```

```
## [1] 61 72
```

All in summary()

Mean, **median**, and **range** will all be reported by calling `summary()` on a numeric vector, such as `Height_inches`. In addition, the lower and upper quartiles will be reported, along with the number of NA responses.

NOTE: `summary()` does NOT require special handling for NA values, in fact - it expects them!

```
summary(data$Height_inches)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##    61.00  63.25   67.00   66.75  70.50   72.00      48
```

Mode Code

One common measure of data reported is the mode, or most frequently occurring value. For whatever reason, this is not a default function in R, but we can easily write our own function like so:

```
my_mode <- function(x){
  tt <- table(x) ## find frequencies
  tt <- tt[order(tt, decreasing = TRUE)] ## resort based on freq

  ## check number of modes
  max <- max(tt)
  n_max <- sum(tt==max)

  if(n_max > 1 ){
    warning("More than one mode detected")
    return(tt[tt==max])
  } else {
    ## return only the first value
    return(tt[1]) ## return whatever the highrst frequency is
  }
}
```

Mode Results

With this function, it's easy to find the **mode**

```
my_mode(data$Height_inches)
```

```
## Warning in my_mode(data$Height_inches): More than one mode detected

## x
## 61 64 70 72
##  1  1  1  1
```

Visualizing Data

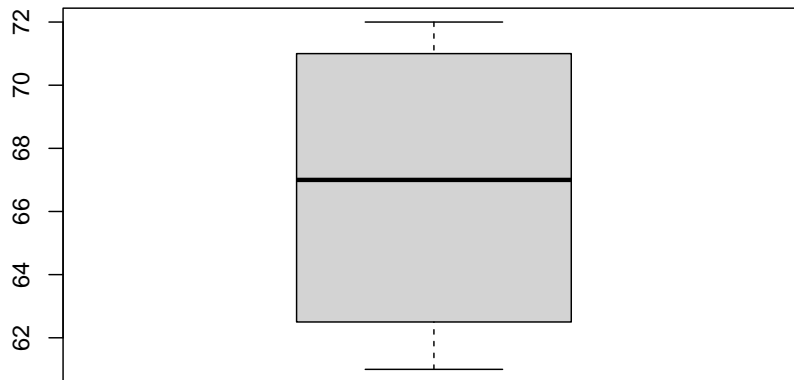
The above summaries describe data with numbers, but we can also describe data visually.

Continuous Data - Boxplots

Univariate continuous data, like height, can be visualized using a box and whisker plot, which shows many of the components of summary:

- the **median** is the black bar in the middle
- the **quartiles** (25th and 75th percentiles) are represented by the extents of the boxes
- The **range** is shown by the whiskers, with outliers shown individually, if needed.

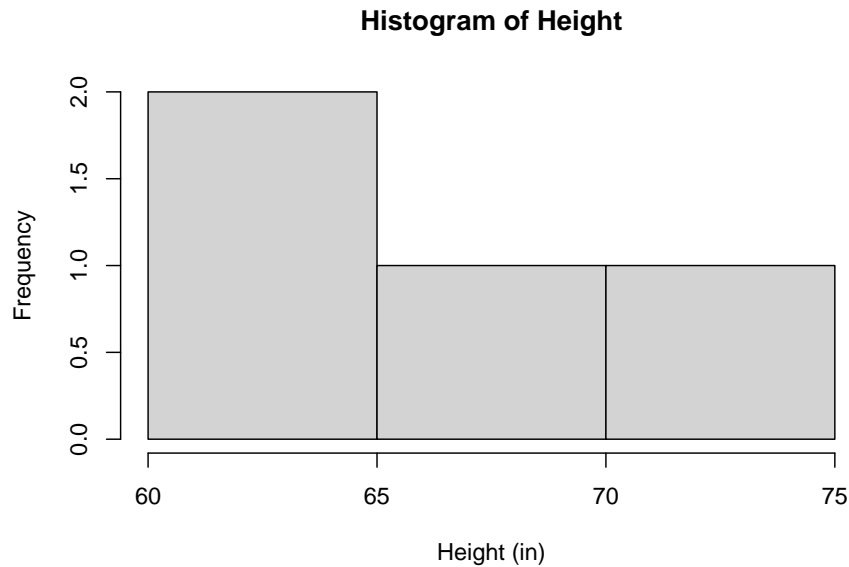
```
boxplot(data$Height_inches)
```



Continuous Data - Histograms

Continuous data, can also be broken into **bins** and plotted as a **histogram**. The `hist()` function will attempt to find the optimum number of bins for you, but you can specify a different number with the `breaks` argument.

```
hist(data$Height_inches, main = "Histogram of Height", xlab = "Height (in)")
```



Categorical Data

Categorical data is already in discrete units. In general with categorical data, we want to count the **frequency** of unique values. There are many ways to do this, but one of the easiest is the `table()` function. Saving the results of the table to an object, `birth_freq`, allows you to save and print the results at any time.

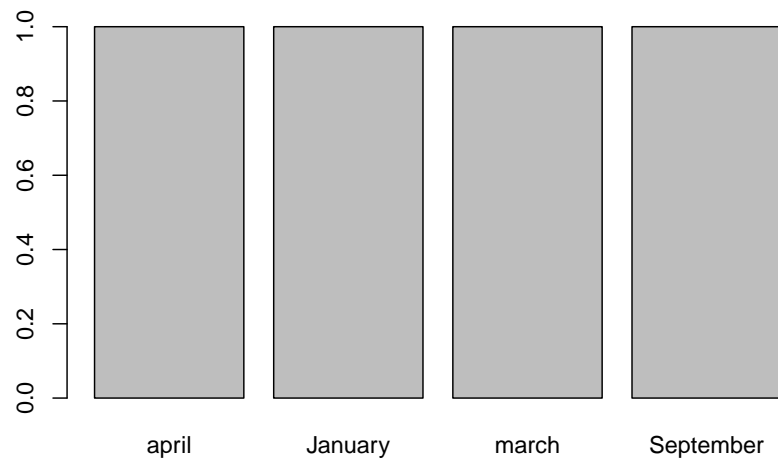
```
birth_freq <- table(data$Birth_Month)
```

```
birth_freq
```

```
##  
##    april  January    march  September  
##      1      1      1      1
```

We can also visualize our tabulated results using a **barplot** as below.

```
barplot(birth_freq)
```



Chapter 3

Lesson 2: Intro to R, data types, data structures

3.1 INTRO TO R

3.1.1 What is R?

R is a statistical programming language. Using R, researchers can write instructions (code) that performs just about any data manipulation, analization, or visualization.

Some highlights from the **Comprehensive R Archive Network** (cran):

“R is a language and environment for statistical computing and graphics...

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is > highly extensible.

The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to > participation in that activity.

One of R’s strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

R is available as Free Software under the terms of the Free Software Foundation’s GNU General Public License in source code form. It

compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS.”

3.1.2 Why use R?

3.1.3 Data Structures

Chapter 4

Lesson 3: Distributions

4.1 Data Distributions

4.1.1 Normal Distributions

First we'll generate a normal distribution with the `rnorm()` function. This takes 3 arguments: `n`, `mean`, `sd`, which you can see filled in below. While we could print out a list of all these values, it's not easy to *understand* a list of numbers

```
normal_dist <- rnorm(n = 100, ## 100 samples
                     mean = 10, ## with a mean of 10
                     sd = 1 ## and a standard deviation of 1
                     )
```

```
normal_dist
```

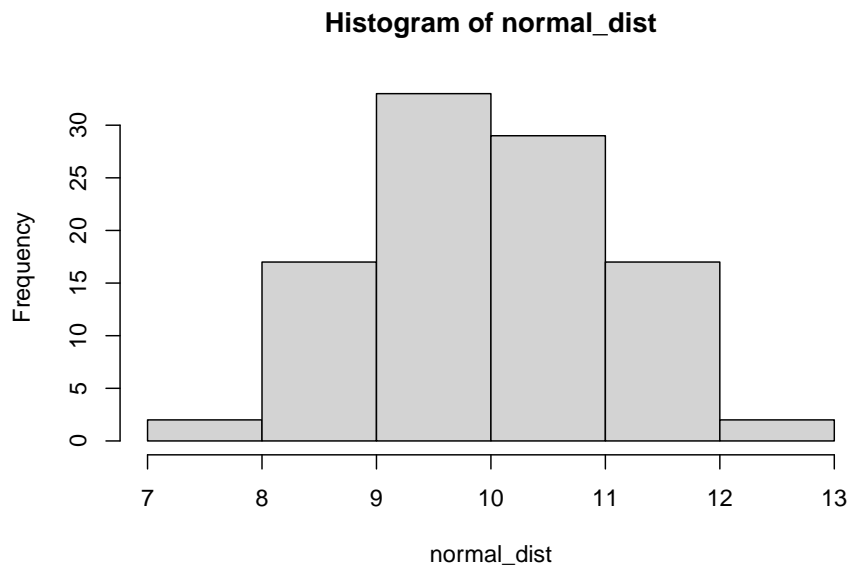
```
##      [1]  8.283281  9.986951 11.126084 10.059244 10.954517  8.551064  9.936085
##      [8]  9.663171  9.476563 10.972992  9.078060  9.955959  8.966456  9.714395
##     [15] 10.617533  8.436985 10.074293 10.269555 11.805408 10.162156 10.229569
##     [22]  9.927926  8.379997 10.005866  7.660421 12.794830 11.987923 11.198557
##     [29] 11.886719  9.601832 11.289638  9.712173  7.967336  9.554024  9.945227
##     [36] 11.025337 11.496550  9.324674 10.592863  9.448456  9.248342 11.007543
##     [43] 11.502264 11.279822 10.146802  9.054626  9.745818 10.299232 11.033905
##     [50] 10.512997  9.295279  8.727555 10.719264  8.758891  9.876770  9.333231
##     [57] 11.479955 10.589533  9.407144 10.427596  8.557797  9.335073  8.728427
##     [64] 10.118108  9.871303  8.955949  9.307462 10.551194 10.133350 10.989681
##     [71] 11.195571 10.159086  8.624714  9.008340  9.189561 10.441537 11.122525
##     [78] 10.199072 10.389668  9.885721  9.980859 11.205213  9.400495  8.570012
```

```
## [85]  9.446789  9.727927 10.147285  8.429662  9.912394 10.442583 12.242780
## [92]  8.932087  8.967939 10.970121 10.095165 10.230579 11.343148  8.130177
## [99]  8.909377  9.699220
```

Another better way to look at data would be to **visualize** or **plot** it. One way to do that is with a **histogram**, which groups **continuous values** into **bins**, then plots the **frequency** for each bin.

In R, we use the `hist()` function to plot a histogram of data. We can (try to) control the number of bins with the `breaks` argument, but note that it doesn't always match up. The `hist()` function will adjust based on the distribution of the data.

```
hist(normal_dist,breaks = 5)
```



Another way to visualize this would be with a d

4.1.2 What *is* normal?

4.1.2.1 Quantitative summaries

5num summary * Min, 25th percentile, median, 75th percentile, Max

```
tab_normal_dist <- summary(normal_dist)
```

We can print the table in R by calling its name.

```
tab_normal_dist
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  7.660   9.304   9.968   9.981  10.599  12.795
```

Mean, standard deviation

4.1.2.2 Meaningful Comparisons

How to compare apples to oranges? Standardize the units / standardize the data

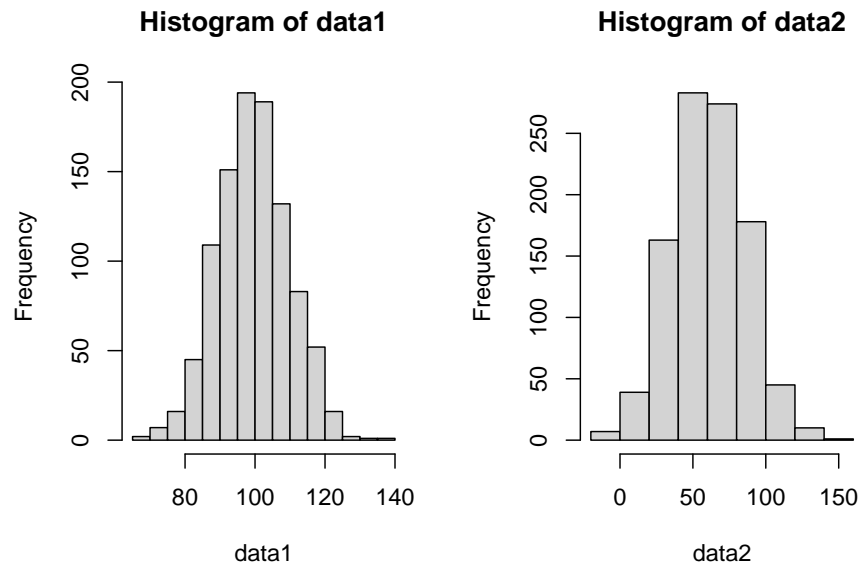
```
data1 <- rnorm(n=1000,
              mean = 100,
              sd = 10)

data2 <- rnorm(n=1000,
              mean = 60,
              sd = 25)
```

Are these the same distribution?

Any issues??

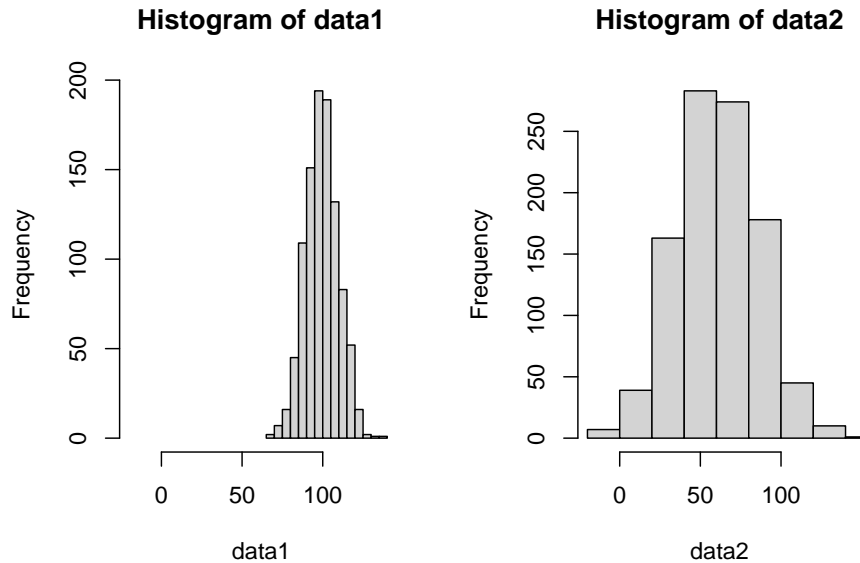
```
layout(matrix(1:2, ncol = 2))
hist(data1)
hist(data2)
```



```
total_range <- range(data1, data2)
```

Are they the same?

```
layout(matrix(1:2, ncol = 2))  
hist(data1, xlim = total_range)  
hist(data2, xlim = total_range)
```



Numerically / tabularly

Often times its important to tables of **summary statistics**

```
norm_comp_tab <- rbind(summary(data1),
                        summary(data2))
```

```
norm_comp_tab
```

```
##           Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## [1,]  68.56394 92.42487 99.29497 99.48844 106.22977 136.5234
## [2,] -19.41310 43.18450 60.50535 61.10462  79.04845 143.2749
```

Making the table a little nicer. Also an example of **conditional programming**.

```
rownames(norm_comp_tab) ## they're null
```

```
## NULL
```

```
if(is.null(rownames(norm_comp_tab))){
  rownames(norm_comp_tab) <- c("data1", "data2")
}
```

When working with **Rmarkdown** we can take advantage of **knitr** and **pandoc** to nice looking tables even easier.

```
knitr::kable(norm_comp_tab)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
data1	68.56394	92.42487	99.29497	99.48844	106.22977	136.5234
data2	-19.41310	43.18450	60.50535	61.10462	79.04845	143.2749

How transform the data

Simple transformation (multiply all values by 100) * to convert units * other examples?

Complex transformations * log-transformation (*DEE: not a fan*) * z-scores (*DEE: a better option*)

Why transform the data? * Real world applications? * Is it always appropriate to transform data?

4.1.3 Skews

What to do if the data are **not** normal?

4.2 Week 3: Statisitcal testing of simple data sets

4.2.1 t-tests, ANOVA, chi2

4.3 Week 4: Relationships between variables in simple data sets

4.3.1 Correlation, Linear Regression

4.3.1.1 Simple LM

4.3.1.2 Complex LM

4.3.2 Genearlized Linear Model

4.4 Week 5:

For now, I have 3 main chapters for each of the main sections: * Basics of data science / R 1 * Applications/critiques using IPUMS data 18 * Student-driven projects 19

Each of these **Chapters** contains multiple sections. We'll likely want to break these sections out into their own `.Rmd` files as they get fleshed out. For now, I'll try to keep the abundance of files limited.

NOTE: As these actually get filled out, we will probably want to insert different **parts** to the book (EG, the content of Unit 1 is covered in **Part I**). * Declare parts with `# (PART) Part I {-}` immediately before the first chapter `#` it contains.

Topics to include: * What is data? * Everything can be data * How do we interpret data * Tables * Plots * Univariate distributions * What can they tell us * Multi-modality in distributions * Categorical vs continuous data * Don't need to get ahead of this yet * Add in a grouping category - multi state/multi-national dataset * Ttest / anova

Type of Data: Age distributions Specifically generate a dataset with old/young folks over-represented to highlight a bimodal distribution

Start with single state/country Add a second state/country to demo ttest Add more to demo anova

Alternatively, income by education level - may be more interesting/relevant to college students (or depressing)

4.5 Intro to R/RStudio

4.6 Reading Data / Distributions

4.6.1 What *is* a normal distribution

4.6.1.1 How normal is it?

show increasingly unclear examples of normal vs not

introduce tests of normality

4.6.1.2 Measuring normality - single sample

reinforce [concept of statistical] **normality**

is a value from a sample? - one way ttest something about tails

4.6.1.3 comparing normality - two samples

standard / two-way t test

4.6.1.4 comparing more than two - ANOVA

#Glossary Data Quantitative Qualitative Discrete Continuous Nominal Ordinal

Chapter 5

Week 2: Collecting Data

Load the Data

Thanks for filling out our simple survey! Below, we'll tabulate results. Make sure to update the `data_template.xlsx` file located in: `inst/unit1_data`

```
dir_path <- file.path("inst","unit1_data")
survey_path <- file.path(dir_path, "data_template.xlsx")

data <- readxl::read_excel(survey_path)
```


Access the Data

What is `data`?? Below we call the `class()` function on `data` and see that it has 3 classes: `tbl_df`, `tbl`, `data.frame`

The first two classes, `tbl_df`, `tbl` indicate it is a special kind of table, in the `tibble` format. In general, you can interact with these like a `matrix` or `data.frame` but they have additional features.

```
class(data)
```

```
## [1] "tbl_df"      "tbl"        "data.frame"
```

We can call `colnames()` on `data`, like a regular `data.frame` or `matrix`. Or we can take advantage of the `tibble` structure and use the `glimpse()` function which provides a succinct summary of your data.

```
colnames(data)
```

```
## [1] "individual"    "Birth_Month"   "Height_inches"
```

```
tibble::glimpse(data)
```

```
## Rows: 52
## Columns: 3
## $ individual    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
## $ Birth_Month   <chr> "January", "September", "march", "april", NA, NA, NA, NA~
## $ Height_inches <dbl> 70, 64, 72, 61, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
```


Summarize Data

Continuous Data

For continuous data, we often want to summarize our data by describing the **mean**, **median**, and/or **range**. **Mean** and **median** describe the *central tendency* of the data, while **range** describes the full extent of the data, as seen below.

NOTE: if NA are present in the data, be sure to use the `na.rm=TRUE` flag for these operations.

```
mean(data$Height_inches, na.rm = T)
```

```
## [1] 66.75
```

```
median(data$Height_inches, na.rm = T)
```

```
## [1] 67
```

```
range(data$Height_inches, na.rm = T)
```

```
## [1] 61 72
```

All in summary()

Mean, **median**, and **range** will all be reported by calling `summary()` on a numeric vector, such as `Height_inches`. In addition, the lower and upper quartiles will be reported, along with the number of NA responses.

NOTE: `summary()` does NOT require special handling for NA values, in fact - it expects them!

```
summary(data$Height_inches)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's
##      61.00   63.25   67.00   66.75   70.50   72.00    48
```

Mode

One common measure of data reported is the mode, or most frequently occurring value. For whatever reason, this is not a default function in R, but we can easily write our own function like so:

```
my_mode <- function(x){
  tt <- table(x) ## find frequencies
  tt <- tt[order(tt, decreasing = TRUE)] ## resort based on freq

  ## check number of modes
  max <- max(tt)
  n_max <- sum(tt==max)

  if(n_max > 1 ){
    warning("More than one mode detected")
    return(tt[tt==max])
  } else {
    ## return only the first value
    return(tt[1]) ## return whatever the highrst frequency is
  }
}
```

With this function, it's easy to find the **mode**

```
my_mode(data$Height_inches)
```

```
## Warning in my_mode(data$Height_inches): More than one mode detected
```

```
## x
## 61 64 70 72
##  1  1  1  1
```


Visualizing Data

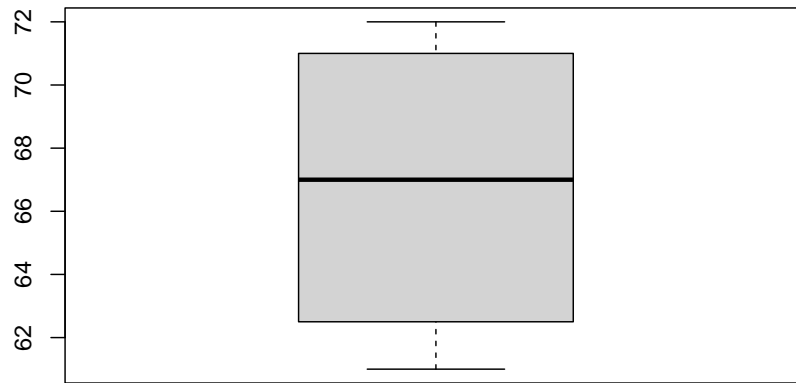
The above summaries describe data with numbers, but we can also describe data visually.

Continuous Data - Boxplots

Univariate continuous data, like height, can be visualized using a box and whisker plot, which shows many of the components of summary:

- the **median** is the black bar in the middle
- the **quartiles** (25th and 75th percentiles) are represented by the extents of the boxes
- The **range** is shown by the whiskers, with outliers shown individually, if needed.

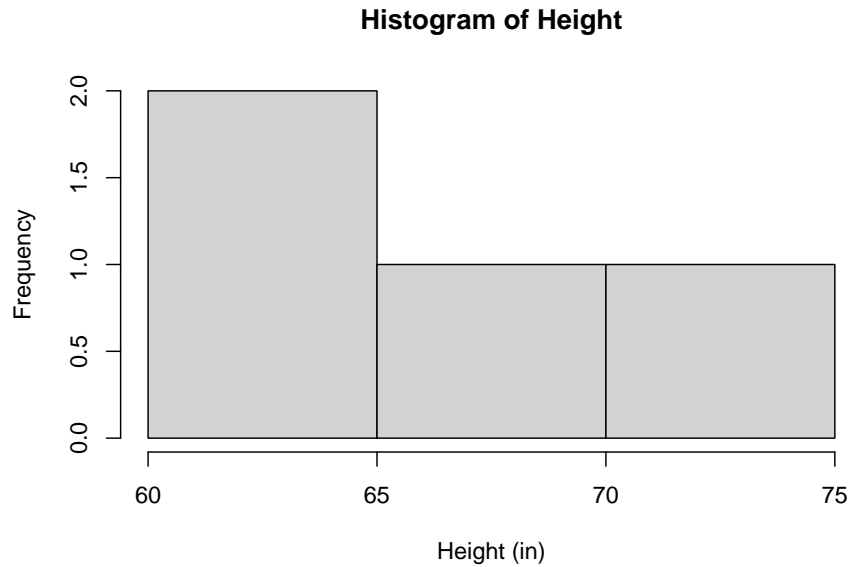
```
boxplot(data$Height_inches)
```



Continuous Data - Histograms

Continuous data, can also be broken into **bins** and plotted as a **histogram**. The `hist()` function will attempt to find the optimum number of bins for you, but you can specify a different number with the `breaks` argument.

```
hist(data$Height_inches, main = "Histogram of Height", xlab = "Height (in)")
```



Categorical Data

Categorical data is already in discrete units. In general with categorical data, we want to count the **frequency** of unique values. There are many ways to do this, but one of the easiest is the `table()` function. Saving the results of the table to an object, `birth_freq`, allows you to save and print the results at any time.

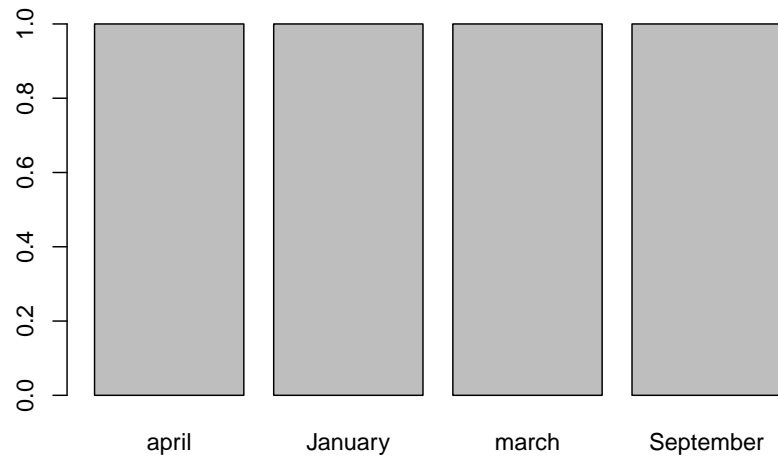
```
birth_freq <- table(data$Birth_Month)
```

```
birth_freq
```

```
##
##      april   January   march September
##         1         1         1         1
```

We can also visualize our tabulated results using a **barplot** as below.

```
barplot(birth_freq)
```



Chapter 6

Intro/Setup

A test edit

NOTE: chapter templates **do not** require a yaml header (most .RMD docs do). Lay out sections using # headers.

Chapter 7

Engage

Brief example of an engaging phenomenon. * Poll students and discuss results + Does the *average* match up with individual experiences? * Real-world research example * Interrogating a commonly held assumption + the average height of humans is 5'5"

Chapter 8

Explore

Explore other examples of this phenomenon.

Build on the first one to show variation. * Other ways to represent the phenomenon/data + tabular + visually * can the example be grouped / broken into subgroups + does the same pattern/phenomenon apply? * Is there a corollary / inverse phenomenon?

Chapter 9

Explain

Discuss/interrogate the pattern of the data * does the shape imply anything *
try to have it student led / guided * with time, students guess at what the data
shows/doesn't show without labels

Chapter 10

Elaborate

Work through a new example, contextualizing the phenomenon using real-world data/examples.

Chapter 11

Evaluate/Exercises

short lab exercises for students to recreate / code in R

Build on past concepts reinforcing basic data manipulations, summaries, etc.

Chapter 12

Intro/Setup

A test edit

more test edits

NOTE: chapter templates **do not** require a yaml header (most .RMD docs do). Lay out sections using # headers.

Chapter 13

Engage

Brief example of an engaging phenomenon. * Poll students and discuss results + Does the *average* match up with individual experiences? * Real-world research example * Interrogating a commonly held assumption + the average height of humans is 5'5"

Chapter 14

Explore

Explore other examples of this phenomenon.

Build on the first one to show variation. * Other ways to represent the phenomenon/data + tabular + visually * can the example be grouped / broken into subgroups + does the same pattern/phenomenon apply? * Is there a corollary / inverse phenomenon?

Chapter 15

Explain

Discuss/interrogate the pattern of the data * does the shape imply anything *
try to have it student led / guided * with time, students guess at what the data
shows/doesn't show without labels

Chapter 16

Elaborate

Work through a new example, contextualizing the phenomenon using real-world data/examples.

Chapter 17

Evaluate/Exercises

short lab exercises for students to recreate / code in R

Build on past concepts reinforcing basic data manipulations, summaries, etc.

Chapter 18

IPUMS

18.1 Week 6 Introduction to IPUMS

18.2 Week 7 Exploratory analysis

If you've just collected a survey, or other raw data, you may not know what you're looking for. This is perfectly ok but goes against *the scientific method* most people learned in grade school (More on that to follow(*include_link*)).

This unit begins by presenting data/distributions and asking students to begin interpreting the data . visual exploration is encouraged and basic of data manipulation are taught * *EG*: how to subset data, how to reshape data, how to recode data, how to convert from one `data type` to another.

Example lab exercise:

Students given a data set (xls, csv, etc) * load data, perform manipulations, basic summaries + cross tabs + group means by a covariate * inspect data visually + *DESCRIBE* the distribution - is it normal? significant? * *FIND* aquestion in the spread of the data + how can you test this (maybe small group work) * write up/ present results + think on confounding factors / biases

18.2.1 Advanced Exploration - Change Over Time

Here we demonstrate an approach to looking at how Family Structure (inferred from household relationships) has changed over time.

18.2.1.1 Setup / Load Data

Install/update R packages

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

Data extract created online using the datacart system.

```
library(ipumsr)
library(dplyr)

ddi <- read_ipums_ddi("Data/ipumsi_00005.xml")
data <- read_ipums_micro(ddi)
```

18.2.1.1.1 Inspect the Data Using haven labeled values.

```
data$RELATE[1:100]
class(data$RELATE)

data %>% count(RELATE)
data %>% count(SEX)
```

What were those codes ??

```
## need to convert this to an image or something similar; kable table?
ipums_view(ddi)
```

18.2.1.1.2 Visualize A simple plot

```
plot(AGE ~ YEAR, data = data)
```

A fancier plot

```
plot(AGE~YEAR, data = data, type = "n", main = "Age by Sex, over Time, CO")
points(data$YEAR[data$SEX==1]-1, data$AGE[data$SEX==1], pch = 16, col = hsv(.6,.6,.8,.2))
points(data$YEAR[data$SEX==2]+1, data$AGE[data$SEX==2], pch = 16, col = hsv(1,.6,.8,.2))
abline(lm(AGE~YEAR, data = data), col = "green")
```

18.2.1.1.3 Asking (logical) questions Here we demonstrate how setting up logical questions can be used to easily filter/subset data.

```
age_test <- data$AGE > 18

class(age_test)

age_test
```

Logical vectors are stored as **TRUE** or **FALSE**, but can also be evaluated numerically as 1 or 0 respectively. We can therefore `sum()` the number of **TRUE** values and divide by total rows for a proportion.

```
sum(age_test)/nrow(data)
```

18.2.1.1.4 HH vs persons A unique characteristic of census and some survey data is the nested-structure with individuals being grouped into households. Often times it is necessary to choose to work at the hh or person level, and data must be appropriately manipulated to fit that case.

```
hh_total <- length(unique(data$SERIAL))
hh_total
ipums_view(ddi)
```

18.2.1.2 Nuclear Family

First we look at a nuclear family, comprising only parents and their immediate children.

```
library(ipumsr)
library(dplyr)

ddi <- read_ipums_ddi("/pkg/ipums/personal/ehrli097/AABA_2022/Data/ipumsi_00005.xml")
all_data <- read_ipums_micro(ddi)

census_years <- c(1860, 1870, 1880, 1900, 1910, 1960, 1970, 1980, 1990, 2000, 2010)

## subset census only
d2 <- all_data %>% filter(YEAR %in% census_years)

## make a household dataframe
hhs <- d2 %>% distinct(YEAR, SERIAL, .keep_all = TRUE) %>% select(YEAR, SERIAL, GEO1_US)

hhs %>% View()
```

```

hhs <- d2 %>% filter(RELATE ==4) %>%

distinct(YEAR, SERIAL) %>% mutate(extended_test=TRUE) %>% right_join(hhs, by = c("YEAR", "SERIAL"))

hhs <- d2 %>% filter(!RELATE %in% c(1, 2, 3) |
                     (RELATE == 3 &
                      MARST %in% c(2, 3, 4))
                     ) %>%

distinct(YEAR, SERIAL) %>% mutate(nuclear_test = FALSE) %>% right_join(hhs, by = c("YEAR", "SERIAL"))

table(hhs$extended_test, hhs$nuclear_test)

```

```

hhs <- d2 %>% filter(RELATED %in% c(4200, 4210, 4211, 4220, 4500, 4510, 4600)) %>% distinct(YEAR, SERIAL)

hhs <- d2 %>% filter(RELATED %in% c(4100, 4110, 4120, 4130, 4300, 4301, 4302)) %>% distinct(YEAR, SERIAL)

res_tabs <- list(
  "nuclear_test" = hhs %>% group_by(YEAR, nuclear_test, GEO1_US) %>% summarize(.groups = "drop"),
  "extended_test" = hhs %>% group_by(YEAR, extended_test, GEO1_US) %>% summarize(.groups = "drop"),
  "parent_test" = hhs %>% group_by(YEAR, parent_test, GEO1_US) %>% summarize(.groups = "drop"),
  "children_test" = hhs %>% group_by(YEAR, children_test, GEO1_US) %>% summarize(.groups = "drop")
)

```



```

collapsed_results <- res_tabs %>% purrr::map(function(x){
  x <- x %>% group_by(across(names(x)[1:3])) %>% summarize(.groups="drop", n = sum(n))
})

collapsed_results <- lapply(collapsed_results, function(x){
  colnames(x)[2] <- "test"
  colnames(x)[3] <- "state"
  return(x)
})

combined <- collapsed_results %>% purrr::reduce(full_join, by = c("YEAR", "test", "state"))

colnames(combined) <- c("YEAR", "test", "state", "n_nuclear", "n_extended", "n_parent", "n_childre")
combined[is.na(combined)] <- 0

to_plot <- combined %>% group_by(YEAR, state) %>% mutate(n_tot = sum(n_nuclear)) %>% ungroup() %>%

```

18.2.1.2.1 Tabulate results

```

to_plot <- to_plot %>% filter(test==TRUE)

plot(to_plot$YEAR, to_plot$pct$n_nuclear, col = hsv(.4, .6, .8), pch = 16, ylim = c(0,1), xlab = "

```

18.2.1.2.2 Visualize Nuclear Families

18.2.1.3 Extended Family

Next we look at hhs with extended families present. IE, any that contain more relationships than just Parent/Child/Sibling (between children only)

```

to_plot <- to_plot %>% filter(test==TRUE)

glm_hist <- glm(pct$n_extended ~ YEAR, data = to_plot[to_plot$YEAR < 1950,], family = "quasibinomial")

glm_hist_x <- seq(from=1860, to = 1910, length.out = 100)
glm_hist_y <- predict(glm_hist, list(YEAR = glm_hist_x), type = "response")

glm_mod <- glm(pct$n_extended ~ YEAR, data = to_plot[to_plot$YEAR > 1950,], family = "quasibinomial")

glm_mod_x <- seq(from = 1960, to = 2010, length.out = 100)
glm_mod_y <- predict(glm_mod, list(YEAR = glm_mod_x), type = "response")

mods <- list("hist"=list(),
            "mod" = list()
            )
mods_plots <- list("hist"=list(),
                  "mod" =list()
                  )

for(i in names(to_plot$pct)){

  hist_x <- to_plot$YEAR[to_plot$YEAR < 1950]
  mod_x <- to_plot$YEAR[to_plot$YEAR > 1950]

  mods$hist[[i]] <- lm(pct[[i]] ~ YEAR, data = to_plot[to_plot$YEAR < 1950,])

  mods_plots$hist[[i]] <-
    data.frame("x" = hist_x,
              "y" = predict(mods$hist[[i]],
                            list(YEAR =hist_x),
                            type = "response")
              )

  mods$mod[[i]] <- lm(pct[[i]] ~ YEAR, data = to_plot[to_plot$YEAR > 1950,])

  mods_plots$mod[[i]] <-
    data.frame("x" = mod_x,
              "y" = predict(mods$mod[[i]],

```

```

    list(YEAR = mod_x),
    type = "response")
  )
}

```

18.2.1.3.1 Generate models

```

plot(to_plot$YEAR, to_plot$pct$n_extended, col = hsv(.95, .6, .8), pch = 16, ylim = c(0, .25), bg =

lines(glm_hist_x, glm_hist_y, col = hsv(.95, .3, 1), lwd = 2)
lines(glm_mod_x, glm_mod_y, col = hsv(.95, .3, 1), lwd = 2, lty = 2)

points(to_plot$YEAR,
       to_plot$pct$n_extended,
       pch = 23,
       bg = hsv(.95, .6, .8))

```

18.2.1.3.2 Visualize

18.2.1.4 Even more DETAIL - maybe remove

```
ipums_view(ddi)
```

```

hhs <- d2 %>% filter(RELATED %in% c(4200, 4210, 4211, 4220, 4500, 4510, 4600)) %>%
distinct(YEAR, SERIAL) %>% mutate(parent_test=TRUE) %>% right_join(hhs, by = c("YEAR", "SERIAL")
hhs <- d2 %>% filter(RELATED %in% c(4100, 4110, 4120, 4130, 4300, 4301, 4302)) %>% distinct(YEA

```

```

plot(to_plot$YEAR, to_plot$pct$n_extended, col = hsv(.95, .6, .8), pch = 16, ylim = c(0, .25), bg =

```

```

lines(glm_hist_x,glm_hist_y, col = hsv(.95, .3, 1), lwd = 2)
lines(glm_mod_x, glm_mod_y, col = hsv(.95, .3, 1), lwd = 2, lty = 2)

lines(mods_plots$hist$n_parent,col = hsv(.8, .3,1), lwd = 2)

lines(mods_plots$mod$n_parent, col = hsv(.8, .3,1), lwd = 2, lty = 2)

points(to_plot$YEAR,
       to_plot$pct$n_parent,
       pch = 23,
       bg = hsv(.8, .6,.8))

points(to_plot$YEAR,
       to_plot$pct$n_extended,
       pch = 23,
       bg = hsv(.95,.6,.8))

```

18.2.1.4.1 Parents Supporting Parents

```

plot(to_plot$YEAR, to_plot$pct$n_extended, col = hsv(.95, .6,.8), pch = 16, ylim =c(0,

lines(glm_hist_x,glm_hist_y, col = hsv(.95, .3, 1), lwd = 2)
lines(glm_mod_x, glm_mod_y, col = hsv(.95, .3, 1), lwd = 2, lty = 2)

lines(mods_plots$hist$n_children, col = hsv(.55,.3,1), lwd = 2)

lines(mods_plots$mod$n_children, col = hsv(.55,.3,1), lwd = 2, lty = 2)

points(to_plot$YEAR,
       to_plot$pct$n_children,
       pch = 23,
       bg = hsv(.55,.6,.8))

points(to_plot$YEAR,
       to_plot$pct$n_extended,

```

```
pch = 23,
bg = hsv(.95, .6, .8))
```

18.2.1.4.2 Parents Supporting (extended) children

18.3 Week 8 Hypothesis Testing

If, on the other hand you have an a pre-existing idea you want to test. We can follow the traditional *scientific method*. With a question in mind, the first question is: where to look. What better place than IPUMS!

Begin introducing navigation of web resources - mainly IPUMS international

Students should become comfortable working through lab exercises: * Define a question (or be presented with one) * Download variables from IPUMS (course downloads possible) * Perform a basic analysis (discussed in Unit 1) * Generate a **visual argument** for your analysis + Include explanation/interpretation/reflection on the question at hand, and the data used + Any obvious biases + Any obvious confounding factors

18.4 Week 9 Statistical Inference

18.5 Week 10 (TBD)

Some text to break up the sub-section headers

18.6 Intro to IPUMS website

18.6.1 background on ipums

18.6.2 navigating website

Find certain (very common) variables to answer (common) social science questions.

We describe our methods in this chapter.

Math can be added in body using usual syntax as follows. This may be useful, particularly for explaining the math side of things.

18.7 math example

p is unknown but expected to be around $1/3$. Standard error will be approximated

$$SE = \sqrt{\left(\frac{p(1-p)}{n}\right)} \approx \sqrt{\frac{1/3(1-1/3)}{300}} = 0.027$$

You can also use math in footnotes like this¹. Footnotes are helpful because they re-link to where you left off.

We will approximate standard error to 0.027^2

The `longnote` footnote seems particularly useful.

¹where we mention $p = \frac{a}{b}$

² p is unknown but expected to be around $1/3$. Standard error will be approximated

$$SE = \sqrt{\left(\frac{p(1-p)}{n}\right)} \approx \sqrt{\frac{1/3(1-1/3)}{300}} = 0.027$$

Chapter 19

Unit 3: Independent Research

Students will select their own research question that can be answered with the IPUMS data set and will spend five weeks conducting a research project complete with data analysis, visualization, and interpretation.

In this section we encourage the instructor to provide ample time for independent student/small-group research. Some class time should be devoted to modeling healthy discussion and critique of methods. Students should learn to discuss not just *how* to answer a research question but *why* they are asking/answering it. What impact does the question/answers have. Is the question relevant/meaningful, and importantly, Is this research question perpetuating racist ideas.

We provide some examples here but encourage instructors (or students) to bring in recent journal/popular articles that do (or do not) apply data science methods well.

- 19.1 **Week 11: Students develop research Question**
- 19.2 **Week 12: Students find relevant variables from IPUMS**
- 19.3 **Week 13: Students test and evaluate results**
- 19.4 **Week 14: Students prepare presentations of results**
- 19.5 **Week 15: Students present work (slides, poster, podium, etc)**

By this point, students should be familiar with basic concepts from Chapter 1. These include:

- Basic Coding
 - read/write data in/out of R
 - basic manipulations
- Theoretical Basis
 - looking at data distributions
 - formal assessment of distributions

Students will also be familiar with how these concepts are applied from Chapter 18. Hopefully students will be able to:

- Come up with a social science question they are interested in
 - Critically think about target variable(s) of interest. Any *a priori* covariates? confounders?
 - Acquire relevant data from IPUMS
 - Analyze, Summarize, Visualize Data
 - * scope and complexity at student/teach discretion
 - Present research to class
 - * **potentially** critically discuss/evaluate each others work.

- * **science is collaborative** everyone should be out to do their best work and represent the data as best we can. We all have conscious and unconscious biases, and the best way to confront them is share and receive (respectful) feedback.

During this Unit, we suggest giving ample class time for independent student research, peer-to-peer collaboration, and basic R/stats troubleshooting. This would also be a great time to model how to give respectful criticism by discussing recent research papers. * We could maybe come up with 1-2 seed examples, with a few talking points

19.6 Example one

19.7 Example two

Chapter 20

Example RMD code

For now, this chapter is a bit of a placeholder. I'm not sure what/how the `references.Rmd` file actually fits in to the code/construction (it looks automatic) so I want to keep that in place and need a section to note that.

I also want a more centralized reference point to put any example code I find helpful while working in R/bookdown. This section could get really unruly really fast, but oh well.

20.1 Core

`index.Rmd` is required and treated as file 00. Chapters *should* be numbered for ease of sorting but custom orders are possible by specifying filenames somewhere **in this file**

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#. + IE` beyond the YAML header this file functions as a normal chapter since it starts with a top level header. + Note that `index.Rmd` has its own YMAL in addition to the various .yaml files...not sure exactly how these relate.

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure `@ref(fig:norm_dist_plot)`. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table `@ref(tab:norm_summary_tab)`. * Again, this prints an auto-generated numeral * also leaving this in the context of the plots in Chapter 18

You can write citations, too. See `knitr::write_bib()` for more on this. Quick example from `demo/index` (may not work without `write_bib()` though): we are using the **bookdown** package (Xie, 2022) in this sample book, which was built

on top of R Markdown and **knitr** (Xie, 2015). * If included, “Refernces” section gets added to each chapter. * Not exactly sure where

Embed html renders (EG, fancy tables (IPUMS_var_desc), or any shiny app) with **webshot** R package and **phantomJS**.

```
install.packages("webshot")
webshot::install_phantomjs()
```

Embed figures from a folder.

For this, it’s usually best to use a code-chunk and **knitr**. There are a number of graphical paramerters you can set (or ignore) **out.width** will scale your image accordingly - irrespective of unit/display **fig.align** should be “left”, “right”, or “center” **fig.cap** allows you to provide “mouse over” captions for the image. **echo=FALSE** is important if you ONLY want the image (IE the result of the code). If you want the code itself to show, (IE, or echo) set **echo=TRUE**.



Figure 20.1: the ipums logo

20.2 Tips

***Autonumber sections** Note the {-} used to indicate “do not number this section” eg: preface.

LABEL EVERYTHING you’ll likely want to reference it later * code chunks that produce figures can be referenced via `@\ref{fig:[LABEL]}`

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 1. If you do not manually label them, there will be automatic labels anyway, * No idea how the automatic references work, so always be sure to declare them. * **NOTE** these display as the relevant Chapter **numeral**.

20.3 Syntax

italics or *italics* (can handle spaces) **bold** code *equations*

20.3.1 Math

Randal Pruim features an extensive list of common math expression on their github page. Here are some quick notes:

In-line equations can be written within `$` and will be displayed right there: $a^2 + b^2 = c^2$. In contrast, you can also add equation chunks by using `$$`

This can be coded in-line,

$$\sum_{n=1}^{10} n^2$$

, but will result in a page break.

Alternatively, a more “classic” equation chunk:

`$$` Plain text doesnt get spaces

how

very

odd

`$$`

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.

Chapter 21

Intro/Setup

NOTE: chapter templates **do not** require a yaml header (most .RMD docs do). Lay out sections using # headers.

Chapter 22

Engage

Brief example of an engaging phenomenon. * Poll students and discuss results + Does the *average* match up with individual experiences? * Real-world research example * Interrogating a commonly held assumption + the average height of humans is 5'5"

Chapter 23

Explore

Explore other examples of this phenomenon.

Build on the first one to show variation. * Other ways to represent the phenomenon/data + tabular + visually * can the example be grouped / broken into subgroups + does the same pattern/phenomenon apply? * Is there a corollary / inverse phenomenon?

Chapter 24

Explain

Discuss/interrogate the pattern of the data * does the shape imply anything *
try to have it student led / guided * with time, students guess at what the data
shows/doesn't show without labels

Chapter 25

Elaborate

Work through a new example, contextualizing the phenomenon using real-world data/examples.

Chapter 26

Evaluate/Exercises

short lab exercises for students to recreate / code in R

Build on past concepts reinforcing basic data manipulations, summaries, etc.

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2022). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.30.