STATS R FUN, #3

Data Visualization

AN INTRODUCTION TO PRODUCING AND INTERPRETING STATISTICAL GRAPHICS

DS Collab, Fall 2024

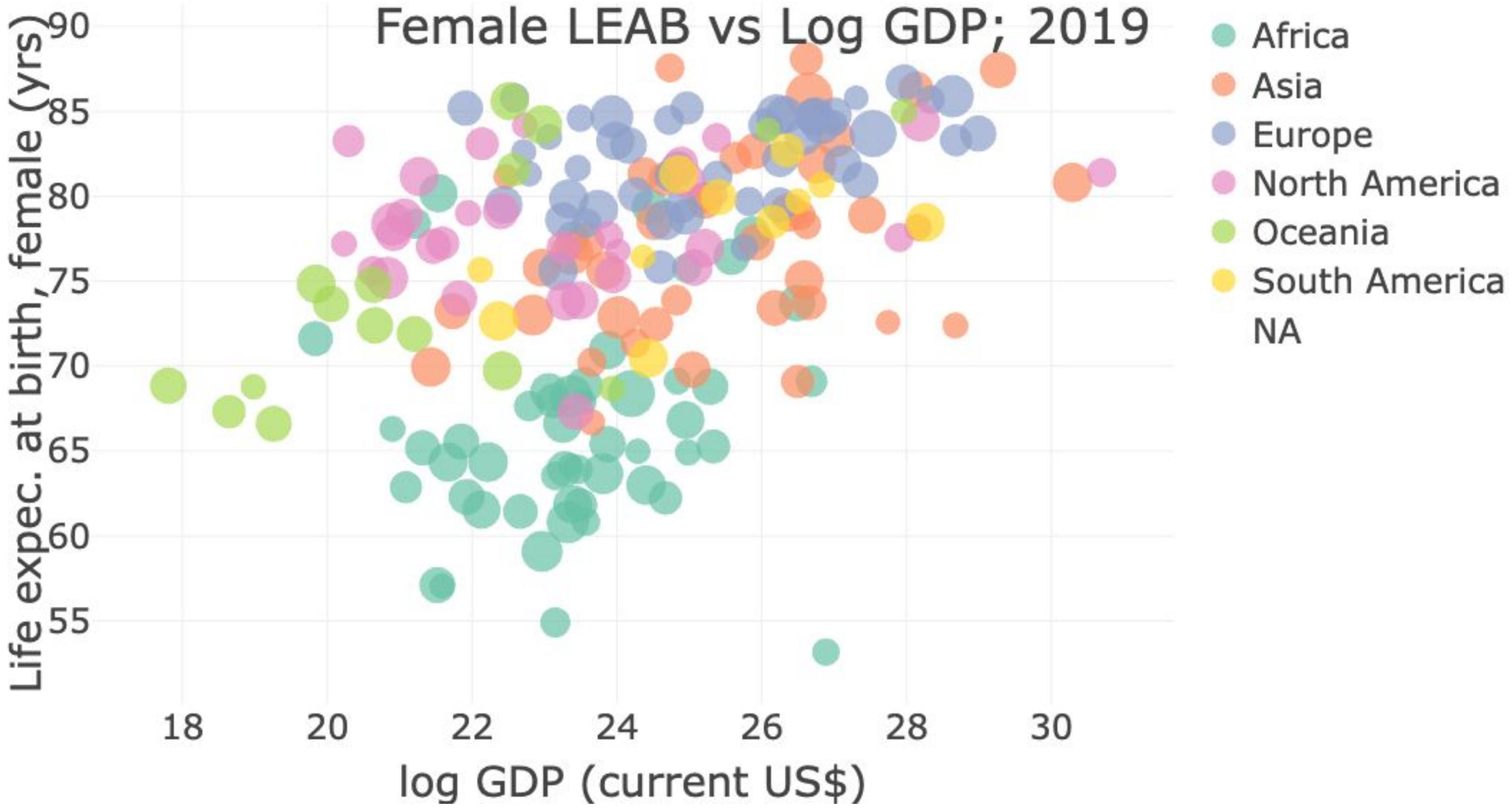
Presented by: Ethan P. Marzban

AGENDA

- Motivate the need for statistical visualizations
- Walk through a concrete example of generating a plot
- Discuss the fundamentals of statistical visualizations
- Critique published graphics

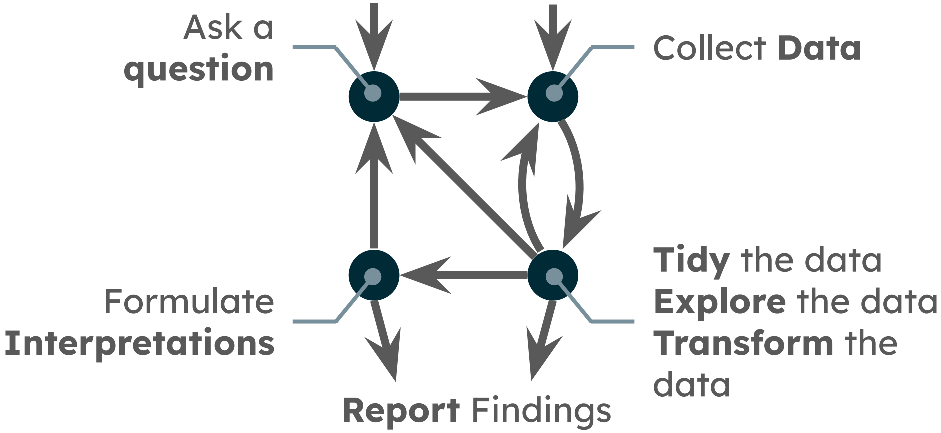
Visualizations and Data

Why Visualize?



Why Visualize?

- The corresponding data has 217 observations on 11 variables, consisting of numbers, words, missing values, and more - very difficult to read and interpret!
- Our brains are much more adept at identifying visual patterns
  - Consequently, a great deal of information can be conveyed to a reader/viewer by producing appropriate visualizations
- **Statistical visualizations** are a part of **Descriptive Statistics** (the branch of statistics dedicated to summarizing and describing data), and often appear in many parts of the **Data Science Lifecycle**:



Ask a **question**

Collect **Data**

Tidy the data
Explore the data
Transform the data

Formulate **Interpretations**

Report Findings

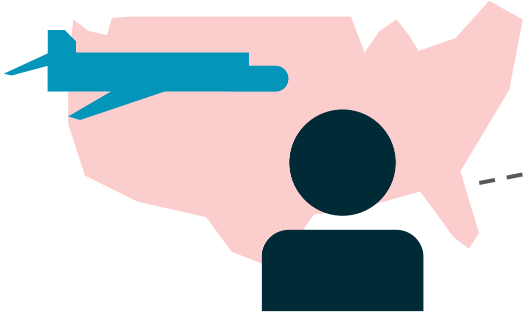
- The DSL seeks to describe the start-to-finish process of a typical Data Science project.
  - You'll learn more about it in PSTAT 100, and possible in future workshops

Data Semantics

DEFINITION

Data/Dataset : a collection of **observations** taken on **observational units**, consisting of **values** measured on a set of **variables.**

Observational Unit: the entity being measured



Observation: collection of **values** measured on various **attributes** (aka **variables**)

```
 Attribute 1     Attribute 2   …
(<value 1>    ,  <value 2> ,   …)
```

- Admittedly, the terminology used to describe different aspects of data is a little scattered, and varies depending on who you talk to. I like to use the convention adopted by **Hadley Wickham.**

Tidy Data

- The above pertains to data **semantics**: the meaning associated with the values in the dataset
- **Structure** refers to the way the values in the dataset are actually displayed
  - I.e. the specification of rows, columns, and number of tables (yes, sometimes we need multiple tables to express a particular dataset!)
  - Can also refer to the decision of how to encode particular values (e.g. should we use ‘high’, ‘medium’, ‘low’, or ‘3’, ‘2’, ‘1’?)
- Mapping the semantics of a particular dataset to its structure is absolutely crucial, to ensure computers understand our data. One such framework: the **tidy** framework of data:

DEFINITION

In **Tidy** data:

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

Let’s Make a Plot!

Dataset: Course Enrollments

Description: Enrollments in undergraduate PSTAT courses, from Winter 2022 through Summer 2024. (Scraped from GOLD)

Variables:

- Course Number (e.g. “PSTAT 100”)
- Course Title (e.g. “Data Science: Concepts and Analysis”)
- Quarter (e.g. “M24”)
- Course Enrollment (e.g. 200)

Course	Title	M24	S24	W24	F23	M23	S23	W23	F22	M22	S22	W22
PSTAT 5A	Understanding Data	77	260	283	553	112	215	279	499	116	194	278
PSTAT 5H	Statistics	60	305	299	26	NA	31	22	4	NA	NA	NA
PSTAT 5LS	Stat. Life Sciences	60	305	299	NA	39	300	294	NA	63	294	284

*Only the first 3 rows are displayed; the dataset contains info from **all** undergrad PSTAT courses*

Desired Plot

Desired Plot: Visualize the changes in enrollments over time among only lower-division PSTAT courses (PSTAT 5A, 5H, 5LS, 8, and 10)

First Problem

- We only want lower-division courses.
- **Resolution:** we need to **slice** the data frame

```
ld_names = ["PSTAT 5A", "PSTAT 5LS", "PSTAT 8", "PSTAT 10"]
lower_divs = enrollments[enrollments.Course.isin(ld_names)]

lower_divs
```

```
      Course                 Title   M24    S24  ...    F22    M22    S22    W22
0   PSTAT 5A  UNDERSTANDING DATA  77.0  260.0  ...  499.0  116.0  194.0  278.0
2  PSTAT 5LS  STAT LIFE SCIENCES  60.0  305.0  ...    NaN   63.0  294.0  284.0
3    PSTAT 8  TRANS DS PROB STAT  22.0   83.0  ...   70.0   23.0   59.0    NaN
4   PSTAT 10  DATA SCIENCE PRINC  93.0  234.0  ...  257.0   87.0  278.0  287.0

[4 rows x 13 columns]
```

Second Problem

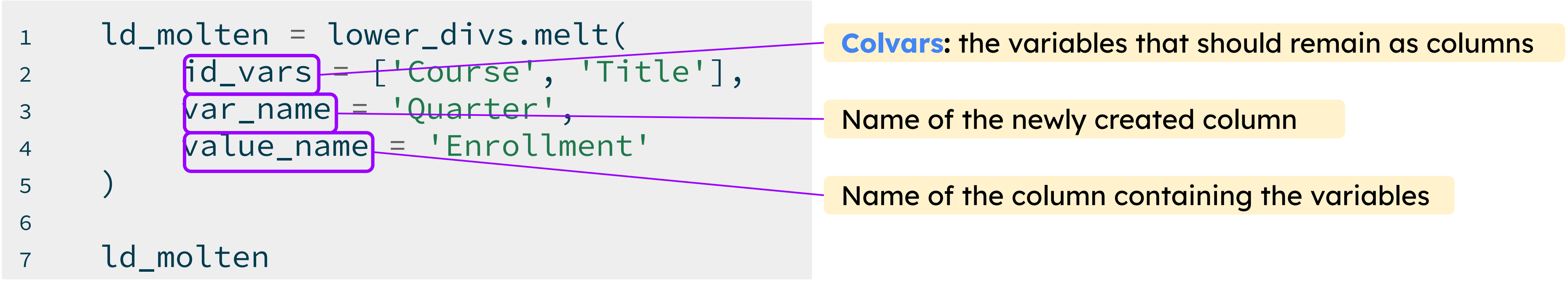
- The dataset is not tidy.
  - Specifically, the `Quarter` variable does not appear as a column, but rather its values are spread out as column headers

```
     Course               Title   M24    S24  ...    F22    M22    S22    W22
0   PSTAT 5A  UNDERSTANDING DATA  77.0  260.0  ...  499.0  116.0  194.0  278.0
2  PSTAT 5LS  STAT LIFE SCIENCES  60.0  305.0  ...    NaN   63.0  294.0  284.0
3    PSTAT 8  TRANS DS PROB STAT  22.0   83.0  ...   70.0   23.0   59.0    NaN
4   PSTAT 10  DATA SCIENCE PRINC  93.0  234.0  ...  257.0   87.0  278.0  287.0
```

```
      Course               Title Quarter  Enrollment
0    PSTAT 5A  UNDERSTANDING DATA     M24        77.0
1   PSTAT 5LS  STAT LIFE SCIENCES     M24        60.0
2     PSTAT 8  TRANS DS PROB STAT     M24        22.0
3    PSTAT 10  DATA SCIENCE PRINC     M24        93.0
4    PSTAT 5A  UNDERSTANDING DATA     S24       260.0
5   PSTAT 5LS  STAT LIFE SCIENCES     S24       305.0
6     PSTAT 8  TRANS DS PROB STAT     S24        83.0
7    PSTAT 10  DATA SCIENCE PRINC     S24       234.0
8    PSTAT 5A  UNDERSTANDING DATA     W24       283.0
...
```

Second Problem

- This operation is called **melting**, and is a common operation to transform messy (the opposite of tidy) data to tidy data.



```
ld_molten = lower_divs.melt(
        id_vars = ['Course', 'Title'],
        var_name = 'Quarter',
        value_name = 'Enrollment'
)

ld_molten
```

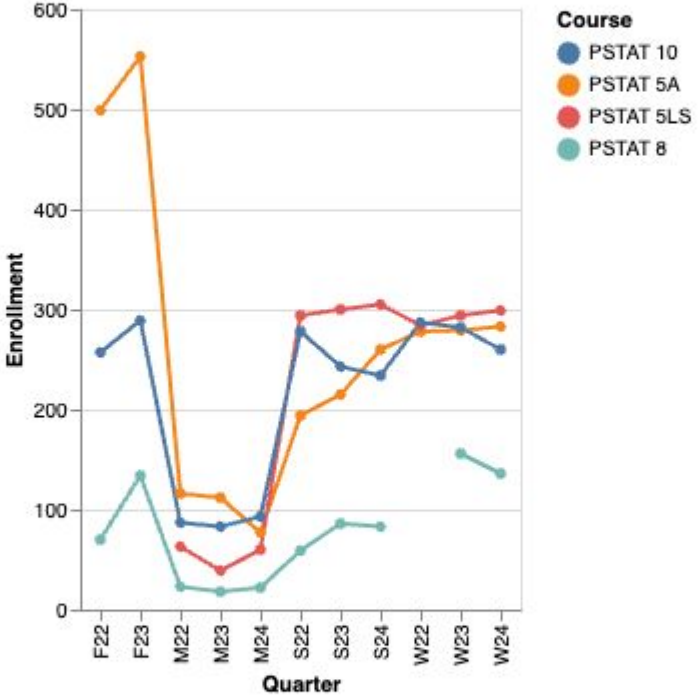
Colvars: the variables that should remain as columns

Name of the newly created column

Name of the column containing the variables

```
      Course                 Title Quarter  Enrollment
0    PSTAT 5A   UNDERSTANDING DATA     M24        77.0
1   PSTAT 5LS   STAT LIFE SCIENCES     M24        60.0
2     PSTAT 8   TRANS DS PROB STAT     M24        22.0
3    PSTAT 10   DATA SCIENCE PRINC     M24        93.0
4    PSTAT 5A   UNDERSTANDING DATA     S24       260.0
5   PSTAT 5LS   STAT LIFE SCIENCES     S24       305.0
6     PSTAT 8   TRANS DS PROB STAT     S24        83.0
7    PSTAT 10   DATA SCIENCE PRINC     S24       234.0
8    PSTAT 5A   UNDERSTANDING DATA     W24       283.0
```

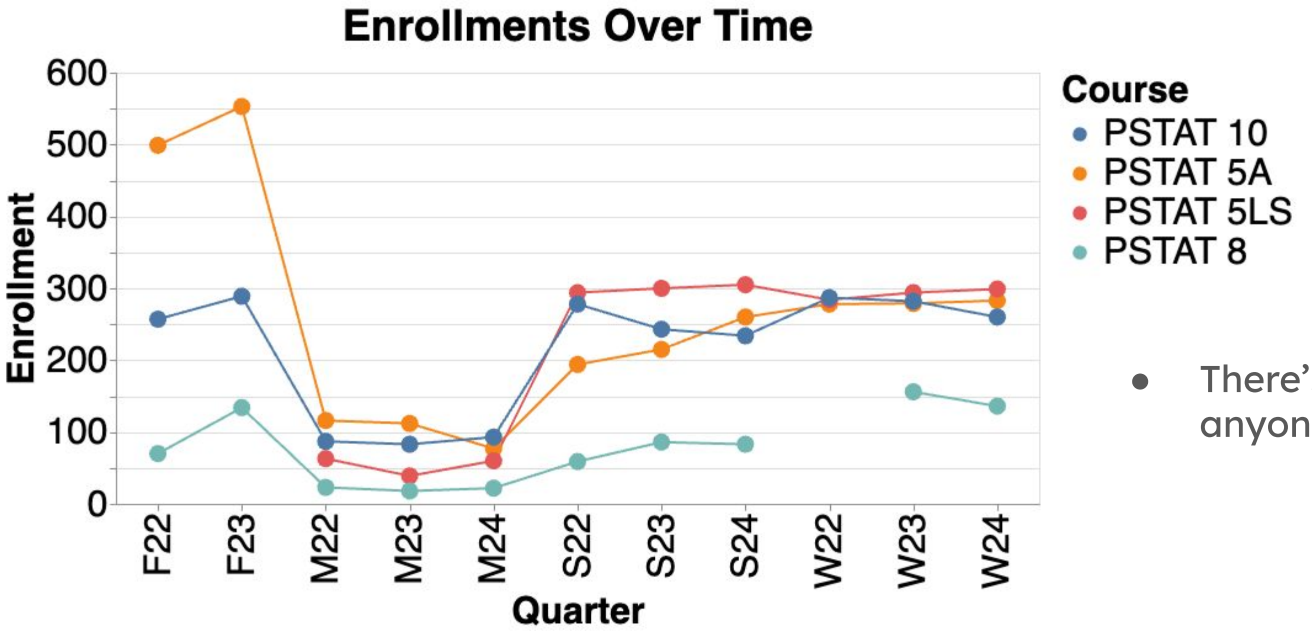
Plot: First Attempt



A few issues:

- Dimensions are off
- Font sizes are too small
- Points are too small
- Missing Title

Plot: Second Attempt



- There’s still one more issue - can anyone spot it?

Some Takeaways

- The act of making visualizations is **highly iterative** - you'll often start with a plot, then realize that certain things need to be changed, which will lead you to create another plot, and so on and so forth.
    - To that end, it's useful to distinguish between **exploratory visualizations** and **presentation-quality visualizations**.
    - Exploratory visualizations are meant for "internal use" and can be less formatted; they're meant to be "quick and dirty"
    - Presentation-quality visualizations are those destined for an audience, and should be more carefully crafted
- It often pays to have an idea of the type of visualization you want before starting on the coding!
- In fact, let's talk a bit about which plot to use when.

What Type of Plot, When?

Numerical vs. Categorical Variables

- At the highest level of classification, variables can be labeled as either **numerical** or **categorical** (sometimes called **quantitative** and **qualitative**, respectively).
- Numerical variables are those whose observations take the form of numbers.
    - Examples include height, weight, population size, GDP, etc.
- Categorical variables are those whose observations take the form of categories.
    - Examples include: species, opinions, letter grades, etc.

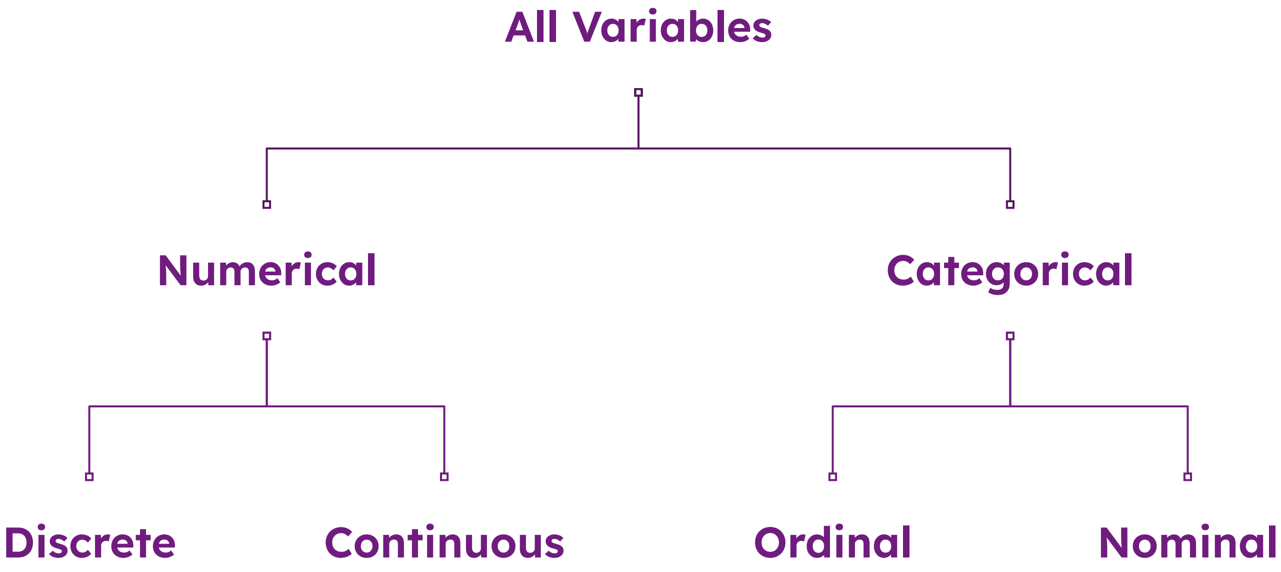
CAUTION

It is a *faux-pas* to conclude that a variable is numerical solely based on the fact that its observations are numbers.

- For instance, we can encode observations on months (e.g. months in which we have rain in Santa Barbara) using 01 for January, 02 for February, etc.
- This does not make month a numerical variable; 1 + 2 is 3, but January + February is not March!

Second Level of Classification

- There is actually a second level of classification that is sometimes used to further subdivide numerical and categorical variables:



All Variables

Numerical

Categorical

Discrete

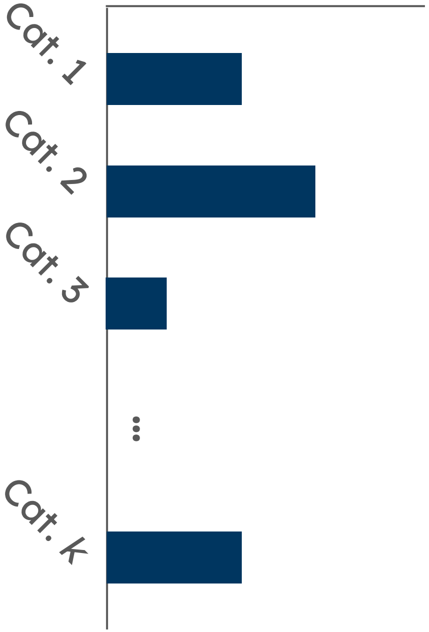
Continuous

Ordinal

Nominal

Bargraphs

- The ideal type of graph to visualize the distribution of a categorical variable is the **barplot/bargraph.**
- In general, if we have *k* categories we will have *k* bars, one for each category and with height proportional to the frequency of the corresponding category.

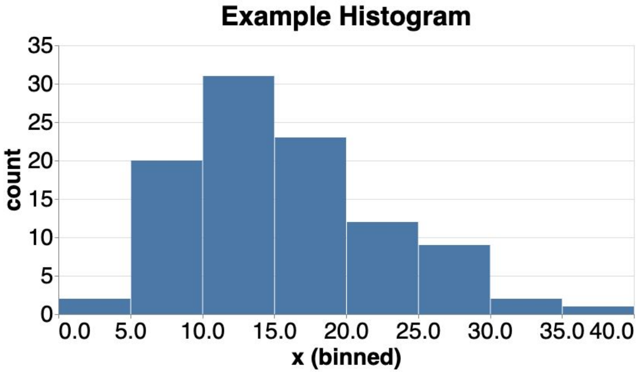


- A tabular specification of the categories and their corresponding frequencies is often called a **frequency table**.

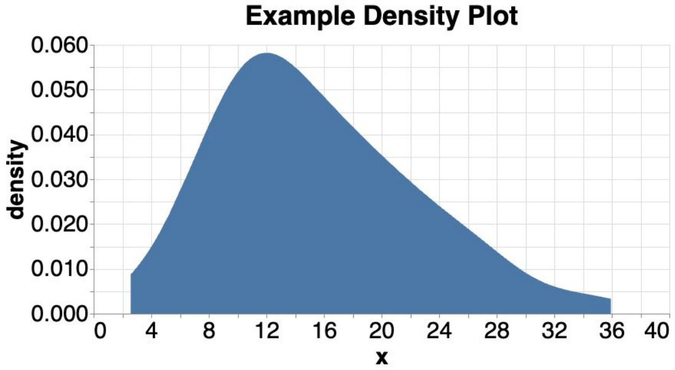
Cat. 1	Cat. 2	…	Cat. *k*
f_1	f_2		f_k

Numerical Variables

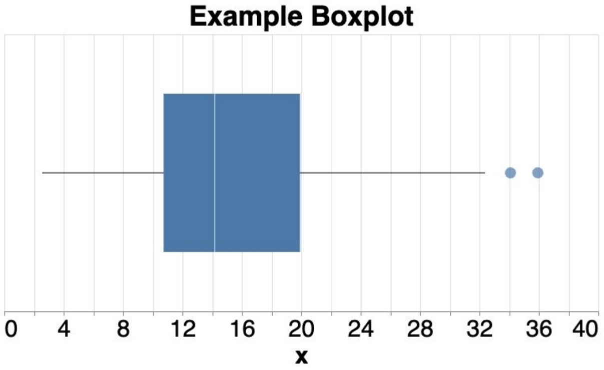
- There are many different types of plots that can be used to visualize the distribution of a numerical variable.



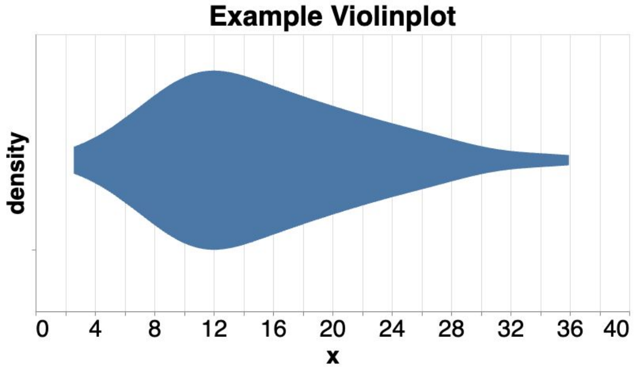
Example Histogram



Example Density Plot



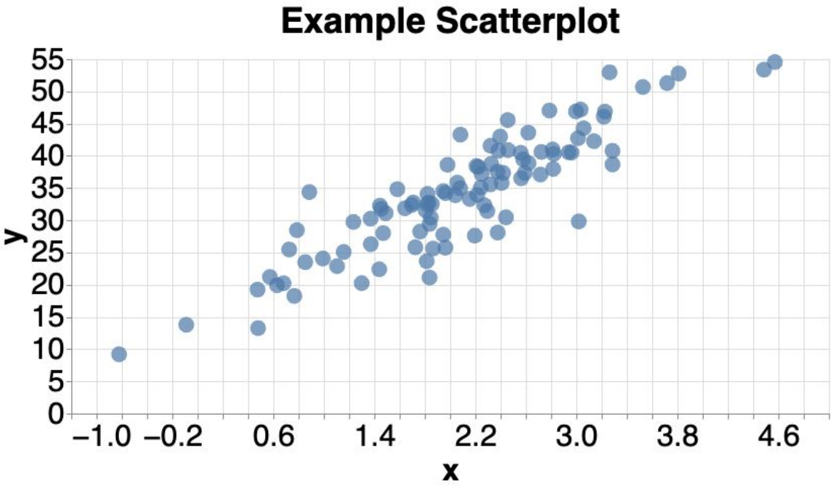
Example Boxplot

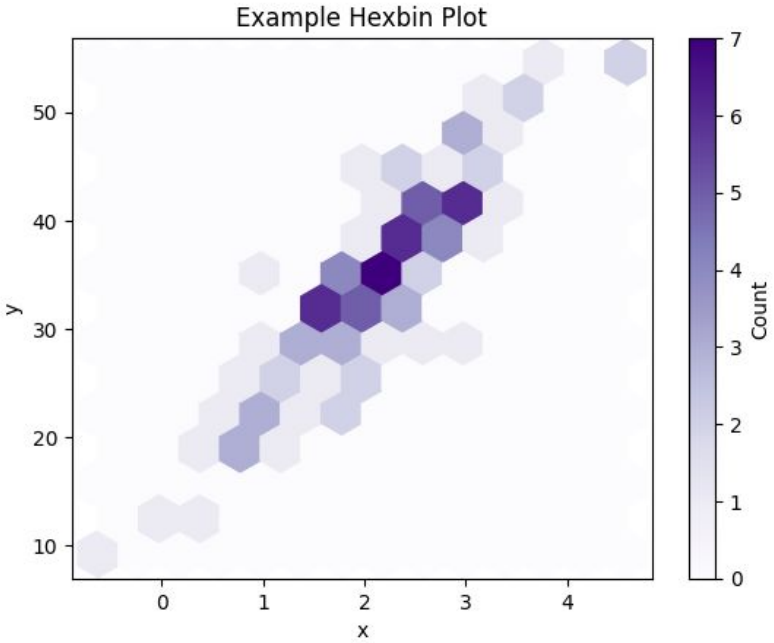


Example Violinplot

Bivariate Plots

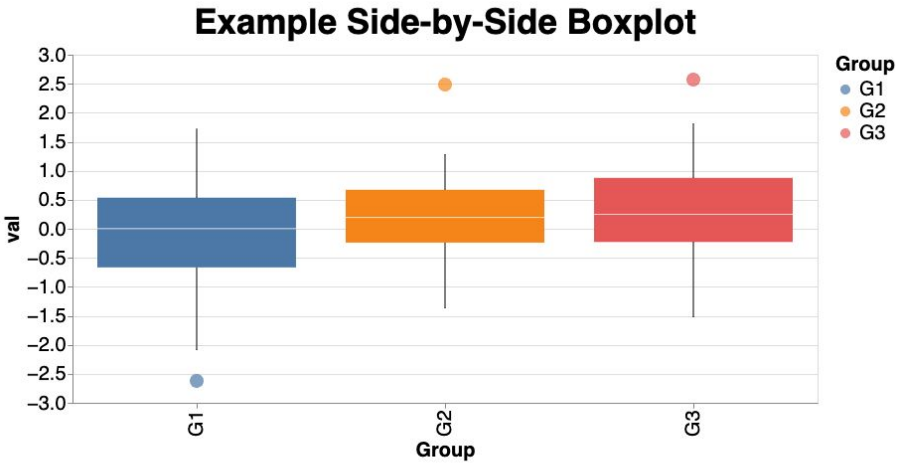
- To compare two numerical variables, we most often use a **scatterplot**
  - If one of the variables is time, we typically use a **lineplot** instead
- If desired, we can also produce a **hexagonal heatmap** (or **hexbin plot**)

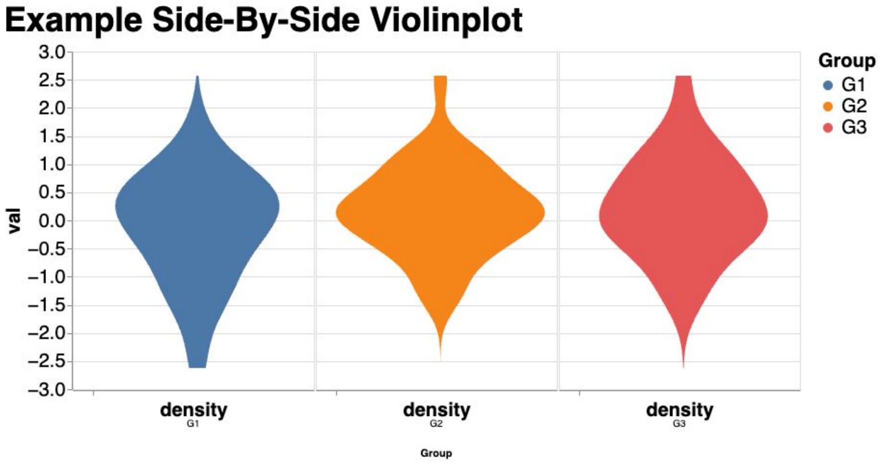




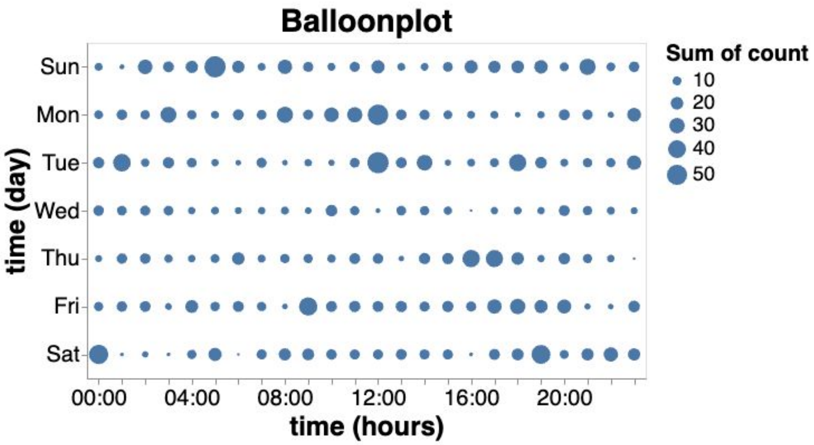
Bivariate Plots

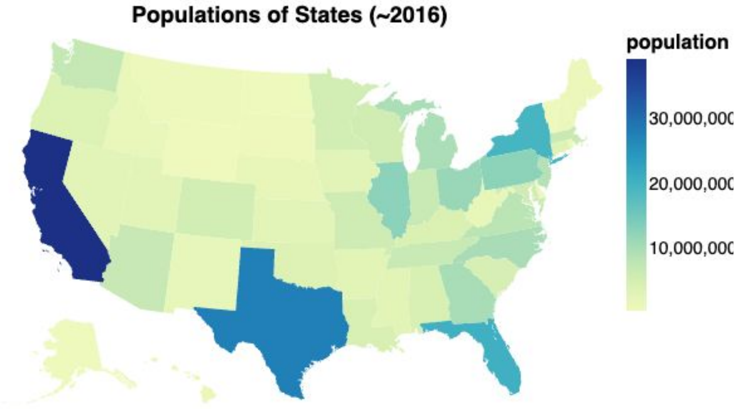
- Two compare a numerical and a categorical variable, we use either a **side-by-side boxplot** or a **side-by-side violinplot**

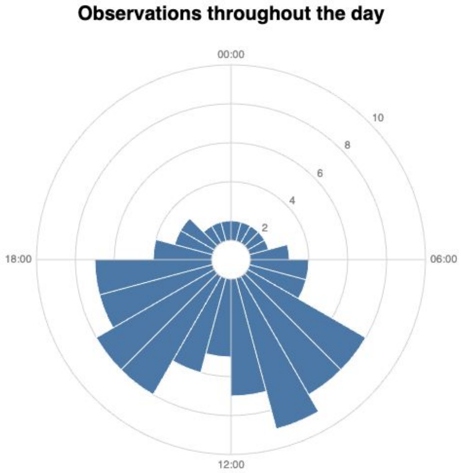


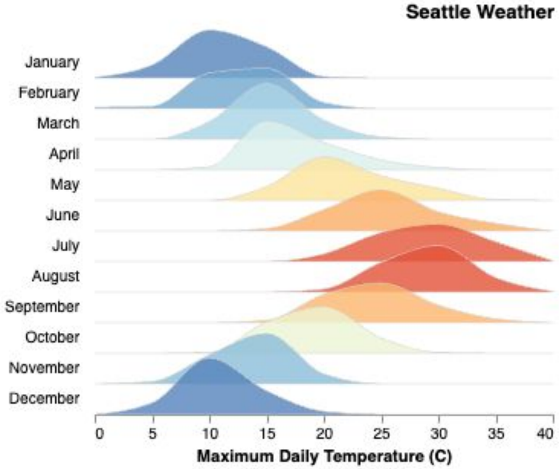


Lots of Other Plots Too!


Balloonplot


Populations of States (~2016)


Observations throughout the day


Seattle Weather

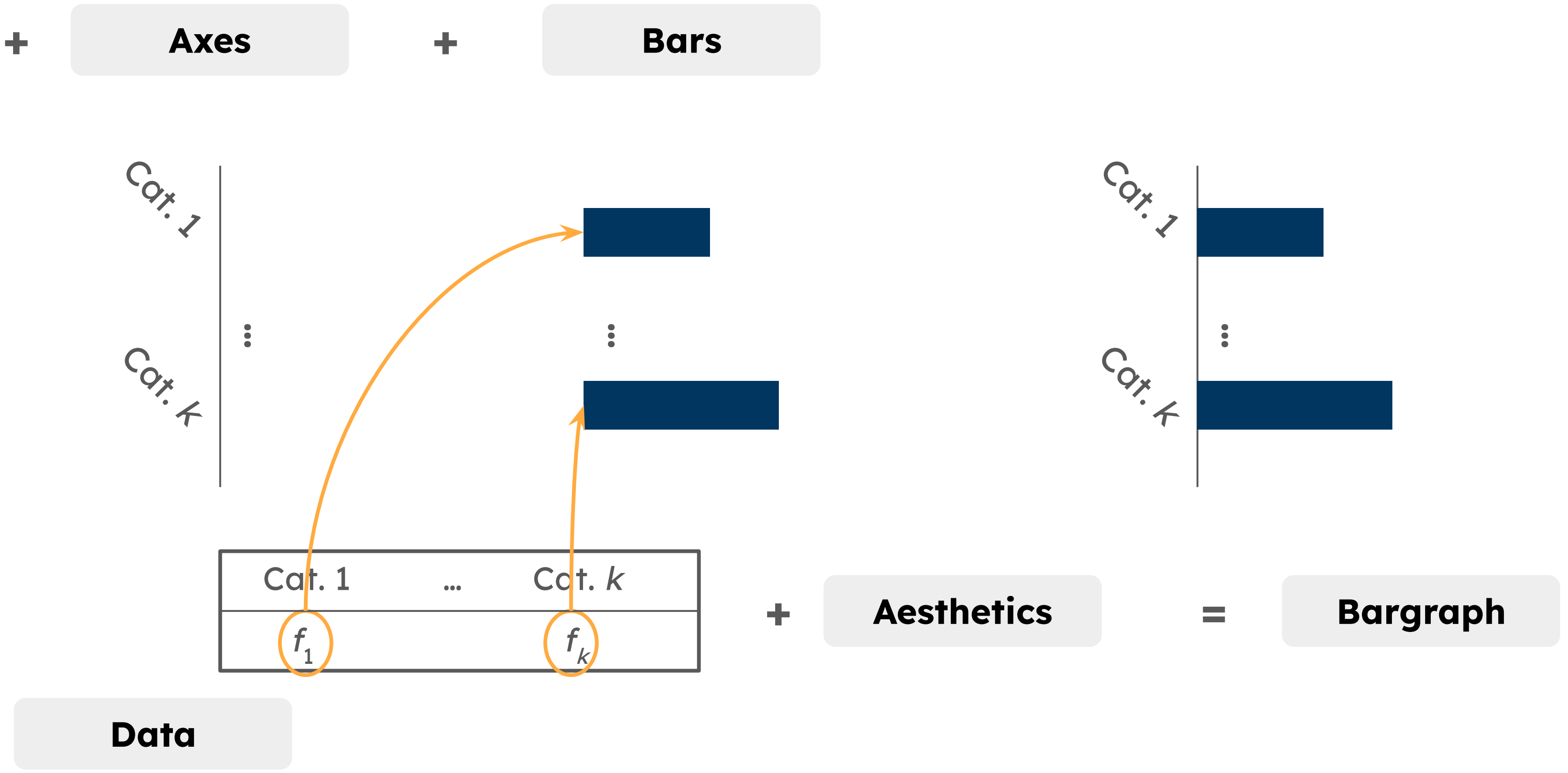
https://altair-viz.github.io/gallery/index.html

Grammar of Graphics

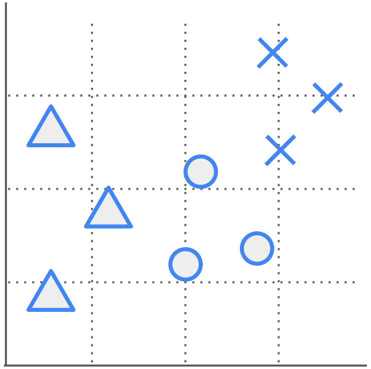
Grammar of Graphics

- Most of us can probably make a graph pretty easily, with a pen, paper, and a ruler.
- But what are the actual components that go into a visualization?
  - It is important to consider this question to be able to figure out how to make *computers* generate graphs and visualizations
- First, we need **data**.
- Next, we need to specify **axes** / a **coordinate system**.
  - What variable goes on the *x*-axis? What about the *y*-axis? Should we include a radial axis (useful for **directional data**)? Should we make a **map**?
- Finally we need **geometric objects** (shortened to **geoms**)
  - Do we need bars, or points? Lines, or sectors? Etc.
- To map our data to our geoms, we need to specify **aesthetics** (e.g. coordinates, heights of bars, etc.)

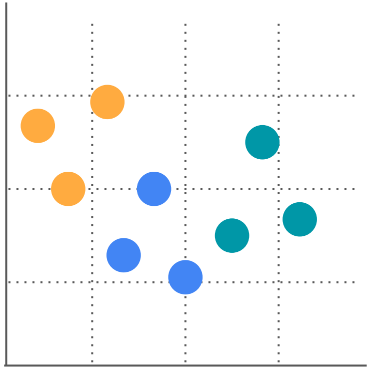
Grammar of Graphics



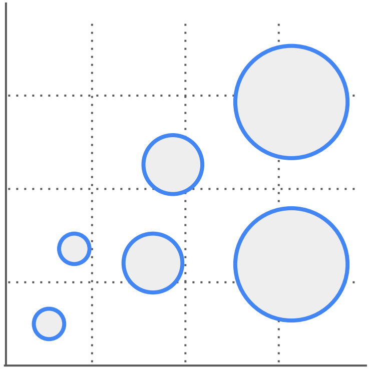
Common Aesthetics



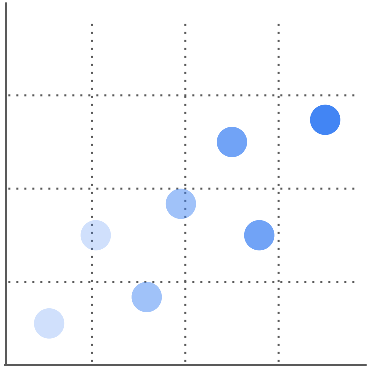
Shapes (points)



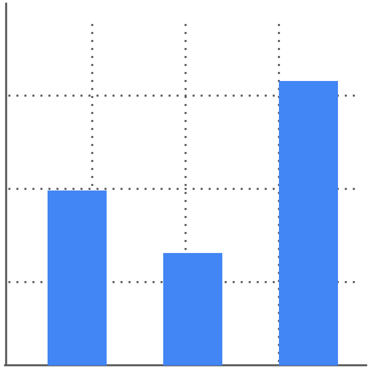
Color



Size



Opacity



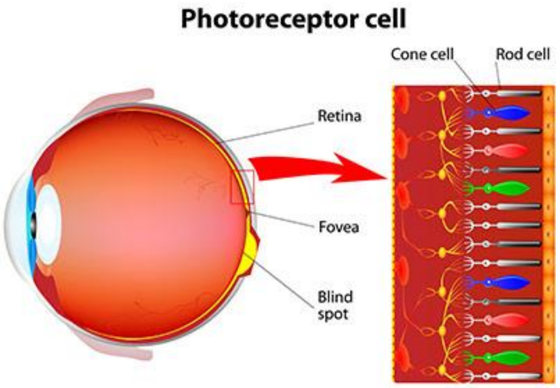
Height

Some Notes on Color

CVD and Accessibility

- Especially when it comes to presentation-oriented graphics, accessibility is key.
- For one thing, it is important to keep in mind that many readers may suffer from **Color-Vision Deficiency** (**CVD**; aka colorblindness), and may not be able to easily perceive differences in colors.
    - **Deuteranomaly**: difficulty perceiving green
    - **Protanomaly**: difficulty perceiving red
    - **Tritanomaly**: difficulty perceiving blue → "Blue-yellow colorblindness"

"Red-green colorblindness"



Trichromatic persons (i.e. people with no colorblindness) possess all three retinal cone cell types (and have cone cell types that function "as expected", and are therefore able to process and perceive red, green, and blue light

Fig. Source: https://www.aao.org/eye-health/anatomy/cones

CVD and Accessibility

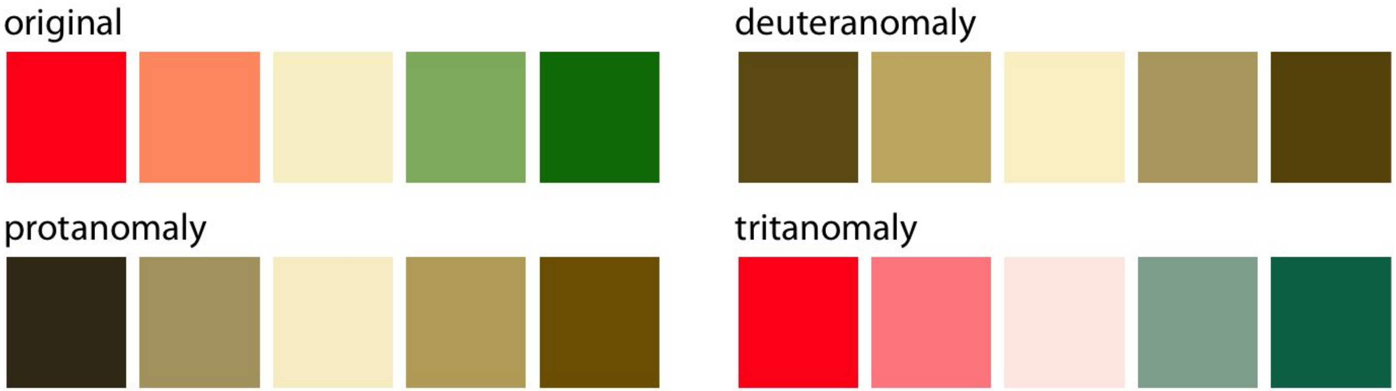


Figure 19.7 from *Fundamentals of Data Visualization*: A red–green contrast becomes indistinguishable under red–green cvd (deuteranomaly or protanomaly).

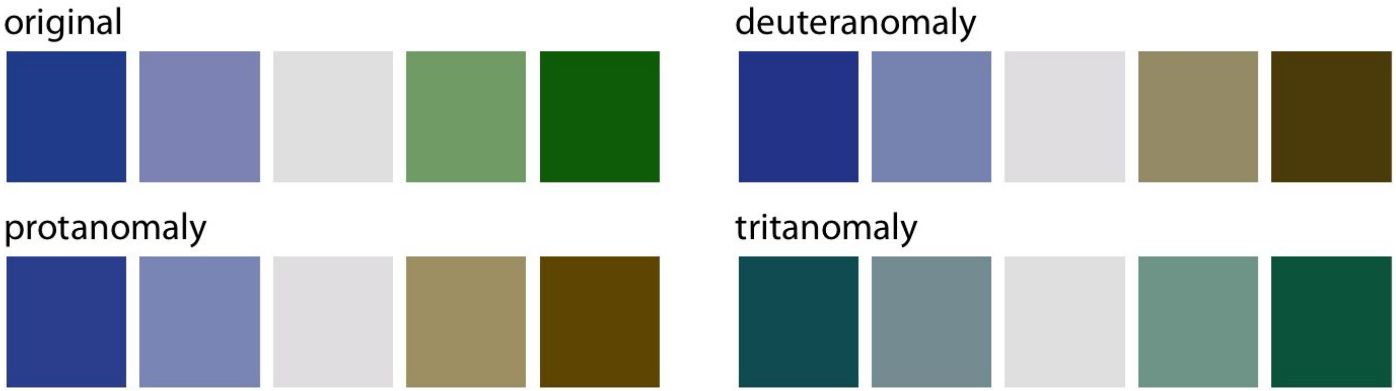


Figure 19.8 from *Fundamentals of Data Visualization*: A blue–green contrast becomes indistinguishable under blue–yellow cvd (tritanomaly).

CVD and Accessibility

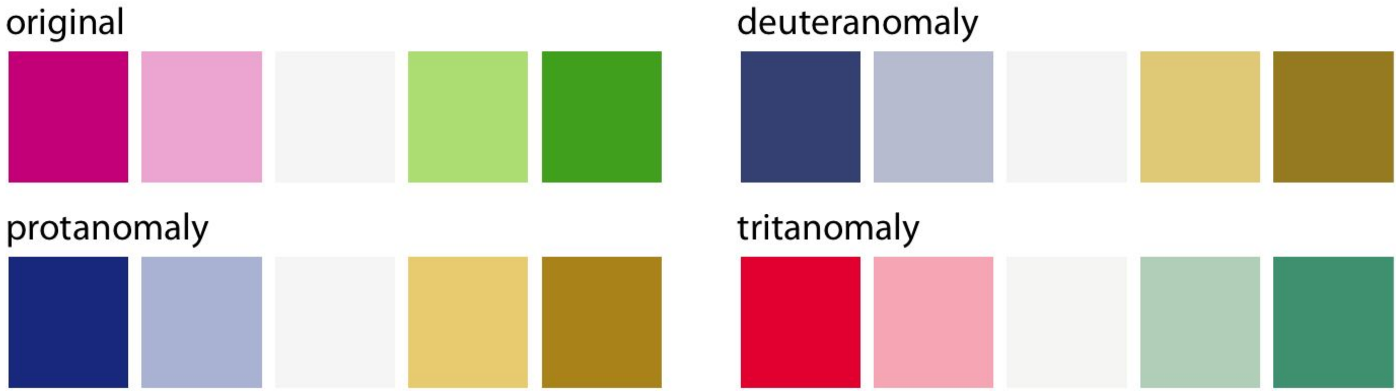


Figure 19.9 from *Fundamentals of Data Visualization*: The ColorBrewer PiYG (pink to yellow-green) scale from Figure 4.5 looks like a red–green contrast to people with regular color vision but works for all forms of color-vision deficiency. It works because the reddish color is actually pink (a mix of red and blue) while the greenish color also contains yellow. The difference in the blue component between the two colors can be picked up even by deutans or protans, and the difference in the red component can be picked up by tritans.

CVD and Accessibility

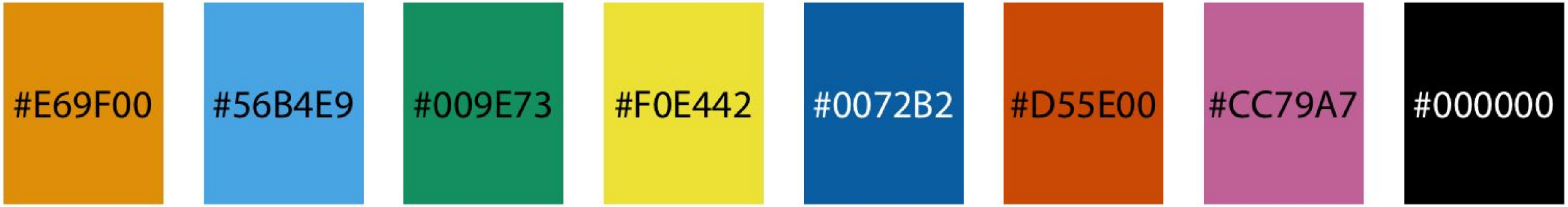


Figure 19.10 from *Fundamentals of Data Visualization*: Qualitative color palette for all color-vision deficiencies (Okabe and Ito 2008). The alphanumeric codes represent the colors in RGB space, encoded as hexadecimals. In many plot libraries and image-manipulation programs, you can just enter these codes directly. If your software does not take hexadecimals directly, you can also use the values in Table 19.1.

Color Scales

- There are three main types of color scales: **qualitative**, **sequential**, and **diverging**.
- Colors in a qualitative scale are meant to look visually different from one another, with no order, while still being “equal”.
  - Good when trying to color according to a categorial (i.e. *qualitative*) variable

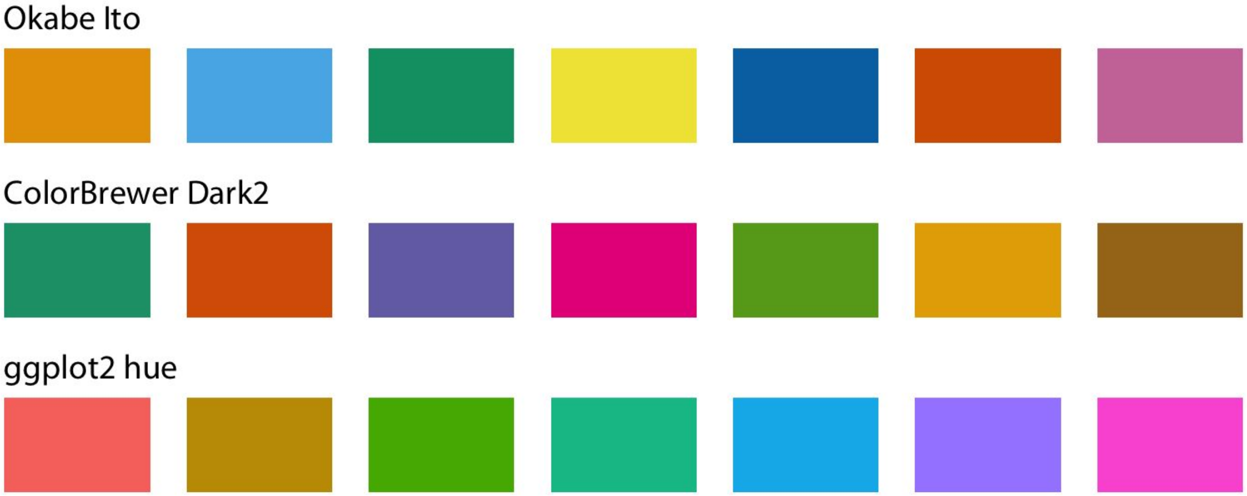


Figure 4.1 from *Fundamentals of Data Visualization*: Example qualitative color scales. The Okabe Ito scale is the default scale used throughout this book (Okabe and Ito 2008). The ColorBrewer Dark2 scale is provided by the ColorBrewer project (Brewer 2017). The ggplot2 hue scale is the default qualitative scale in the widely used plotting software ggplot2.

Color Scales

- Colors in a sequential scale are designed to convey an *ordering*, or *hierarchy* of values.
  - Good to use when low and high values have equal emphasis.
  - Can be based on a single hue, or across multiple hues
  - More akin to we call “gradients”, colloquially

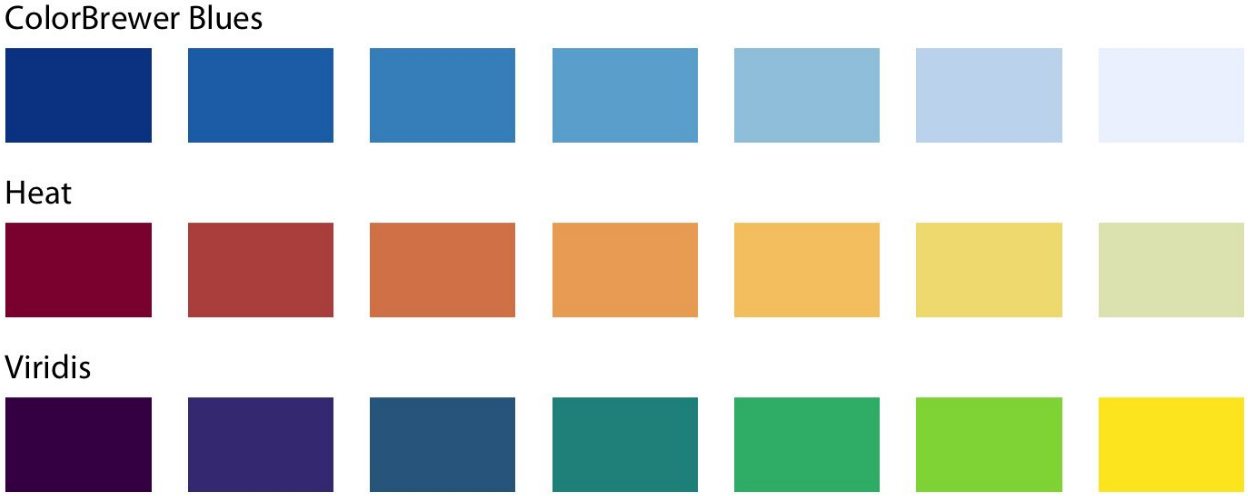


Figure 4.3 from *Fundamentals of Data Visualization*: Example sequential color scales. The ColorBrewer Blues scale is a monochromatic scale that varies from dark to light blue. The Heat and Viridis scales are multi-hue scales that vary from dark red to light yellow and from dark blue via green to light yellow, respectively.

Color Scales

- Colors in a diverging scale are designed to convey an *ordering*, or *hierarchy* of values *in two directions* (positive or negative)
    - Accomplished by effectively “stitching” together two sequential scales at a common, neutral midpoint.

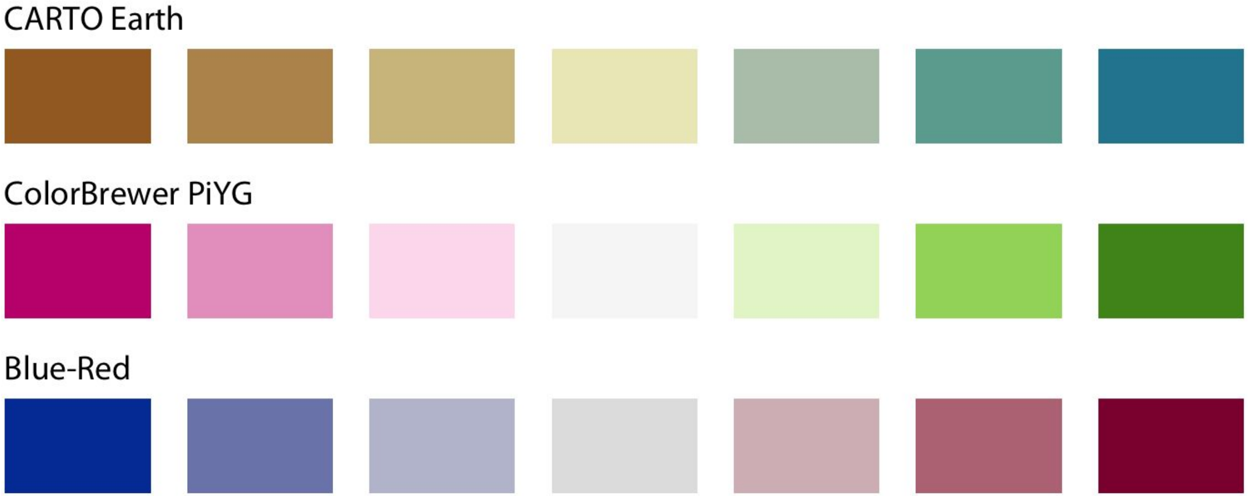
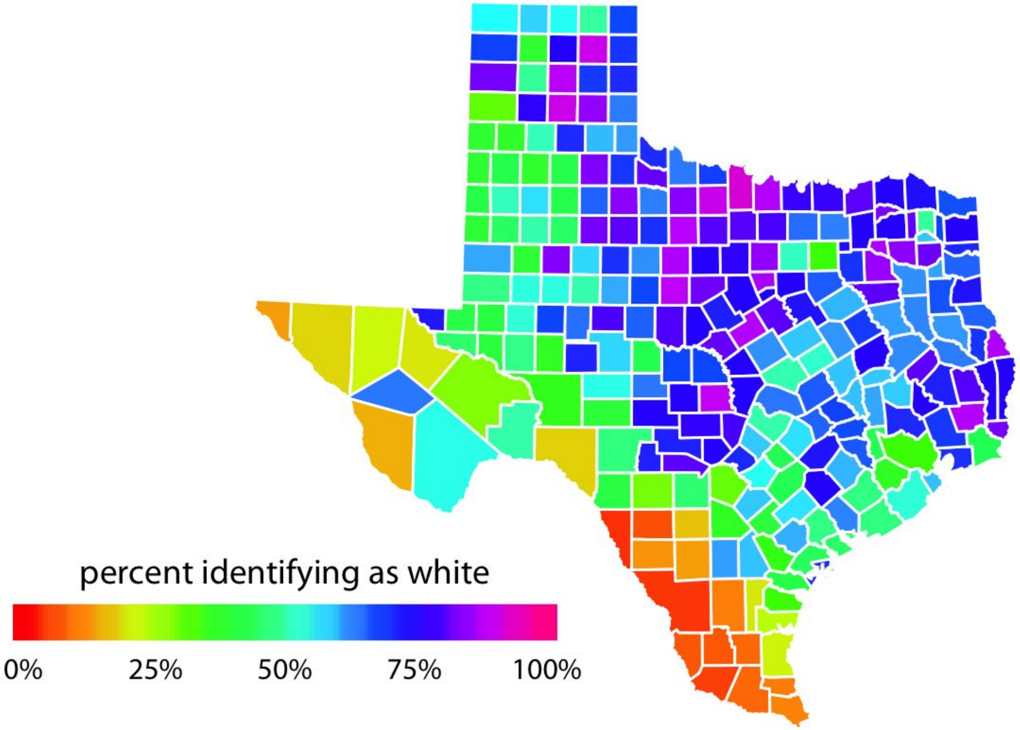


Figure 4.5 from *Fundamentals of Data Visualization*: Example diverging color scales. Diverging scales can be thought of as two sequential scales stiched together at a common midpoint color. Common color choices for diverging scales include brown to greenish blue, pink to yellow-green, and blue to red.

Color Scales



bad

Which scale is being used? How could we do better?

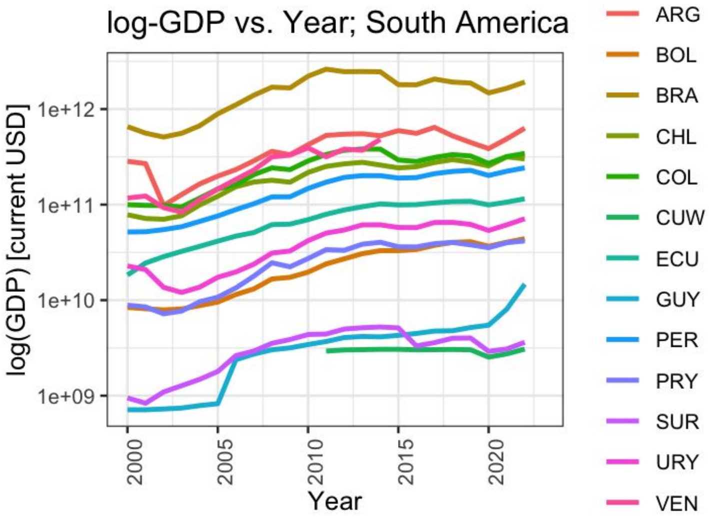
CVD and Accessibility

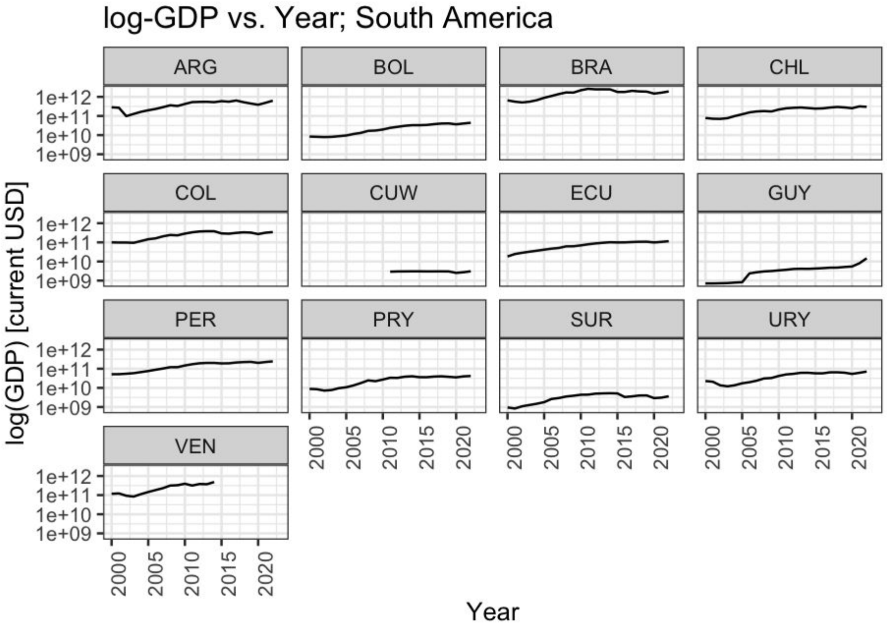
- It’s usually a good idea to test your figures using a CVD simulator.
- E.g. https://www.color-blindness.com/coblis-color-blindness-simulator/

- Another thing to consider is putting alt text in your figures (wherever possible).
  - This helps visually impaired readers, who most often utilize text-to-speech software.

Faceting

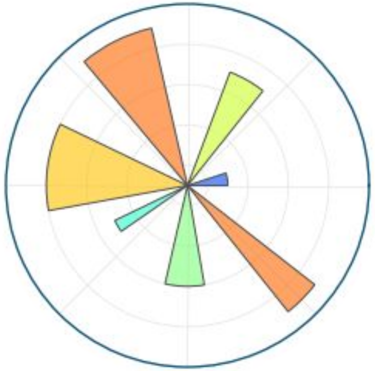
- If you have a categorical variable with more than 5-7 categories, using color to encode information about the variable may not be the best idea.
- Instead, consider **faceting**





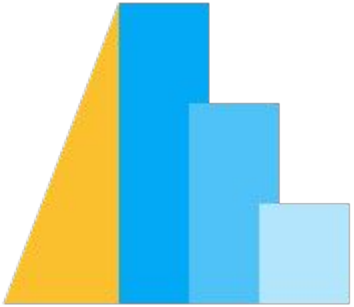
Plotting in Python

Plotting Libraries



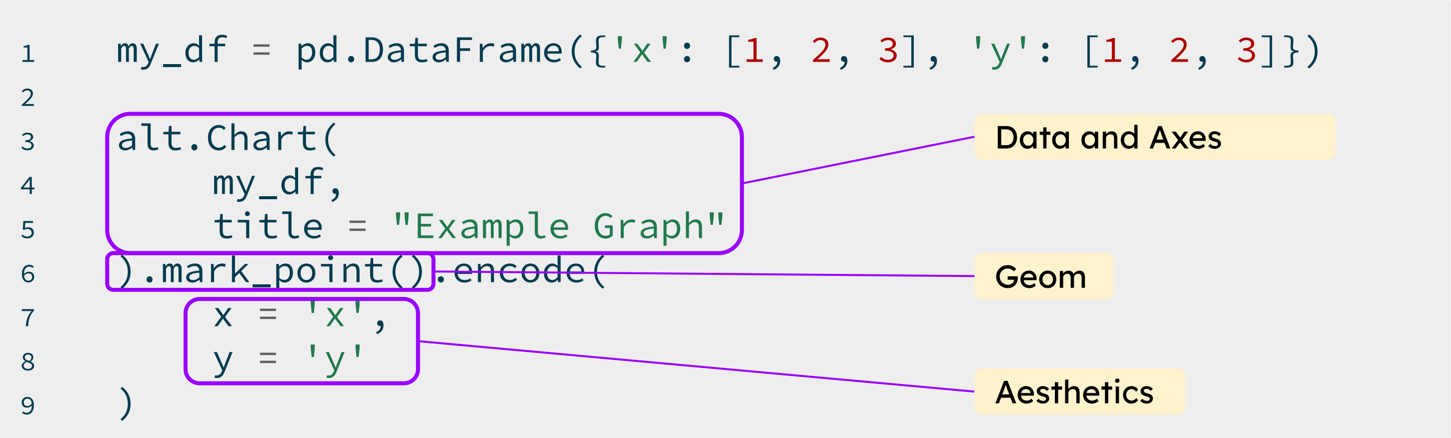
Matplotlib

Seaborn



Altair

Altair



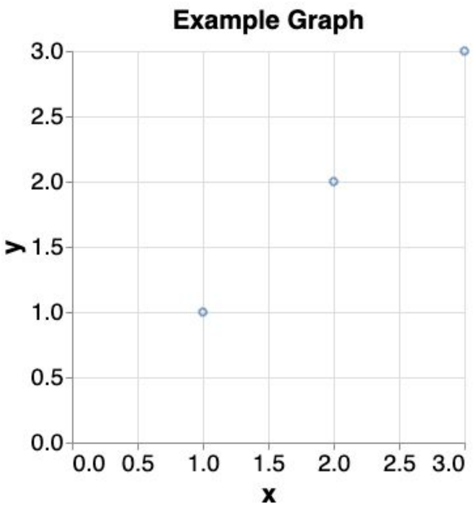
```
my_df = pd.DataFrame({'x': [1, 2, 3], 'y': [1, 2, 3]})

alt.Chart(
     my_df,
     title = "Example Graph"
).mark_point().encode(
     x = 'x',
     y = 'y'
)
```

Data and Axes

Geom

Aesthetics

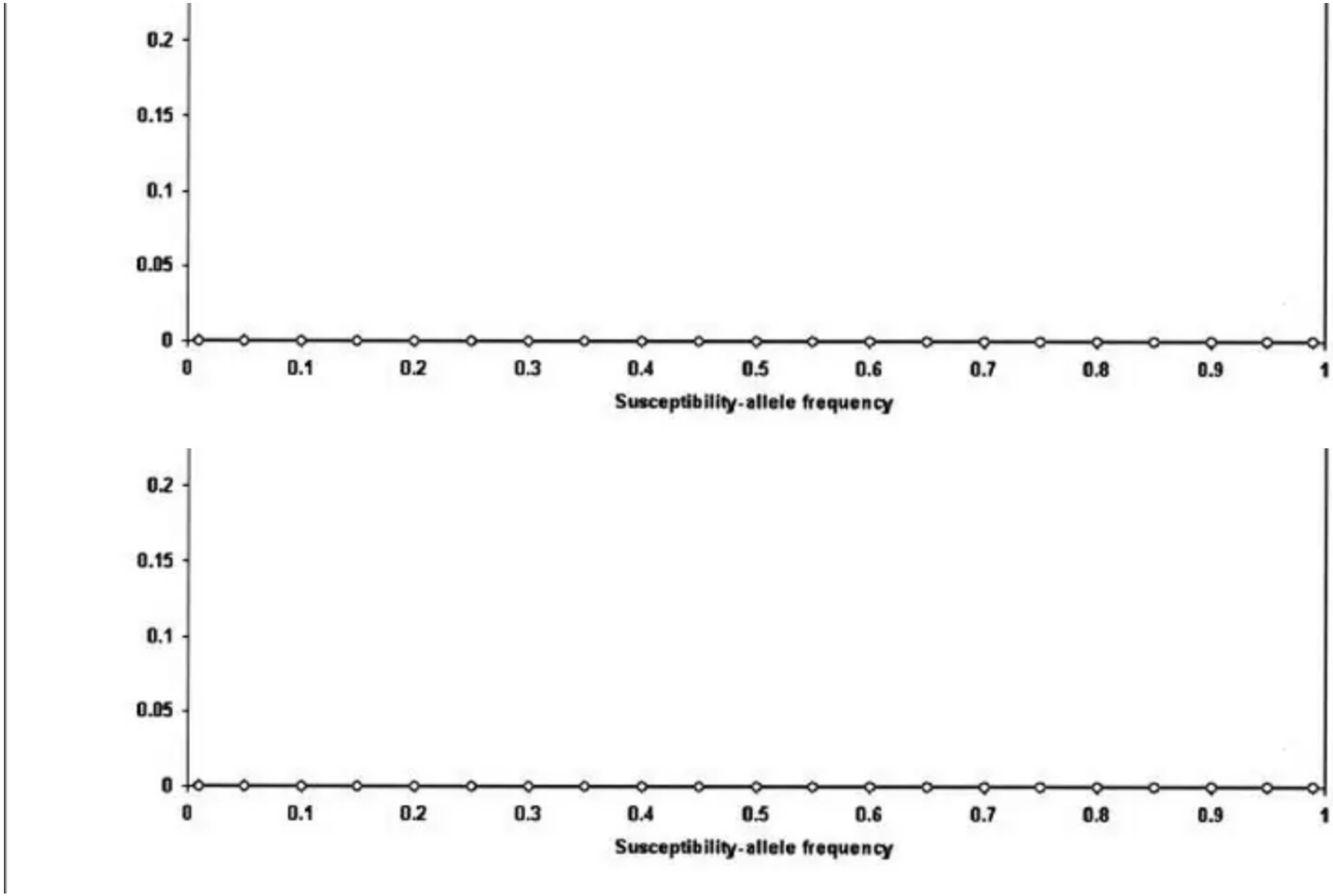


https://bit.ly/dsc-datavis1

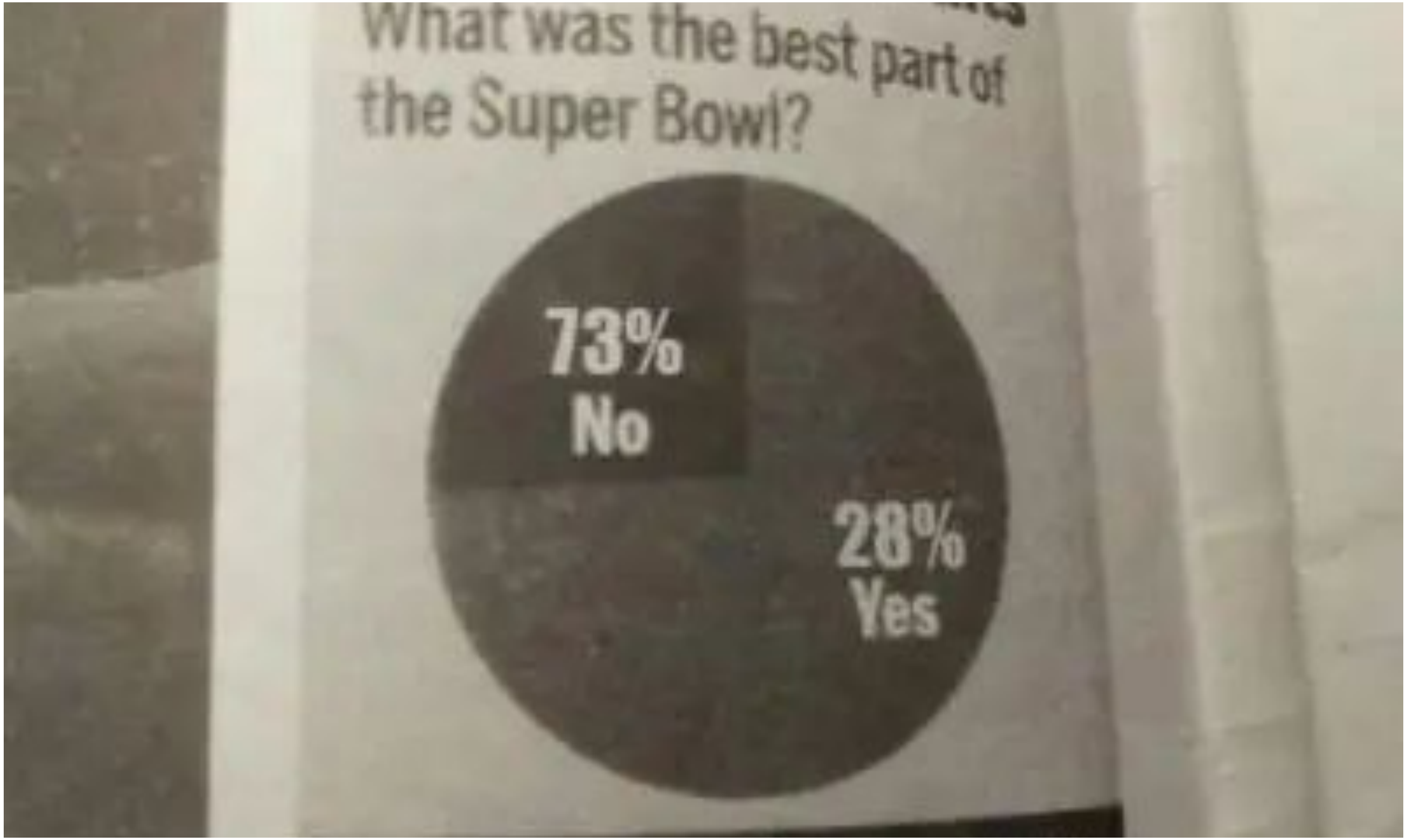
Now Presenting…

The Roast of Graphics!

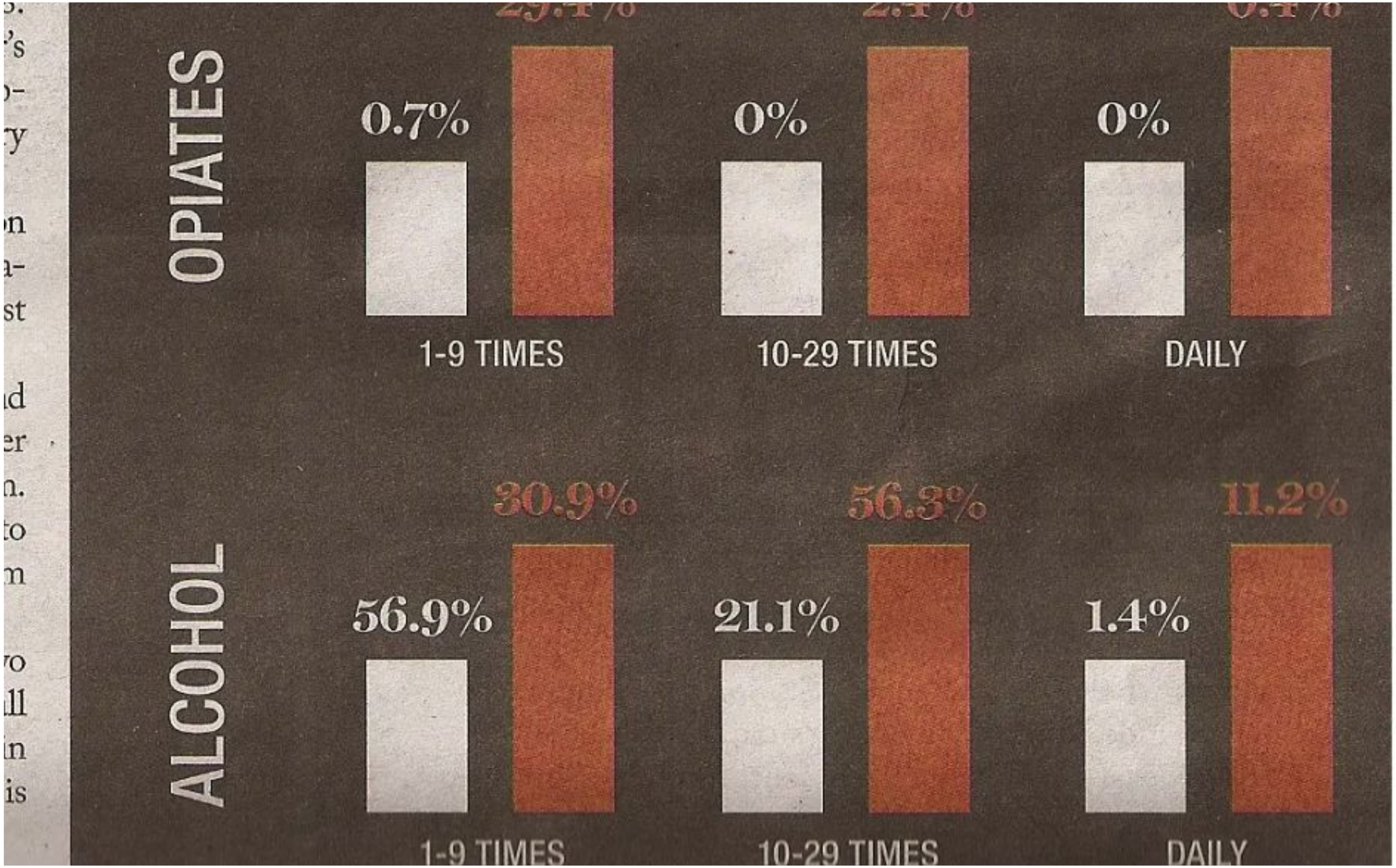
Plot 1


0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Susceptibility-allele frequency
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Susceptibility-allele frequency

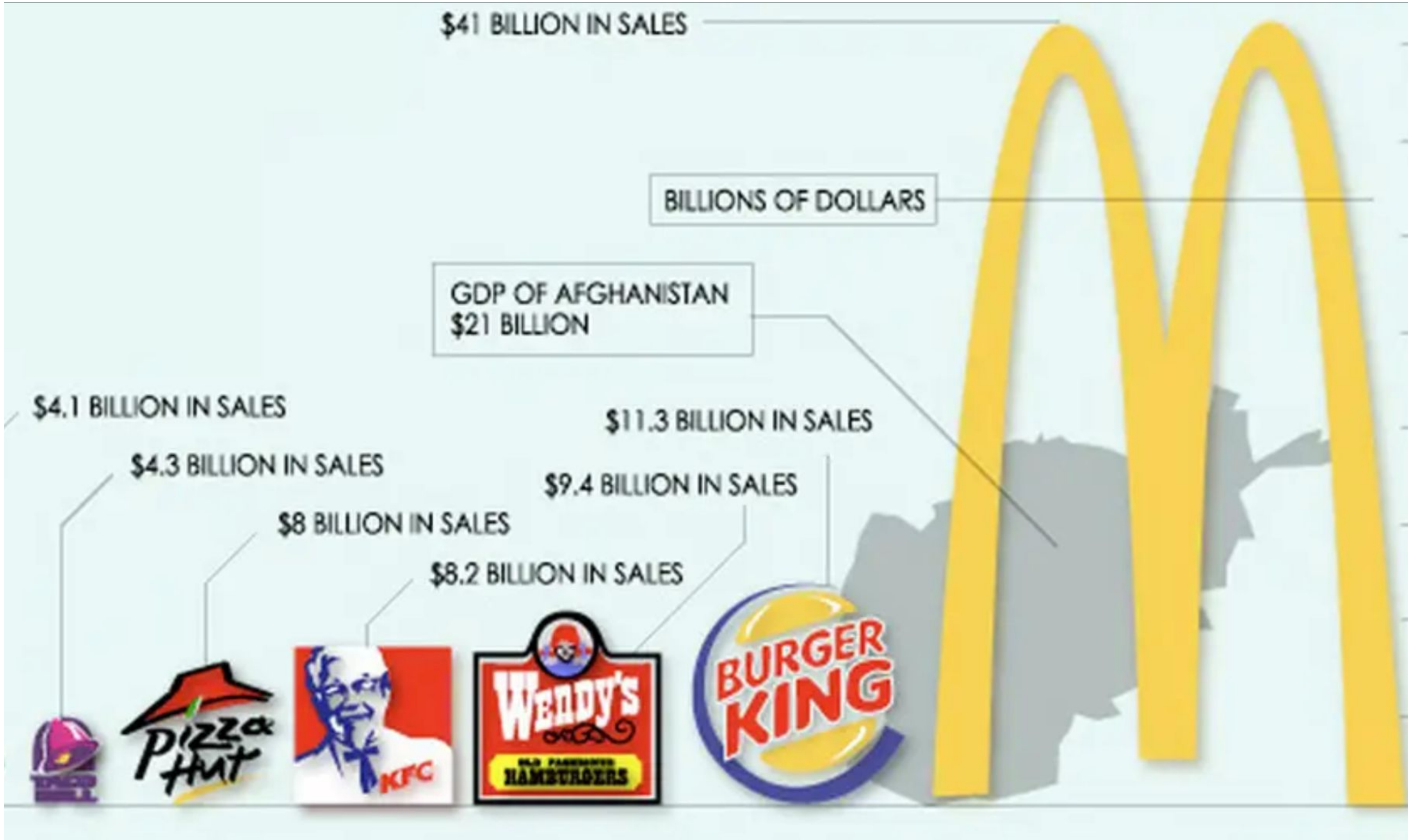
Plot 2



Plot 3

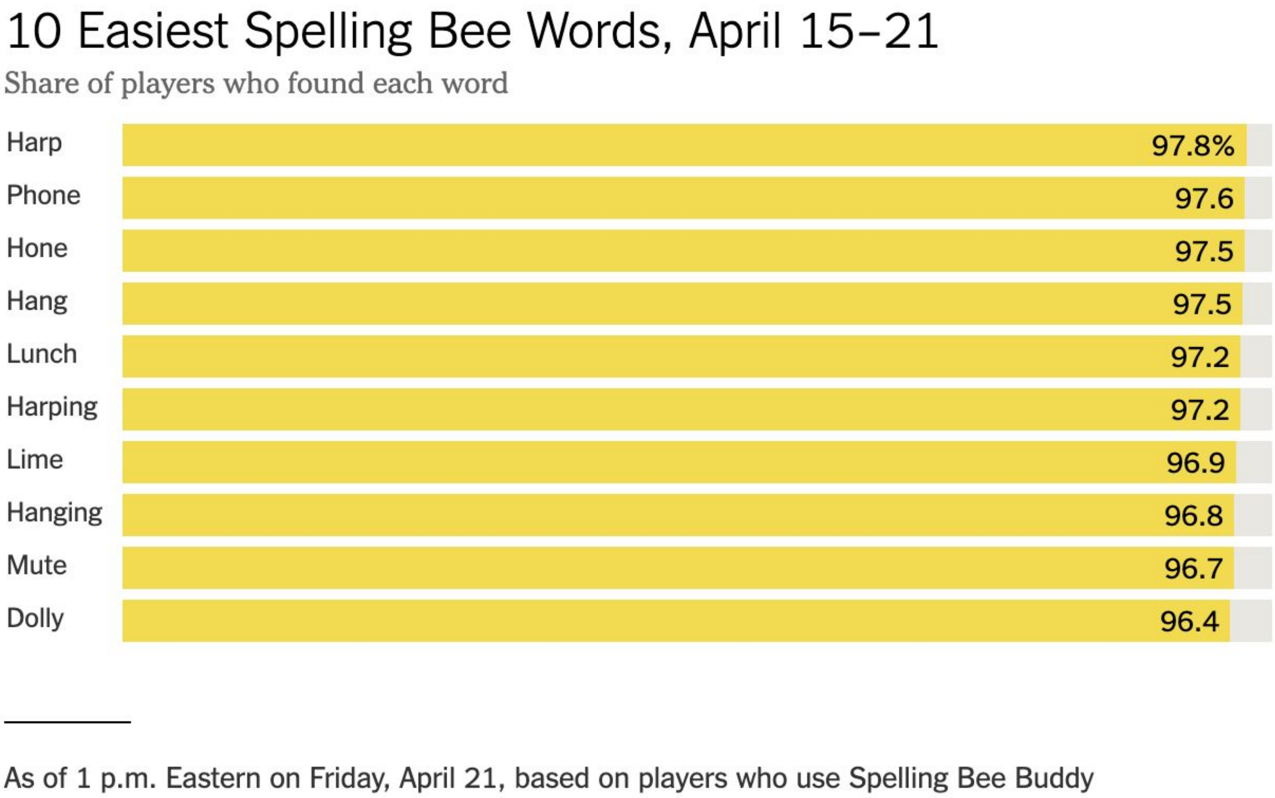

OPIATES
29.4%
0.7%
2.4%
0%
0.4%
0%
1-9 TIMES
10-29 TIMES
DAILY
ALCOHOL
30.9%
56.9%
56.3%
21.1%
11.2%
1.4%
1-9 TIMES
10-29 TIMES
DAILY

Plot 4



$41 BILLION IN SALES
BILLIONS OF DOLLARS
GDP OF AFGHANISTAN
$21 BILLION
$4.1 BILLION IN SALES
$4.3 BILLION IN SALES
$8 BILLION IN SALES
$8.2 BILLION IN SALES
$9.4 BILLION IN SALES
$11.3 BILLION IN SALES
Pizza Hut
KFC
Wendy's
OLD FASHIONED HAMBURGERS
BURGER KING

Plot 5



10 Easiest Spelling Bee Words, April 15–21

Share of players who found each word

As of 1 p.m. Eastern on Friday, April 21, based on players who use Spelling Bee Buddy

Source: https://www.nytimes.com/2023/04/24/upshot/spelling-bee-words.html

Principles of Visualizations

- I admit, making graphics is more of an art than a science!
- Here are a couple of principles I myself have found useful to keep in mind:

1. **Keep things simple**. You can (and in many cases *should*) try to communicate as much information as is effective. But, don't take it to an extreme.
    a. **3D-Styling is almost NEVER effective**. As neat and "cool" as 3D barplots might be, they just obfuscate their meaning too much to be truly effective graphics.
2. **Beware of Scales and Areas.** We'll talk about this one more in a bit - spoiler alert, pie charts are a notoriously bad graphic!
3. **Label Axes, and Title your Plots.** This one should (hopefully) be self-explanatory, but make sure you are using descriptive (but not overly complex) labels for your axes, and make sure your plots are titled.
4. **Interpret your plots**. All too often I see "floating" plots - that is, figures that appear mysteriously and suddenly with no explanation whatsoever. No matter how self-explanatory you think your plot is, make sure you actively describe it and its conclusions somewhere in your report.

Acknowledgements

- GIFs courtesy of giphy.com.
- Palmer penguins dataset courtesy of https://allisonhorst.github.io/palmerpenguins/
- Many “bad” visualizations taken from https://www.businessinsider.com/the-27-worst-charts-of-all-time-2013-6 ; all rights attributed to the original author
- Many figures are sourced from *Fundamentals of Data Visualization* by Claus O. Wilke - all rights attributed to the original author.
- The majority of plots appearing in this slide deck were generated using Altair (https://altair-viz.github.io/index.html), © Copyright 2016-2024, Vega-Altair Developers. All rights reserved.
- Materials adapted from the Spring 2024 iteration of PSTAT 100 (taught by Ethan P. Marzban); material accessible at https://ucsb-pstat100.github.io/