Williams College ECON 370:

Data Science for Economic Analysis

Lecture 1: Data

Professor: Pamela Jakiela

Outline: A Crash Course in Data Cleaning

- Rectangular data, types of variables, data types
- What does it mean for data to be clean, and why is it important to clean your data?
- The steps in the data preparation pipeline: import, clean, shape, define, check

What Is Data?

There are (increasingly) many sources of data that can be used by economists

- Much of “data science” focuses on widening the set of data sources available

Economists and data scientists typically analyze data that is stored as a rectangular **data frame**

- Each column of the data frame is a **variable**
- Each row of the data frame is an **observation**

Spreadsheets, Stata data sets, and matrices are examples of data frames (more or less)

- Text, images, etc. are often transformed into data frames

Rectangular Data Frames

Country	**GDP per Capita**	**Life Expectancy**	**World Bank Lending Group**
Afghanistan	529.14	62.58	Low Income Countries
Albania	4,418.66	76.99	Upper Middle Income Countries
Algeria	3,873.51	74.45	Lower Middle Income Countries
American Samoa	14,214.65	..	High Income Countries
Andorra	34,394.43	..	High Income Countries
Angola	2,435.02	62.26	Lower Middle Income Countries
Antigua and Barbuda	14,803.77	78.84	High Income Countries
Argentina	11,346.65	75.89	Upper Middle Income Countries

Source: World Development Indicators data for the year 2020

Types of Variables

Data (in data frames) is fundamentally either numbers or text, but we can distinguish between:

- Numeric variables
  - Continuous variables
  - Indicator variables (aka “dummy” variables)
  - Discrete/integer variables (may or may not be stored differently than continuous variables)
- String variables (i.e. text), may not include any non-numeric characters (e.g. zip code)
- Categorical variables
  - Can be text or numeric (usually with labels for categories)

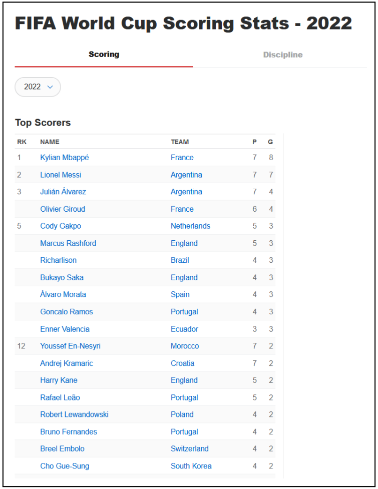
How variables are structured/coded depends on how we plan to analyze the data

Trick Questions

Looking at the data on GDP and life expectancy from slide 3:

- Is **GDP per Capita** a numeric variable or string variables?
- Is **Life Expectancy** a numeric variable or string variables?
- Is **World Bank Lending Group** a numeric variable or a string variable?
- How would you include **World Bank Lending Group** in a regression?

Raw Data from ESPN.com



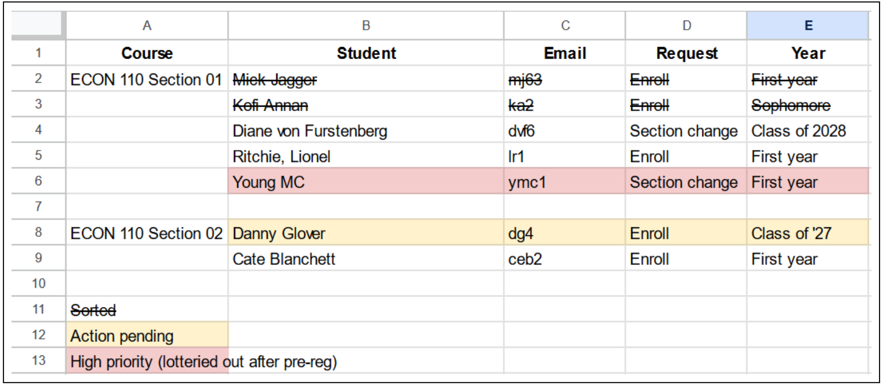
FIFA World Cup Scoring Stats - 2022

Top Scorers

RK	NAME	TEAM	P	G
1	Kylian Mbappé	France	7	8
2	Lionel Messi	Argentina	7	7
3	Julián Álvarez	Argentina	7	4
	Olivier Giroud	France	6	4
5	Cody Gakpo	Netherlands	5	3
	Marcus Rashford	England	5	3
	Richarlison	Brazil	4	3
	Bukayo Saka	England	4	3
	Álvaro Morata	Spain	4	3
	Goncalo Ramos	Portugal	4	3
	Enner Valencia	Ecuador	3	3
12	Youssef En-Nesyri	Morocco	7	2
	Andrej Kramaric	Croatia	7	2
	Harry Kane	England	5	2
	Rafael Leão	Portugal	5	2
	Robert Lewandowski	Poland	4	2
	Bruno Fernandes	Portugal	4	2
	Breel Embolo	Switzerland	4	2
	Cho Gue-Sung	South Korea	4	2

Source: ESPN.com

Almost Exactly the Economics Department's Waitlist for ECON 110



	A	B	C	D	E
1	**Course**	**Student**	**Email**	**Request**	**Year**
2	ECON 110 Section 01	~~Mick Jagger~~	~~mj63~~	~~Enroll~~	~~First year~~
3		~~Kofi Annan~~	~~ka2~~	~~Enroll~~	~~Sophomore~~
4		Diane von Furstenberg	dvf6	Section change	Class of 2028
5		Ritchie, Lionel	lr1	Enroll	First year
6		Young MC	ymc1	Section change	First year
7					
8	ECON 110 Section 02	Danny Glover	dg4	Enroll	Class of '27
9		Cate Blanchett	ceb2	Enroll	First year
10					
11	~~Sorted~~				
12	Action pending				
13	High priority (lotteried out after pre-reg)				

Clean (“Tidy”) Data

Raw data that we find in the wild is not usually ready for analysis

- The process of transforming raw data into usable form is called **data cleaning**

Data can only be analyzed when it is clean (or “tidy” in R-speak):

- Variables are in columns, each is labeled, labels are short and self-explanatory (no spaces)
- Each row is an observation and each observation is a row
- Each cell contains only one value, appropriately formatted (e.g. numbers are not strings)
- Data is only missing when it should be (i.e. when a value is unavailable)
- Values of variables are reasonable, strings are consistent and free of spelling errors

The Data Preparation Pipeline

1. Import/read in/load the data
2. Clean/tidy the data
3. Reshape/pivot/join/merge the data
4. Define/generate/create new variables needed for analysis
5. Exploratory data analysis to confirm that the data is clean and ready for analysis

The Importance of Replicability in Data Science

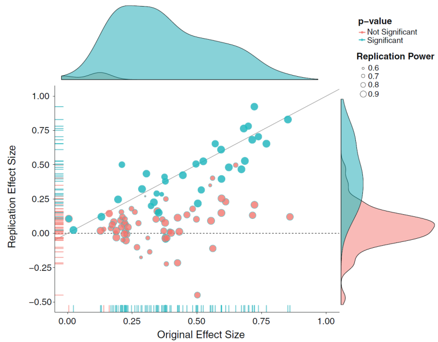


Fig. 3. Original study effect size versus replication effect size (correlation coefficients). Diagonal line represents replication effect size equal to original effect size. Dotted line represents replication effect size of 0. Points below the dotted line were effects in the opposite direction of the original. Density plots are separated by significant (blue) and nonsignificant (red) effects.

Source: Attwood et al. (2015)

LETTERS NATURE HUMAN BEHAVIOUR

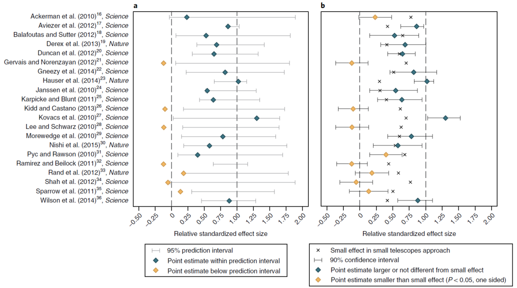


Fig. 2 | Replication results for two complementary replication indicators. a, Plotted are the 95% prediction intervals[47] for the standardized original effect sizes (correlation coefficient r). The standardized effect sizes are normalized so that 1 equals the original effect size. Original and zero effect size are indicated by dashed lines. Fourteen replications out of 21 (66.7%; 95% CI = 44.7–88.7%) are within the 95% prediction interval and replicate according to this indicator. **b**, Plotted are the 90% CIs of replication effect sizes in relation to small-effect sizes as defined by the small telescopes approach[46] (the effect size that the original study would have had 33% power to detect). Effect sizes are standardized to correlation coefficients r and normalized so that 1 equals the original effect size. A study is defined as failing to replicate if the 90% CI is below the small effect. According to the small telescopes approach, 12 out of 21 (57.1%; 95% CI = 34.1–80.2%) studies replicate.

Source: Camerer et al. (2018)

The Importance of Replicability in Data Science

All the steps in the data pipeline are coded so as to be transparent and fully replicable

1. Import/read in/load the data, making sure variables are parsed as correctly as possible
2. Clean the data to make sure variables are named well and in the correct formats, data is not missing unless it should be, and there are no obvious errors apparent in the data values
3. Reshape/pivot/join data frames so that observations are at the appropriate level(s) and the analysis variables are in a single data frame with the correct number of observations
4. Define/generate/create new variables needed for analysis
5. Exploratory data analysis to confirm that the data is/are clean and ready for analysis

Importing Data

Raw data files should be stored in a raw data subfolder within your project folder

- You should never write any files to your raw data folder – it exists to protect the raw data

Raw data is usually stored in csv (comma-separated) or other delimited format, but sometimes in Excel; csv is also how you should save data so that it can be read into R, Python, Stata, etc.

- The function or package you use to load data often has implications for the object created
  - Ex.: `read_csv()` in R loads data as a tibble while `read.csv()` loads a data frame
  - Avoid new packages and stick to data and object types that are widely used
- Iterate with your code to minimize the need for unnecessary data cleaning
  - Ex.: avoid reading in header rows, but try to structure your code so that changes to the raw data (e.g. and increase in the number of populated rows in Excel) won't lead to data errors
  - When possible, read numbers in as numeric variables and text and categoricals as strings

Aside: Variable Types vs. Data Types

Statistical analysis tools other than Stata (here: R) let you define different types of objects

- Individual scalars or strings (like locals/globals in stata), can also be logical, etc.
- **Vectors** are $n \times 1$ column-shaped lists of numeric or string values (variables in stata)
- **Matrices** are multiple $n \times 1$ vectors of the same type grouped together
- **Data frames** and **tibbles** are like matrices, but underlying $n \times 1$ component vectors can be of different types (so data frames are like an entire data set read into stata)
- **Lists** are magical vectors of anything: e.g. a vector of data frames or a vector that contains some character observations and some numeric observations (or other lists)

Each csv file that you load will typically be its own tibble or data frame, and the question of appropriate variable type (i.e. numeric vs. character) is answered at the tibble$vector level

Cleaning Data

The most important, absolutely unbreakable rule for replicable social science:

- **Never** modify your raw data files by hand; do all of your cleaning in your code!

Data cleaning is basically looking at each variable and asking the following questions

1. Is it formatted correctly, i.e. is the variable the appropriate data type?
2. Does the variable/vector/etc have a reasonable name that makes sense?
   - Avoid: `v12`, `‘rep(seq(1:4), each = length(name))‘`, `MeanOfMathTestScore2012`
3. Are there missing values, and if so are they unavoidable? Should some observations be dropped from the analysis because key variables are missing (e.g. incomplete surveys)?
4. Are the observed values reasonable?

The **garden of forking paths**: there are often many reasonable ways to handle problems with the raw data; your goal is to make sensible choices, document them in comments in your code, and learn when to ask for guidance and when to make a decision on your own and move forward

Shaping Data: Groups, Pivots, and Joins

Clean data always has one observation per row, but what constitutes an observation?

- Ex.: transaction-level sales data might be analyzed at the day or month level

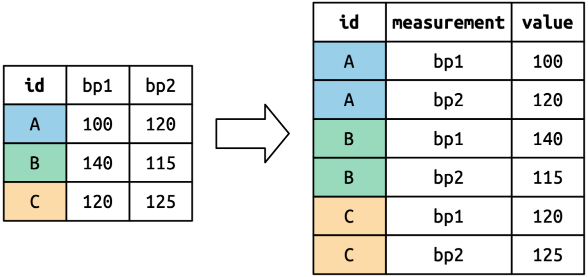
Grouping/collapsing data involves a loss of specific detail (e.g. keeping only day-level mean), but sometimes we want to change the level/unit of analysis without losing any information

- Ex.: difference-in-differences with two rounds of state-level data

Reshaping or pivoting a data frame converts it from **wide** format to **long** or vice versa

- In wide format, we might have observations of outcome Y at the state-year level
- In long format, we would then have observations at the state level with distinct Y_t variables for each of the different years (different values of t) included in the analysis

Shaping Data: Groups, Pivots, and Joins



id	bp1	bp2
A	100	120
B	140	115
C	120	125

id	measurement	value
A	bp1	100
A	bp2	120
B	bp1	140
B	bp2	115
C	bp1	120
C	bp2	125

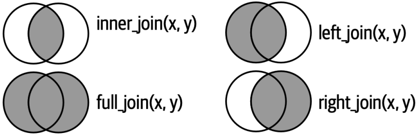
Source: Wickham et al. (2023)

Shaping Data: Groups, Pivots, and Joins

We often find ourselves wanting to combine two sources of data

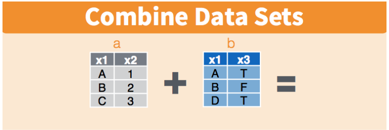
- Ex.: merging World Bank data in GDP per capita with data on educational attainment

We do this by **joining** (or merging) two data fames using a common variable (the **key**)



Source: Wickham et al. (2023)

Shaping Data: Groups, Pivots, and Joins



Combine Data Sets

a

x1	x2
A	1
B	2
C	3

+

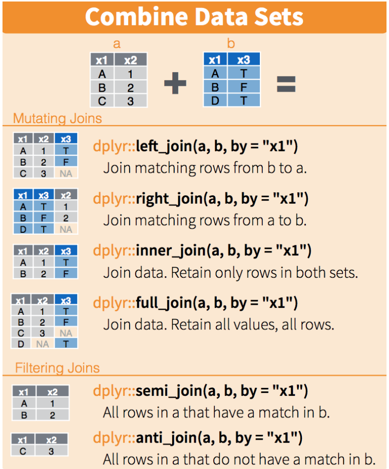
b

x1	x3
A	T
B	F
D	T

=

Source: Irizarry (2024)

Shaping Data: Groups, Pivots, and Joins



Combine Data Sets

a

x1	x2
A	1
B	2
C	3

+

b

x1	x3
A	T
B	F
D	T

=

Mutating Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NA

dplyr::**left_join(a, b, by = "x1")**
Join matching rows from b to a.

x1	x3	x2
A	T	1
B	F	2
D	T	NA

dplyr::**right_join(a, b, by = "x1")**
Join matching rows from a to b.

x1	x2	x3
A	1	T
B	2	F

dplyr::**inner_join(a, b, by = "x1")**
Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T

dplyr::**full_join(a, b, by = "x1")**
Join data. Retain all values, all rows.

Filtering Joins

x1	x2
A	1
B	2

dplyr::**semi_join(a, b, by = "x1")**
All rows in a that have a match in b.

x1	x2
C	3

dplyr::**anti_join(a, b, by = "x1")**
All rows in a that do not have a match in b.

Source: Irizarry (2024)

Defining New Variables

Defining the variables needed for analysis is typically the most straightforward aspect of the data preparation pipeline, and variable formulas are often well specified by the research design

- Ex.: log GDP per capita, incumbent vote share, hours worked

A few rules-of-thumb for constructing controls/covariates:

- Categorical variables need to be converted to dummies (one hot encoding) for analysis
  - Who is the reference group? One hot encoding does not choose reference group ex ante.
- Many variables are converted to normalized z-scores with mean $= 0$ and SD $= 1$
  - Who is the reference group? Normalizing in entire sample vs. control/pre-treatment group.
  - Many machine learning techniques expect variables measured on comparable scales
- Create a dummy variable to indicate when covariates are missing/imputed

Lab #1

Objective: combine country-level World Bank data on GDP per capita in 2010 with data on educational attainment (also in 2010) from the Barro-Lee Educational Attainment Data Set

- Install packages, specify file path
- Download Barro-Lee data set and World Development Indicators data on GDP per capita (constant 2015 US$), saving both as csv files in your working directory ("raw")
- Read in Barro-Lee data, keep only observations from 2010
- Read in World Bank data
- Combine the two data sets, making sure to match as many countries as possible
- Explore the relationship between the two variables