Data Science for Social Scientists

An applied course using IPUMS data

Daniel E. Ehrlich, IPUMS, University of Minnesota
Anna Tremblay, Dept of Soc, Anth, & CJ, Clemson University

Compiled on: 2022-09-09

Contents

```
knitr::include_graphics("")
```

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 $_{\text{(nearly)}}$ 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 soial 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
  - Alerts about notable changes in variable definition, universe, or coding

IPUMS data is free to use for education and research purposes. Researchers just need 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 set up classroom accounts, making it easy to get your students registered and share IPUMS data with them.

Why make this course

While we (DEE) may be slightly biased, we think IPUms is a fantastic resource for **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 meany instructors may be teaching across multiple disciplines, in large departments, or be the only "data person" at their instutition. We think IPUMS data will be useful to virtually any social science field. We provide some example lessons, and encourage instructors to develop their own, using our `template`, 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 ~ EDUATTAIN ; SEX ~ EDATTAIN + EMPSTAT
- Students learn to perform exploratory analysis, hypothesis testing, and statistical inference.
- Students learn to navigate IPUMS website,and find relevant data to thier 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: Statisitcal 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 2.2, 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 2.3, 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 releveant/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

0.0.0.1.1 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?

0.0.0.1.2 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

0.0.0.1.3 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

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

1.2 Week 2: Plotting Data

1.2.1 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
```

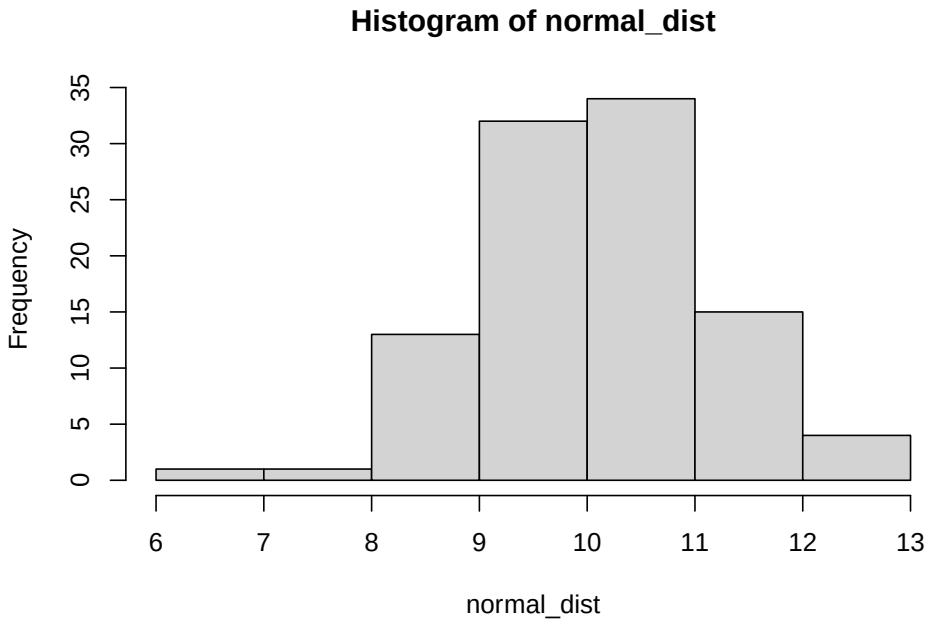
```
##   [1] 11.478646 10.331263  8.834902  9.961564 11.245215 10.369800 12.182108
##   [8] 10.815619 10.642496 10.096741  8.851044 12.193397  9.312289 12.325273
##  [15] 10.157898 11.168569  9.052855 10.244365 10.082199  9.333650 10.446365
##  [22]  6.294574  9.947701  9.238648  9.361780 10.661817 10.096049  8.037339
##  [29]  9.678354 10.437560 10.493063  8.848172  8.714679  7.635926  9.761703
##  [36] 10.571953 11.249762  8.977454  9.173229  9.076643 10.317893 11.758586
##  [43] 11.166208 10.369127  9.882664 10.129603 11.888215  9.829739 10.621154
##  [50]  9.800394  8.624825 12.424222 10.088494  9.911126 10.110451  9.222761
```

```
##  [57] 10.605988  9.911060  9.447371  9.589187 10.587178 11.837349 11.117509
##  [64] 10.851895  9.778163  9.457002 10.237196  8.658112  9.543992 10.223956
##  [71]  9.559867 11.513343 10.143615  9.124337  8.739889  9.598971  9.450454
##  [78]  9.931935 10.681716 11.309528 10.119992  9.655199  9.869252  8.987831
##  [85] 11.168759  9.680862 10.471553 10.048201  9.936379 11.064880  8.564822
##  [92]  8.543090 10.316112 10.674261  9.569134 11.190667 11.713518 10.348422
##  [99] 10.639166  8.913141
```

Another better way to look at data would be to **visualize** or **plot** it. One way to to 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

1.2.2 What *is* normal?

1.2.2.1 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.
##   6.295   9.450  10.092  10.069  10.640  12.424
```

Mean, standard deviation

1.2.2.2 Meaningful Comparisons

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

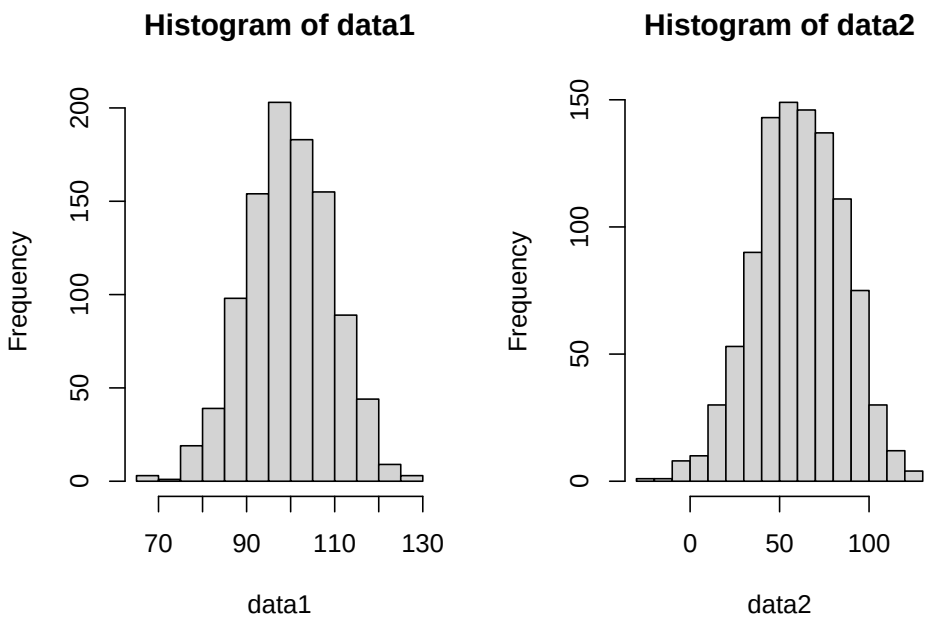
```
data1 <- rnorm(n=1000,
               mean = 100,
               sd = 10)

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

Are these the same distribution?

Any issues??

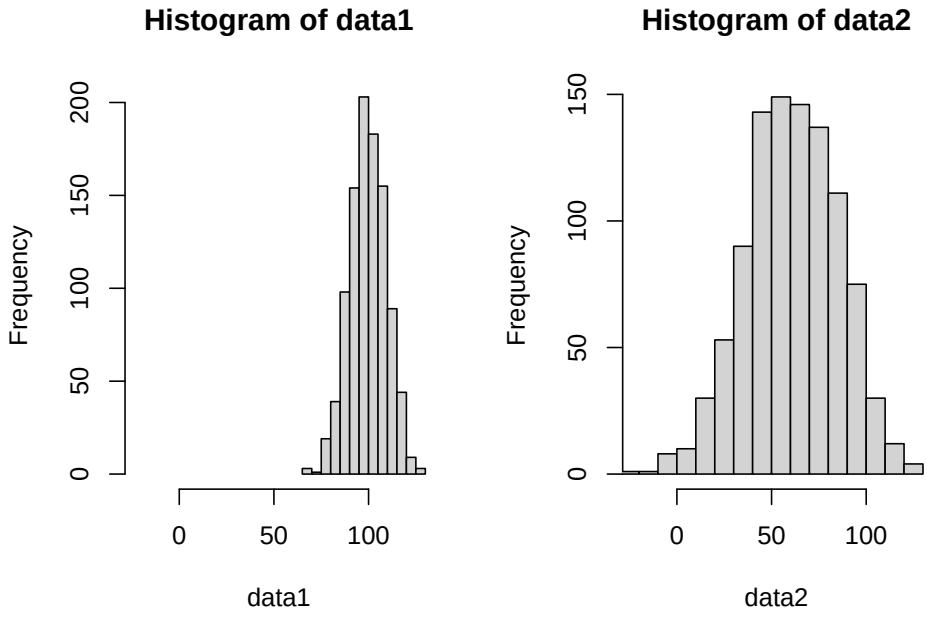
```
layout(matrix(1:2, ncol = 2))
hist(data1)
hist(data2)
```




```
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,]  68.91153 93.14111 99.53670 99.69213 106.19819 127.9187
## [2,] -22.63174 44.29881 60.90239 60.89345  78.79853 127.2767
```

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	68.91153	93.14111	99.53670	99.69213	106.19819	127.9187
data2	-22.63174	44.29881	60.90239	60.89345	78.79853	127.2767

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?

1.2.3 Skews

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

1.3 Week 3: Statisitcal testing of simple data sets

1.3.1 t-tests, ANOVA, chi2

1.4 Week 4: Relationships between variables in simple data sets

1.4.1 Correlation, Linear Regression

1.4.1.1 Simple LM

1.4.1.2 Complex LM

1.4.2 Genearlized Linear Model

1.5 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 2 * Student-driven projects 3

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)

1.6 Intro to R/RStudio

1.7 Reading Data / Distributions

1.7.1 What *is* a normal distribution

1.7.1.1 How normal is it?

show increasingly unclear examples of normal vs not

introduce tests of normality

1.7.1.2 Measuring normality - single sample

reinforce [concept of statistical] **normality**

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

1.7.1.3 comparing normality - two saples

standard / two-way t test

1.7.1.4 comparing more than two - ANOVA

Chapter 2

IPUMS

2.1 Week 6 Introduction to IPUMS

2.2 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

2.2.1 Advanced Exploration - Change Over Time

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

2.2.1.1 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)
```

2.2.1.1.1 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)
```

2.2.1.1.2 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,.1

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")
```

2.2.1.1.3 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)
```

2.2.1.1.4 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)
```

2.2.1.2 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("YE
```

```
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("
```

```
table(hhs$extended_test,hhs$nuclear_test)
```

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

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

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

```
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() %>
```

2.2.1.2.1 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 = "'
```

2.2.1.2.2 Visualize Nuclear Families

2.2.1.3 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 = q



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 = qua

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")
                )
}
```

2.2.1.3.1 Gernerate 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))
```

2.2.1.3.2 Visualize

2.2.1.4 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))
```

2.2.1.4.1 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))
```

2.2.1.4.2 Parents Supporting (extended) children

2.3 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

2.4 Week 9 Statistical Inference

2.5 Week 10 (TBD)

Some text to break up the sub-section headers

2.6 Intro to IPUMS website

2.6.1 background on ipums

2.6.2 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.

2.7 math example

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

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

You can also use math in footnotes like this[1]. 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.

[1] where we mention $p = \frac{a}{b}$

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

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

Chapter 3

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 releveant/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.

3.1 Week 11: Students develop research Question

3.2 Week 12: Students find relevant variables from IPUMS

3.3 Week 13: Students test and evaluate results

3.4 Week 14: Students prepare presentations of results

3.5 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 2. 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

3.6 Example one

3.7 Example two

Chapter 4

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 unrully really fast, but oh well.

4.1 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 .yml 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 2

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()
```

4.2 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`.

4.3 Syntax

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

4.3.1 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/.

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 0.28.