Williams College ECON 370:

Data Science for Economic Analysis

Lecture 1: Preprocessing and Exploratory Data Analysis

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 preprocessing pipeline
- Visualizing one variable
- Visualizing relationships between variables

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)

- Collections of text(s), images, etc. are (sometimes implicitly) transformed into rectangle(s) (i.e. lists of units and associated attributes) in order to conduct statistical analysis
- Computer scientists, big data users, etc. think a lot about computational efficiency, but economists are usually constrained by data sets that are too small (rather than too big)
  - Feasible to store, analyze most data sets in rectangular form

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)
  - Examples: respondent name, head of state, book title, phone number
- Categorical variables
  - Can be text or numeric (usually with labels for categories)

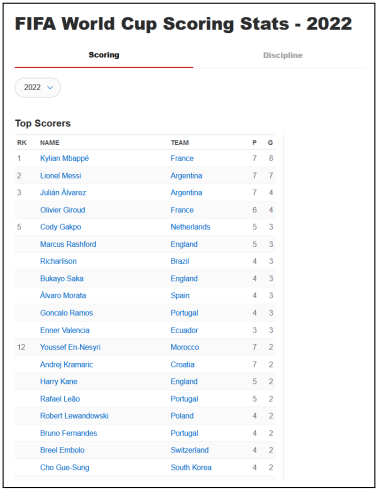
Variable types vs. data types: the same information can often be stored in several different ways, and how variables are structured/coded depends on how we plan to analyze the data

Trick Questions

Looking at the data on GDP and life expectancy from Slide 4:

- 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

Scoring | Discipline

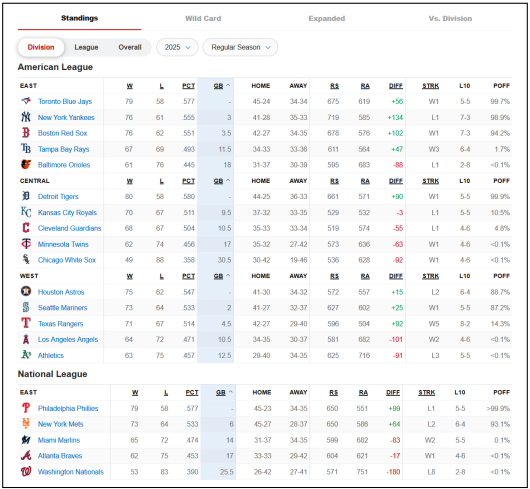
2022

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

Raw Data from ESPN.com



Standings | Wild Card | Expanded | Vs. Division

Division | League | Overall | 2025 | Regular Season

American League

EAST	W	L	PCT	GB	HOME	AWAY	RS	RA	DIFF	STRK	L10	POFF
Toronto Blue Jays	79	58	.577	-	45-24	34-34	675	619	+56	W1	5-5	99.7%
New York Yankees	76	61	.555	3	41-28	35-33	719	585	+134	L1	7-3	98.9%
Boston Red Sox	76	62	.551	3.5	42-27	34-35	678	576	+102	W1	7-3	94.2%
Tampa Bay Rays	67	69	.493	11.5	34-33	33-36	611	564	+47	W3	6-4	1.7%
Baltimore Orioles	61	76	.445	18	31-37	30-39	595	683	-88	L1	2-8	<0.1%

CENTRAL	W	L	PCT	GB	HOME	AWAY	RS	RA	DIFF	STRK	L10	POFF
Detroit Tigers	80	58	.580	-	44-25	36-33	661	571	+90	W1	5-5	99.9%
Kansas City Royals	70	67	.511	9.5	37-32	33-35	529	532	-3	L1	5-5	10.5%
Cleveland Guardians	68	67	.504	10.5	35-33	33-34	519	574	-55	L1	4-6	4.8%
Minnesota Twins	62	74	.456	17	35-32	27-42	573	636	-63	W1	4-6	<0.1%
Chicago White Sox	49	88	.358	30.5	30-42	19-46	536	628	-92	W1	4-6	<0.1%

WEST	W	L	PCT	GB	HOME	AWAY	RS	RA	DIFF	STRK	L10	POFF
Houston Astros	75	62	.547	-	41-30	34-32	572	557	+15	L2	6-4	88.7%
Seattle Mariners	73	64	.533	2	41-27	32-37	627	602	+25	W1	5-5	87.2%
Texas Rangers	71	67	.514	4.5	42-27	29-40	596	504	+92	W5	8-2	14.3%
Los Angeles Angels	64	72	.471	10.5	34-35	30-37	581	682	-101	W2	4-6	<0.1%
Athletics	63	75	.457	12.5	29-40	34-35	625	716	-91	L3	5-5	<0.1%

National League

EAST	W	L	PCT	GB	HOME	AWAY	RS	RA	DIFF	STRK	L10	POFF
Philadelphia Phillies	79	58	.577	-	45-23	34-35	650	551	+99	L1	5-5	>99.9%
New York Mets	73	64	.533	6	45-27	28-37	650	586	+64	L2	6-4	93.1%
Miami Marlins	65	72	.474	14	31-37	34-35	599	682	-83	W2	5-5	0.1%
Atlanta Braves	62	75	.453	17	33-33	29-42	604	621	-17	W1	4-6	<0.1%
Washington Nationals	53	83	.390	25.5	26-42	27-41	571	751	-180	L8	2-8	<0.1%

Source: ESPN.com

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, with names, names 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

Also important to make sure that information encoded in raw data (e.g. color coding) is not lost

Steps in the Data Preprocessing Pipeline

Preparing data for analysis involves these steps, in some order:

- **Import/load the data**
- **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
- **Reshape and merge data sets** so that observations are at the appropriate level(s) and the analysis variables are in a single data set with the correct number of observations
- **Generate new variables** needed for analysis
- **Exploratory data analysis** to confirm that the data is/are clean and ready for analysis

All the steps in the data preprocessing pipeline must be transparent and fully replicable

The Importance of Replicability in Social Science

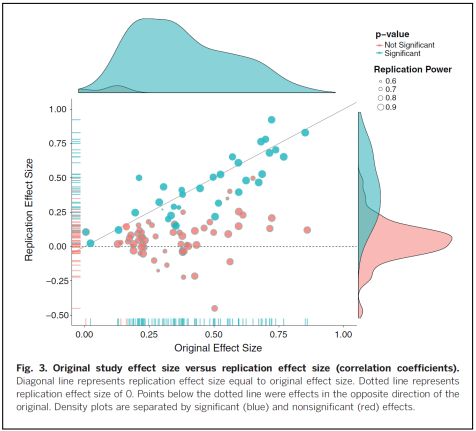


Fig. 3. Original study effect size versus replication effect size (correlation coefficients). Diagonal line represents replication effect size equal to original effect size. Dotted line represents replication effect size of 0. Points below the dotted line were effects in the opposite direction of the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

Source: Attwood et al. (2015)

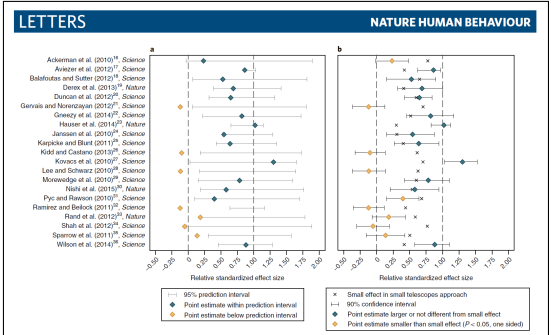


Fig. 2 | Replication results for two complementary replication indicators. a. Plotted are the 95% prediction intervals[42] for the standardized original effect sizes (correlation coefficient *r*). The standardized effect sizes are normalized so that 1 equals the original effect size. Original and zero 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[43] (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)

Steps in the Data Preprocessing Pipeline

Preparing data for analysis involves these steps, in some order:

- **Import/load the data**
- **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
- **Reshape and merge data sets** so that observations are at the appropriate level(s) and the analysis variables are in a single data set with the correct number of observations
- **Generate new variables** needed for analysis
- **Exploratory data analysis** to confirm that the data is/are clean and ready for analysis

All the steps in the data preprocessing pipeline must be transparent and fully replicable

Importing Data

Raw data should be stored in a raw data subfolder within your project folder (or on github)

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

Raw data is often stored in csv (comma-separated values) or other delimited format, but sometimes in Excel, Stata, or SPSS format; any of these 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? Do correlations with other variables make sense?

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

Reshaping Data: Pivoting

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 long format, we might have observations of outcome Y at the state-year level
- In wide 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

Reshaping Data: Pivoting

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

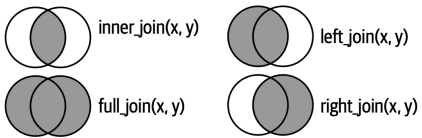
Source: Wickham et al. (2023)

Reshaping Data: Merging

We often find ourselves wanting to combine two sources of data

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

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



Source: Wickham et al. (2023)

Reshaping Data: Merging

id	var1
101	8
102	7
103	12

+

id	var2
101	14
102	15
104	92

=

inner join:

id	var1	var2
101	8	14
102	7	15

full join:

id	var1	var2
101	8	14
102	7	15
103	12	·
104	·	92

left join:

id	var1	var2
101	8	14
102	7	15
103	12	·

right join:

id	var1	var2
101	8	14
102	7	15
104	·	92

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 (often) expect variables measured on comparable scales

Handling Missing Data

Information on particular variables is often missing for some observations, but statistical analysis requires a consistent analysis data set that does not have any missing values

- Missing outcome variables vs. missing control variables
- Is "missingness" at random?

Two options: drop variables with missing values or impute missing values and flag observations

- When outcome data is missing, only reasonable option is to exclude observations
- In OLS, norm is to impute missing values and include a dummy for imputed data
  - Typical imputed value is the mean, but there are many alternatives
- This approach may not be appropriate for machine learning techniques that select a subset of independent variables to be included in a model (e.g. lasso, random forests)

The Most Important Step in Data Analysis

The last step in the preprocessing pipeline and the first step in analysis is looking at the data

- Tabulating the values, looking at summary statistics, visualizing distributions of variables

Exploratory data analysis serves two purposes:

- Detecting errors, problems, outliers, etc.
- Looking for patterns, regularities, relationships in the data

There is almost nothing worse than finding a bug in your cleaning/preparation code after you've analyzed the data, written up your results, presented your findings, published, etc.

What's Wrong With This Picture?

Statistic	N	Mean	St. Dev.	Min	Max
Female	812	1.49	0.50	1	2
Age	812	35.72	22.70	−99	60
Education	812	3.00	1.41	1	5
Married	812	0.84	0.37	0	1
Income	683	55.98	26.42	20.18	180.58

Not All Data Issues Appear in Summary Statistics Tables

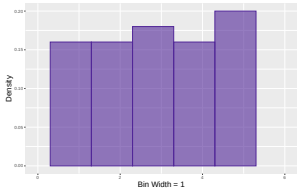
A **histogram** is a bar graph that plots the distribution of a variable X by:

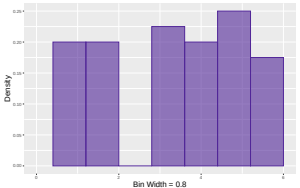
- Partitioning the support of X into equally-spaced bins
- Counting the number of observations in each bin
- Using bars to plot the relationship between the range of X value(s) included in each bin and the number (or the proportion/density) of observations that fall within that bin

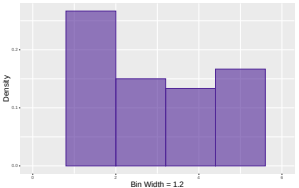
With histograms, there is only one statistical decision to be made: how many bins?

- How many bins is also one of many aesthetic decisions

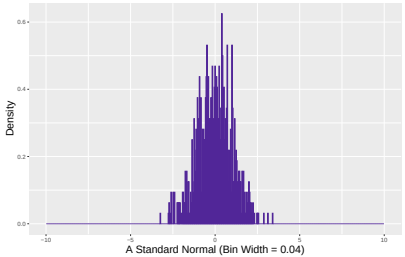
Choosing the Correct Bin width

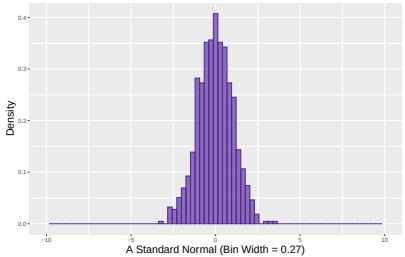


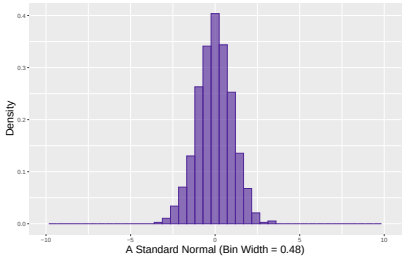


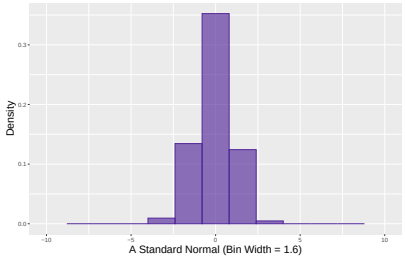


Choosing the Correct Bin width









Kernel Density Estimation

Histogram can depend on bin width and the starting point for the first/lowest bin

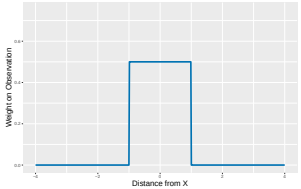
- An alternative would be to define a function $f(x)$ that counted up the number of observations "near" x (i.e. within $h > 0$ of x) for all values in the support of x
- We could then scale the function $f(x)$ so that the area under the curve sums to one

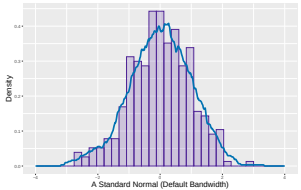
Kernel density estimation generalizes this approach for different weighting functions (kernels)

- The example above is kernel density estimation with a rectangular/uniform kernel
  - The rectangular kernel puts equal weight on all data points within bandwidth h of x
- We can instead calculate a weighted count of observations near x
  - Commonly used kernel include: Gaussian (i.e. normal), Epanechnikov (parabolic)

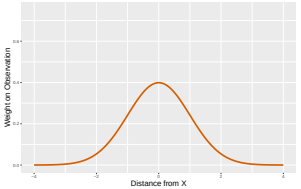
Kernel Density Estimation in Practice

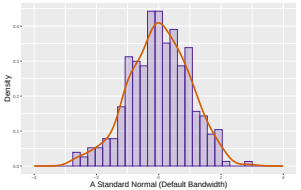
Rectangular Kernel



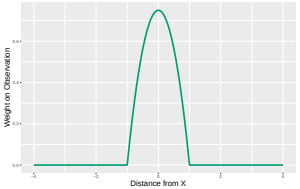


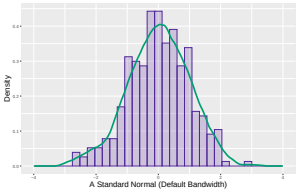
Gaussian Kernel





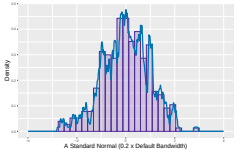
Epanechnikov Kernel



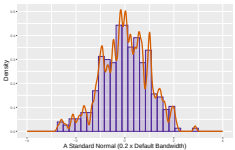


Kernel Density Estimation in Practice

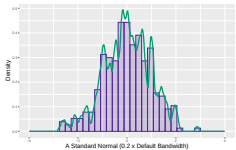
Rectangular Kernel



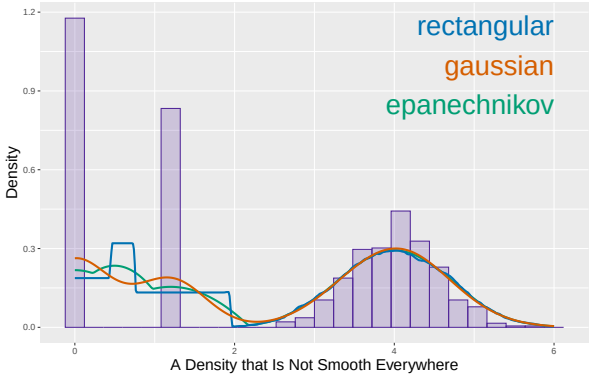
Gaussian Kernel



Epanechnikov Kernel



Q: When Shouldn't You Use a Kernel Density?

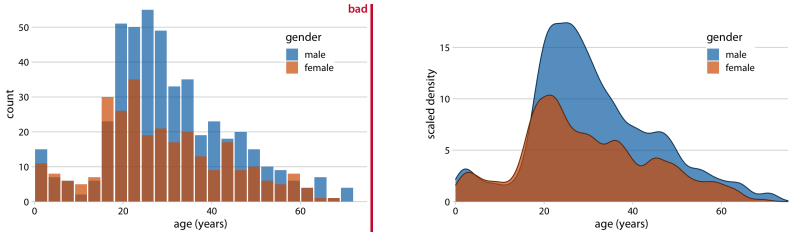


Q: When Shouldn't You Use a Histogram?

There is usually nothing wrong with using a histogram as long as you choose the size and placement of the bins carefully (though see earlier examples of how this can go wrong)

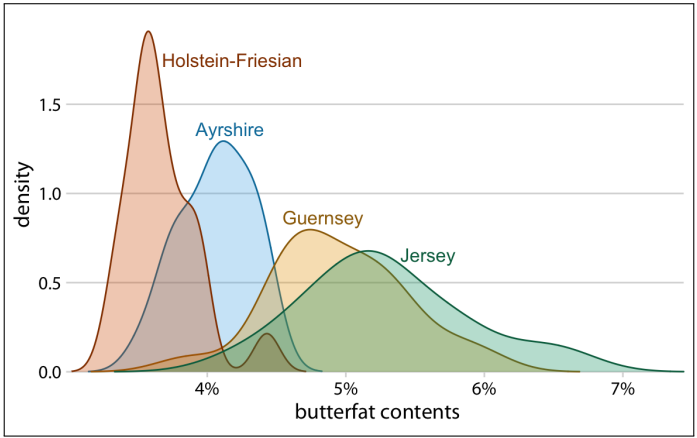
- However, it is usually better to use a kernel density plot if (you believe) the underlying density is smooth, as the bins add little to our understanding (see Slide 30 for an example)

Kernel density plots also work much better when you want to show more than one distribution



source: Wilke (2019)

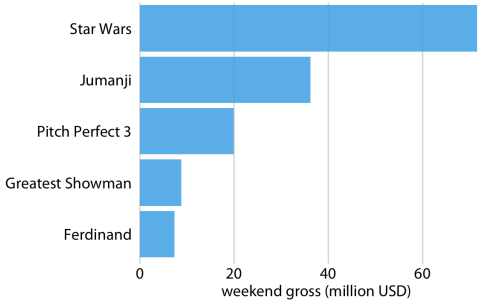
Q: When Shouldn't You Use a Histogram?



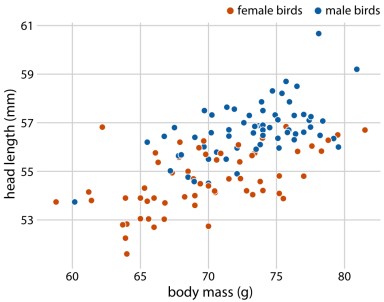
source: Wilke (2019)

Visualizing Relationships Between Two Variables

1 continuous variable + 1 categorical variable

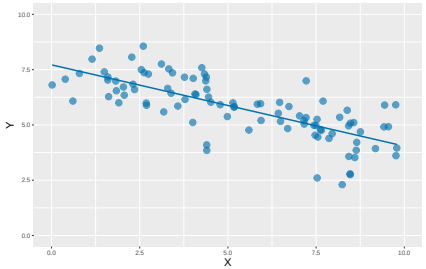


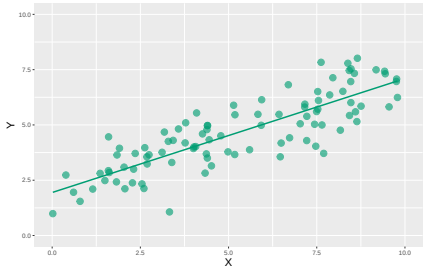
2 continuous variables

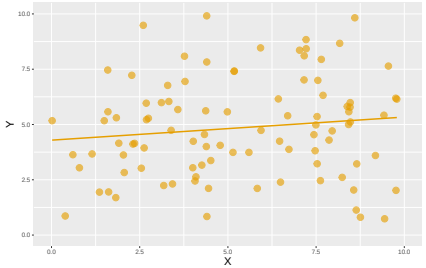


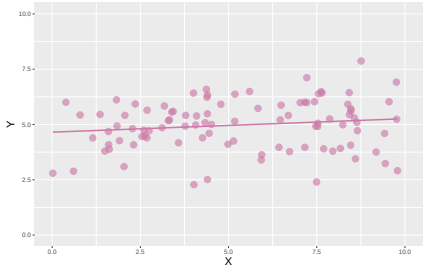
source: Wilke (2019)

A Scatter Plot Is Worth a Thousand Words

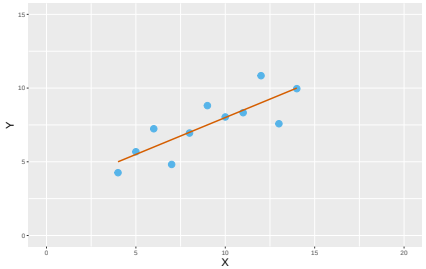


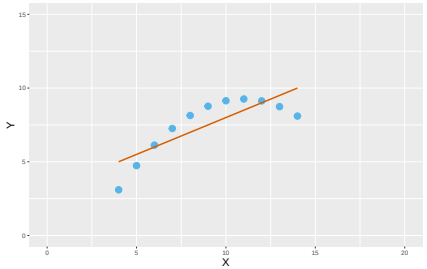


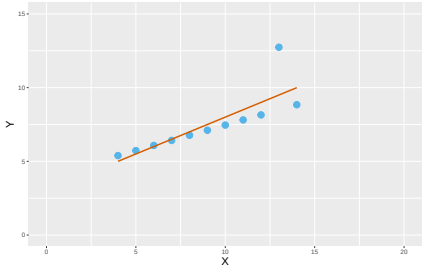


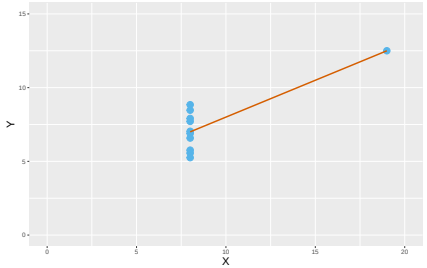


Anscombe's Quartet

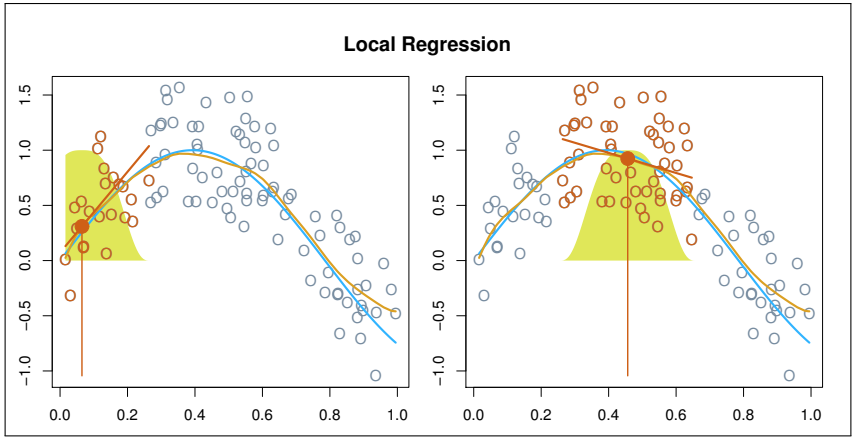






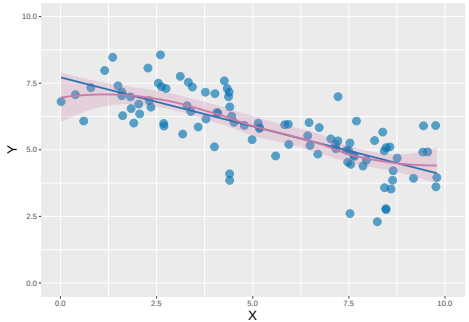


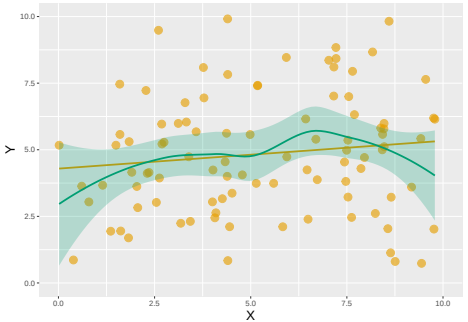
Local Linear Regression



source: James et al. (2021)

Your Workhorse Exploratory Scatter Plot





Summary of (Simple) Exploratory Data Analysis Techniques

- Summary statistics table: mean, standard deviation, minimum, maximum, count
- One variable: histogram, kernel density, or both
- Two variables: scatter plot with a linear and/or local polynomial fit

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/libraries
- Download datasets directly from github (avoids needing to set a file path)
- Clean and preprocess the data
- Combine the two data sets, making sure to match as many countries as possible
- Summarize the means of a few of the variables
- Make a histogram (of GDP per capita) and a scatter plot (of GDO and education)

The catch is that you will write both an R script and a Python script to do this

Lab #1

Separate versions of the lab for R and Python, identical except for language-specific hints

- `ECON370-lab1-template.R` and `ECON370-lab1-template.py` identical except for hints
  - If you know more of one (R or Python), start there
  - If you are committed to R, you can do the lab as a google colab
  - If you are new to Python but want to learn it, consider working out the R code first and then asking chatgpt to translate each line of code for you (and then check its suggestion)

Make sure that R and Python give you the same answers

Lab #1: Data Visualizations