LECTURE 11

Visualization, Part 2

Principles of sound visualizations; smoothing and transformations.

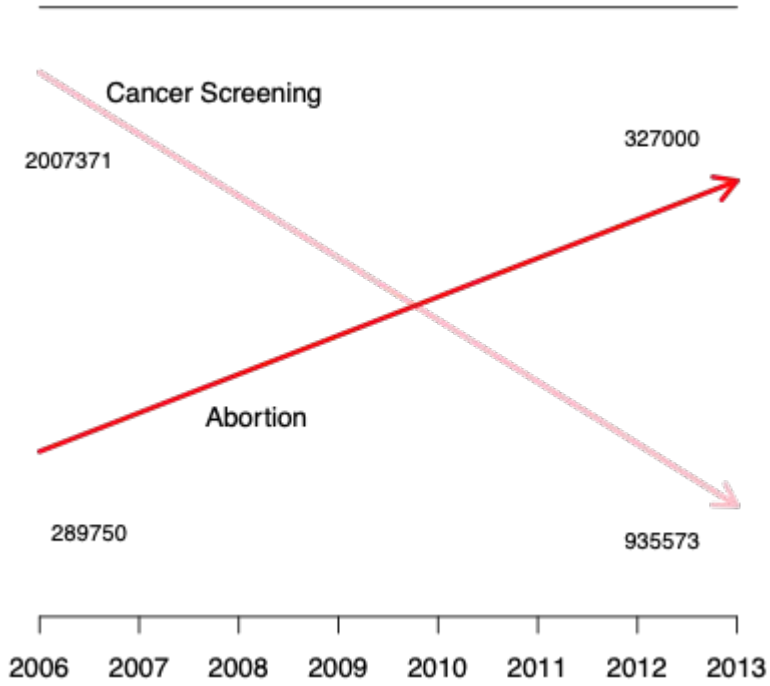
Data 100/Data 200, Fall 2020 @ UC Berkeley
Fernando Pérez and Alvin Wan
Content credit: Suraj Rampure, Ani Adhikari, Sam Lau, Yifan Wu, Deborah Nolan

Overview

- In the first visualization lecture we talked about how to actually make visualizations.
- In this lecture, we will examine visualizations through the following four principles:
  - Scale.
  - Conditioning.
  - Perception.
  - Context.
- We'll also look at how kernel density estimates (KDEs) work, as a method of smoothing histograms.
- We will finish off by looking at transformations as a means to linearize relationships.

Scale

Case Study: Planned Parenthood Hearing

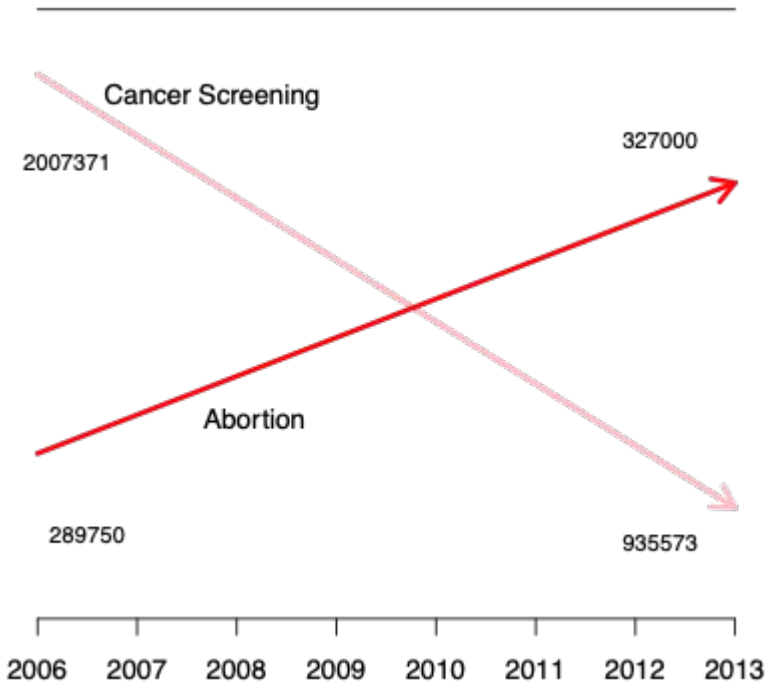


In 2015, Planned Parenthood was accused of selling aborted fetal tissue for profit.

Congressman Chaffetz (R-UT) showed this plot which originally appeared in a report by Americans United for Life.

- What is this graph plotting?
- What message is this plot trying to convey?
- Is anything suspicious?

Keep axis scales consistent



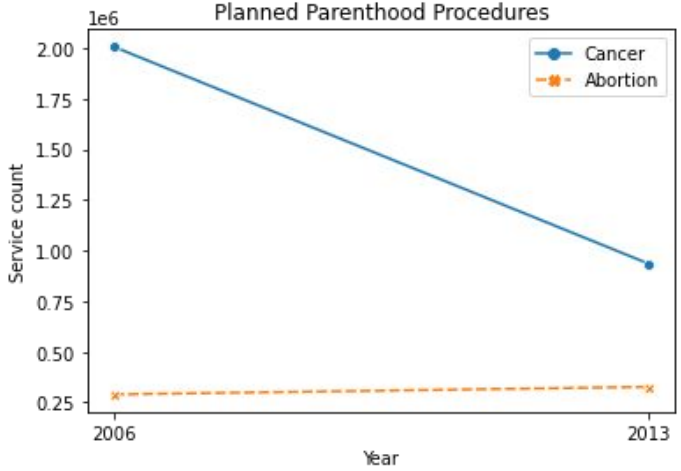
The scales for the two lines are completely different!

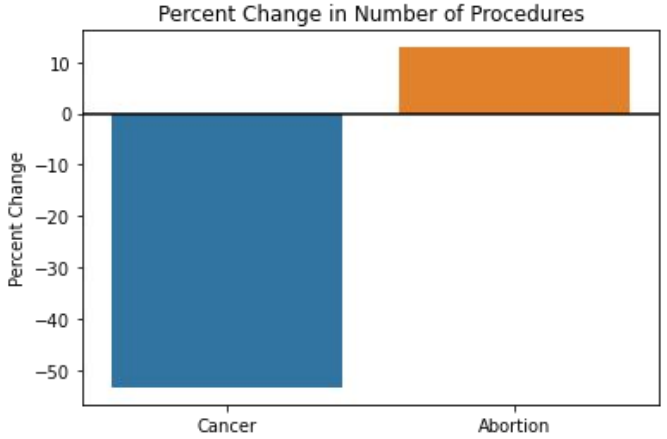
- 327000 is smaller than 935573, but appears to be way bigger.
- **Do not use two different scales for the same axis!**

Consider scale of the data

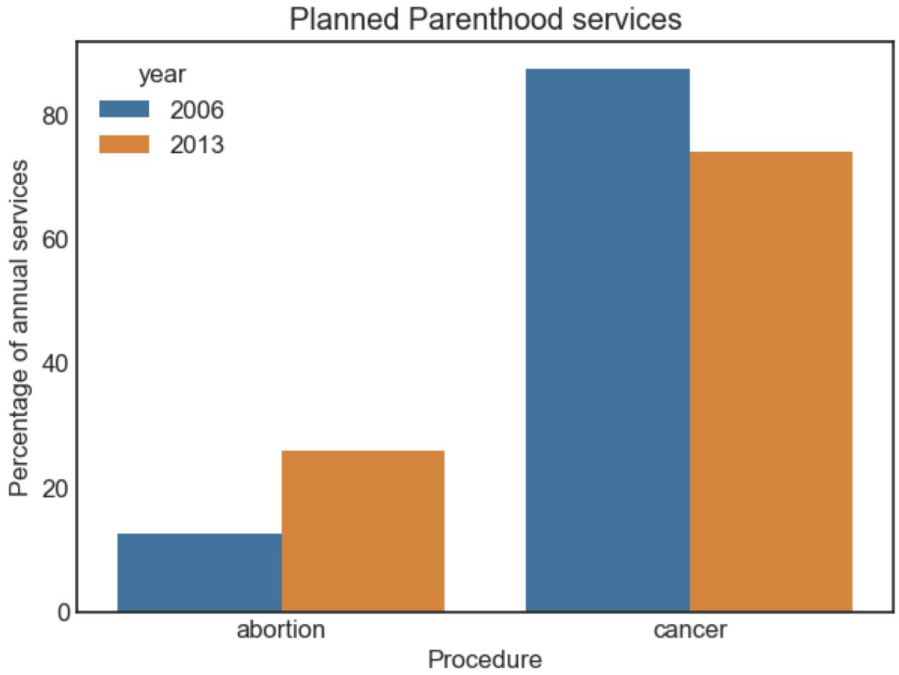
The top plot draws all of the data on the same scale.

- It clearly shows there was a dramatic drop in cancer screenings by PP.
- But there are still far more cancer screenings than abortions.
- Can plot percentage change instead of raw counts (bottom). This shows that cancer screenings have decreased and abortions have increased, without being misleading.





Consider scale of the data

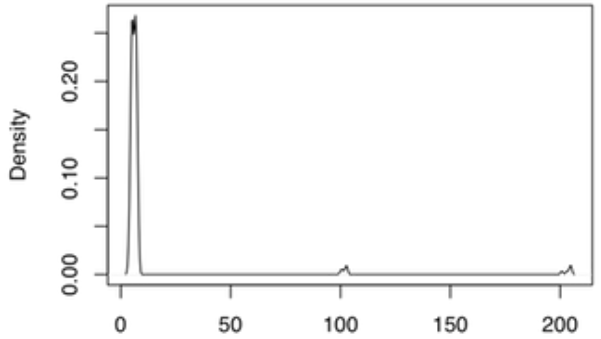


We could also visualize abortions and cancer screenings as a percentage of total procedures.

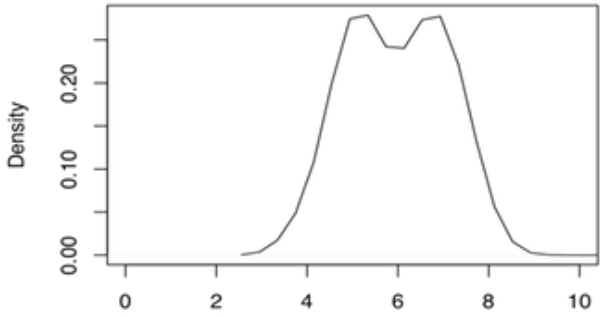
- Abortions increased from 13% to 26% of total procedures.

Reveal the data

- Choose axis limits to fill the visualization.
- If necessary:
  - Zoom in on the bulk of the data.
  - Create multiple plots to show different regions of interest.

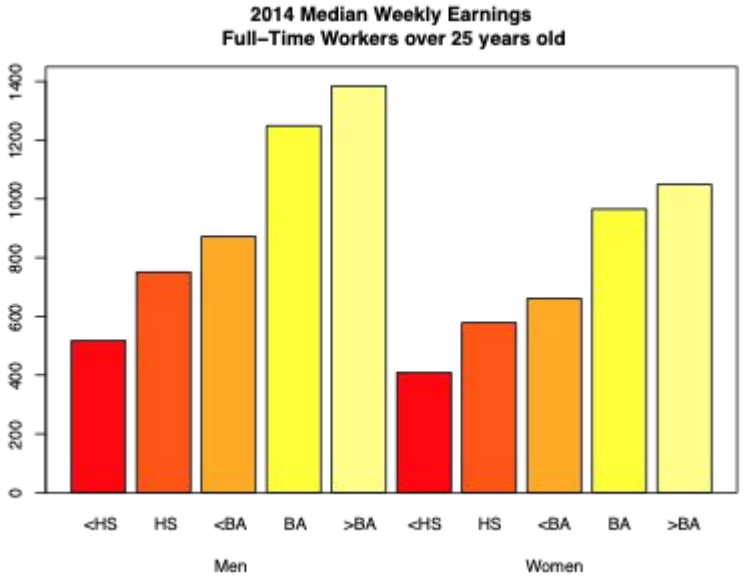


On the left, the bulk of the data is in the [0, 10] range on the x-axis.



Conditioning

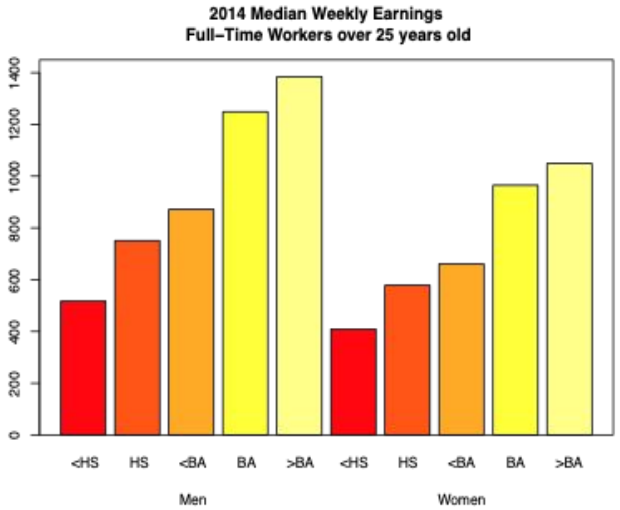
Case Study: Median Weekly Earnings

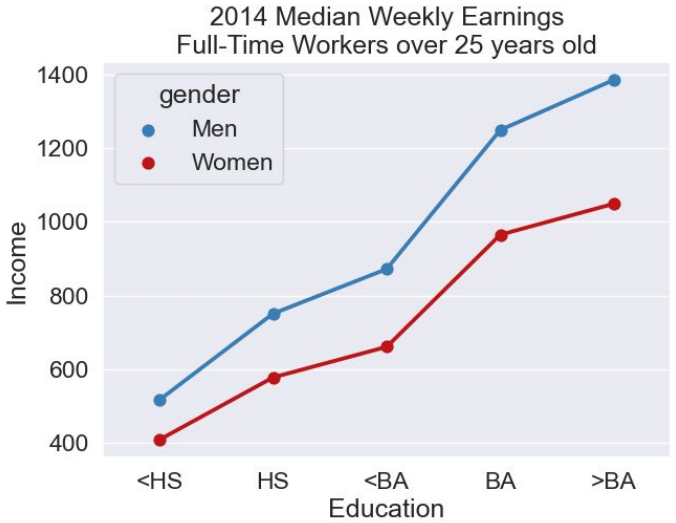


This data comes from the Bureau of Labor Statistics, who oversees surveys regarding the economic health of the US. They have plotted median weekly earnings for men and women by education level.

- What comparisons are made easily with this plot?
- What comparisons are most interesting and important?

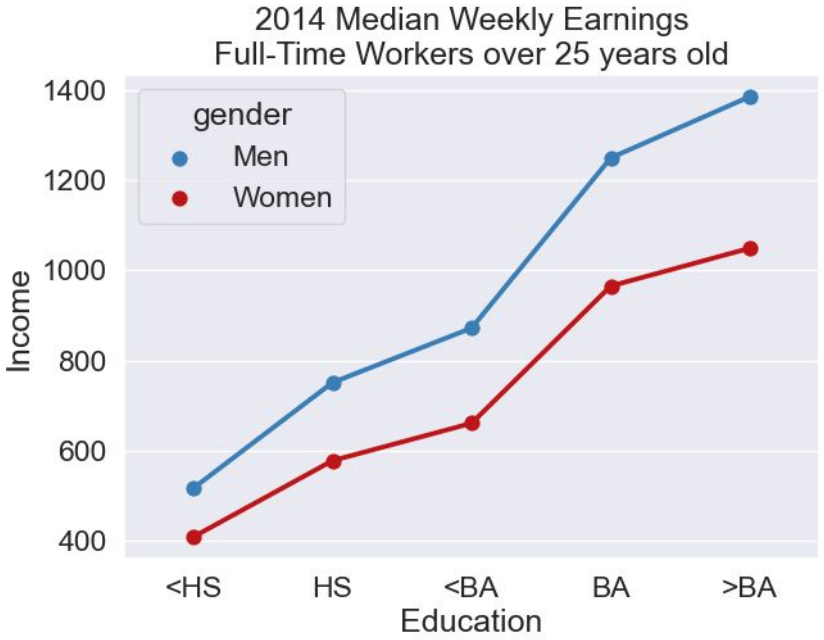
Use conditioning to aid comparison

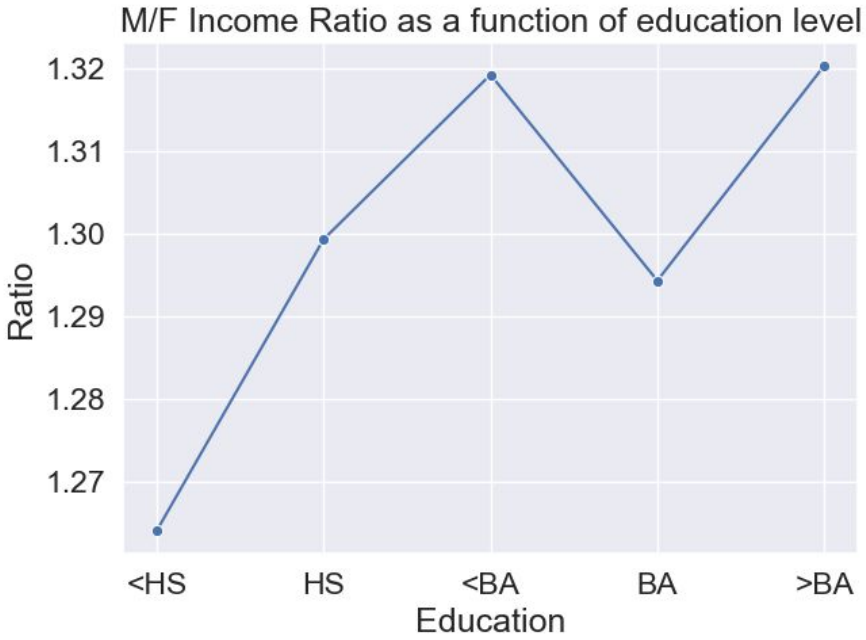




- Lines make it easy to see the large effect of having a BA on weekly earnings.
- Having two separate lines makes clear the wage difference between men and women.
  - It also highlights the fact that the wage difference increases, as education level does.

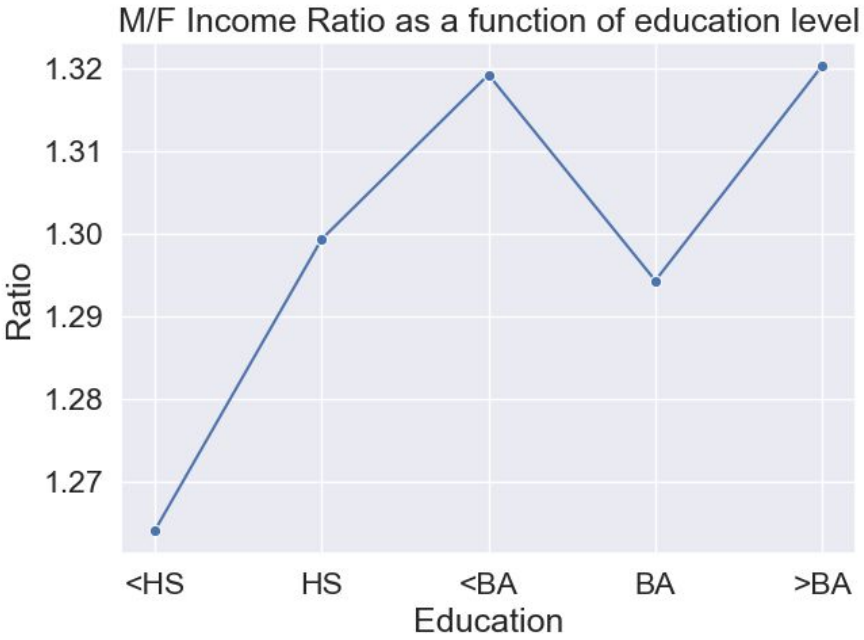
How does the income gap increase with education?

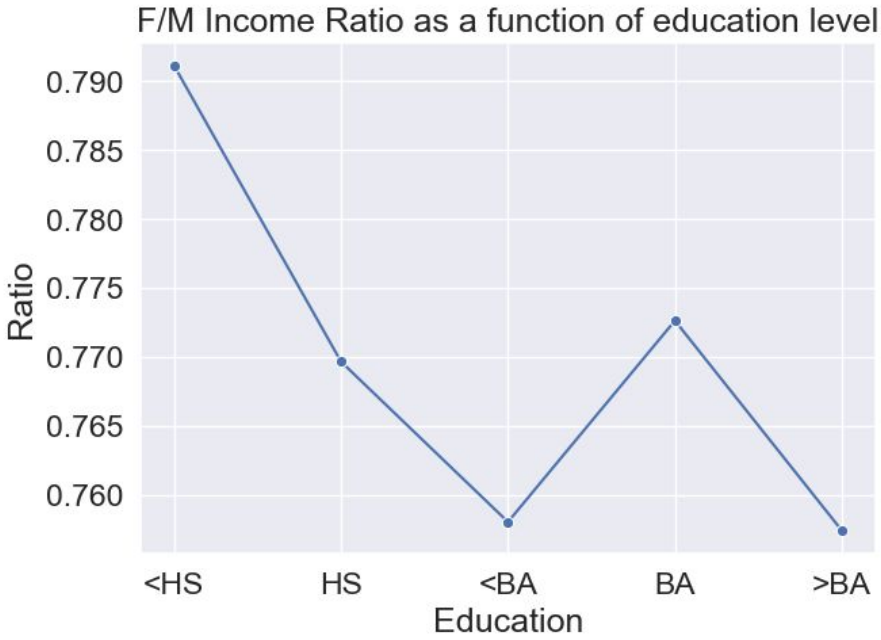




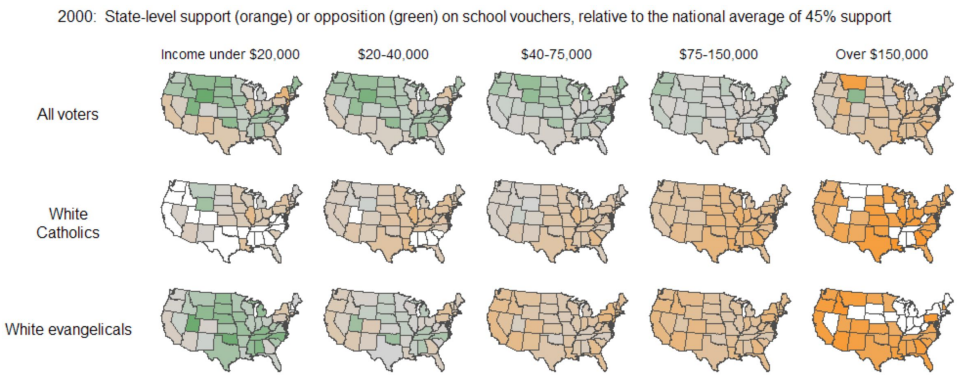
See notebook for how to get this figure with groupby!

But… which ratio should we pick? M/F or F/M?





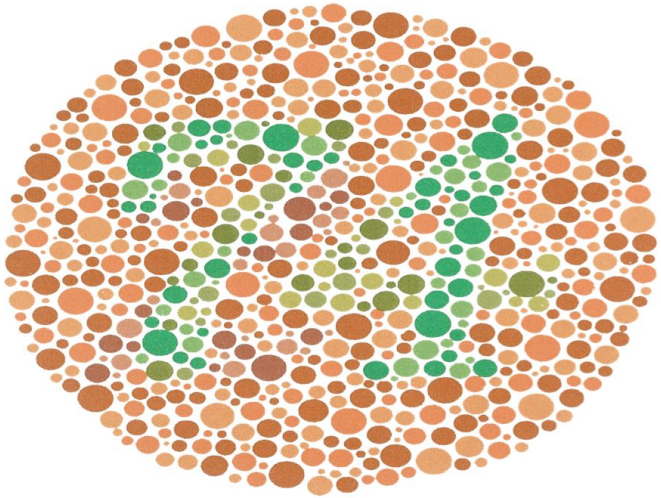
Distributions and relationships in subgroups



An example of **small multiples**.

- Juxtaposition: placing multiple plots side by side, with the same scale (called “small multiples”).
- Superposition: placing multiple density curves, scatter plots on top of each other (previous lec)
- Use color and shapes to represent additional variables.
    - See more in discussion.

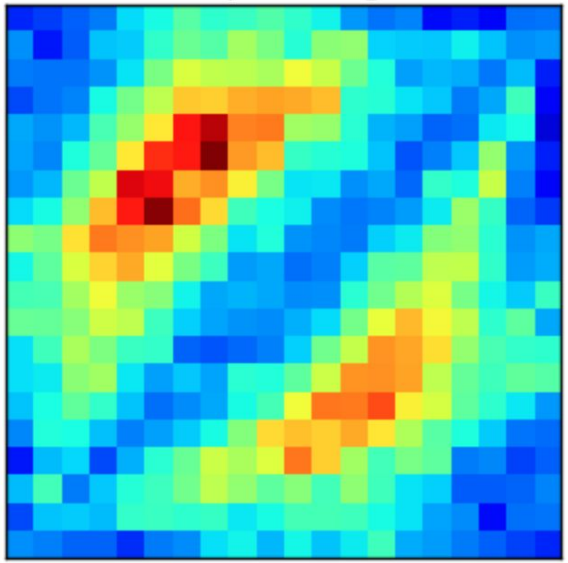
Perception



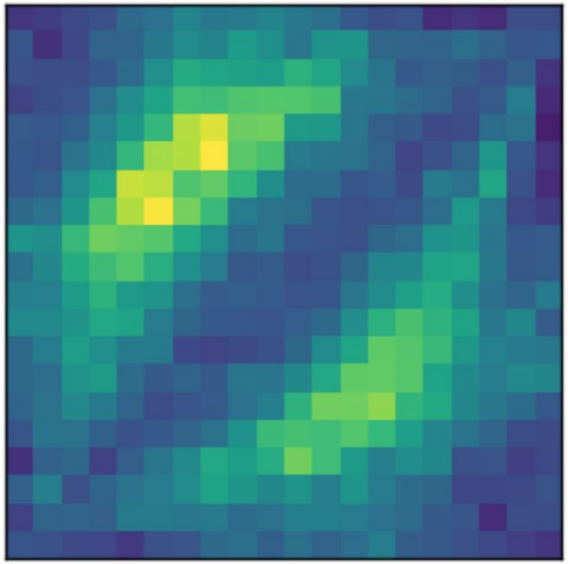
Choosing a set of colors which work together is a challenging task!

Perception of Color

Colormaps

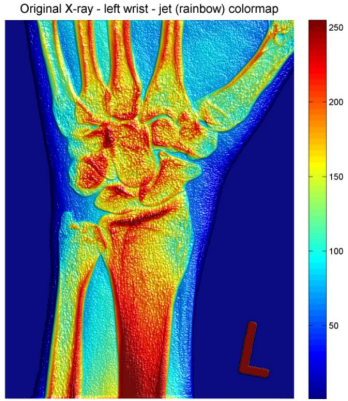


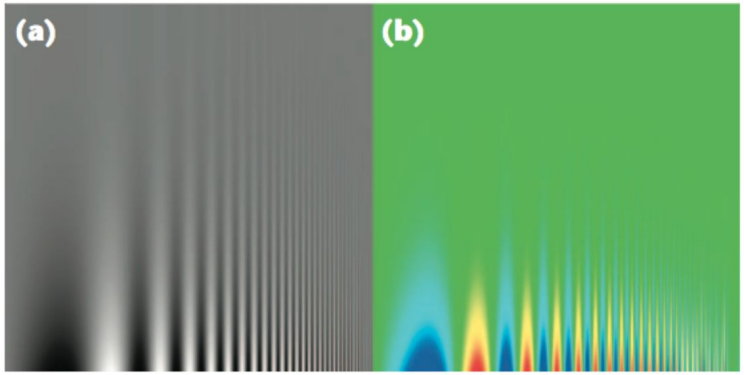
Jet

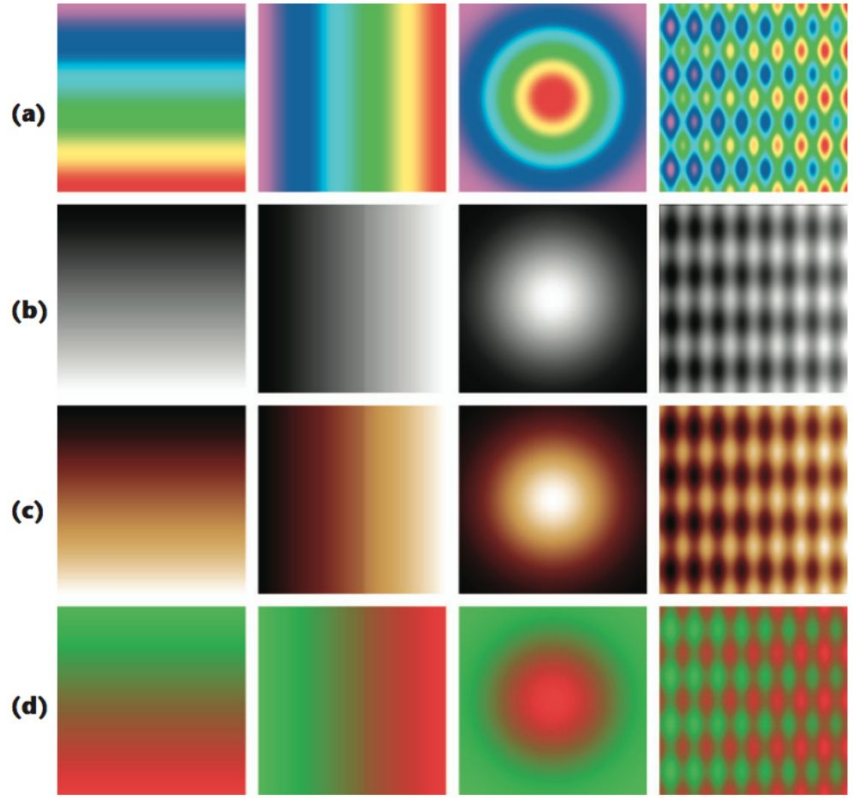


Viridis

The jet/rainbow colormap actively misleads



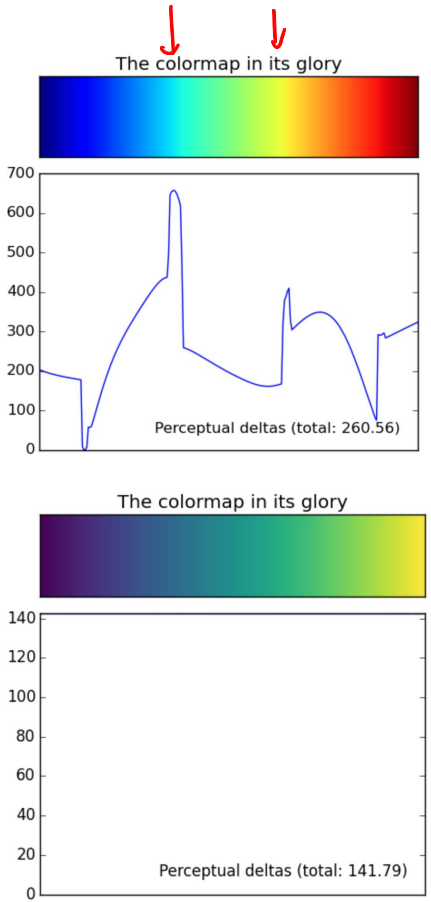




"Rainbow Colormap (Still) Considered Harmful", Borland and Taylor, 2007.

Use a perceptually uniform colormap!

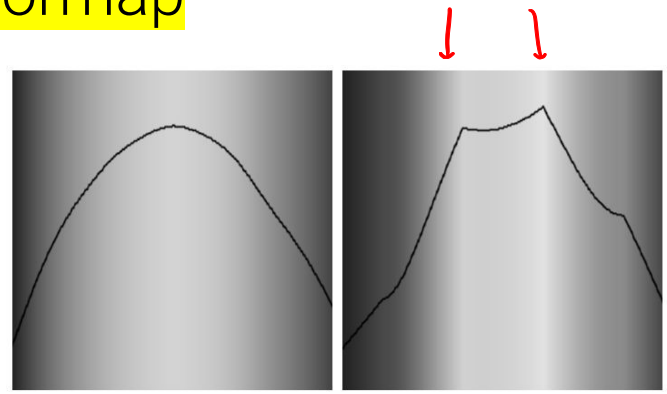
- **Perceptually uniform colormaps** have the property that if the data goes from 0.1 to 0.2, the **perceptual change** is the same as when the data goes from 0.8 to 0.9.
- Jet, the old matplotlib default, was far from uniform.
- Viridis, the new default colormap, is.
  - It was created by folks at the Berkeley Institute of Data Science!
  - https://bids.github.io/colormap/
- Avoid combinations of red and green, due to red-green color blindness.



Except when not :) The Google Turbo Colormap

Turbo

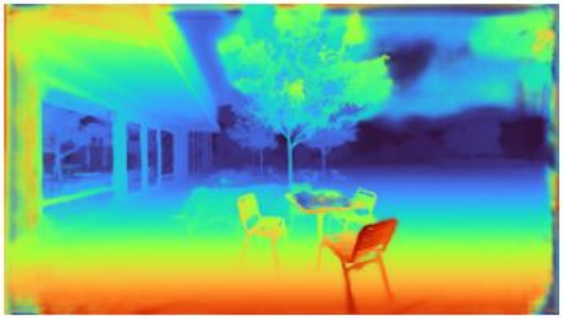
Jet



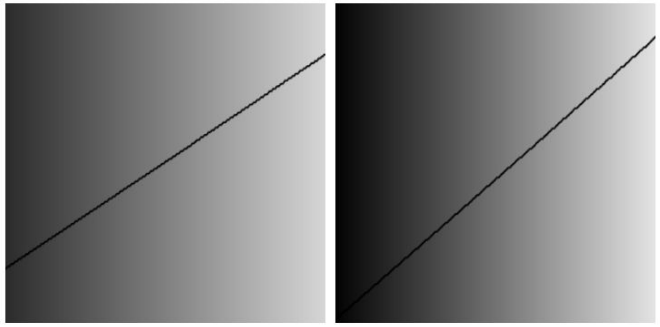
Turbo

Jet

Inferno



Turbo



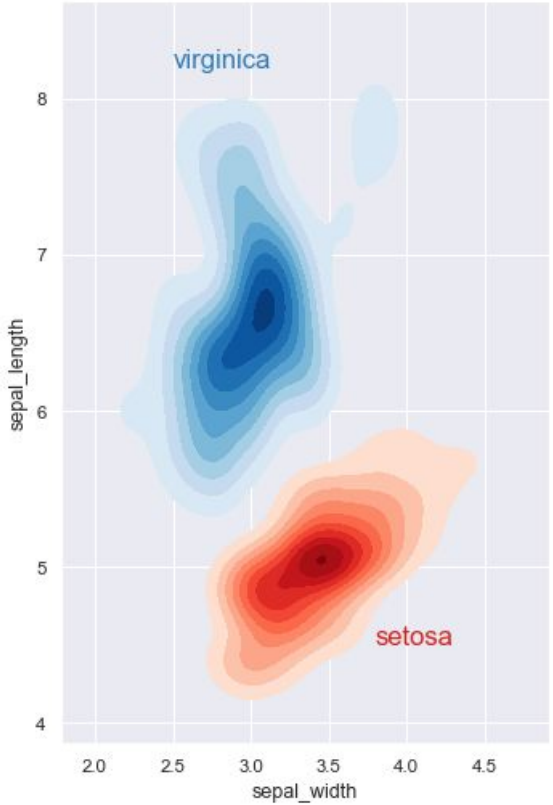
Viridis

Inferno

https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html

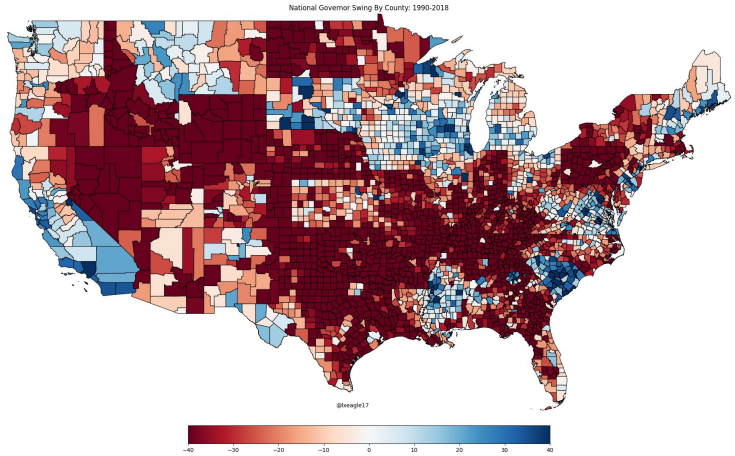
Use color to highlight data type

- **Qualitative**: Choose a qualitative scheme that makes it easy to distinguish between categories.
  - One category isn't "higher" or "lower" than another.
- **Quantitative**: Choose a color scheme that implies magnitude.
  - More on this in the next slide.
- The plot on the right has both!



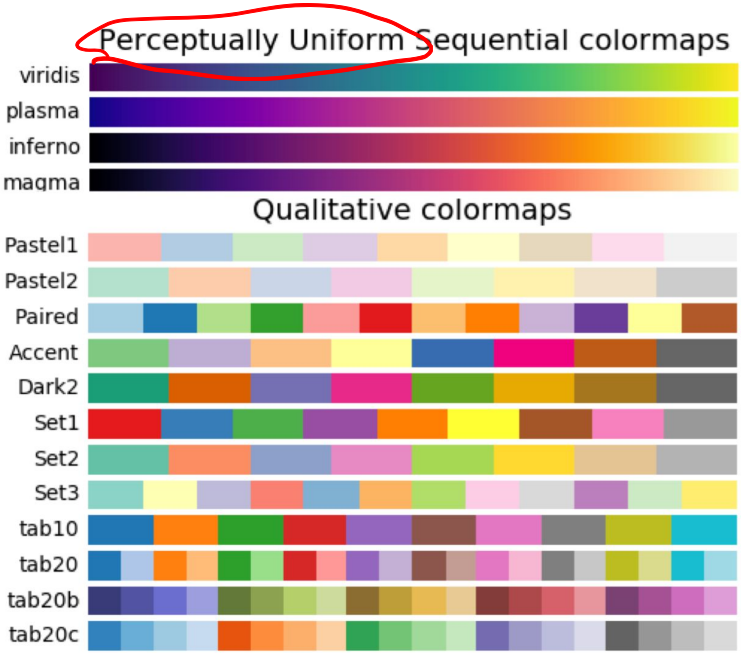
Sequential vs. diverging colormaps for quantitative data

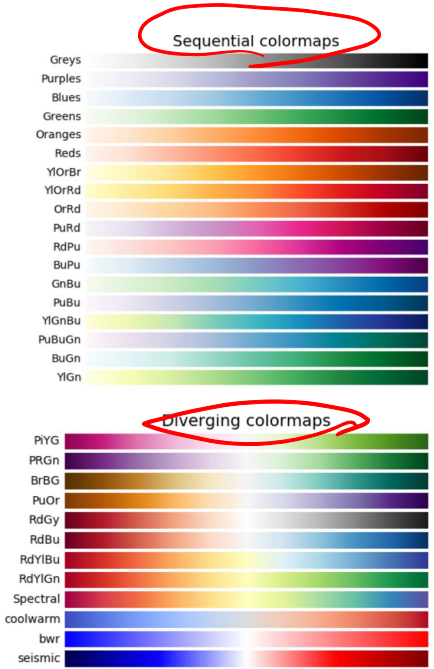
If the data progresses from low to high, use a **sequential** scheme where lighter colors are for more extreme values.



If low and high values deserve equal emphasis, use a **diverging** scheme where lighter colors represent middle values.

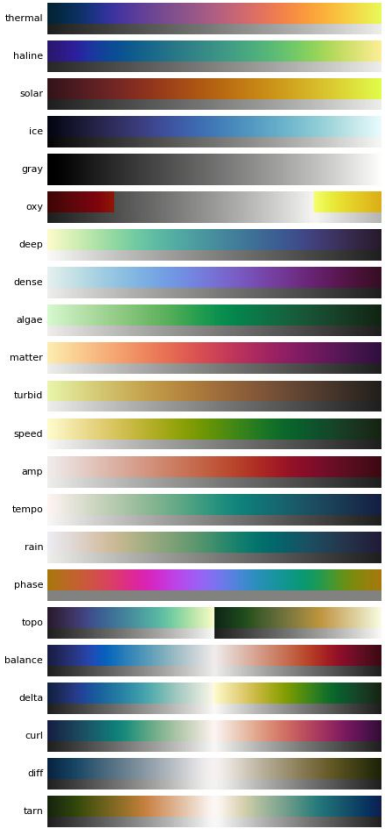
Default matplotlib colormaps

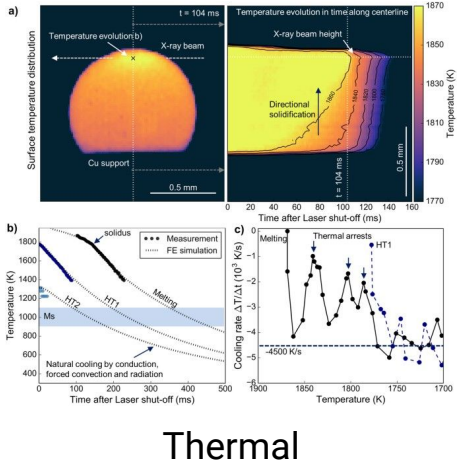




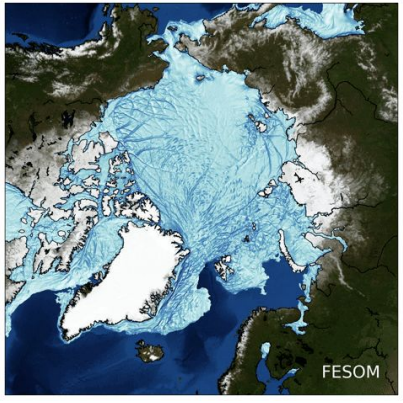
Taken from matplotlib documentation.

Domain specific colormaps: cmocean
(beautiful colormaps for oceanography, by Kristen Thyng)

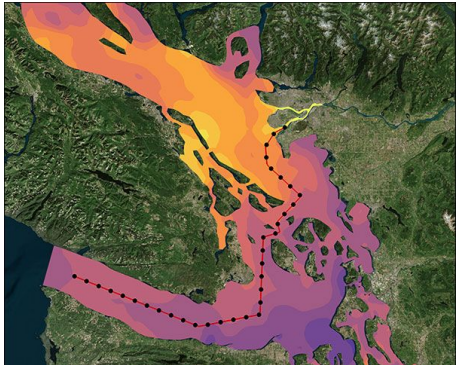


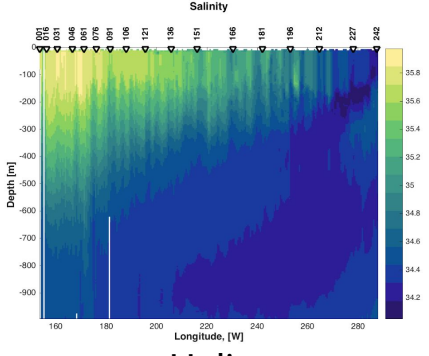


Thermal



Ice



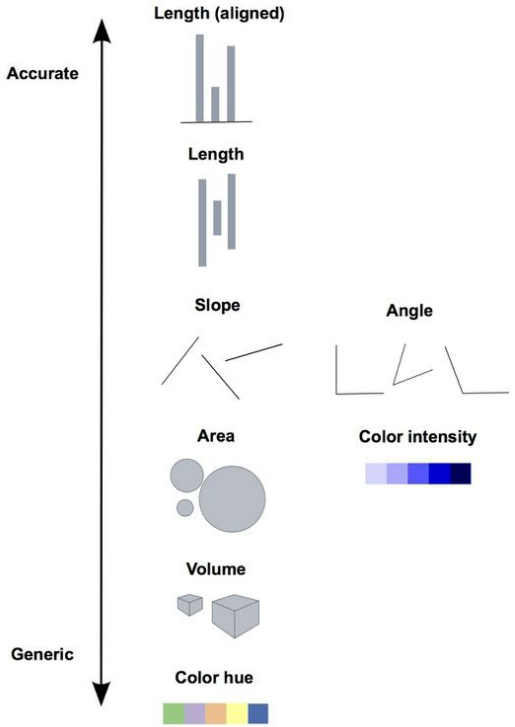


Haline

Extra reading

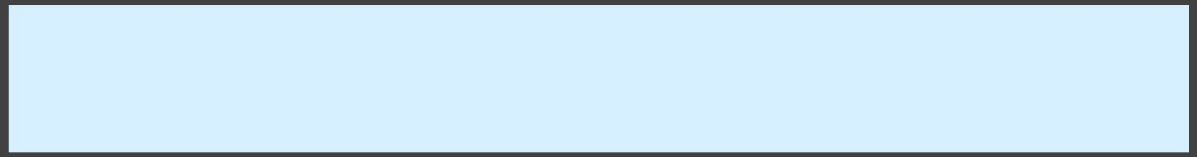
You may want to refer to these articles, which also discuss colormaps.

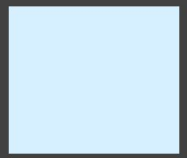
- Rainbow Colormap (Still) Considered Harmful - paper and presentation slides.
- https://eagereyes.org/basics/rainbow-color-map
- https://everydayanalytics.ca/2017/03/when-to-use-sequential-and-diverging-palettes.html
- https://web.natur.cuni.cz/~langhamr/lectures/vtfg1/mapinfo_2/barvy/colors.html



The accuracy of our judgements depend on the type of marking.

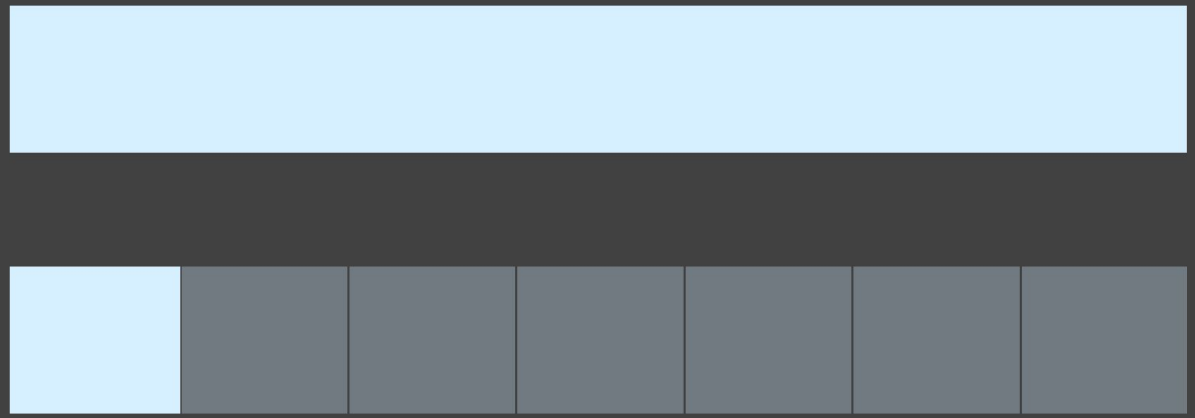
Perception of Markings



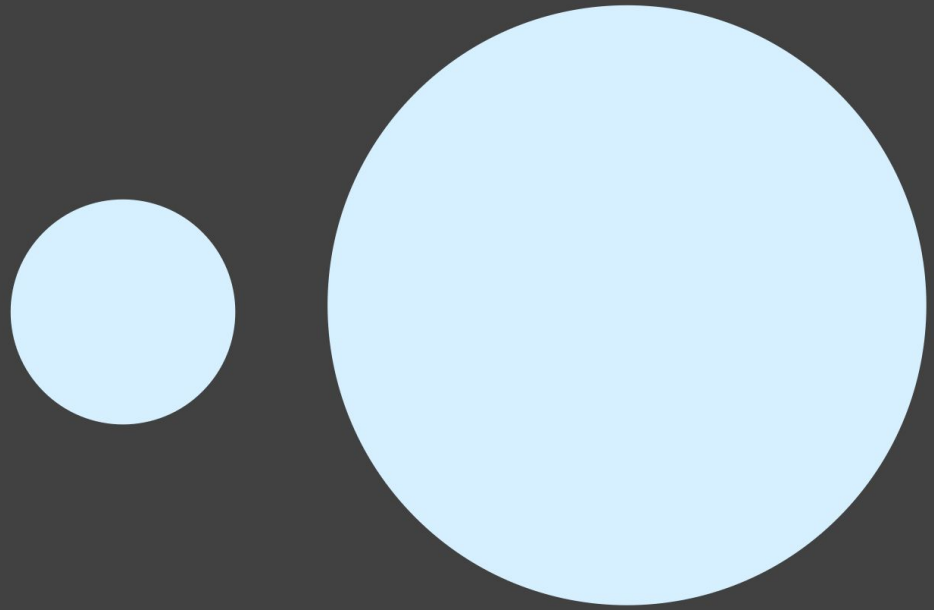


x 8 ?

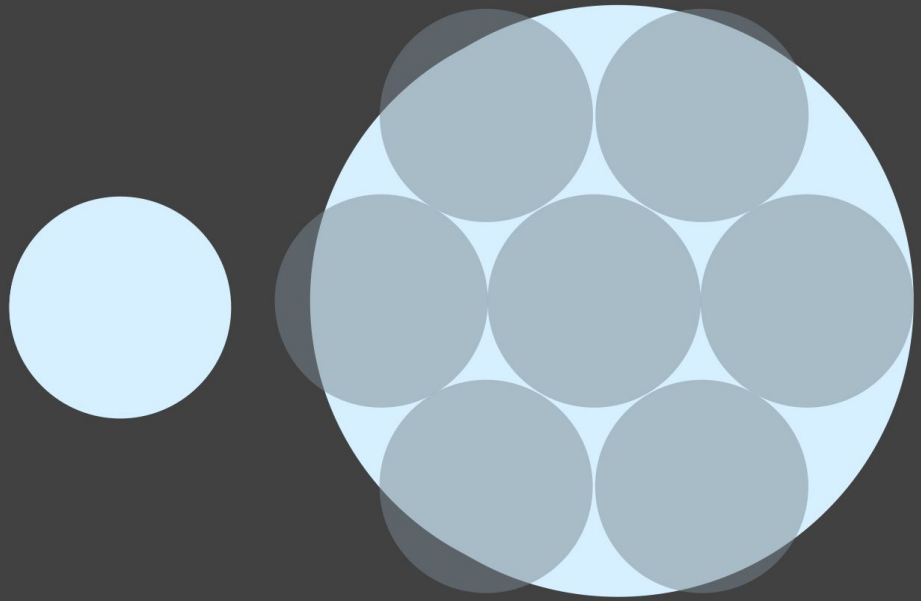
How much longer is the top bar?



The top bar is 7 times longer than the bottom bar.

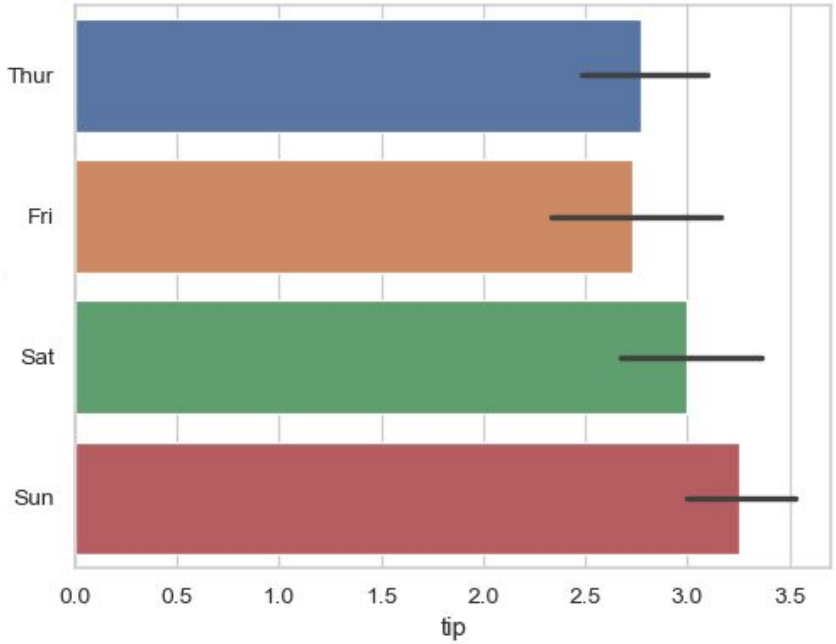


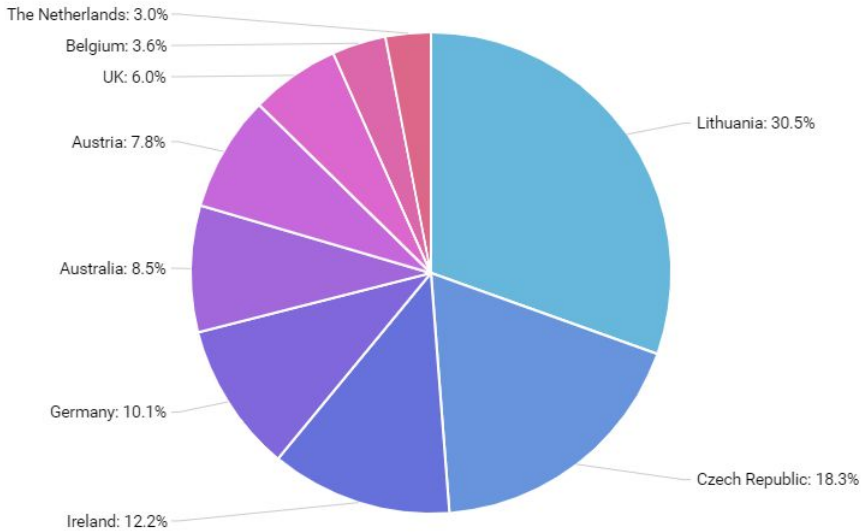
How much bigger is the big circle?



The area of the big circle is 7 times larger than the area of the small circle.

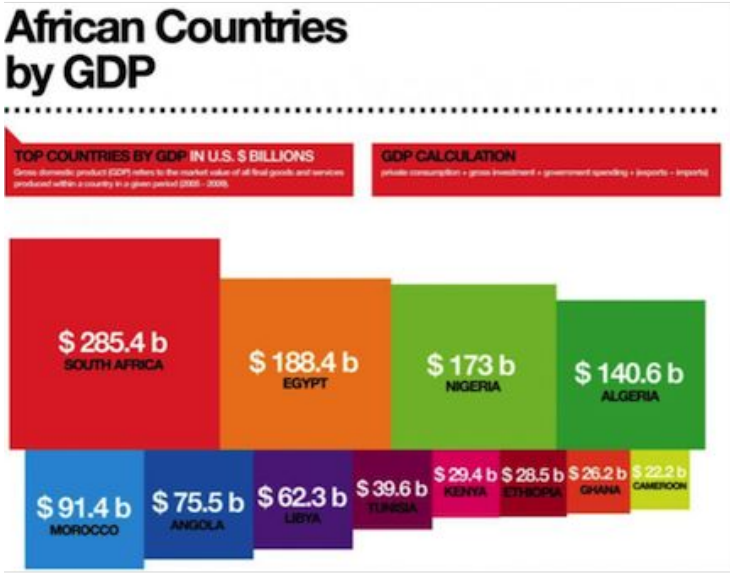
Lengths are easy to distinguish; angles are hard

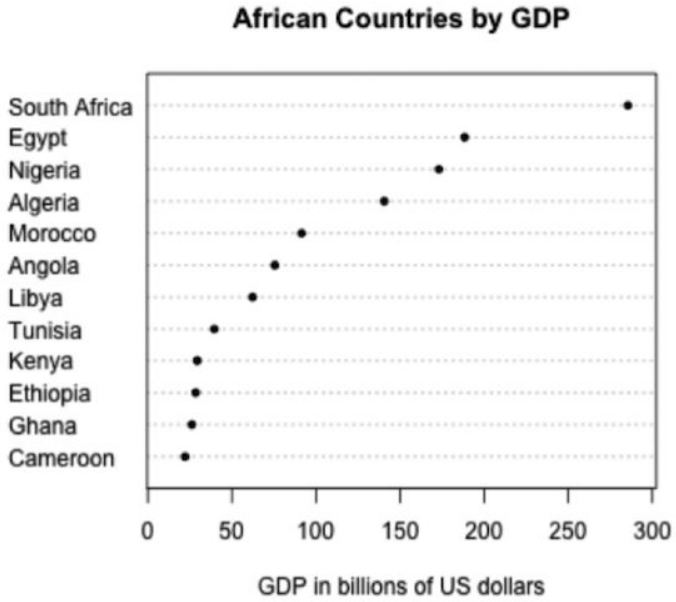




Don’t use pie charts! Angle judgements are inaccurate.

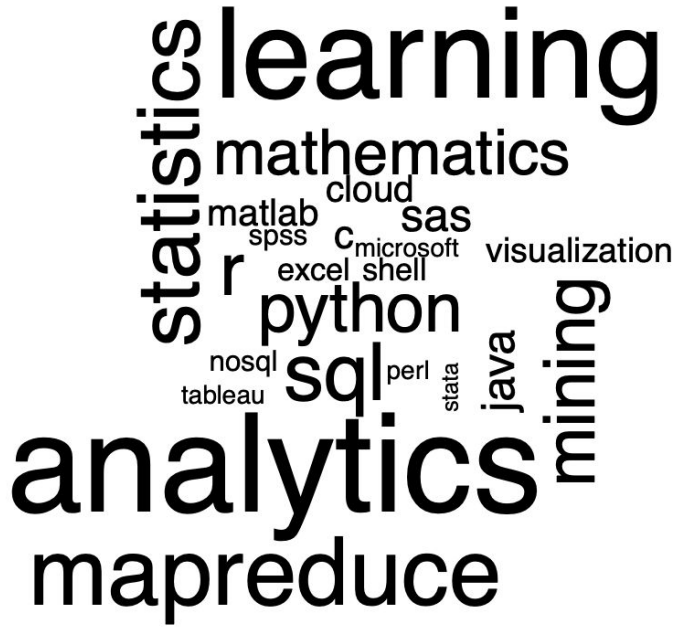
Areas are hard to distinguish

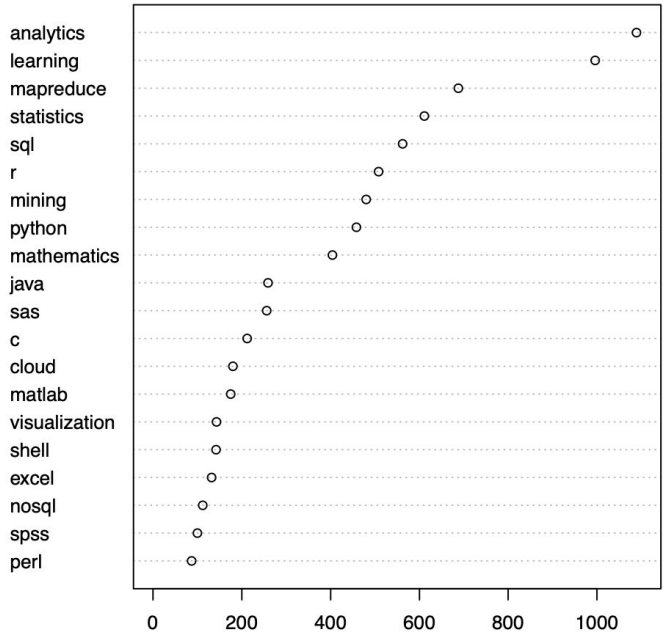




Avoid area charts! Area judgements are inaccurate. (For instance, South Africa has twice the GDP of Algeria, but that isn't clear from the areas.)

Areas are hard to distinguish



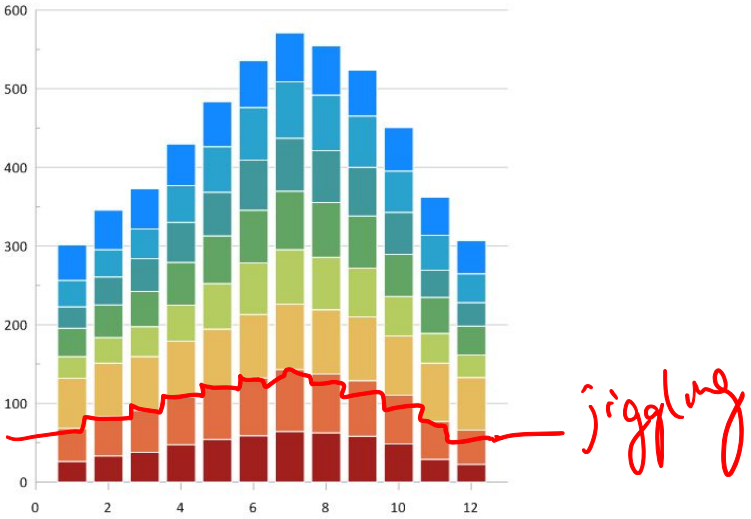


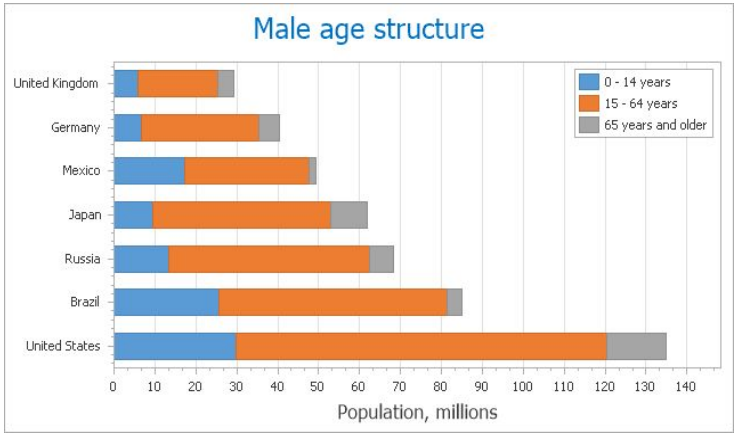
Avoid word clouds too! It's hard to tell the area taken up by a word.

Avoid jiggling the baseline

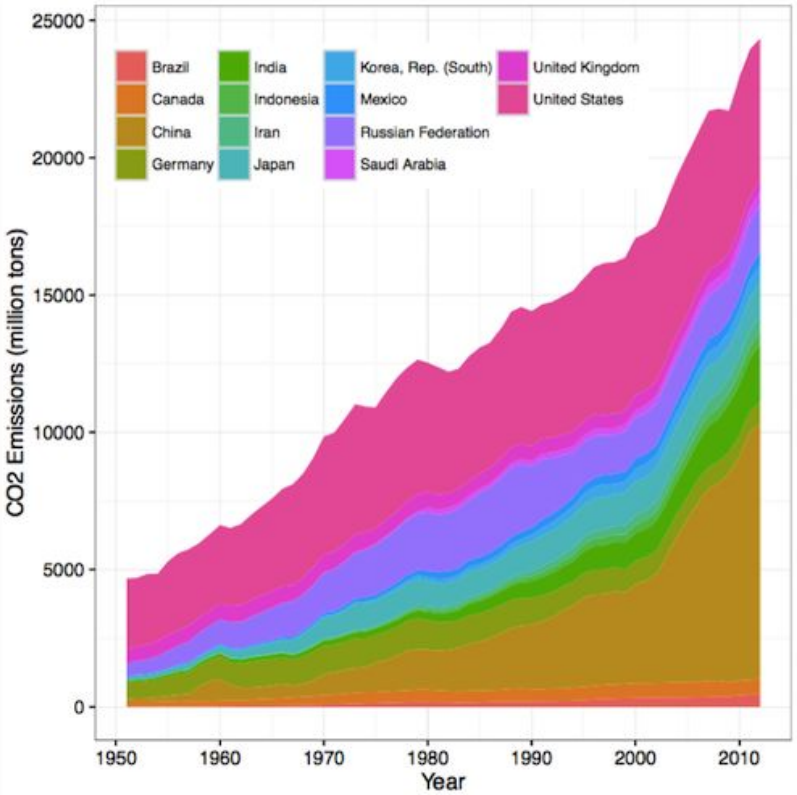
Stacked bar charts, histograms, and area charts are hard to read because the baseline moves.

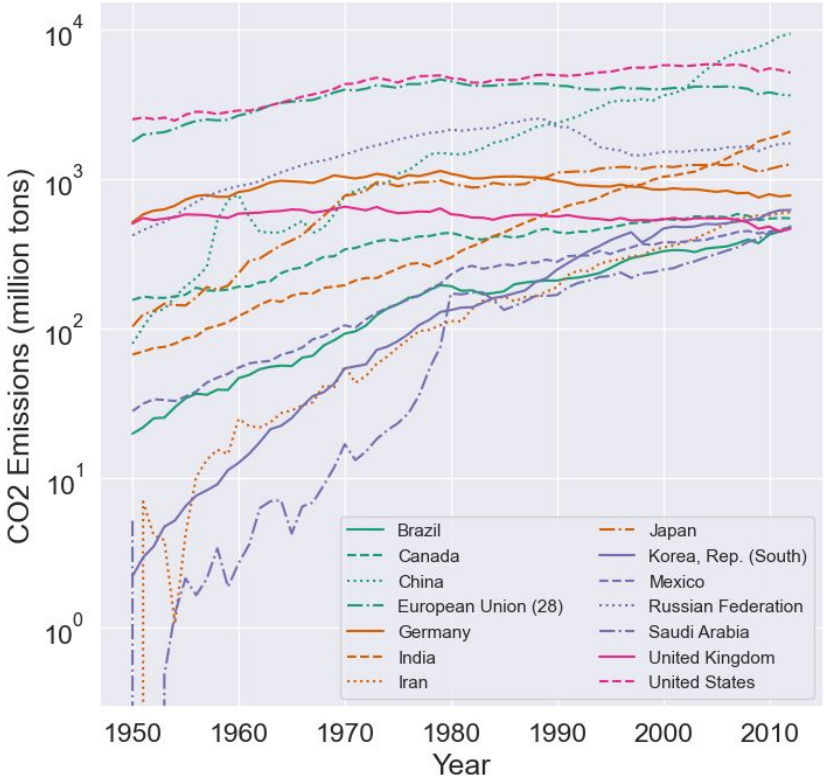
- In the first plot, the top blue bars are all roughly of the same length. But that's not immediately obvious!
- In the second plot, comparing the number of 15-64 year old males in Germany and Mexico is difficult.





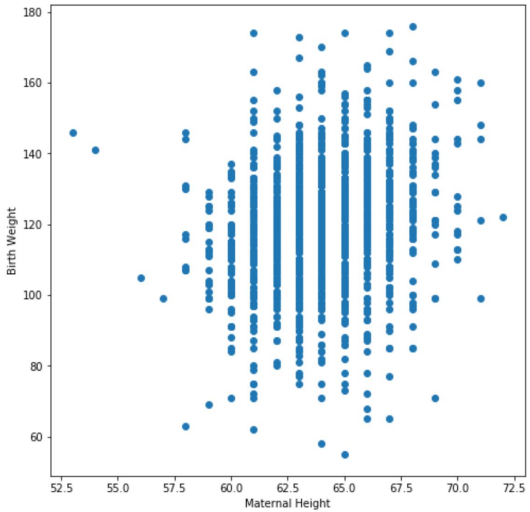
Avoid jiggling the baseline

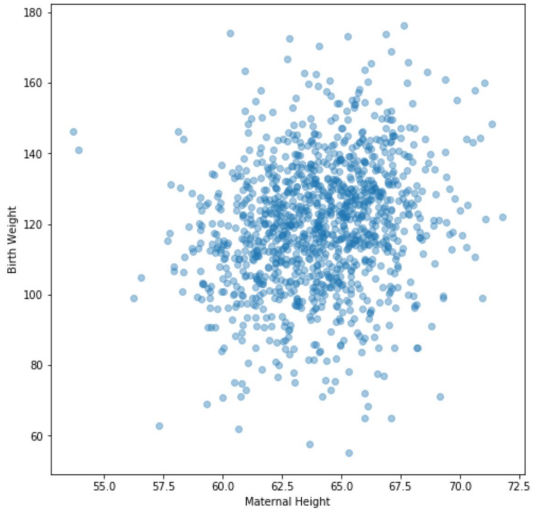




Here, by switching to a line plot, comparisons are made much easier.

Related – overplotting





In the plot on the left, it's hard to tell exactly how many points are being visualized.

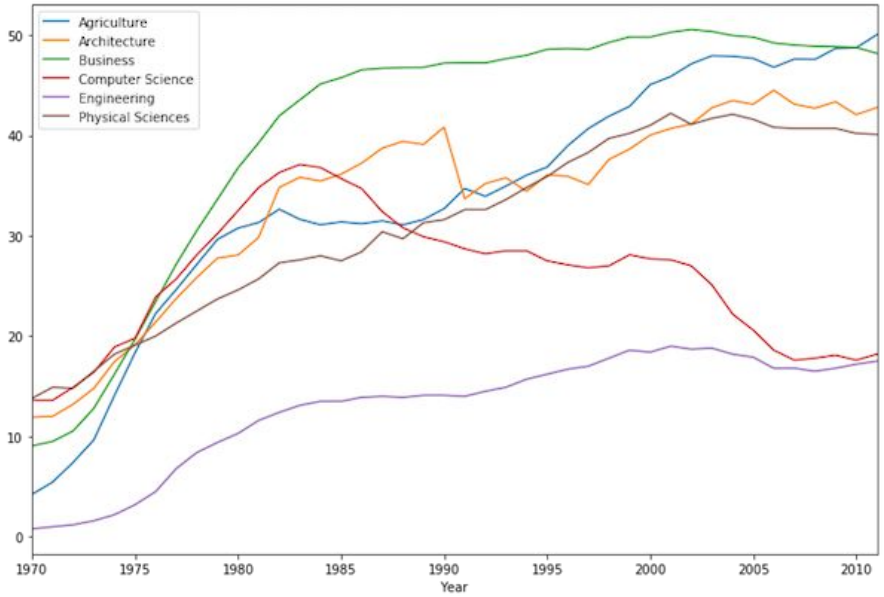
- Many on top of one another.
- Observations only on lattice points.

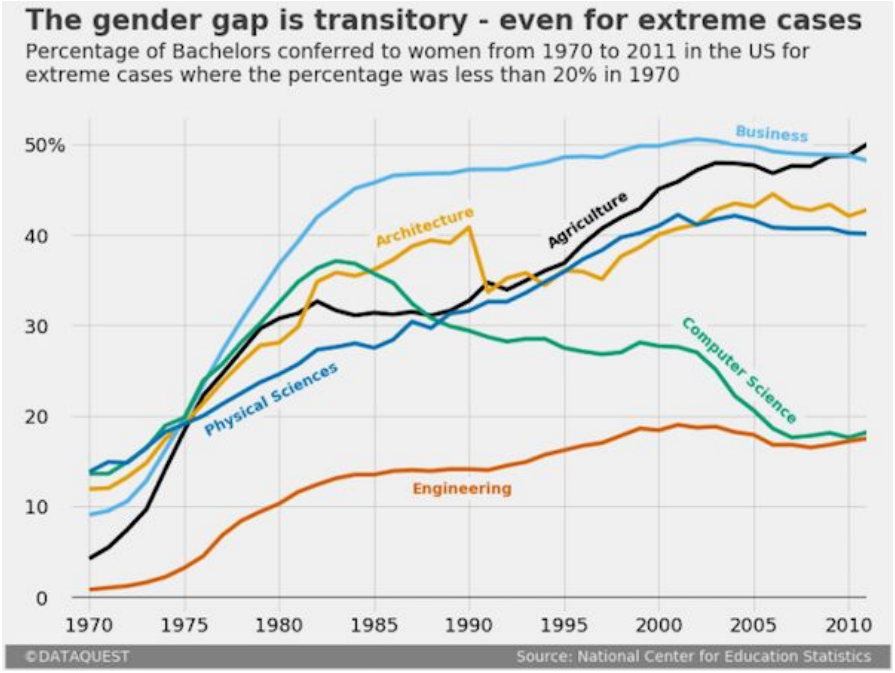
Some solutions:

- Add small random noise to both x and y ("jittering").
- Make points smaller (wouldn't help here though).

Context

all the necessary info to readers





Add context directly to plot

A publication-ready plot needs:

- Informative title (takeaway, not description).
    - “Older passengers spend more on plane tickets” instead of “Scatter plot of price vs. age”.
- Axis labels.
- Reference lines, markers, and labels for important values.
- Legends, if appropriate.
- Captions that describe the data.

The plots you create in this class always need titles and axes labels, too.

Captions

A picture is worth a thousand words, but not all thousand words you want to tell may be in the picture. In many cases, we need captions to help tell the story.

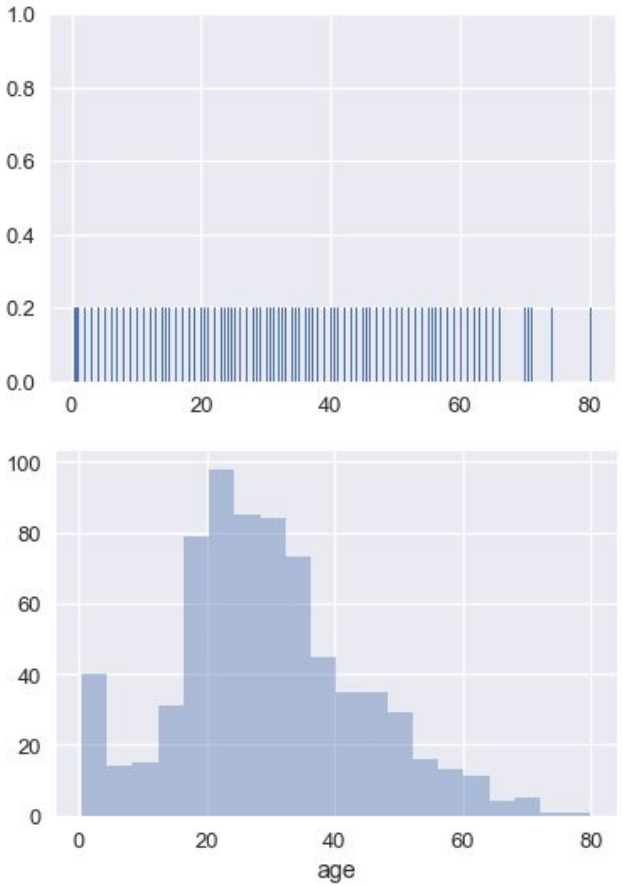
Captions should be:

- Comprehensive and self-contained.
- Describe what has been graphed.
- Draw attention to important features.
- Describe conclusions drawn from graph.

Smoothing

Smoothing

- Histograms are a smoothed version of rug plots.
- We smooth if we want to focus on general structure rather than individual observations.

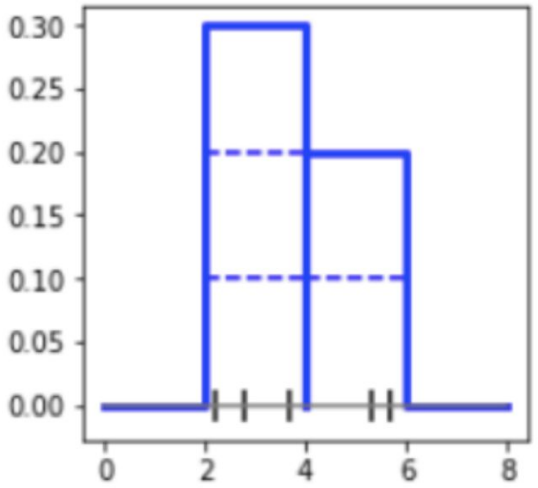


Spreading proportion uniformly

Points: [2.2, 2.8, 3.7, 5.3, 5.7]

Bins: [0, 2), [2, 4), [4, 6), [6, 8]

- Each of the 5 points is a proportion ⅕ of the list.
- In a histogram, **area = proportion**.
- Each point:
  - Contributes an area 1/5 to the histogram.
  - Rectangular area of 1/5 has a width 2.
  - Rectangle has width 2 and thus height 1/10.
- Kernel density estimates follow similar guidelines.



In each bin, add a rectangle with area 1/5 for each point in that bin.

Kernel density estimation (KDE)

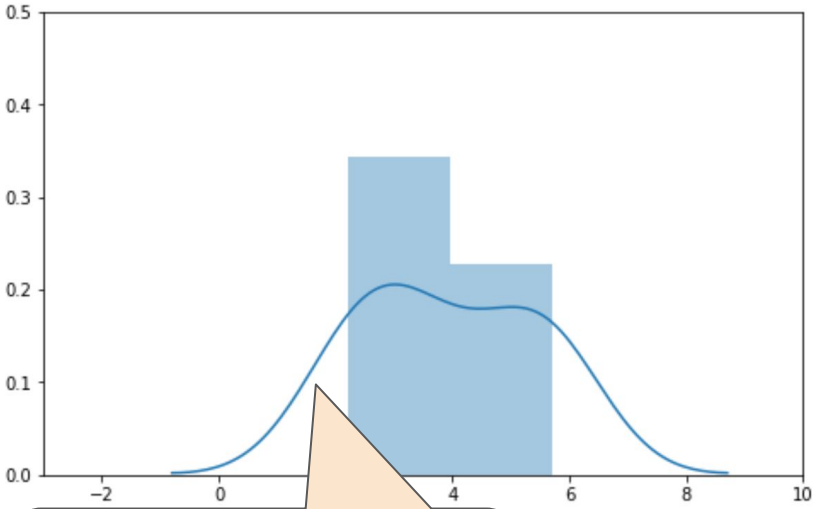
Kernel Density Estimation is used to estimate a **probability density function** (or density curve) from a set of data.

- Just like a histogram, a density function's total area must sum to 1.

To create a KDE:

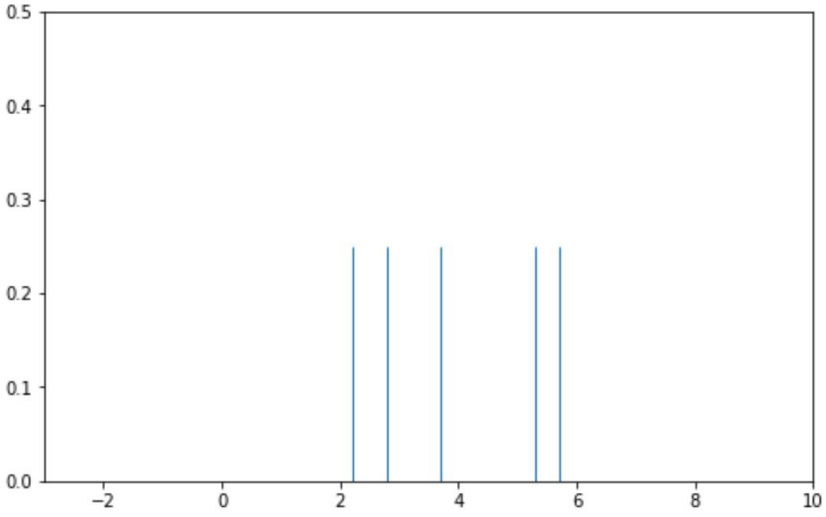
- Place a **kernel** at each data point.
- Normalize kernels so that total area = 1.
- Sum all kernels together.

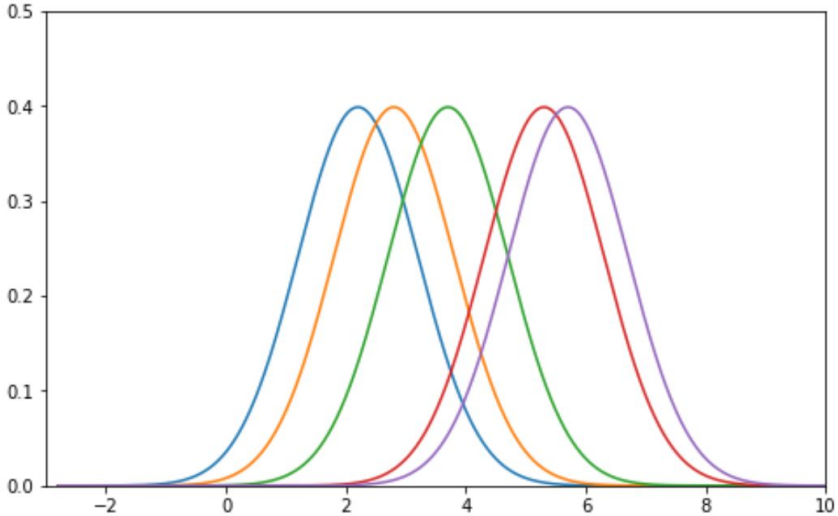
We also need to choose a kernel and **bandwidth.**



Step 1 – place a kernel at each data point

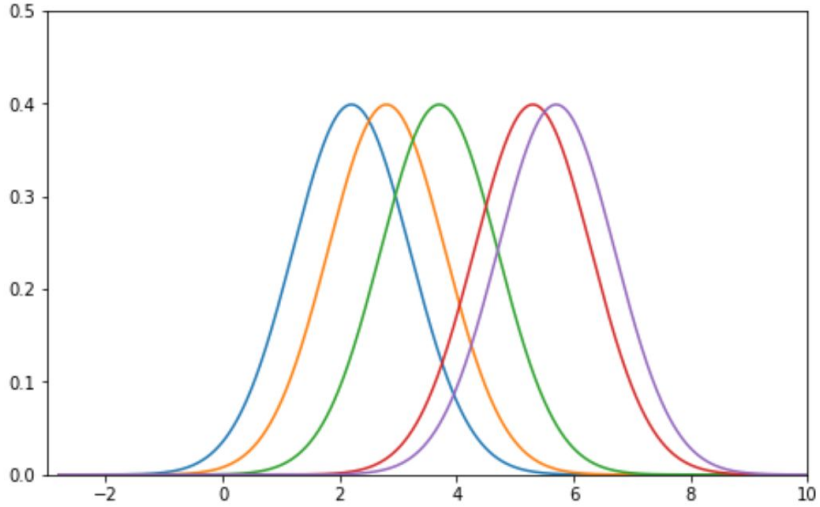
At each of our 5 points (depicted in the rug plot on the left), we've placed a **Gaussian** kernel with **alpha = 1**. The idea is that there is a higher density near the points we've already seen.

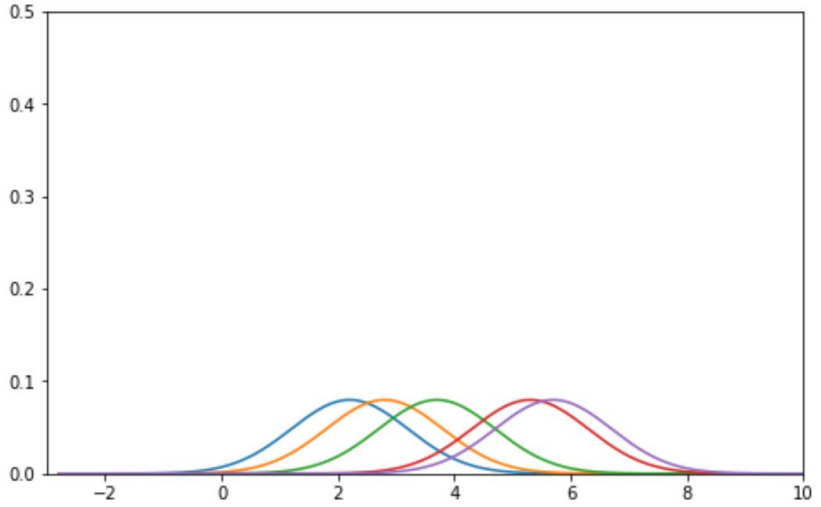




Step 2 – normalize kernels

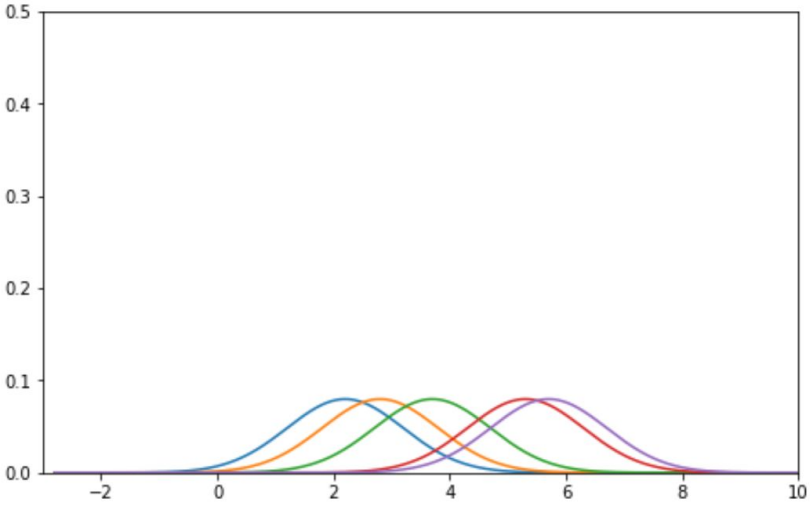
In Step 3, we will be summing each of these kernels. We want the result to be a valid density, that has area 1. Right now, we have 5 different kernels, each with an area 1. So, we **multiply each by 1/5**.

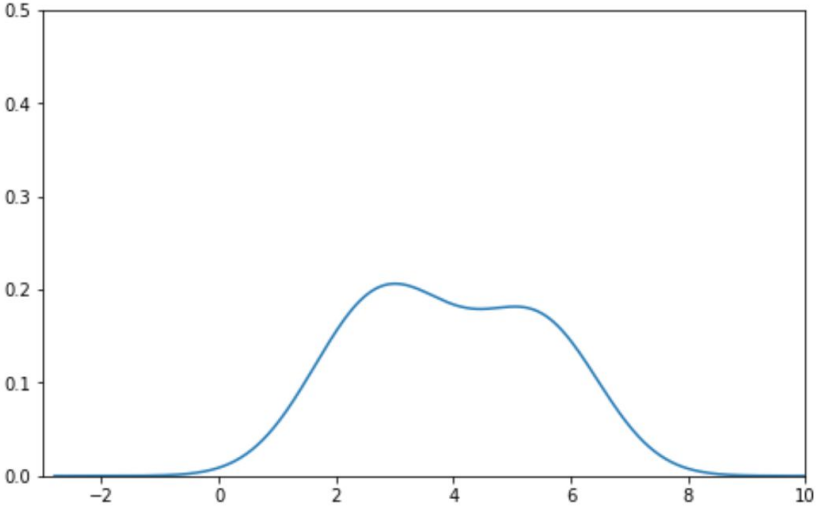




Step 3 – sum kernels

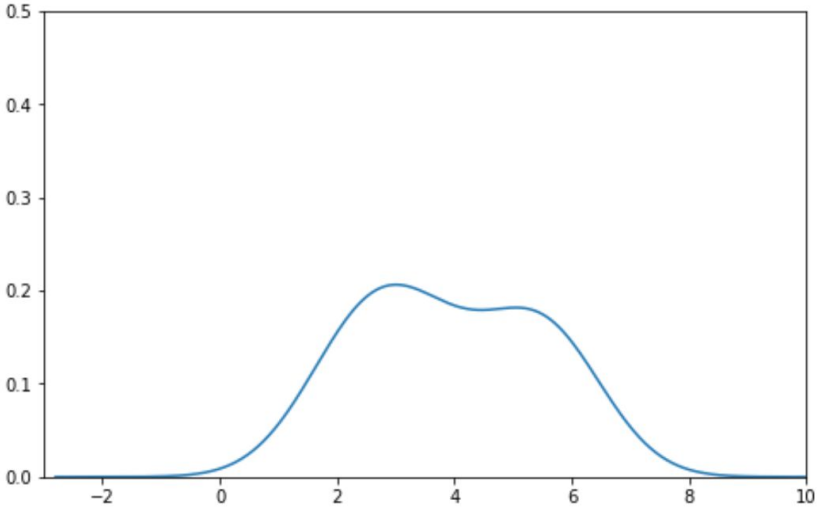
Our **kernel density estimate** is the **sum of the normalized kernels at each point.** It is depicted below on the right.

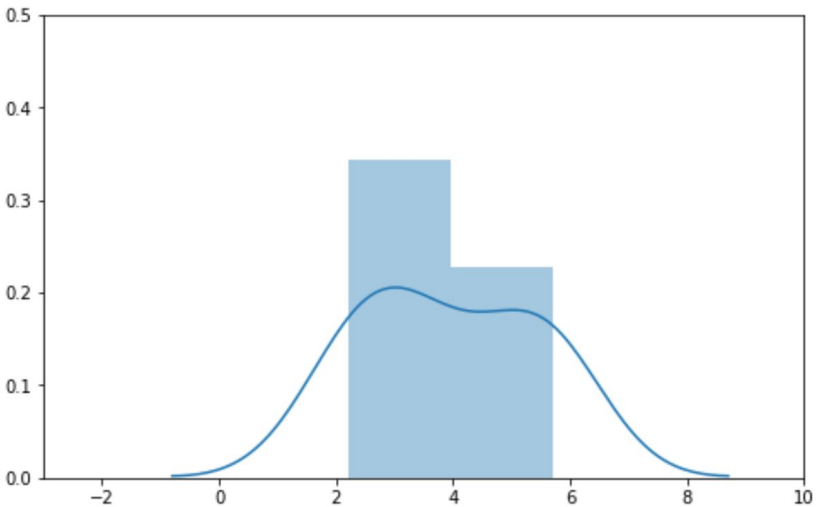




Kernel density estimates

The curve we manually created (left) exactly matches the one that `sns.distplot` creates for us (right)!





Kernels

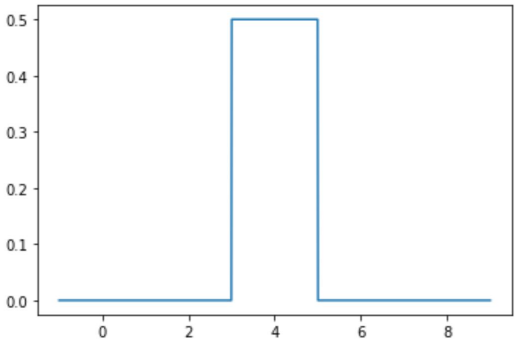
- A kernel (for our purposes) is a valid density function. That means it:
  - Must be non-negative for all inputs.
  - Must integrate to 1.
- The most common kernel is the **Gaussian** kernel.
  - Here, x represents any input, and x_i represents the ith observed value. The kernels are centered on our observed values (and so the mean of this distribution is x_i).
  - **α** is the **bandwidth parameter**. It controls the smoothness of our KDE. Here, it is also the standard deviation of the Gaussian.

$$K_\alpha(x, x_i) = \frac{1}{\sqrt{2\pi\alpha^2}} e^{-\frac{(x-x_i)^2}{2\alpha^2}}$$

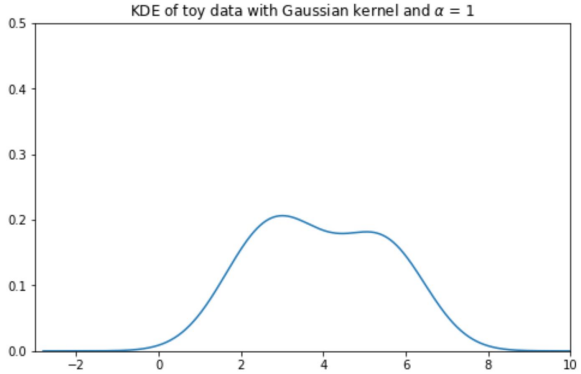
Kernels

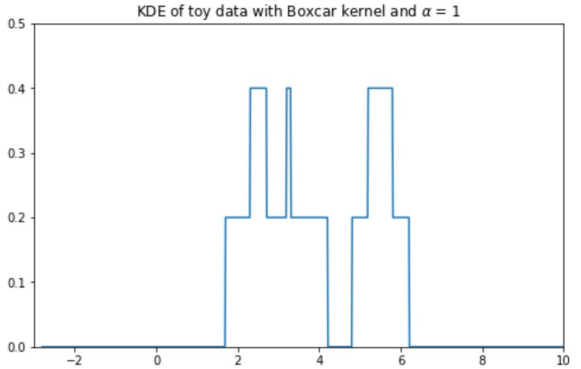
- Another common kernel is the **boxcar** kernel.
  - It assigns uniform density to points within a "window" of the observation, and 0 elsewhere.
  - Resembles a histogram… sort of.

$$K_\alpha(x, x_i) = \begin{cases} \frac{1}{\alpha}, & |x - x_i| \leq \frac{\alpha}{2} \\ 0, & \text{else} \end{cases}$$

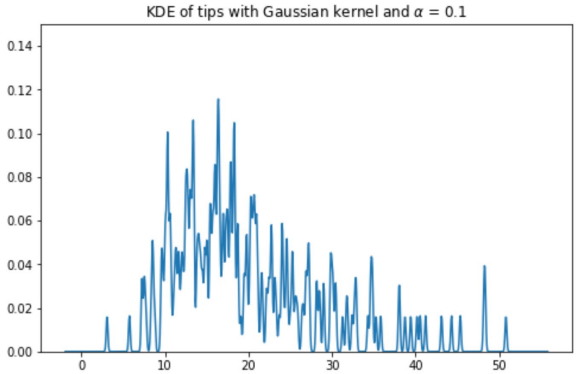


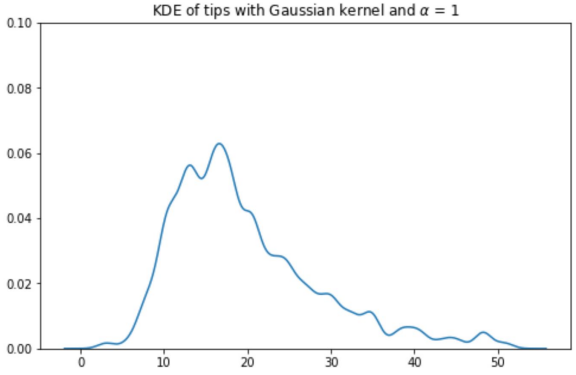
A boxcar kernel centered on $x_i = 4$ with $\boldsymbol{\alpha} = 2$.

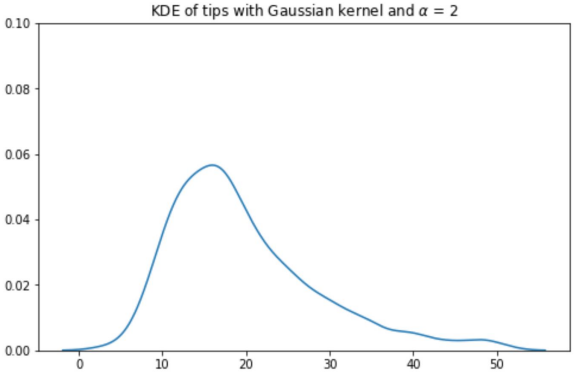


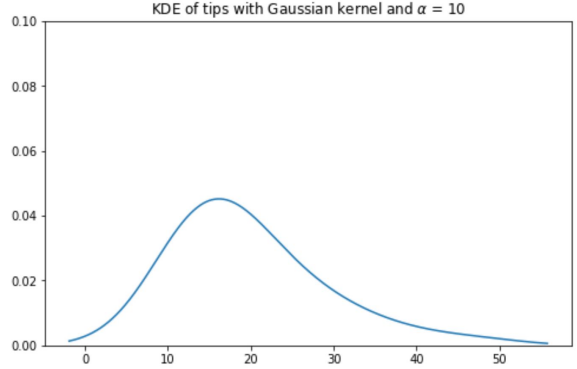


Effect of bandwidth on KDEs



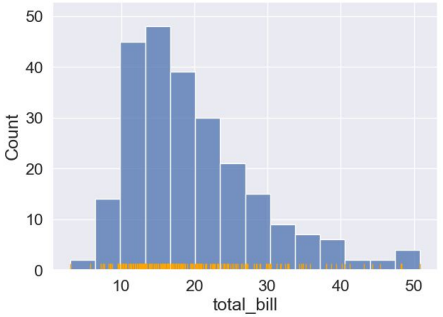






Bandwidth is analogous to the width of each bin in a histogram.

- As **α** increases, the KDE becomes more smooth.
- Simpler to understand, but gets rid of potentially important distributional information.
- We call **α** a **hyperparameter**. Be familiar with this term!



Summary of KDE

P.47

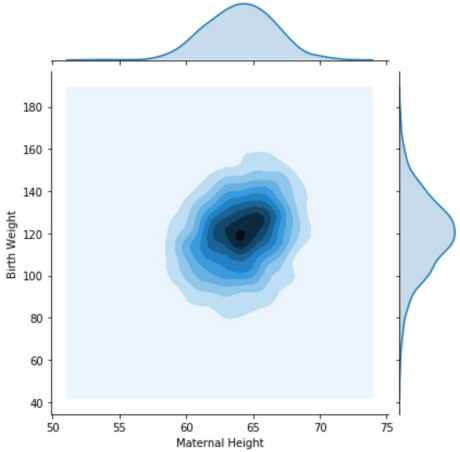
$$f_\alpha(x) = \frac{1}{n}\sum_{i=1}^{n} K_\alpha(x, x_i)$$

The “KDE formula” is above.

- $\mathbf{x}$ represents any number on the number line. It is the input to our function.
- $\mathbf{n}$ is the number of observed data points that we have.
- Each $\mathbf{x}_i$ ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$) represents an observed data point. These are what we use to create our KDE.
- $\boldsymbol{\alpha}$ is the bandwidth or smoothing parameter**.**
- $K_\alpha(x, x_i)$ is the kernel centered on the observation i.
    - Each kernel individually has area 1. We multiply by $1/\mathbf{n}$ so that the total area is still 1.

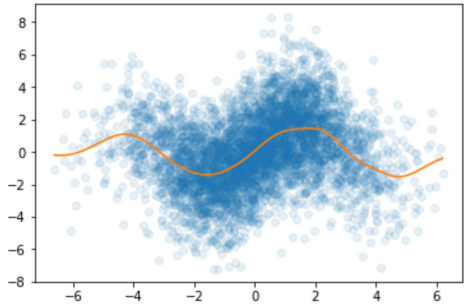
Extensions

- One can extend the idea of kernel density estimation to two dimensions.
  - A contour plot is a two dimensional KDE (top).
- One can also use kernels to create smoothed versions of scatterplots (bottom).
  - Won't do that in Data 100.



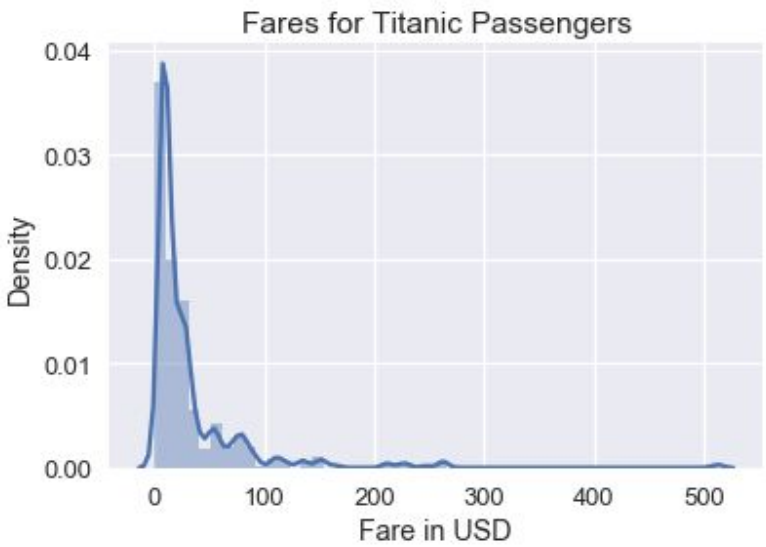
In the next lecture, we will talk about **models**.

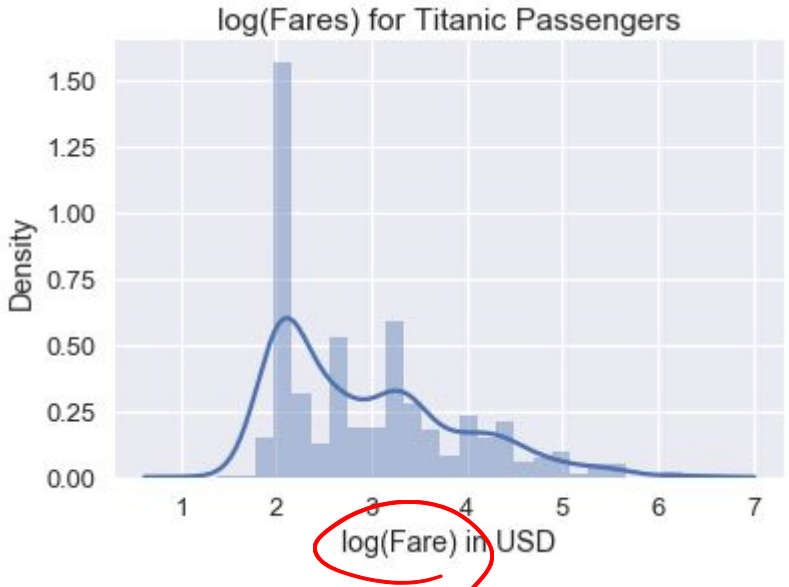
- We will focus on parametric models.
- Kernel density estimates are an example of **non-parametric models**.



Transformations

Transforming data can reveal patterns



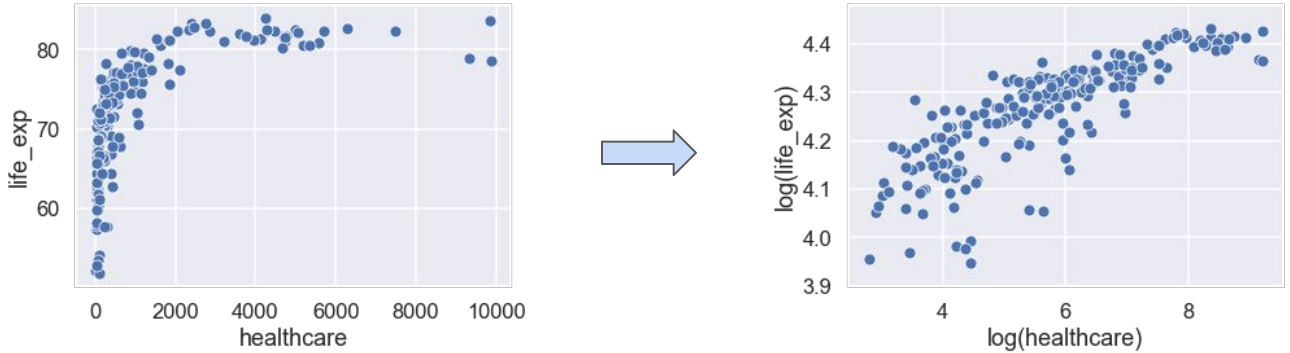


When a distribution has a large dynamic range, it can be useful to take the log.

Why straighten relationships?

Now, we will look at how to **linearize** the scatter plot of two variables. Why?

- If we know what transformation made our plot of y vs. x linear, we can “backtrack” to figure out the exact relationship between x and y.
- Linear relationships are particularly simple to interpret.
  - We know what slopes and intercepts mean.
  - We will be doing a lot of linear modeling – starting next lecture!



Log of y-values

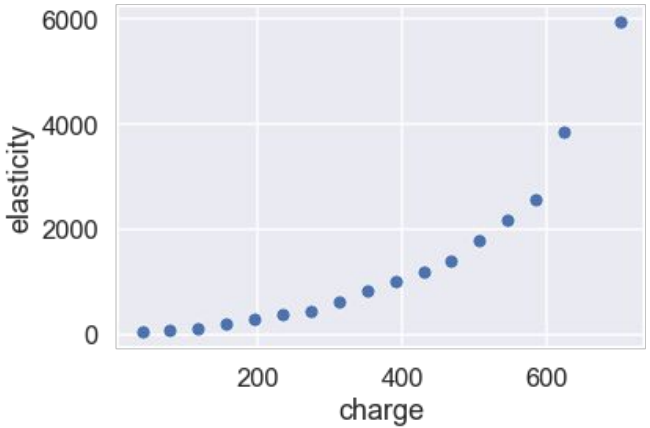
If we take the log of our y-values and notice a linear relationship, we can say (roughly) that

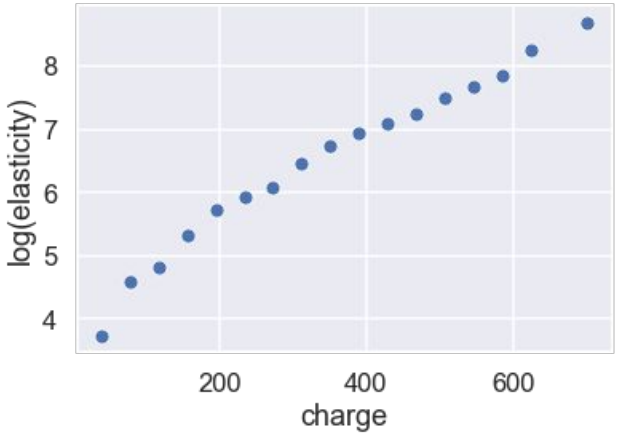
$$\log y = ax + b$$

Working backwards:

$$\log y = ax + b$$
$$y = e^{ax+b}$$
$$y = e^{ax}e^{b}$$
$$y = Ce^{ax}$$

This implies an **exponential** relationship in the original plot.





Log of both x and y-values

If we take the log of both axes and notice a linear relationship, we can say (roughly) that

$$\log y = a \cdot \log x + b$$

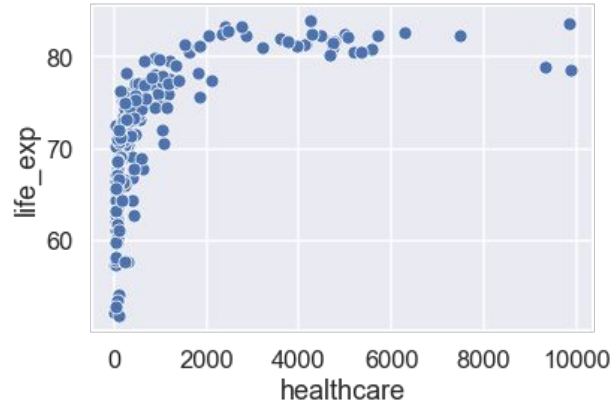
Working backwards:

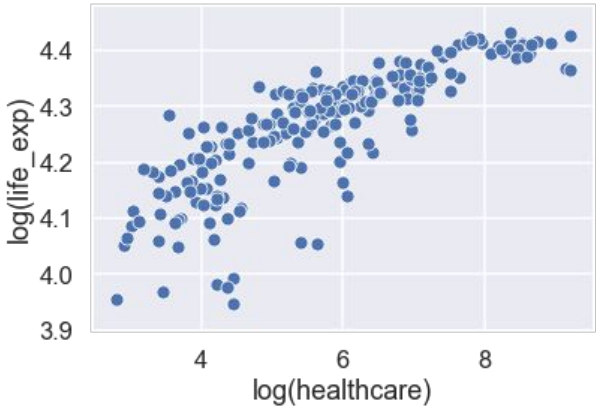
$$y = e^{a \cdot \log x + b}$$

$$y = Ce^{a \cdot \log x}$$

$$y = Cx^a$$

This implies a **power** relationship in the original plot (a one-term **polynomial**)





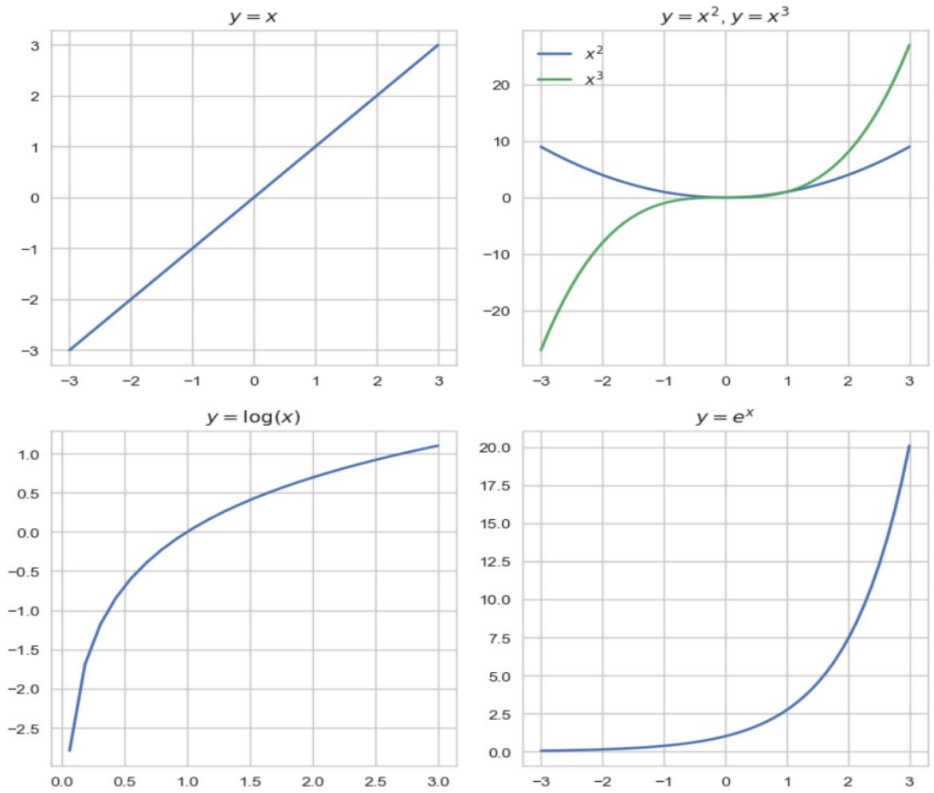
Log transform as a “Swiss army knife”

$$y = a^x \rightarrow \log(y) = x\log(a)$$
$$y = ax^k \rightarrow \log(y) = \log(a) + k\log(x)$$

Properties of logarithms make them very powerful!

Basic functional relations

Knowing the general shapes of polynomial, exponential, and logarithmic curves (regardless of base) will go a long way.

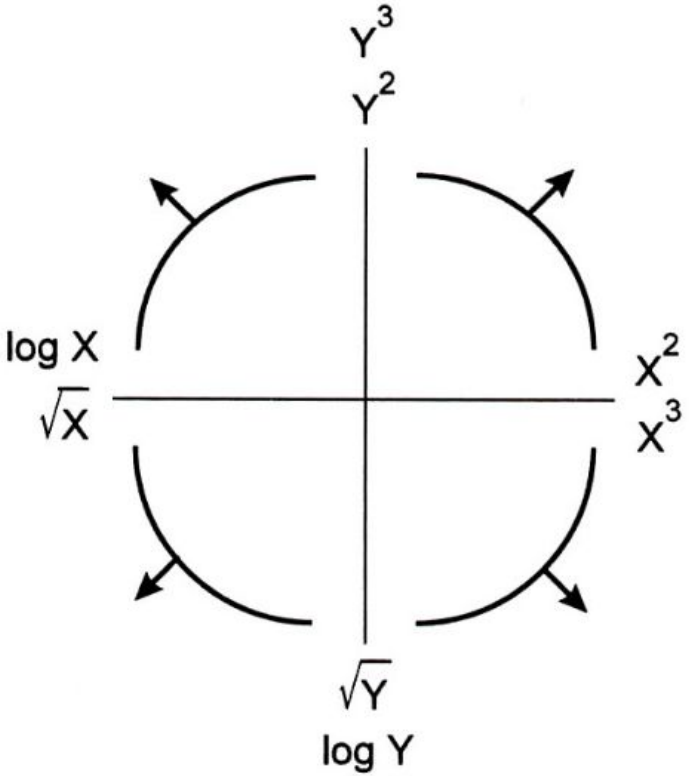


Tukey-Mosteller Bulge Diagram

This diagram can help us choose which transformation(s) to apply to our data in order to linearize it.

- There are multiple solutions. Some will fit better than others.
- sqrt and log make a value “smaller”. Raising to a value to a power makes it “bigger”.
- Each of these transformations equates to increasing or decreasing the scale of an axis.

(demo)



Summary

- Choose appropriate scales.
- Condition in order to make comparisons more natural.
- Choose colors and markings that are easy to interpret correctly.
- Add context and captions that help tell the story.
- Smoothed estimates of distributions help with big-picture interpretation.
    - Kernel Density Estimates are a method of smoothing data.
- Transforming our data can linearize relationships.
    - Helpful when we start linear modeling next lecture.
- **More generally – reveal the data!**
    - Eliminate anything unrelated to the data itself – “chart junk.”
    - It’s fine to plot the same thing multiple ways, if it helps fit the narrative better.