LECTURE 8

Visualization, Part II

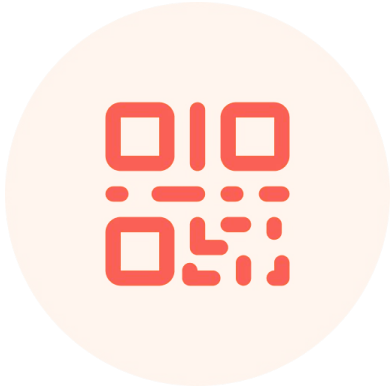
KDEs, Transformations, and Visualization Theory

Data 100/Data 200, Spring 2023 @ UC Berkeley

Narges Norouzi and Lisa Yan

Content credit: Josh Hug, Suraj Rampure, Ani Adhikari, Sam Lau, Fernando Pérez

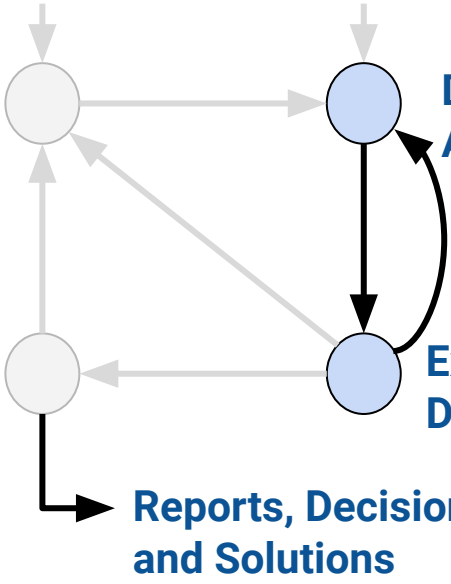
slido

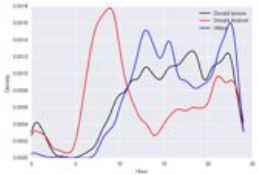


Join at slido.com
#3557130

ⓘ Start presenting to display the joining instructions on this slide.

Plan for Weeks 3 and 4





(today)

Data Wrangling Intro to EDA	Working with Text Data Regular Expressions	Plots and variables Seaborn	KDE / Smoothing Viz principles Transformations

Last week (Part I: Processing Data)

This week (Part II: Visualizing and Reporting Data)

KDE Mechanics

Lecture 08, Data 100 Spring 2023

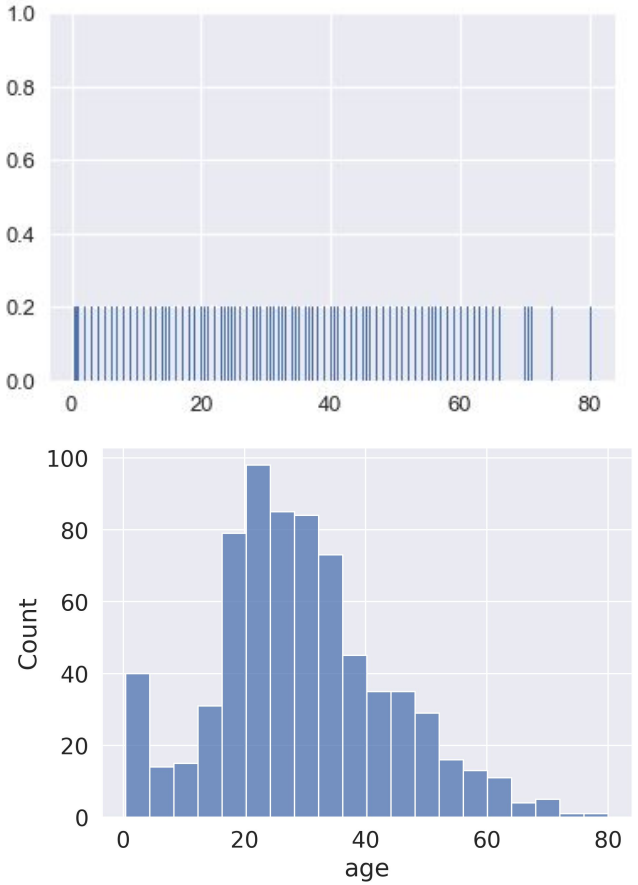
- **Kernel Density Functions**
  - **KDE Mechanics**
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - Transformations
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context

Smoothing in 1D (Histograms)

Arguably, histograms are a “smoothed” version of rug plots.

- Many (sometimes overlapping) data points collected into a single bin or category.

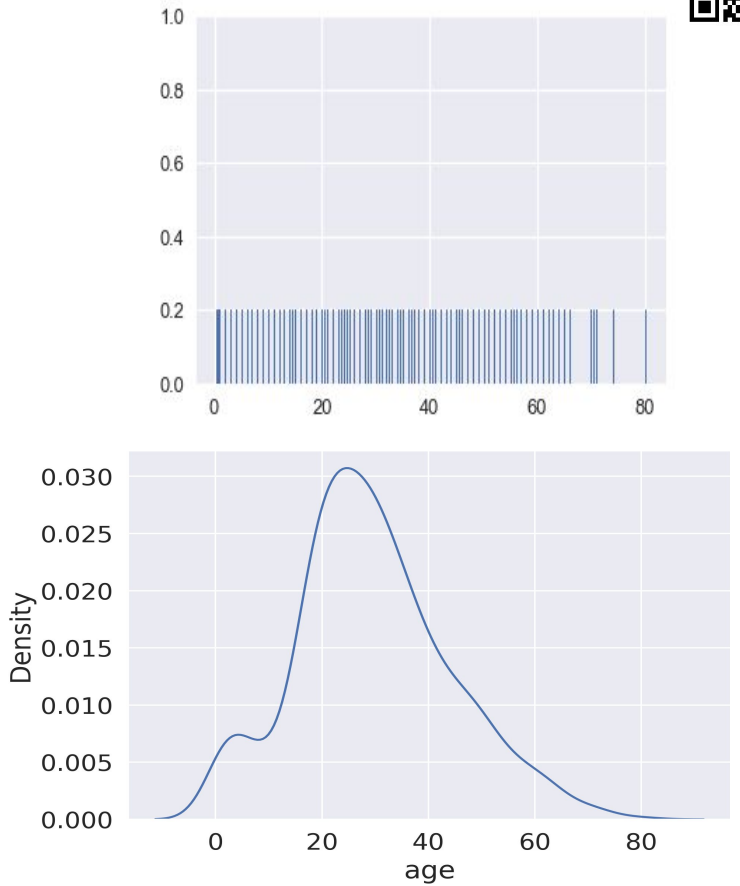
In general, we smooth if we want to focus on general structure rather than individual observations.



Smoothing in 1D (KDEs)

An alternate technique for smoothing 1D data is to use a Kernel Density Estimate.

We've already seen this in the previous lecture, but let's spend some time demystifying this function curve.

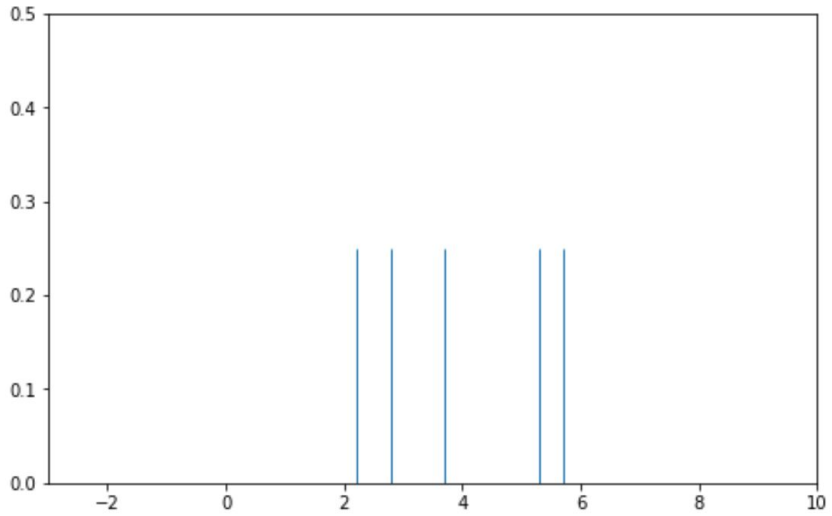


Note: The domain for our KDE is wider than the rug plot axes (e.g. x<0, x>80).

(Intuition) Histogram: Proportional Areas

N=5 points**:** [2.2, 2.8, 3.7, 5.3, 5.7]

Rug plot:

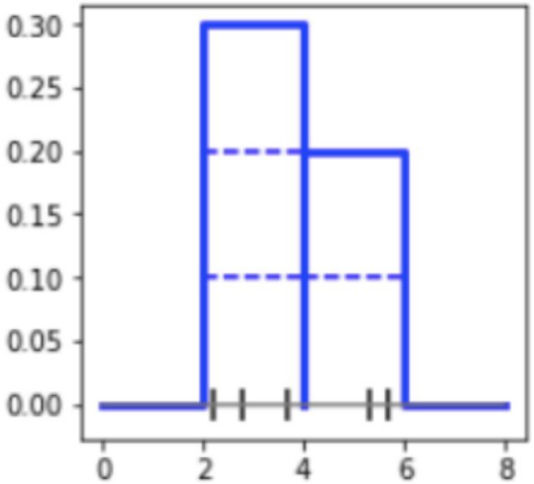


(Intuition) Histogram: Proportional Areas

N=5 points**:** [2.2, 2.8, 3.7, 5.3, 5.7]

In a histogram, **area = proportion**.

Bins	Points
[0, 2)	{}
[2, 4)	{2.2, 2.8, 3.7}
[4, 6)	{5.3 ,5.7}
[6, 8]	{}



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

Each of the N = 5 points:

- Is a 1/5 proportion of the sample.
- Contributes a rectangular area 1/5 to the histogram.
  - Rectangle (bin) Width: 2
  - Rectangle Height: 1/10

The **total area under the curve** is 1.

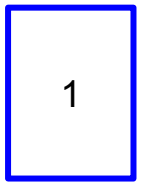
Kernel density estimates follow similar guidelines.

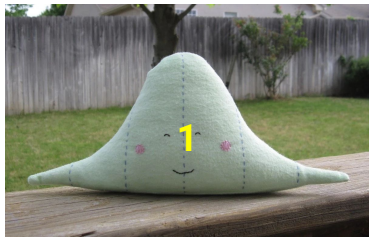
(Intuition) Kernel Density Estimate (KDE): Smoothed Proportional Areas

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

(Data 140 formally defines probability density function.)

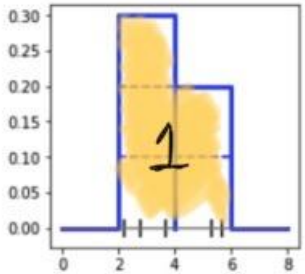
- Just like a histogram, a density curve's **total area must sum to 1**.

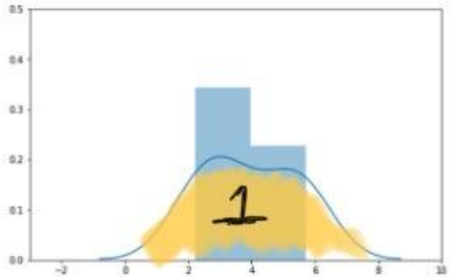




curve with area 1

squash per datapoint





Sum together to make a curve

(Intuition) Kernel Density Estimate (KDE): Smoothed Proportional Areas

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 curve's **total area must sum to 1**.

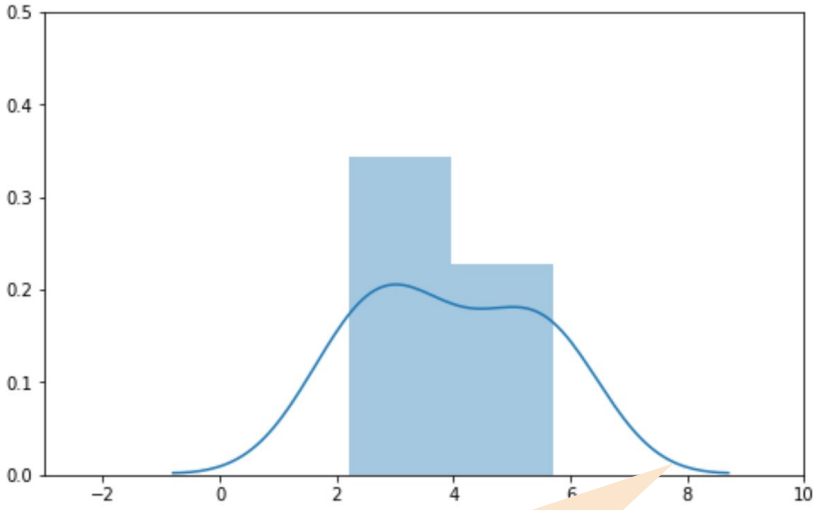
(Data 140 formally defines probability density function.)

KDEs are constructed from the data:

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

To generate a curve we need to choose:

- A **kernel function** (curve), and
- A **bandwidth** (smoothing parameter α).

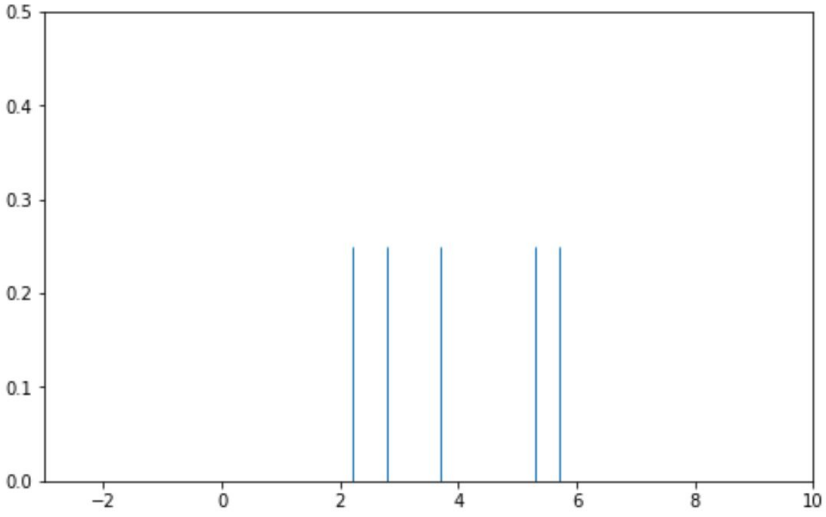


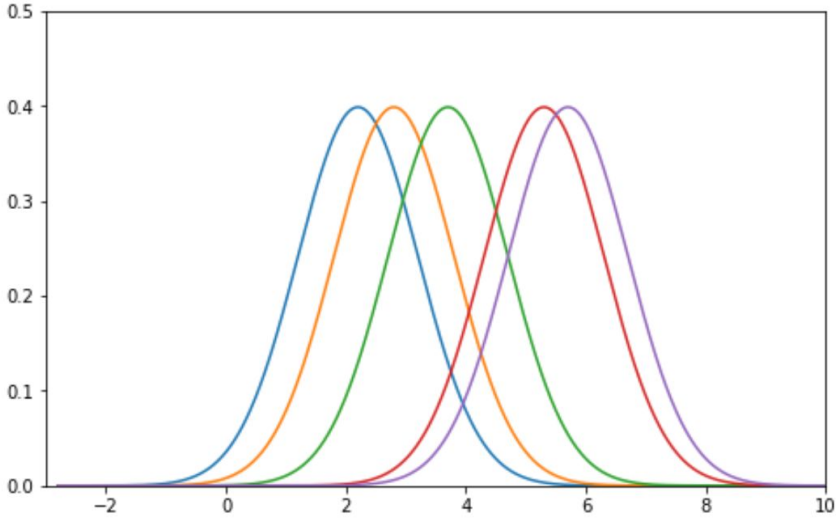
Our goal now is to generate this smooth curve.

Step 1 – Place a kernel at each data point

At each of our 5 points:

- Place a **Gaussian kernel** with **bandwidth** of **alpha = 1**.
- Idea:There is a higher density near the points we've already seen.
- We will precisely define both the **Gaussian kernel** and **bandwidth** in a few slides.

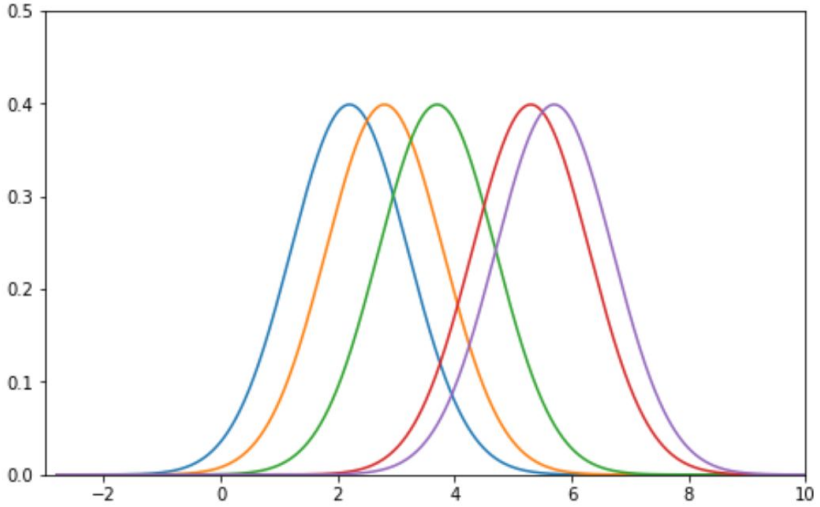


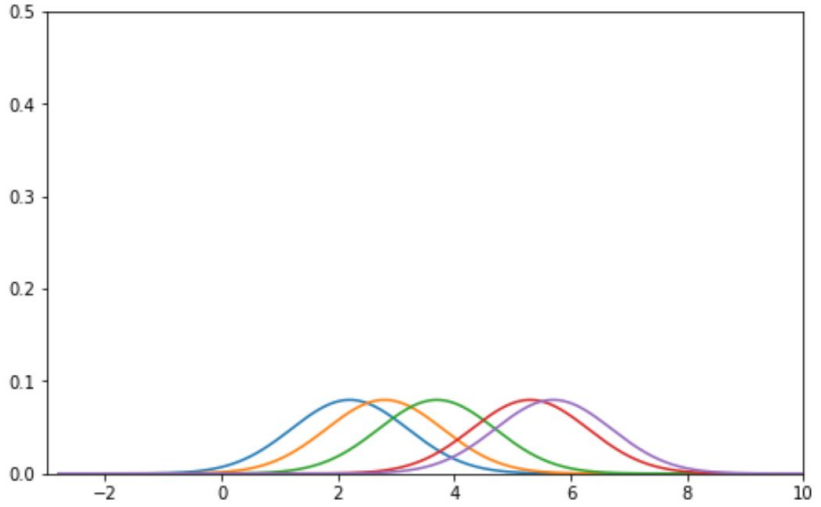


Step 2 – Normalize kernels

In Step 3, we will be summing each of these **kernels**.

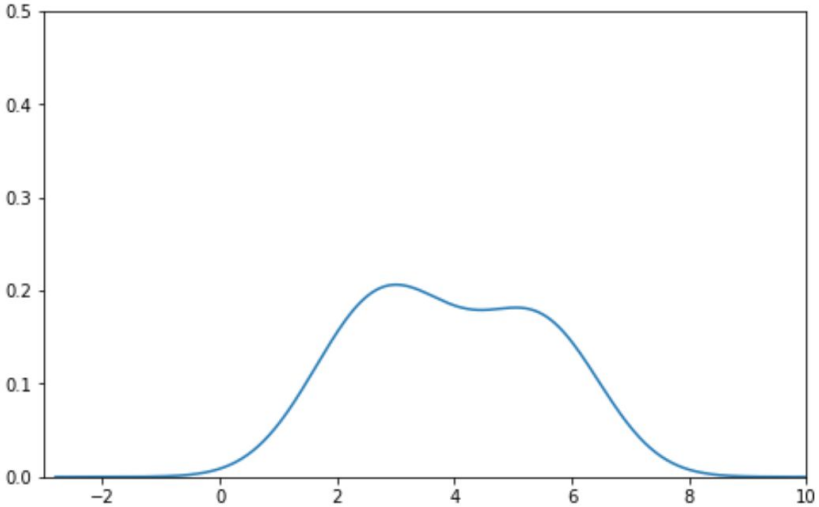
- We want the result to be a valid density, that has area 1.
- We have 5 different **kernels**, each with an area 1.
- So, we **normalize** by multiplying each **kernel** by 1/5.

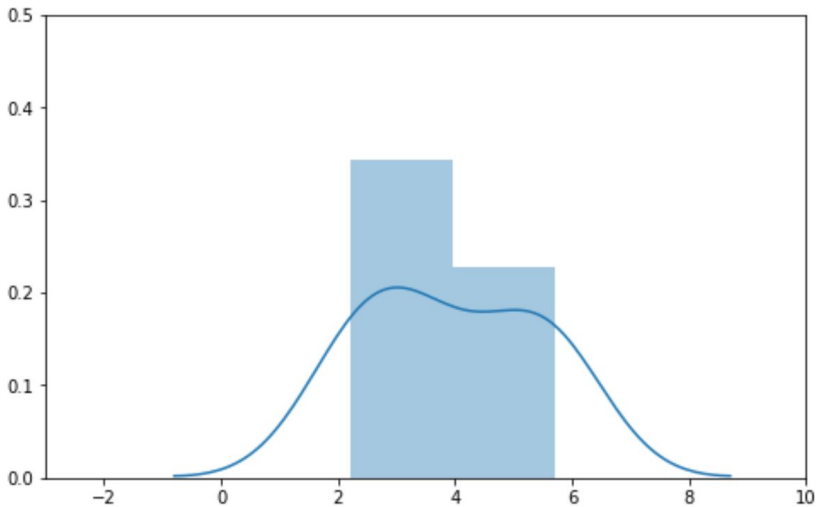




Step 3 - Sum Normalized Kernels

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

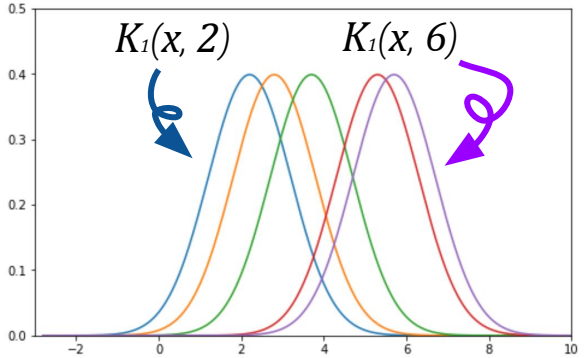




Summary of KDE

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

2 3 1



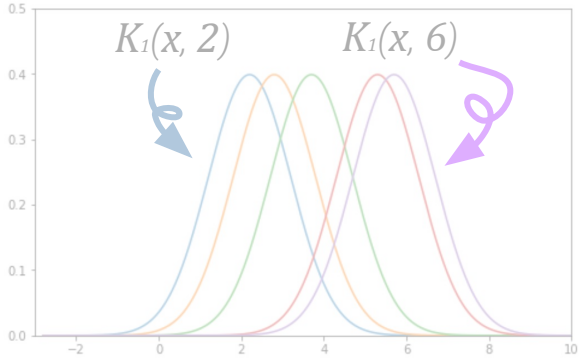
A general "KDE formula" function is given above.

- $K_\alpha(x, x_i)$ is the **kernel** centered on the observation i.
  - Each kernel individually has area 1.
  - **x** represents any number on the number line. It is the input to our function.

Summary of KDE

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

(2) (3) (1)



A general "KDE formula" function is given above.

1. $K_\alpha(x, x_i)$ is the **kernel** centered on the observation i.
   - Each kernel individually has area 1.
   - $\mathbf{x}$ represents any number on the number line. It is the input to our function.
2. $\mathbf{n}$ is the number of observed data points that we have.
   - We multiply by $\mathbf{1/n}$ so that the total area of the KDE is still 1.
3. 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 by summing multiple shifted kernels.

α is the **bandwidth** or **smoothing parameter**.

Kernel Functions and Bandwidth

Lecture 08, Data 100 Spring 2023

- Kernel Density Functions
  - KDE Mechanics
  - **Kernel Functions and Bandwidth**
- Relationships between Quantitative Variables
  - Transformations
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context

Kernel (Functions)

A **kernel** (for our purposes) is a valid density function, meaning:

- It must be non-negative for all inputs.
- It must integrate to 1(area under curve = 1).

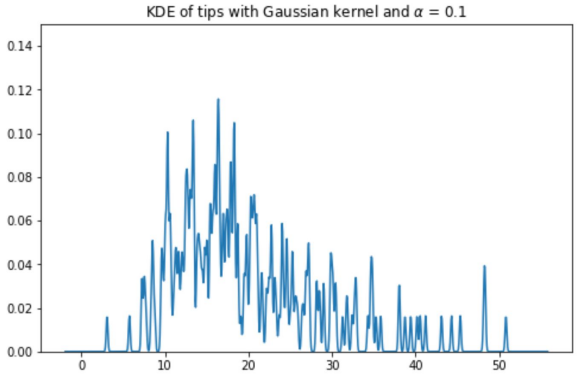
The most common kernel is the **Gaussian kernel**.

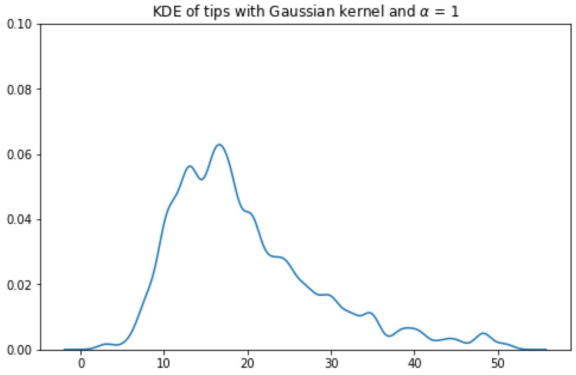
- Here, x represents any input, and x_i represents the ith observed value (datapoint).
- Each kernel is **centered** on our observed values (and so its distribution mean is x_i).
- **α** is the **bandwidth parameter**. It controls the smoothness of our KDE. Here, it is also the standard deviation of the Gaussian.

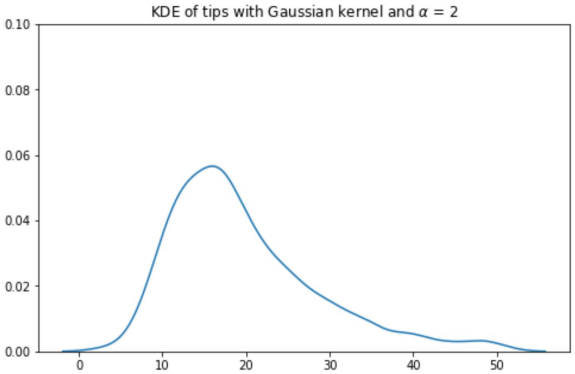
$$K_\alpha(x, x_i) = \frac{1}{\sqrt{2\pi\alpha^2}} e^{-\frac{(x-x_i)^2}{2\alpha^2}}$$

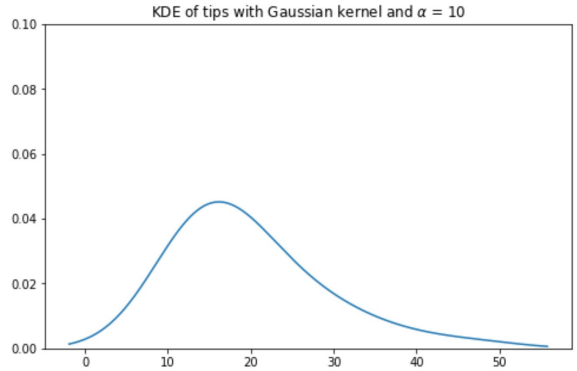
Memorizing this formula is less important than knowing the shape and how the bandwidth parameter **α** smoothes the KDE.

Effect of bandwidth on KDEs



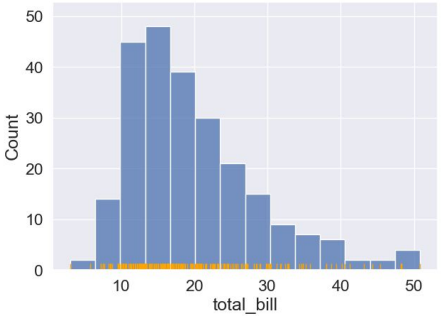






Bandwidth is analogous to the width of each bin in a histogram.

- As **α** increases, the KDE becomes more smooth.
- Large **α** KDE is simpler to understand, but gets rid of potentially important distributional information (e.g. multimodality).



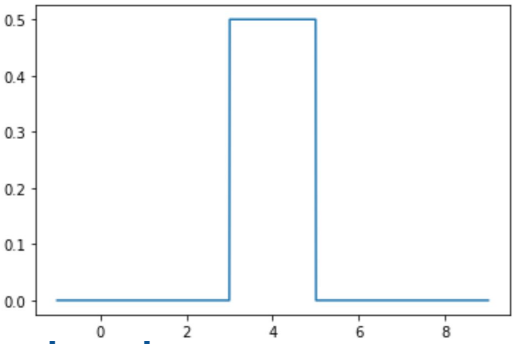
Other Kernels: Boxcar

As an example of another **kernel**, consider the **boxcar kernel**.

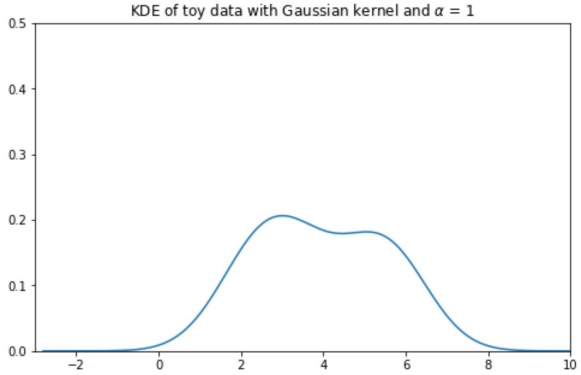
- It assigns uniform density to points within a "window" of the observation, and 0 elsewhere.
- Resembles a histogram… *sort of*.

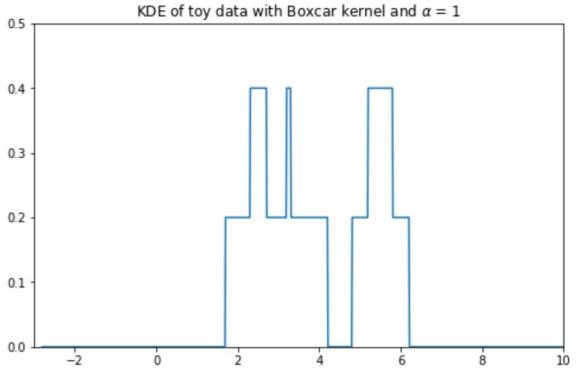
$$K_\alpha(x, x_i) = \begin{cases} \frac{1}{\alpha}, & |x - x_i| \leq \frac{\alpha}{2} \\ 0, & \text{else} \end{cases}$$

- Not of any practical use in DS100! Presented as a simple theoretical alternative.



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





For even more **kernels**, see https://en.wikipedia.org/wiki/Kernel_(statistics)

Kernels in Practice

Despite a great deal of literature in statistics on kernel properties (e.g. the Epanechnikov kernel has some nice theoretical properties), the libraries we use in this class only support a **Gaussian kernel**.

seaborn.kdeplot ¶

`seaborn.kdeplot` (*x=None, *, y=None, shade=None, vertical=False, kernel=None, bw=None, gridsize=200, cut=3, clip=None, legend=True, cumulative=False, shade_lowest=None, cbar=False, cbar_ax=None, cbar_kws=None, ax=None, weights=None, hue=None, palette=None, hue_order=None, hue_norm=None, multiple='layer', common_norm=True, common_grid=False, levels=10, thresh=0.05, bw_method='scott', bw_adjust=1, log_scale=None, color=None, fill=None, data=None, data2=None, warn_singular=True, **kwargs*) ¶

Plot univariate or bivariate distributions using kernel density estimation.

kernel : *str*

Function that defines the kernel.

Deprecated since version 0.11.0: support for non-Gaussian kernels has been removed.

On the lab, we'll have you implement a KDE yourself.

- The goal is so that you don't think of KDEs as magic.
- Instead, we want you to realize that KDEs are derived from data in a simple, mathematically defined way.

Relationships Between Quantitative Variables

Lecture 08, Data 100 Spring 2023

- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- **Relationships between Quantitative Variables**
  - Transformations
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context

From distributions to individual variables

Up until now, we focused exclusively on visualizing variable distributions…

Now we will visualize **relationships** between variables.

Scatter plots

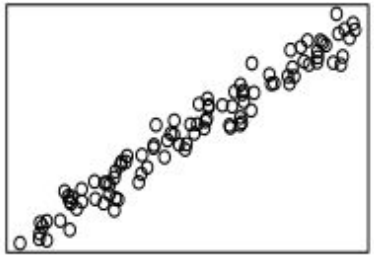
Scatter plots are used to reveal relationships between **pairs** of numerical variables.

- Visual assessment may help us decide how to model these relationships.

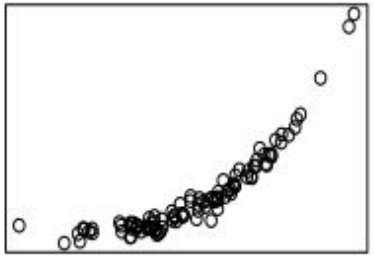
- Example: Linear model
  - Linear Regression (Data 8)
  - Good for the left two, not so much for the right two.
- Reminder: "Correlation does not imply causation." A linear relationship is a mathematical one.

Next lecture, we will precisely define modeling and review the simple linear regression model.

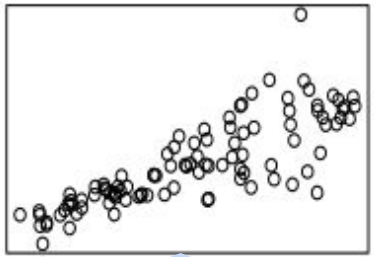
simple linear



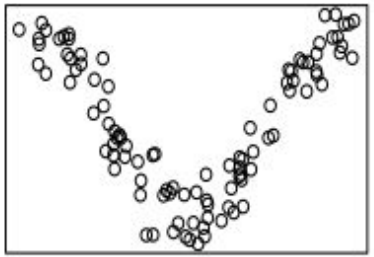
simple nonlinear



linear, spreading

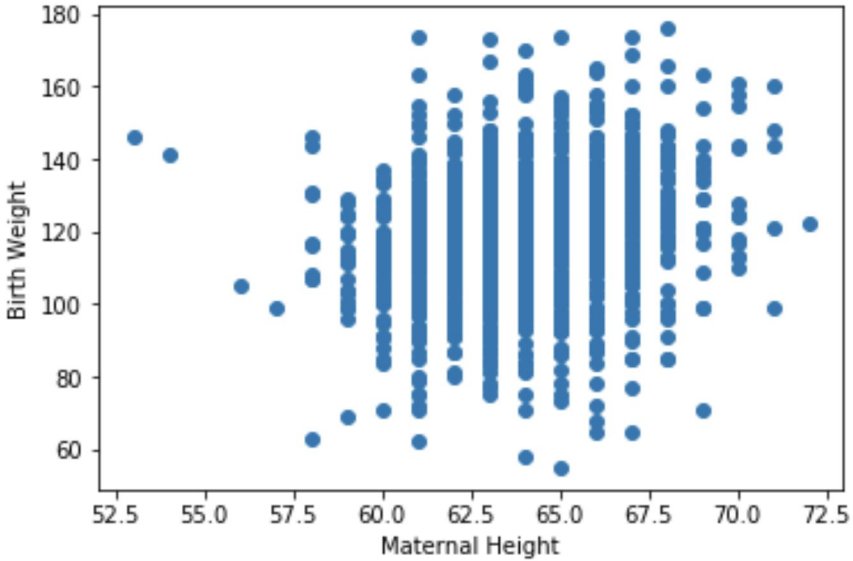


v-shaped



relationship appears linear, but with increasing spread as x gets larger

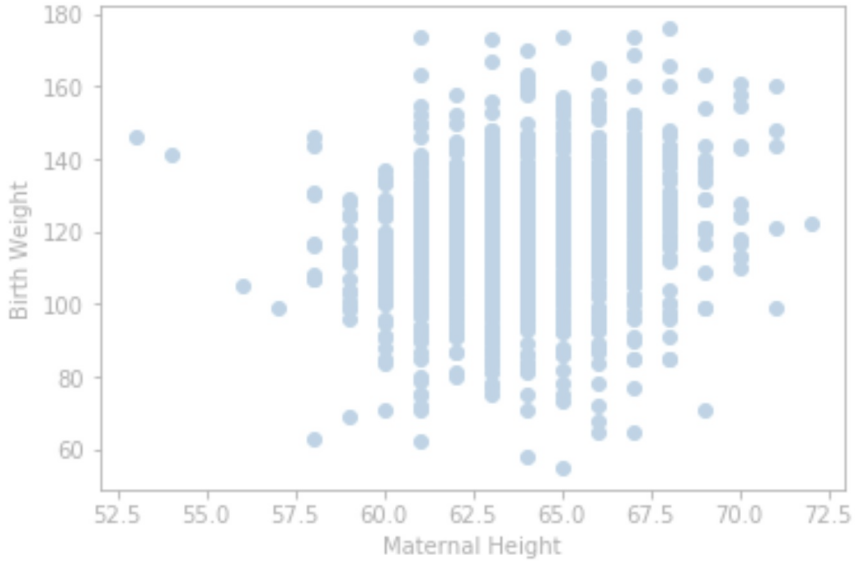
Scatter Plot on our Birth Data (Matplotlib Example)

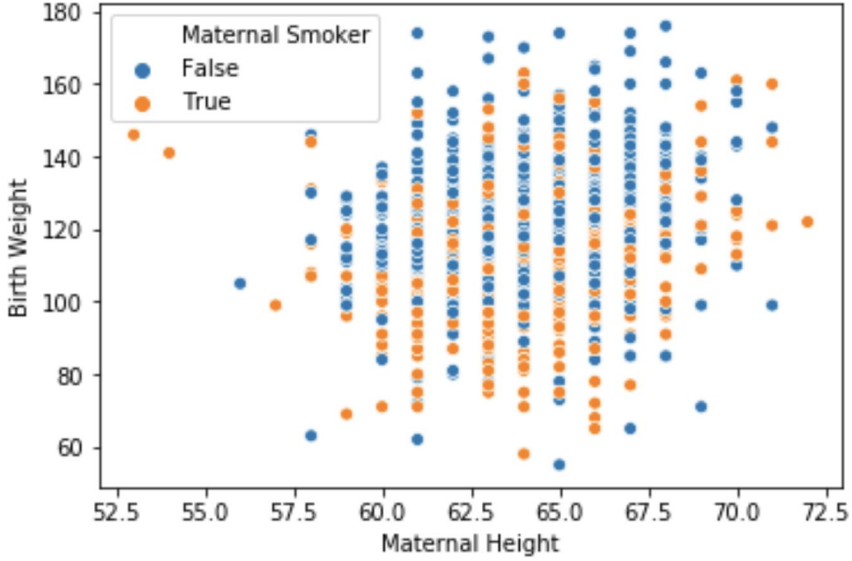



```
plt.scatter(births['Maternal Height'], births['Birth Weight']) # array/series
# OR
plt.scatter(data=births, x='Maternal Height', y='Birth Weight') # dataframe

plt.xlabel('Maternal Height')
plt.ylabel('Birth Weight')
```

Scatter Plot on our Birth Data (Seaborn Example)





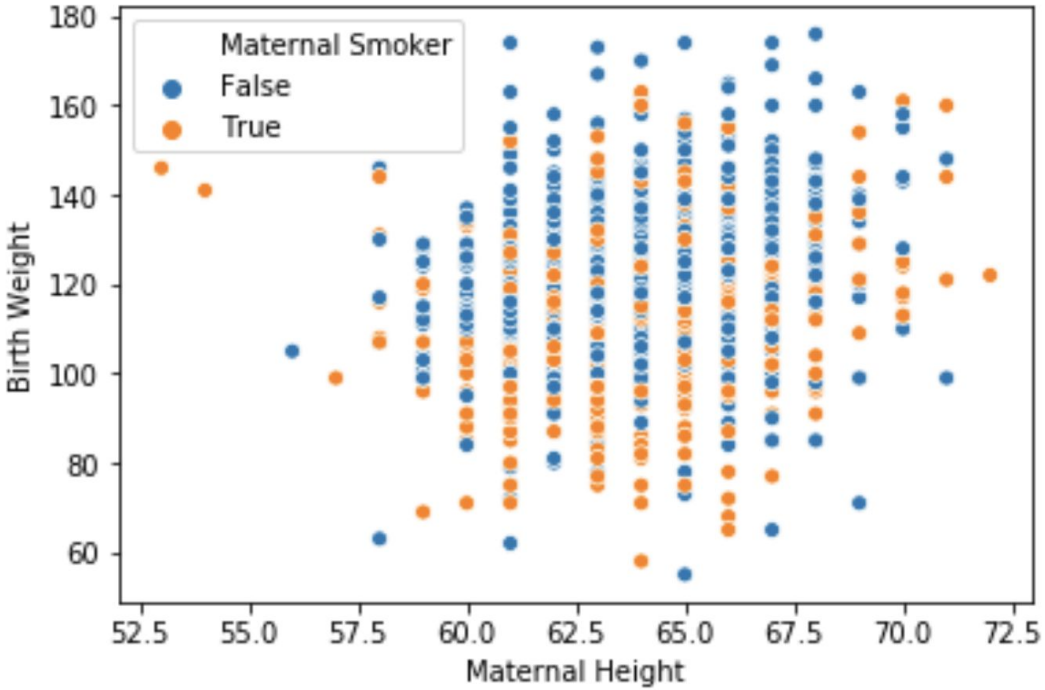

```
plt.scatter(data=births, x='Maternal Height', y='Birth Weight')
plt.xlabel('Maternal Height')
plt.ylabel('Birth Weight')
```

```
sns.scatterplot(data = births,
                x = 'Maternal Height',
                y = 'Birth Weight',
                hue = 'Maternal Smoker')
```

The Seaborn example on the right uses **color** to add a **third dimension** to our plot!

- Unlike earlier plots, the color represents information present nowhere else.
- Note that our box plot and violin plots were MUCH better assessments of these two distributions. Harder to see the lower weight for **True** babies.

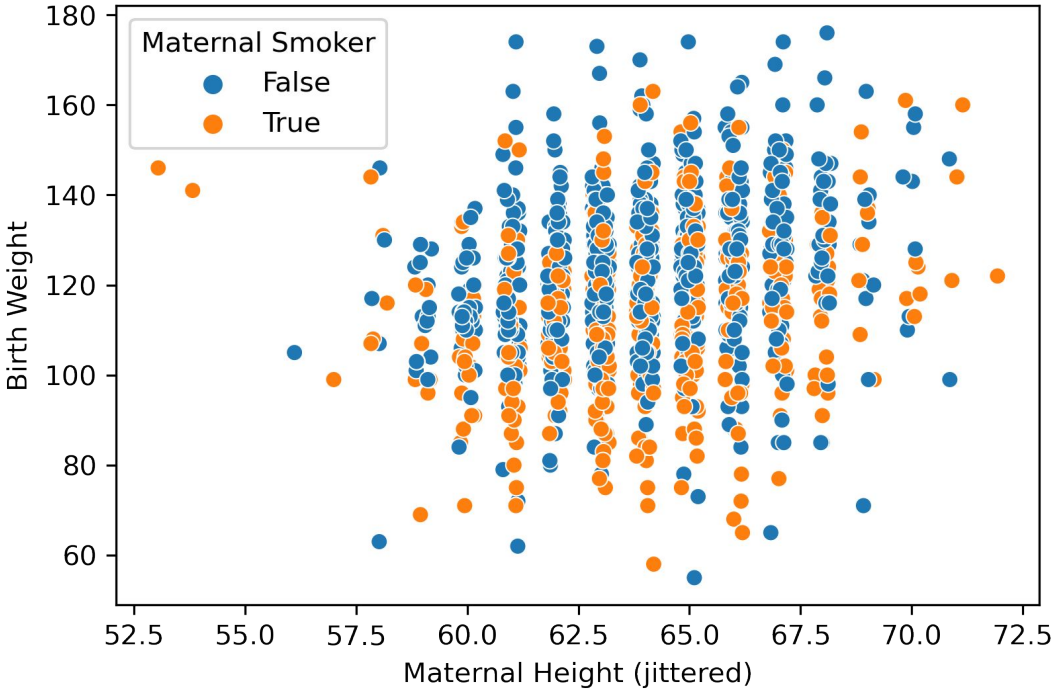
Scatter Plot on our Birth Data (Seaborn Example)



This plots suffers from overplotting – many of the points are on top of one another!

- Any suggestions for fixing this? (still has to be a scatter plot)

Scatter Plot on our Birth Data (Seaborn Example)



One solution: Add some **jitter** (random noise) to the x variable.

- Is this really reasonable???

```
births["Maternal Height (jittered)"] = births["Maternal Height"] + np.random.uniform(-0.2, 0.2, len(births))
```

Reminder: Goals of Data Visualization

Goal 1: To **help your own understanding** of your data/results.

- Key part of exploratory data analysis.
- Useful throughout modeling as well.
- Lightweight, iterative and flexible.

Goal 2: To **communicate results/conclusions to others**.

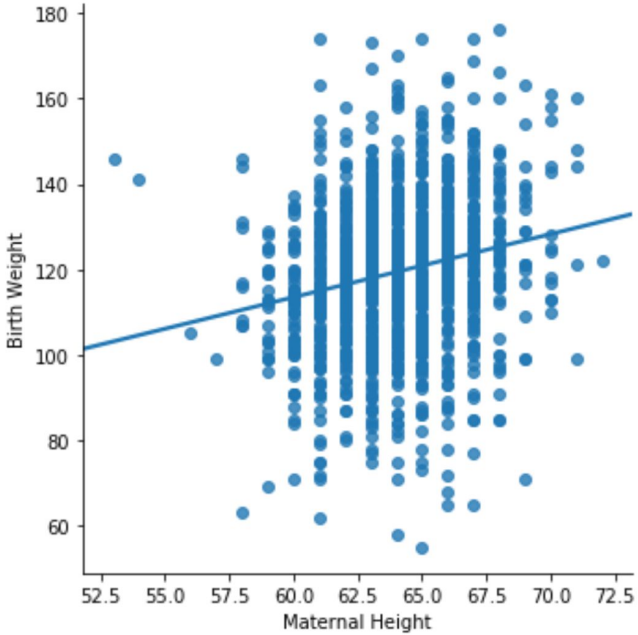
- Highly editorial and selective.
- Be thoughtful and careful!
- Fine tuned to achieve a communications goal.
- Often time-consuming: bridges into design, even art.

Based on this, I'd say jittering is:

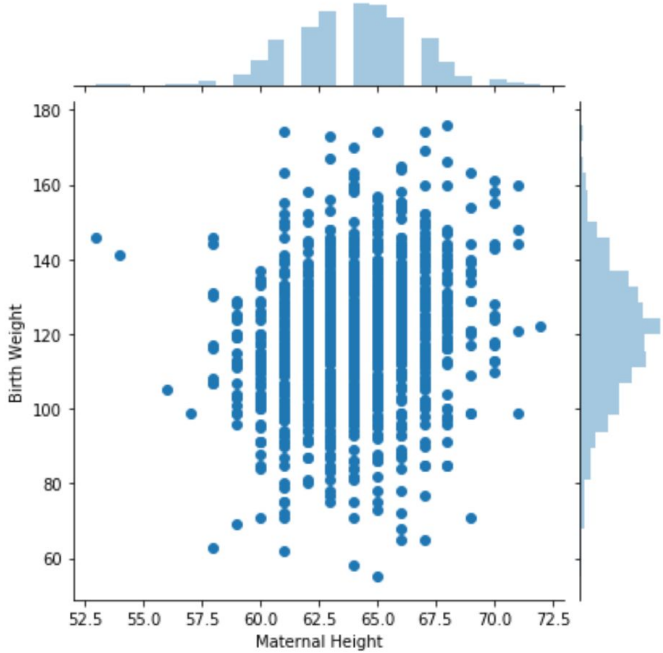
- Completely reasonable for goal 1.
- Possibly reasonable for goal 2 (if you can communicate exactly what you did well).

A constant tool across the lifecycle of data science

Scatter plots




```
sns.lmplot(data=births, x='Maternal Height', y='Birth Weight', ci=False)
```




```
sns.jointplot(data=births, x='Maternal Height', y='Birth Weight')
```

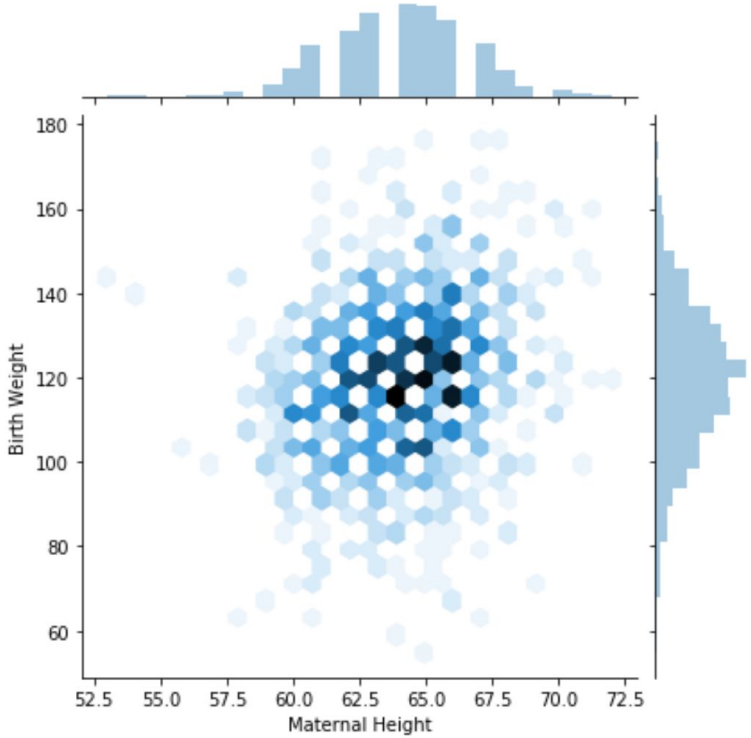
Hex plots

Can be thought of as a two dimensional histogram. Shows the joint distribution.

- The xy plane is binned into hexagons.
- More shaded hexagons typically indicate a greater density/frequency.

Why hexagons instead of squares?

- Easier to see linear relationships.
- More efficient for covering region.
- Visual bias of squares – drawn to see vertical and horizontal lines.

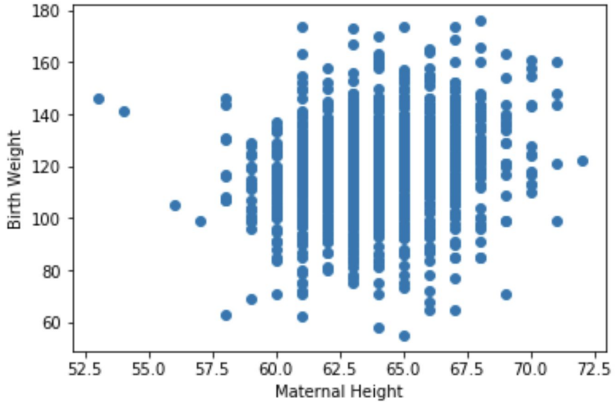


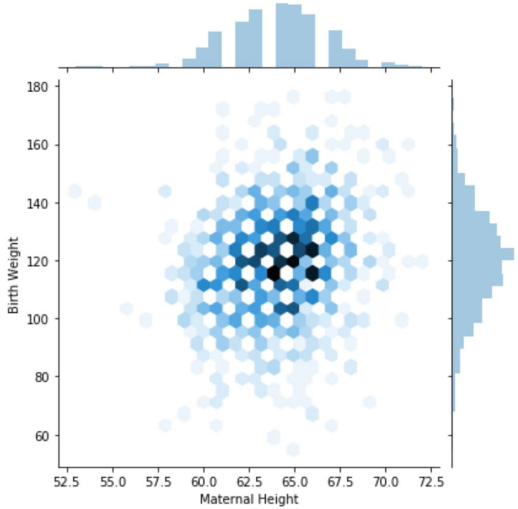

```
sns.jointplot(data=births, x='Maternal Height',
y='Birth Weight', kind='hex')
```

Smoothing in 2D (Hex Plots)

Similarly, we can think of a heatmap as a smoothed version of a scatter plot.

- Color represents each bin's "height".





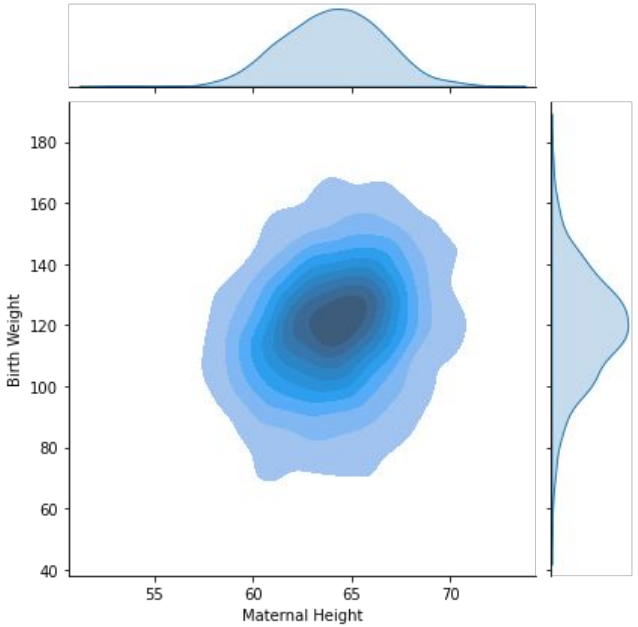
Contour plots

Contour plots are two dimensional versions of density curves.

- Will reappear when we study gradient descent!

Each of the last few plots has been created by **`sns.jointplot`**.

- By default, shows **marginal** distributions on the horizontal and vertical axes.
- These are the histograms/density curves of each variable independently.



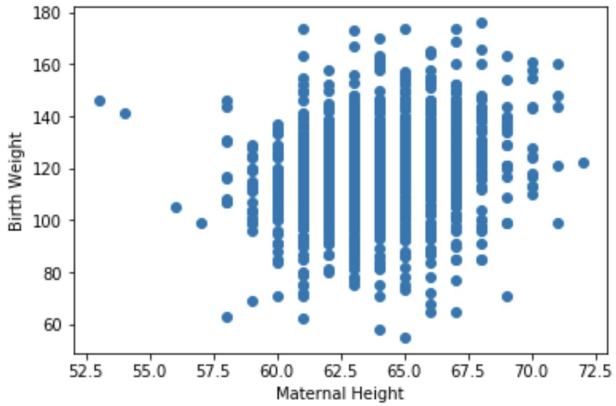

```
sns.jointplot(data=births, x='Maternal Height',
y='Birth Weight', kind='kde', fill=True)
```

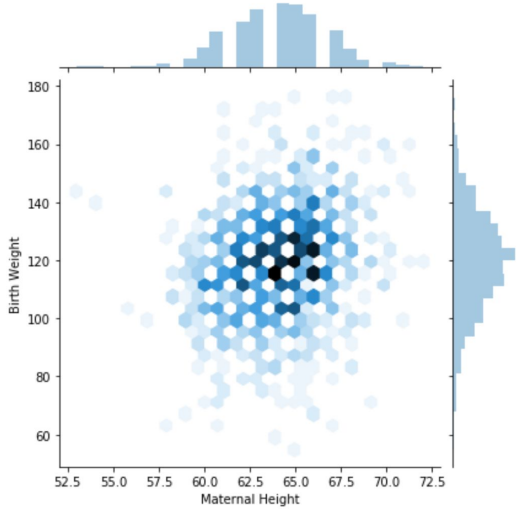
Generalizing to 2D (Contour Map)

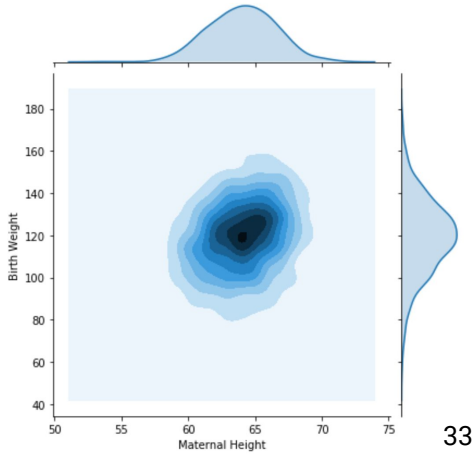
Note the analogies from 1-D to 2-D.

- Rug Plot :: Histogram :: KDE
- Scatter Plot :: Hex Plot :: Contour Map

You'll have a chance to play around with contour maps on the homework.







Summary

- **Visualization requires a lot of thought!**
- Many tools for visualizing distributions.
  - Distribution of a single variable: rug plot, histogram, density plot, box, violin.
  - Joint distribution of two quantitative variables: scatter plot, hex plot, contour plot.
- This class primarily uses seaborn and matplotlib.
  - Pandas also has basic built-in plotting methods.
  - Many other visualization libraries exist. **`plotly`** is one of them.
    - It very easily creates **interactive** plots.
    - plotly will occasionally appear in lecture code, labs, and assignments!

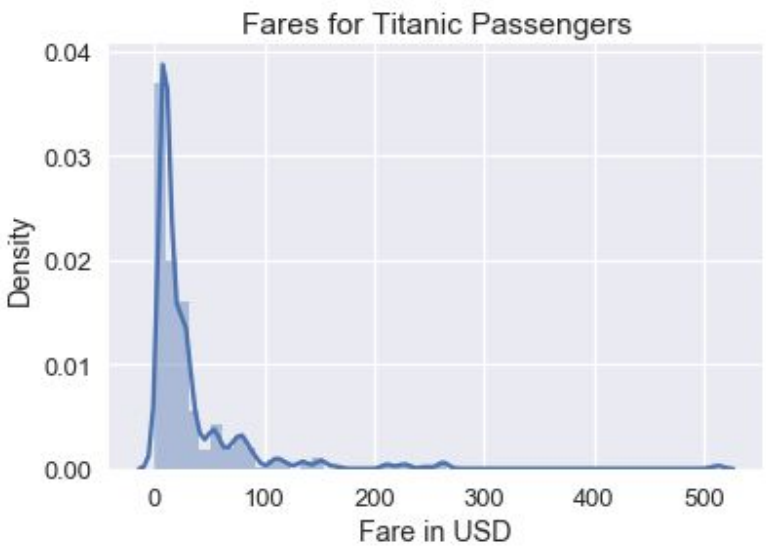
Next, we'll go deeper into the theory behind visualization.

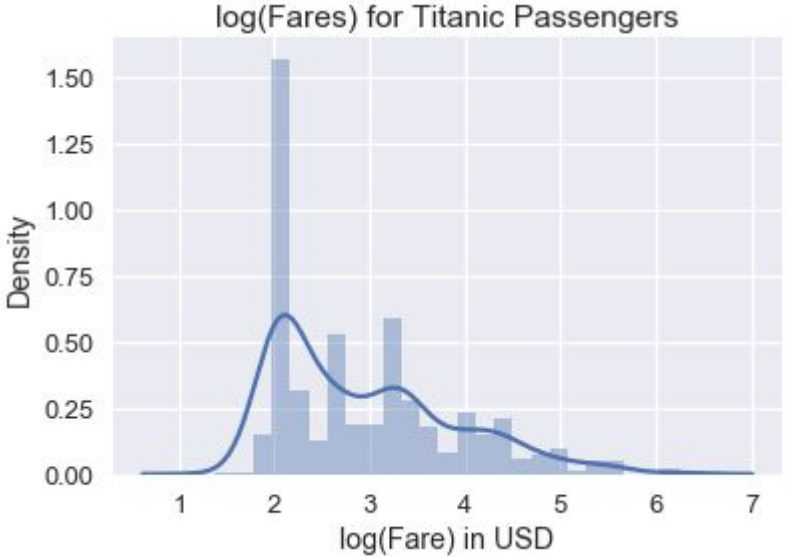
Transformations

Lecture 08, Data 100 Spring 2023

- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - **Transformations**
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context

Transforming data can reveal patterns





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

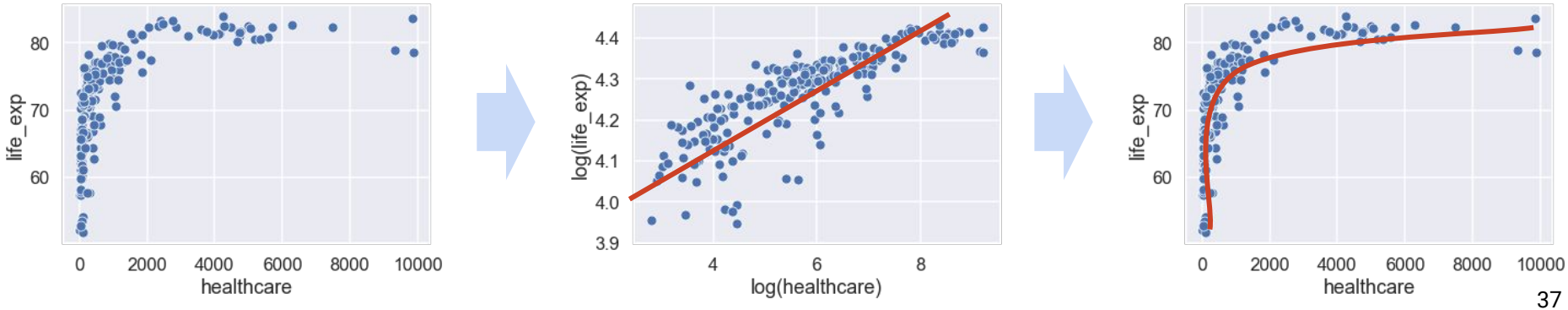
Why straighten relationships?

Now, we will look at how to **transform** the scatter plot of two variables to **linearization**.

- Make a transformation so that y vs. x is linear
- Fit a line to this transformed plot
- "**Backtrack**" to figure out the exact relationship between x and y.

Why?

- Linear relationships are particularly simple to interpret.
- We know what slopes and intercepts mean.
- We will be doing a lot of linear modeling in this course.



Log of just y-values

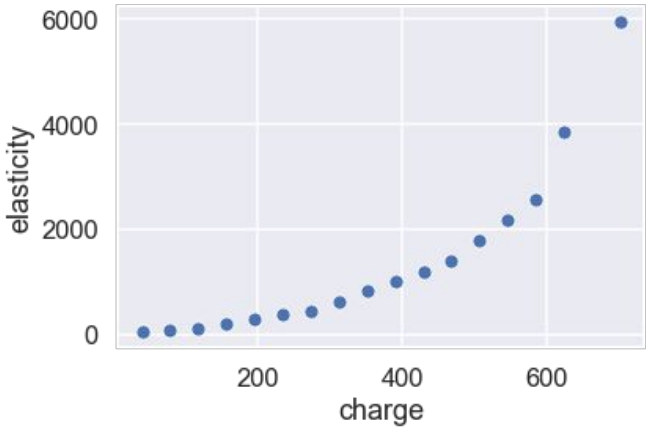
Transform:

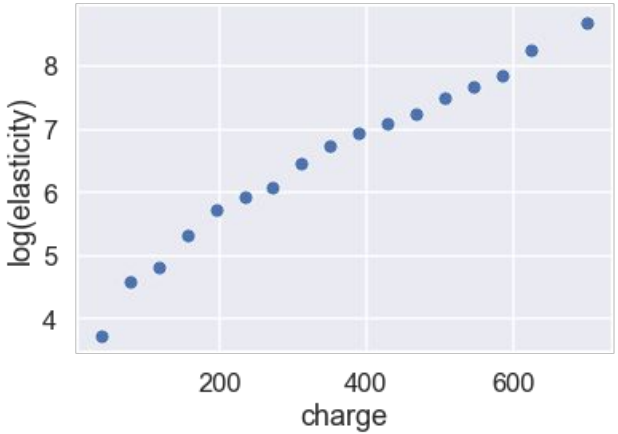
- If we take the **log of our y-values** and notice a **linear** relationship…
- We can say (roughly) that $\log y = ax + b$

"Backtrack":

- If we solve then solve the linear equation for y…
- We see that this implies an **exponential** relationship in the **original plot**.

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

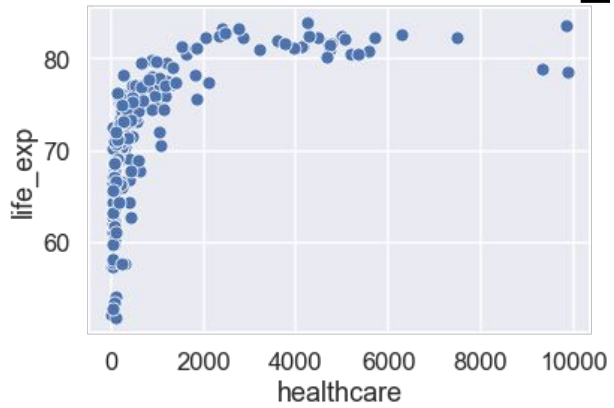




Log-Log plot: Log of both x and y-values

Transform:

- If we take the **log of both axes** and notice a **linear** relationship…
- We can say (roughly) that $\log y = a \cdot \log x + b$



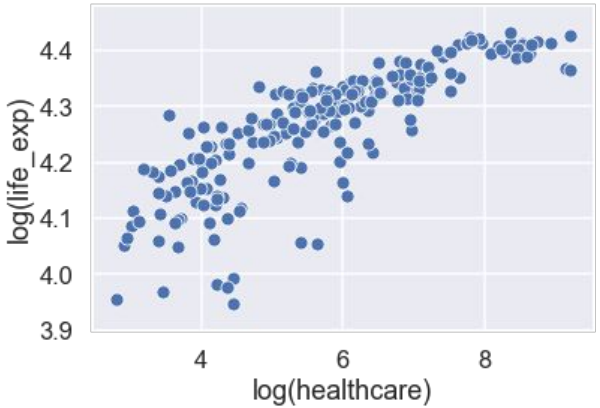
"Backtrack":

- This time if we solve for y,…
- We see that this implies an **power** relationship in the **original plot** (i.e., a one-term **polynomial**).

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

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

$$y = Cx^a$$



For more: https://en.wikipedia.org/wiki/Power_law

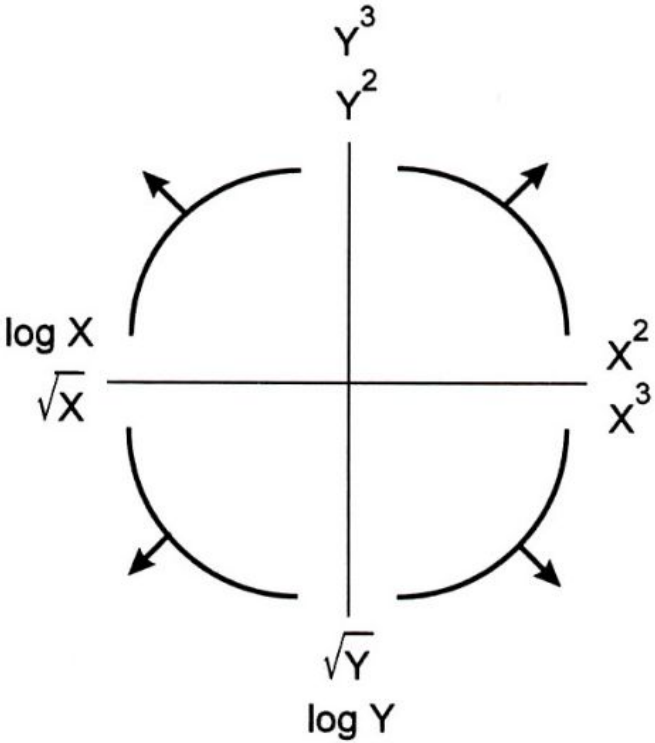
Tukey-Mosteller Bulge Diagram

The **Tukey-Mosteller Bulge Diagram** is a guide to possible transforms to try to get linearity.

- 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.

Other goals other than linearity are possible

- E.g. make data appear more symmetric (see lab).
- Linearity allows us to fit lines to the transformed data



While many use this Bulge Diagram, generally log or log-log plots are the most common first steps to linearizing a visual relationship.

Demo Slides

Johann Kepler and Log Transformations

Log transform as a “Swiss army knife”

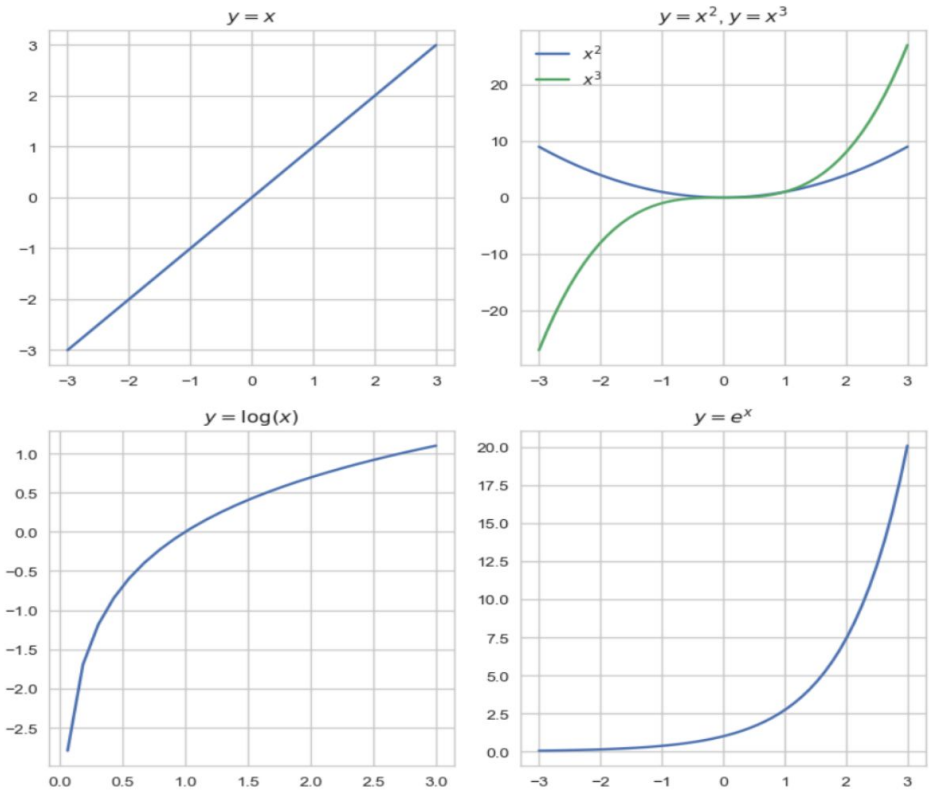
Properties of logarithms make them very powerful!

$$y = a^x \rightarrow \boxed{\log(y) = x\log(a)}$$

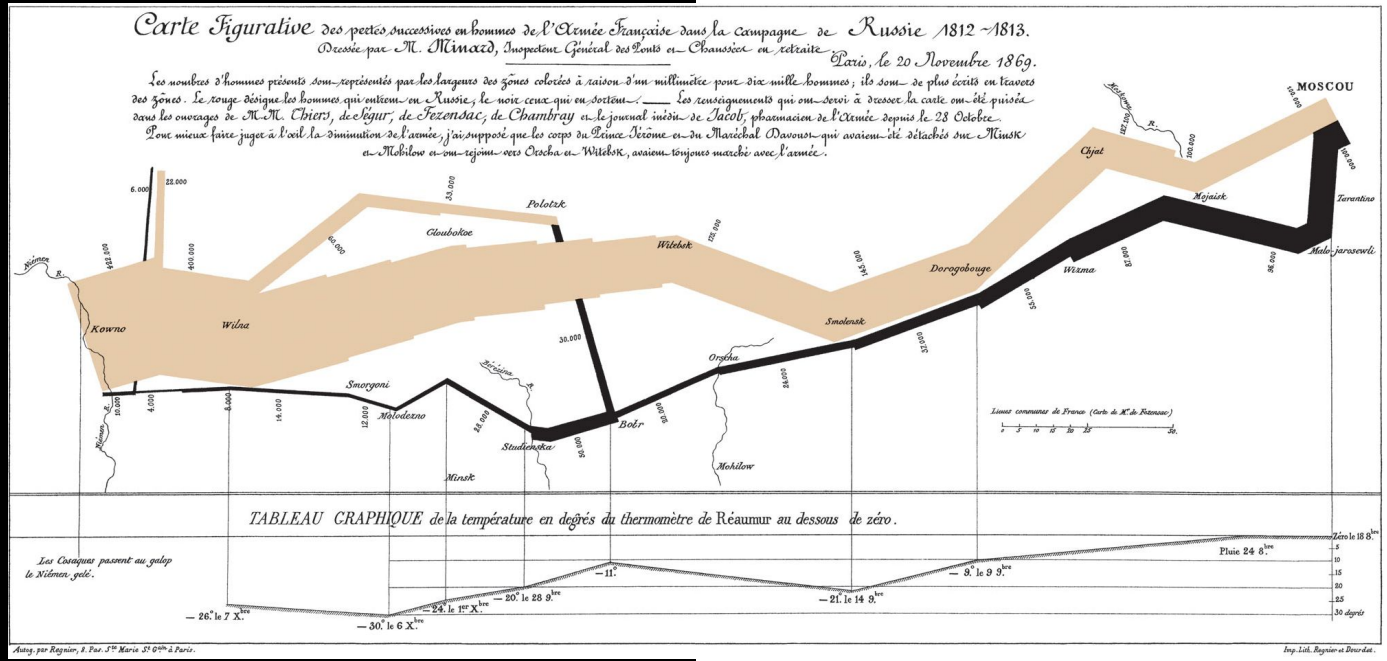
$$y = ax^k \rightarrow \boxed{\log(y) = \log(a) + k\log(x)}$$

Basic functional relations

Knowing the general shapes of polynomial, exponential, and logarithmic curves (regardless of base) can be very helpful for EDA and modeling.



Carte Figurative by Charles Joseph Minard



Interlude

From the mid-1800s: Depicts Napoleon's losses during the Russian campaign of 1812. Visualizes six features in two dimensions: # of troops, distance, temperature, latitude/longitude, direction of travel, date.

slido

How many hours on average have you spent on Homeworks so far?

ⓘ Start presenting to display the poll results on this slide.

Visualization Theory

Lecture 08, Data 100 Spring 2023

- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - Transformations
- **Visualization Theory**
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context

Reminder: Goals of Data Visualization

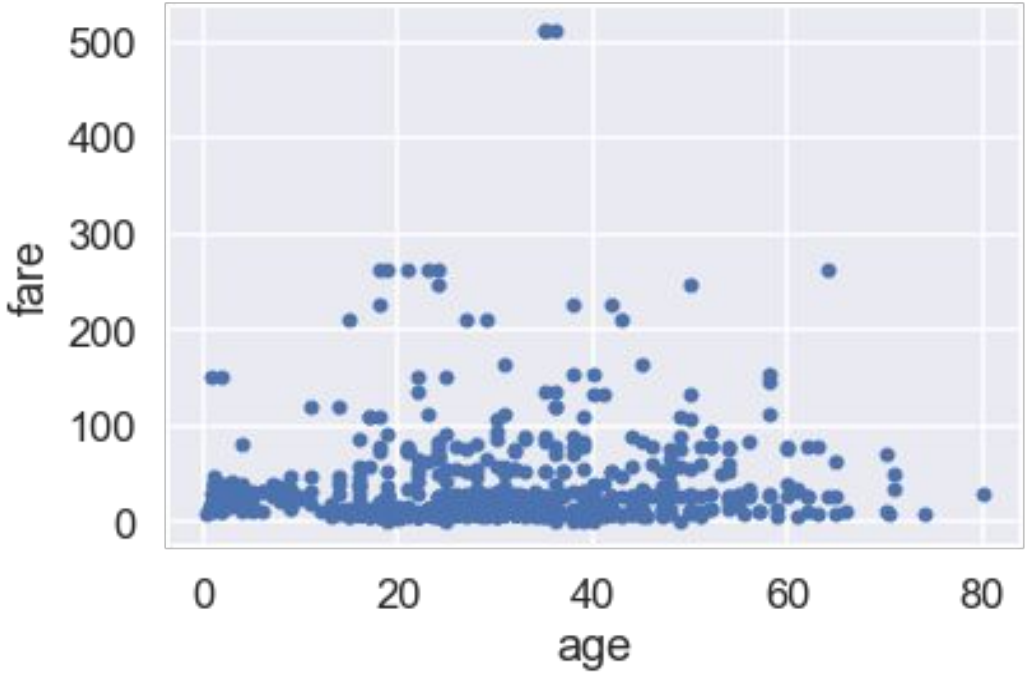
Goal 1: To **help your own understanding** of your data/results.

Goal 2: To **communicate results/conclusions to others**.

An ever-present tool across the data science lifecycle

Visualizations Are For Humans

“Looks like older people didn’t spend more money on tickets for the Titanic than younger people.”

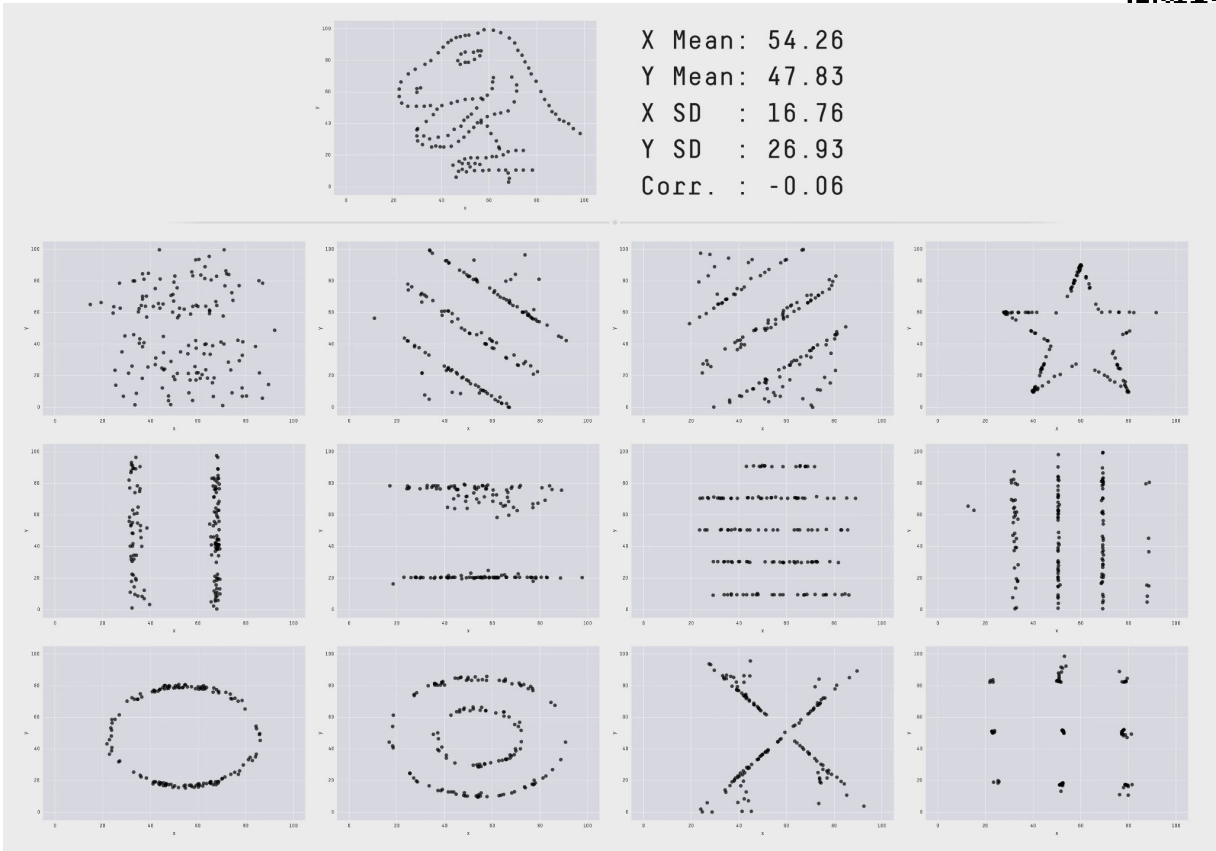


(Note: A histogram or KDE would give stronger evidence than a scatter plot.)

Visualizations Are More Expressive than Summary Statistics

Each of these 13 datasets has the same mean, standard deviation, and correlation coefficient.

Visualizations complement statistics.



https://www.autodesk.com/research/publications/same-stats-different-graphs

But What is Visualization?

(Castlevania meme, circa 2009.)

But What is Visualization?

Visualization is the use of computer-generated, interactive, visual representations of data to ***amplify cognition****.*

Card, Mackinlay, & Shneiderman 1999

…finding the ***artificial memory*** *that best* ***supports*** *our natural means of* ***perception***

[Bertin 1967]

Information Channels

Lecture 08, Data 100 Spring 2023

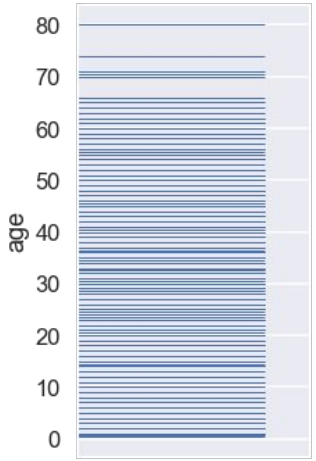
- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - Transformations
- Visualization Theory
  - **Information Channels**
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context

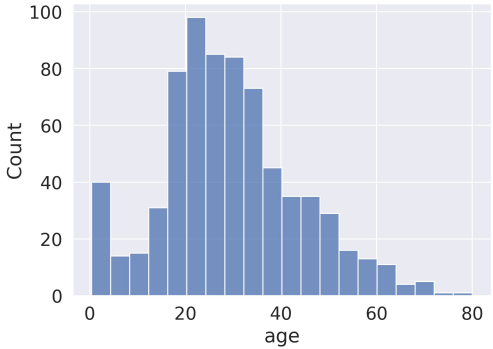
Take advantage of the human visual perception system

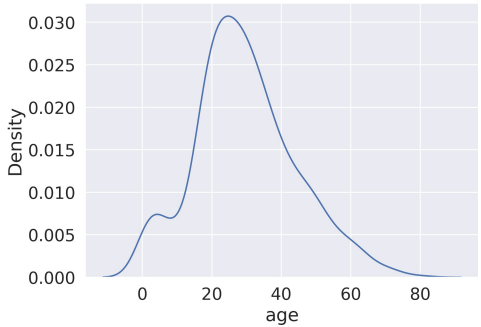
Data can be visualized in many ways!

- Let’s deconstruct the most basic plot types.

	age
0	22.0
1	38.0
2	26.0
...	...
888	NaN
889	26.0
890	32.0

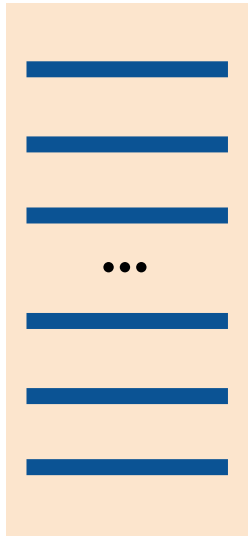






Rug Plot: Encoding 1 Variable

	age
0	22.0
1	38.0
2	26.0
...	...
888	NaN
889	26.0
890	32.0



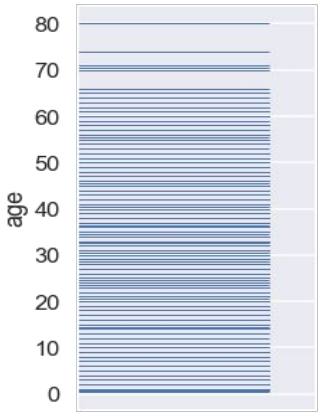
Mark

(Represents a datum)

10px
16px
11px
...
NONE
11px
15px

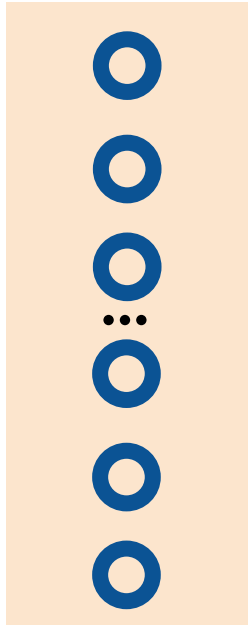
Encoding

(Maps datum to visual position)



Rug Plot: Different Marks

	age
0	22.0
1	38.0
2	26.0
...	...
888	NaN
889	26.0
890	32.0



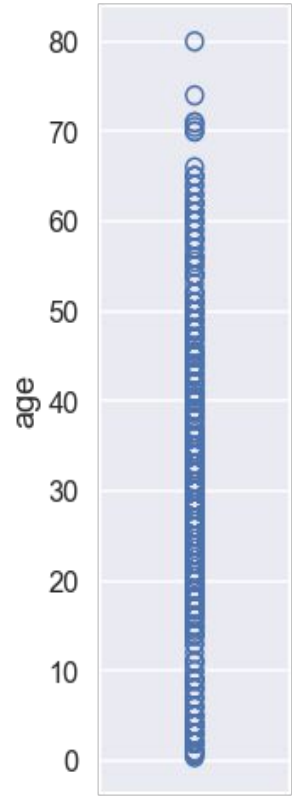
Mark

(Represents a datum)

10px
16px
11px
...
NONE
11px
15px

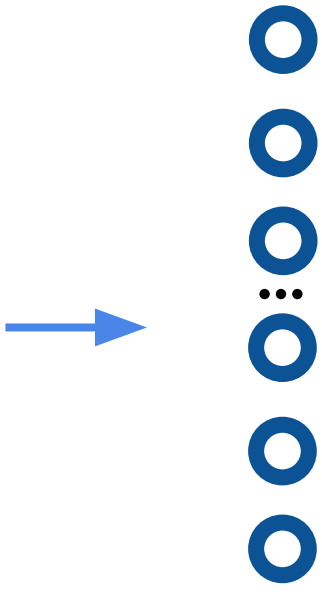
Encoding

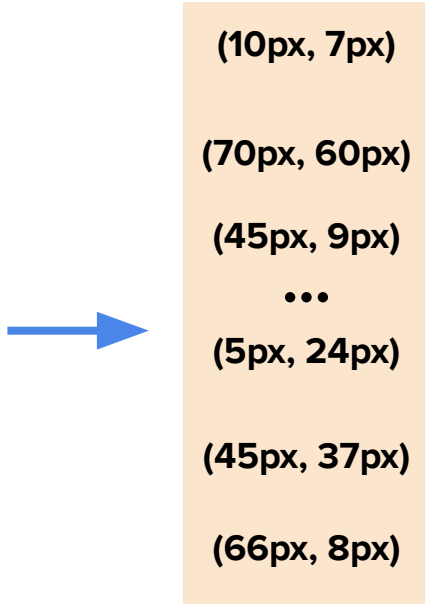
(Maps datum to visual position)

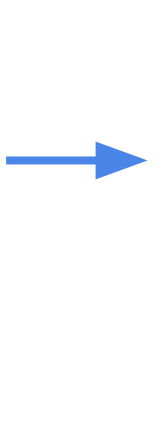


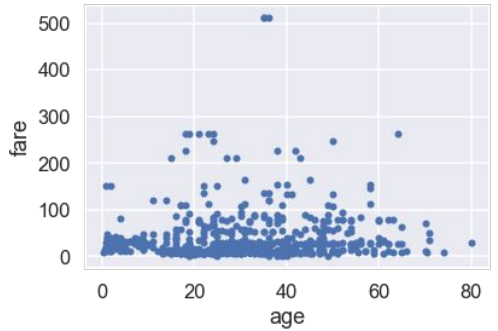
Scatter Plot: Encoding 2 Variables

	age	fare
0	22.0	7.25
1	38.0	71.28
2	26.0	7.92
...	...	...
888	NaN	23.45
889	26.0	30.00
890	32.0	7.75









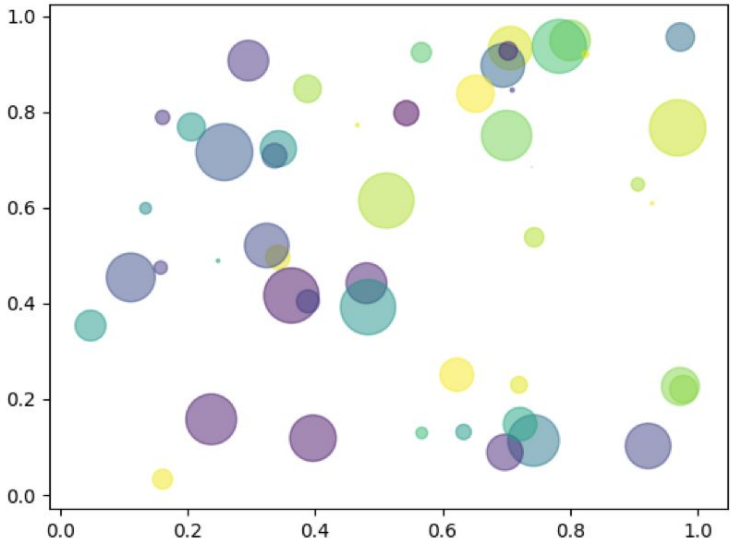
Mark
(Represents a datum)

Encoding
(Maps datum to visual position)

Going Beyond: Encoding 3+ Variables

How many variables are we encoding here?

- In other words, how many “channels” of information are there?



slido

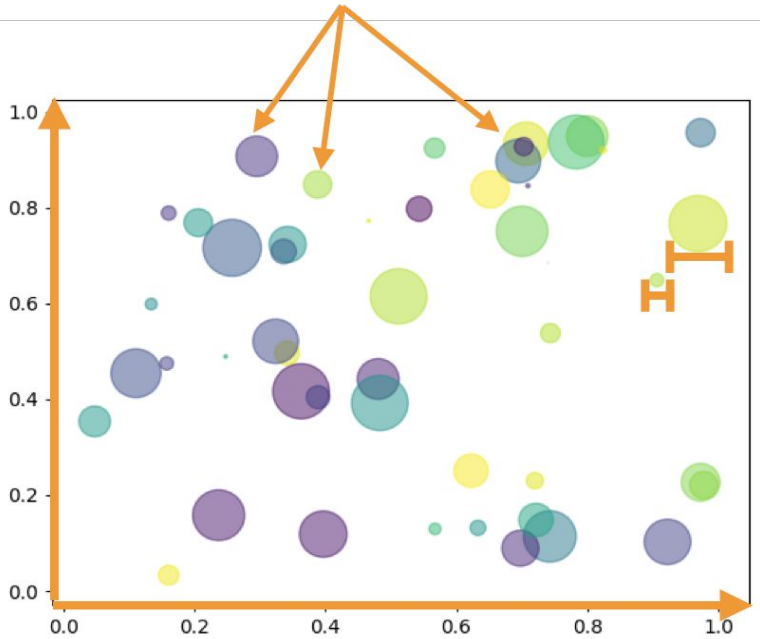
How many variables are we encoding here?

ⓘ Start presenting to display the poll results on this slide.

Going Beyond: Encoding 3+ Variables

How many variables are we encoding here?

- In other words, how many “channels” of information are there?



Answer: 4.

- x
- y
- area
- color

We could add even more: Shapes, outline colors of shapes, shading, etc.
There are infinite possibilities!

Abusing Encodings: Length

There are many things that can go wrong in a visualization. For example, the visualization below abuses the **length channel**:

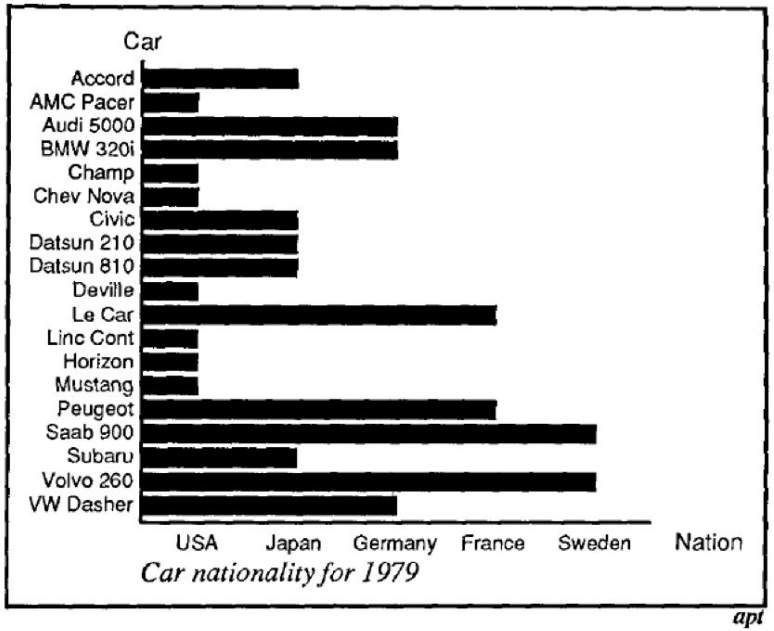


Fig. 11. Incorrect use of a bar chart for the *Nation* relation. The lengths of the bars suggest an ordering on the vertical axis, as if the USA cars were longer or better than the other cars, which is not true for the *Nation* relation.

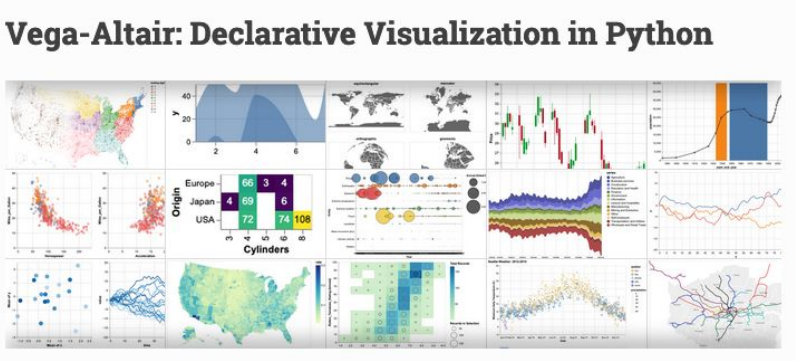
Link

?? This is a very famous paper, but I'm not sure why Mackinlay thinks the bar chart would suggest USA cars are longer ??

For the next huge chunk of today's lecture, we'll dive into ways to properly use other aspects of a visualization:

- x/y
- Color
- Markings
- Conditioning
- Context

Altair: a library designed around this model (with interactivity)

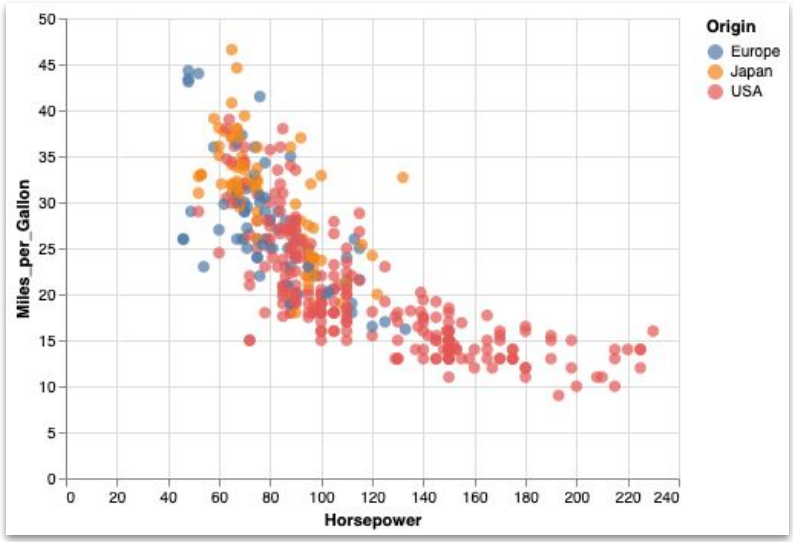


altair-viz.github.io

```
import altair as alt
from vega_datasets import data

source = data.cars()

alt.Chart(source).mark_circle(size=60).encode(
    x='Horsepower',
    y='Miles_per_Gallon',
    color='Origin',
    tooltip=['Name', 'Origin', 'Horsepower', 'Miles_per_Gallon']
).interactive()
```



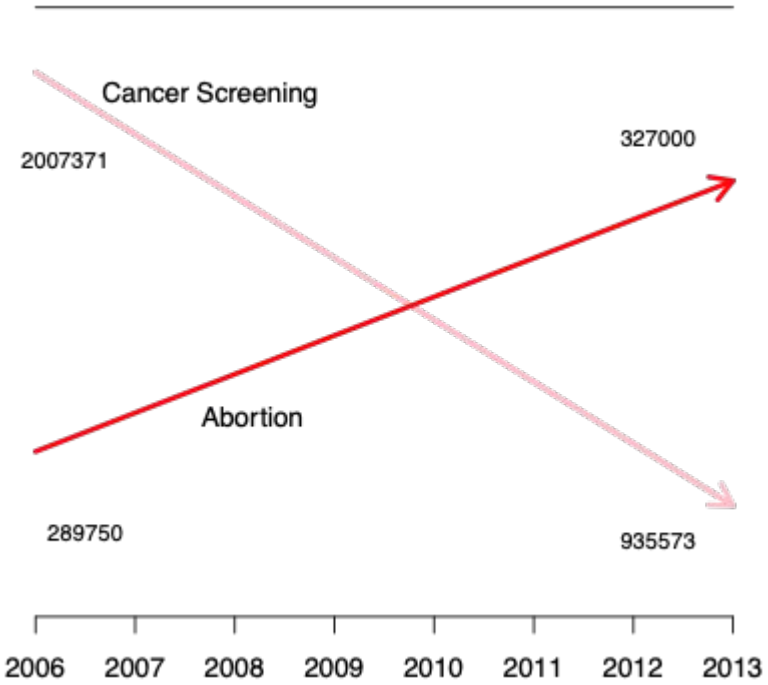
We won't cover it further here. Just FYI.

Harnessing X/Y

Lecture 08, Data 100 Spring 2023

- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - Transformations
- Visualization Theory
  - Information Channels
  - **Harnessing X/Y**
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context

Case Study: Planned Parenthood Hearing

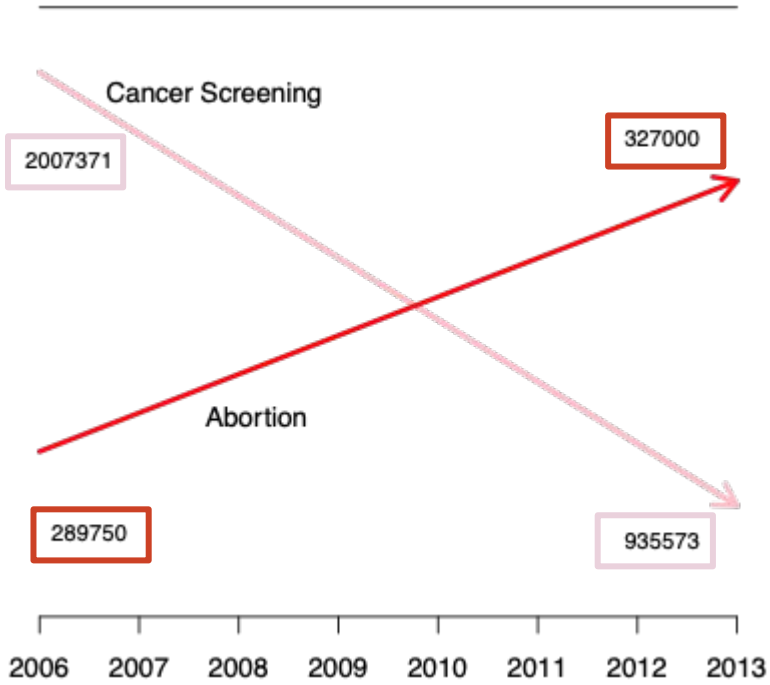


In 2015, Planned Parenthood was accused of selling aborted fetal tissue for profit.

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!

In 2013:

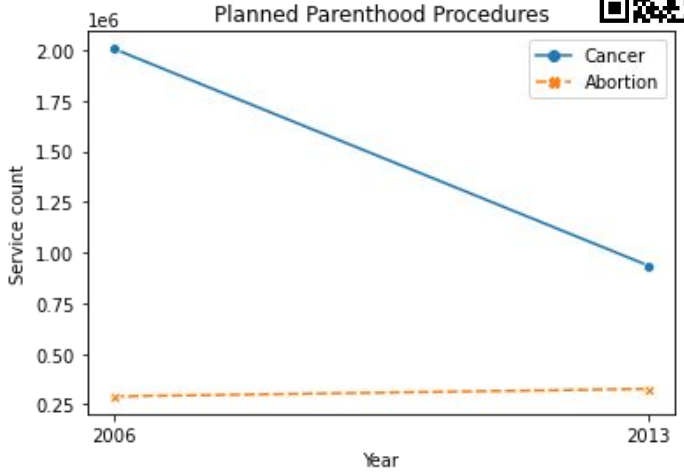
- 327000 is smaller than 935573…
- …but appears to be way bigger??

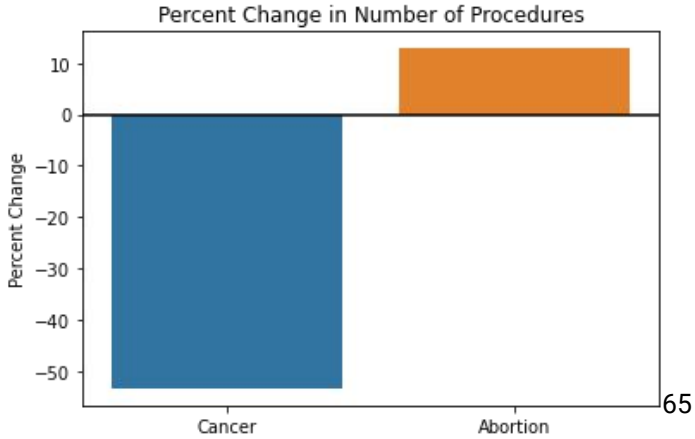
Do not use two different scales for the same axis!

Always consider the scale when comparing “similar” data.

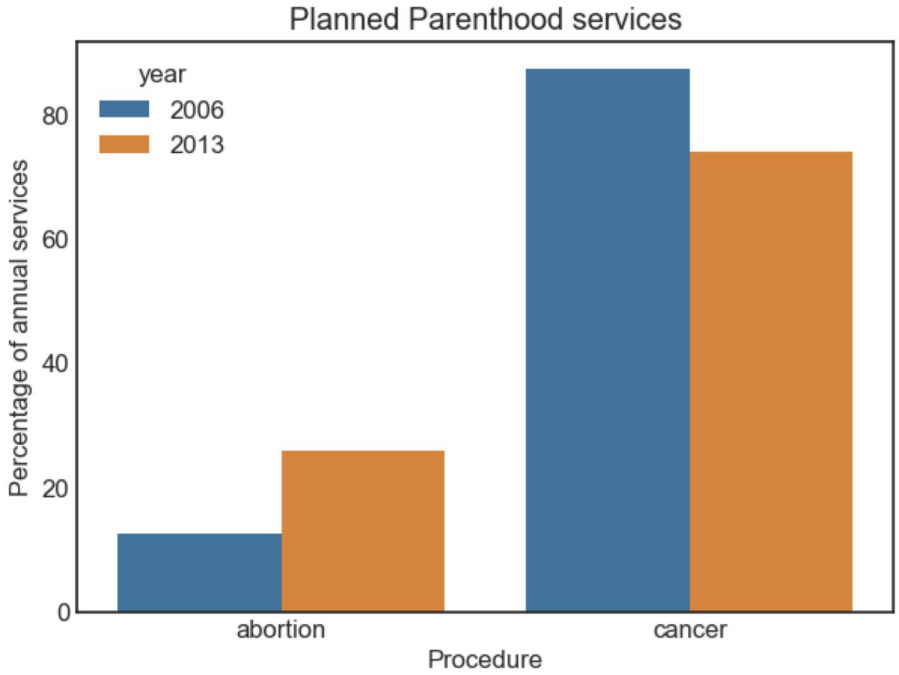
The top plot draws all of the data on the same scale.

- 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.





Always consider the scale when comparing “similar” data.



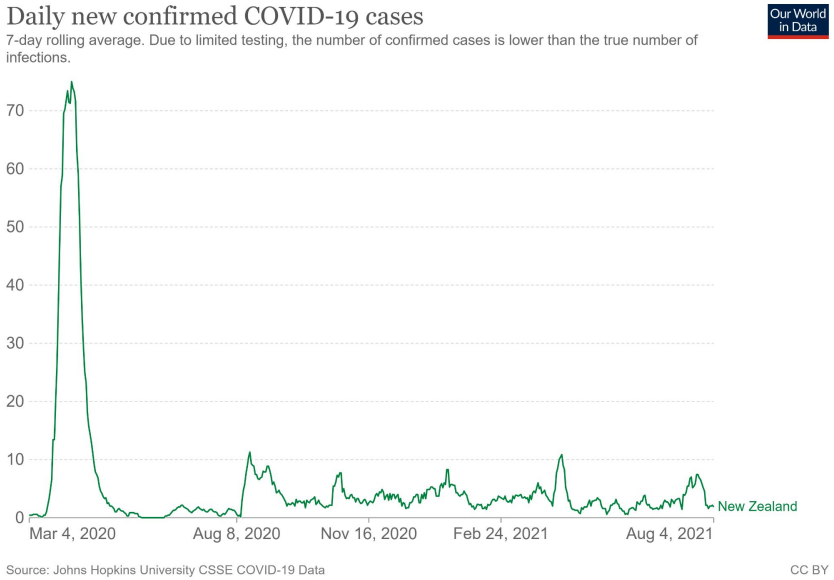
We could also visualize abortions and cancer screenings as a percentage of total procedures.

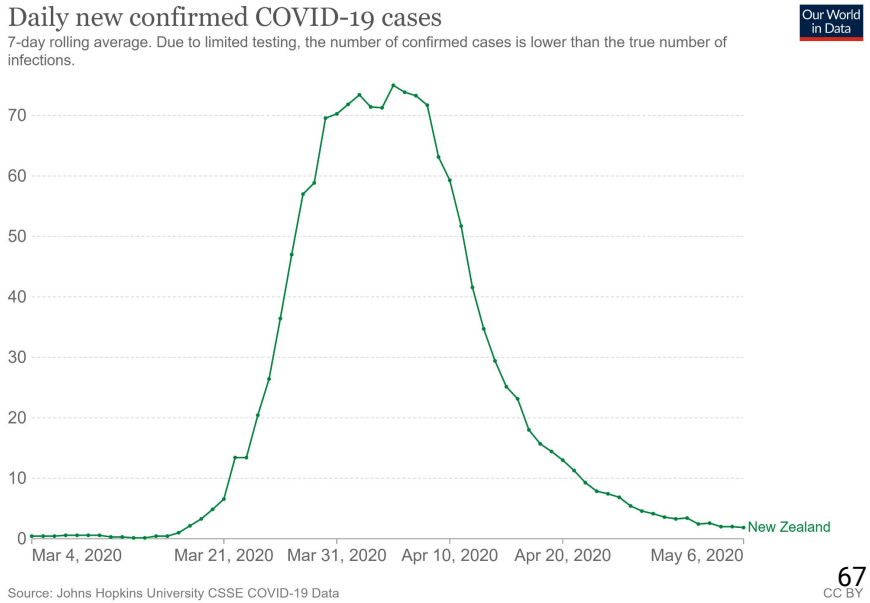
- Abortions increased from 13% to 26% of total procedures.

Reveal the Data

Recommendations:

- Choose axis limits to fill the visualization.
- **You don’t have to visualize all of the data at once:**
  - Zoom in on the bulk of the data (it’s ok to not include 0!) if only one part matters.
  - Can also create multiple plots to show different regions of interest.





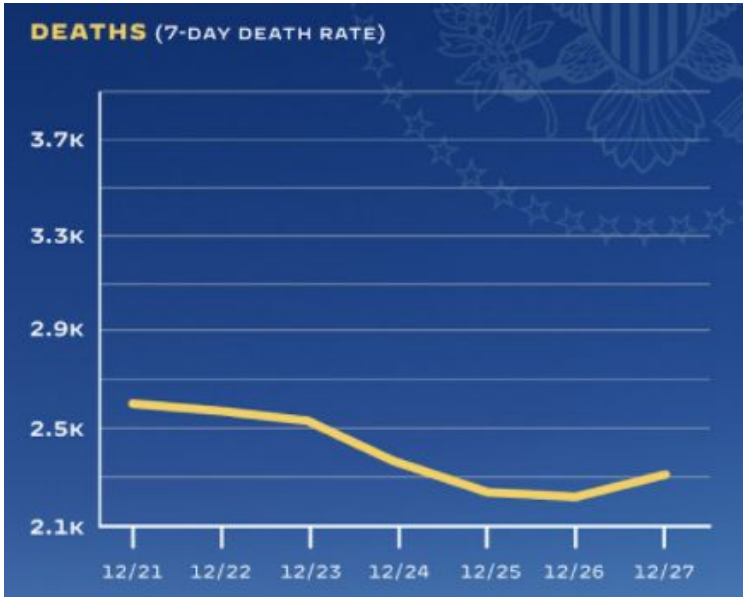
Reveal the Data

Recommendations:

- **Choose axis limits to fill the visualization.**
- You don't have to visualize all of the data at once:
  - Zoom in on the bulk of the data (it's ok to not include 0!) if only one part matters.
  - Can also create multiple plots to show different regions of interest.

Terrible White House COVID-19 visualization:

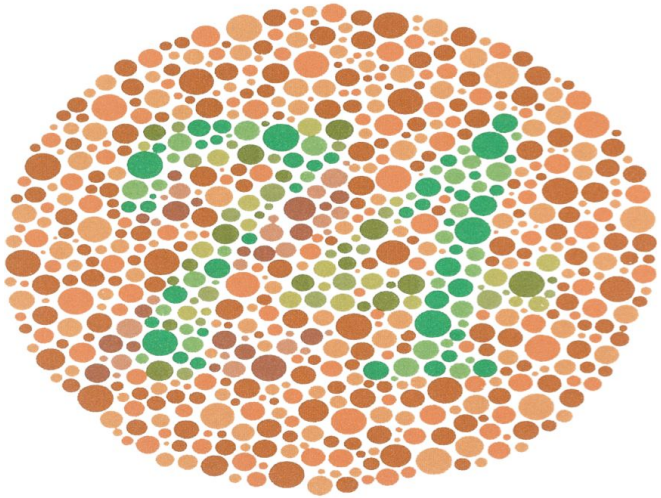
- Mysterious maximum value on y-axis.



Harnessing Color

Lecture 08, Data 100 Spring 2023

- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - Transformations
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - **Harnessing Color**
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context

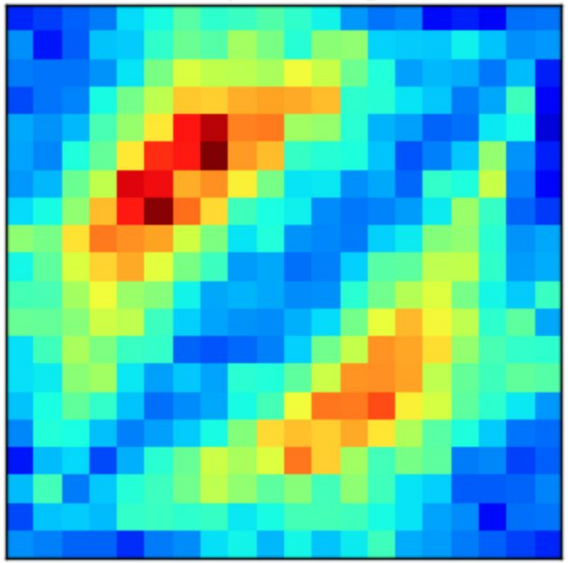


Choosing a set of colors which work together is a challenging task!

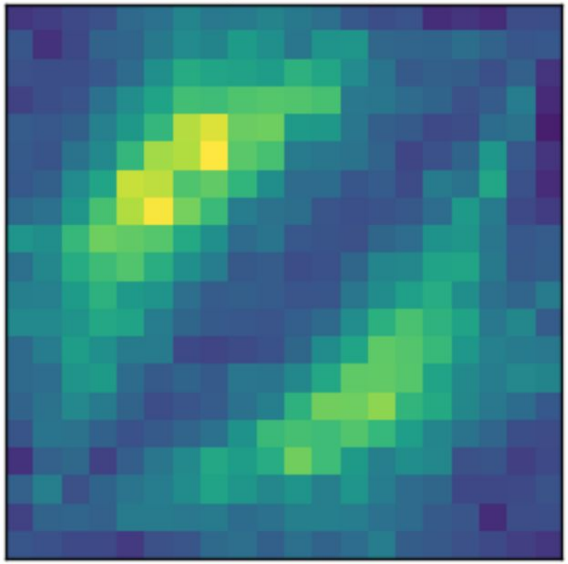
Perception of Color

Download the Color Oracle App to simulate common color vision impairments.

Colormaps

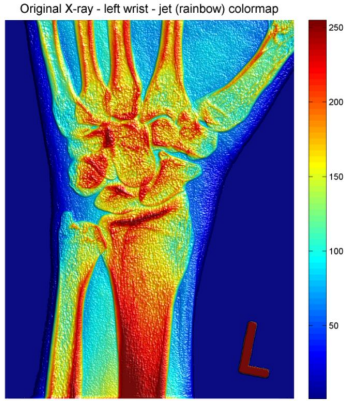


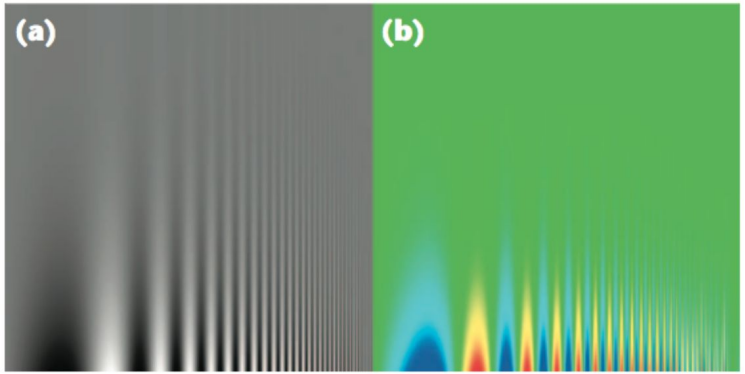
Jet

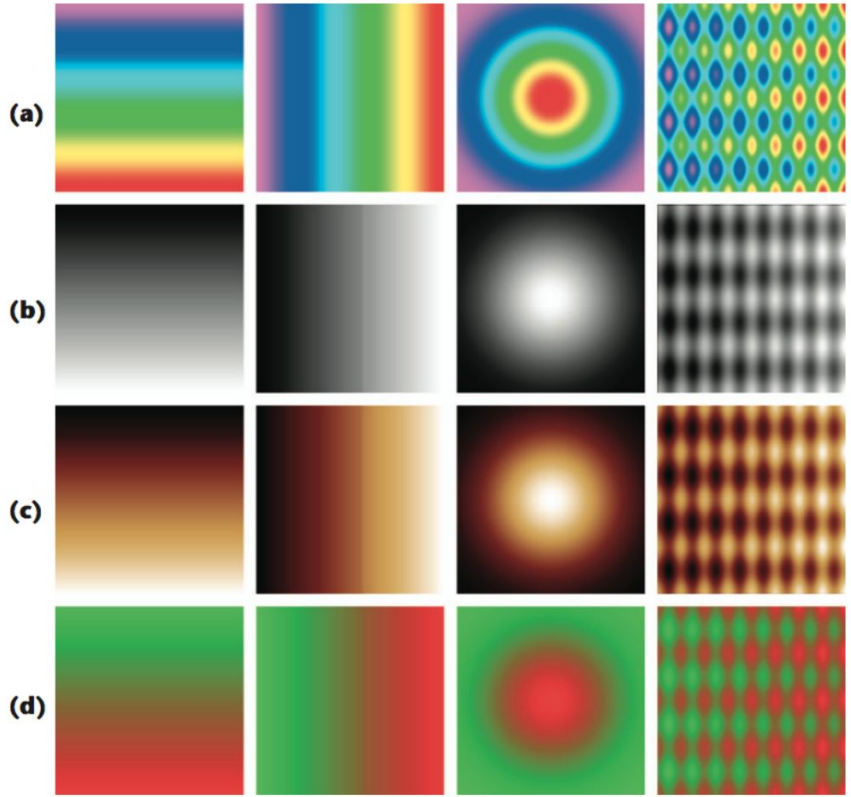


Viridis

The jet/rainbow colormap actively misleads



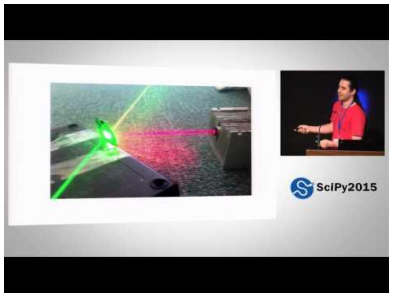


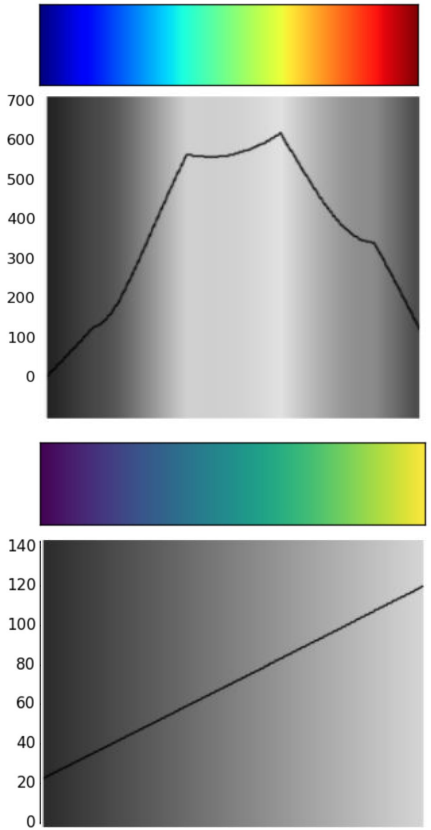


"Rainbow Colormap (Still) Considered Harmful", Borland and Taylor, 2007.

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.





Bounces all over

Slope is constant

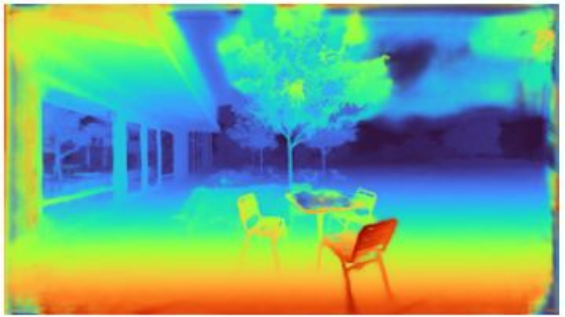
x-axis is color,
y-axis is "lightness"

Except when not :) The Google Turbo Colormap

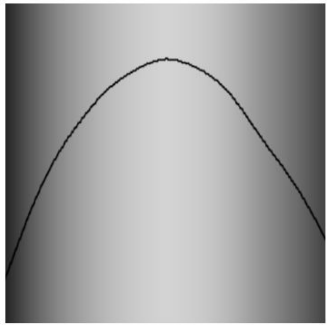
Turbo

Jet

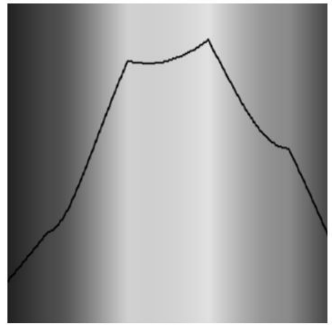
Inferno



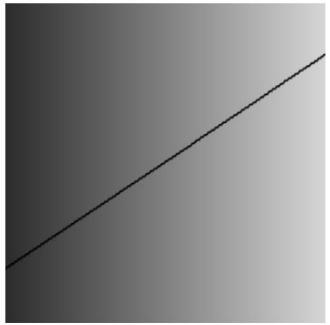
Turbo



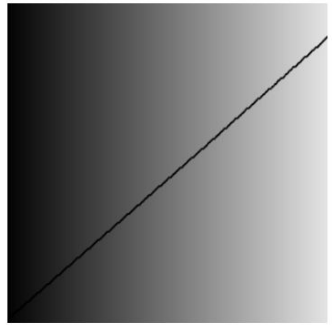
Turbo



Jet



Viridis



Inferno

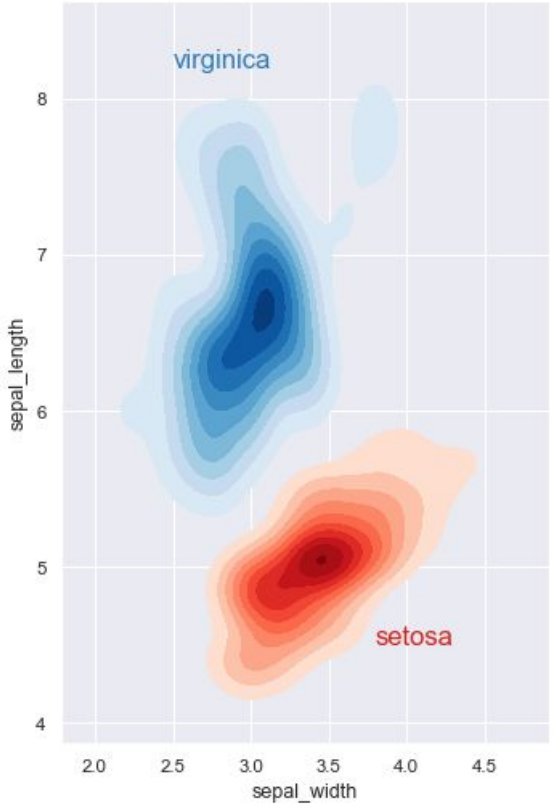
X-axis is color, y-axis is "lightness"

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

Use color to highlight data type

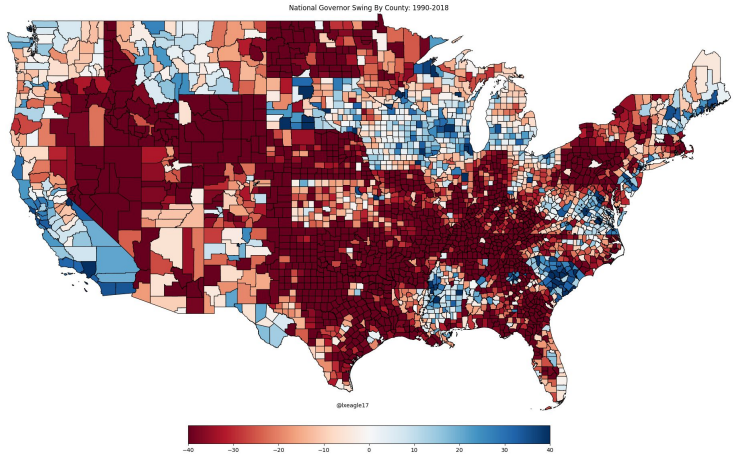
- **Qualitative**: Choose a qualitative scheme that makes it easy to distinguish between categories.
  - One category isn't "higher" or "lower" than another.
- **Quantitative**: Choose a color scheme that visualizes magnitude of change.

The plot on the right has both distinctions!



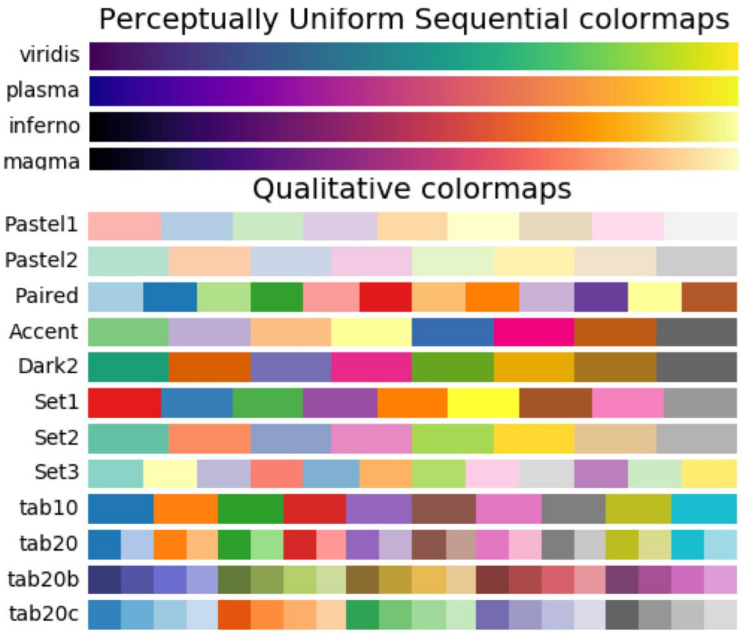
Sequential vs. diverging colormaps for quantitative data

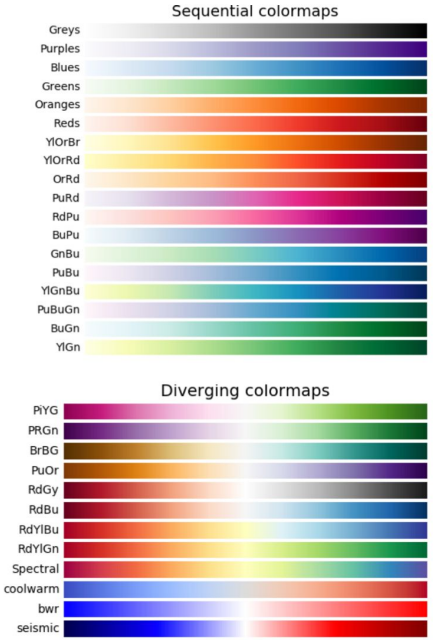
If the data progresses from low to high, use a **sequential** scheme where lighter colors are for more extreme values.



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

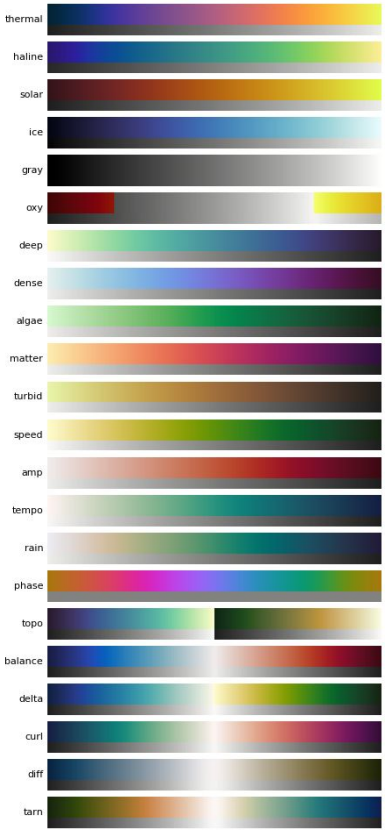
Default matplotlib colormaps

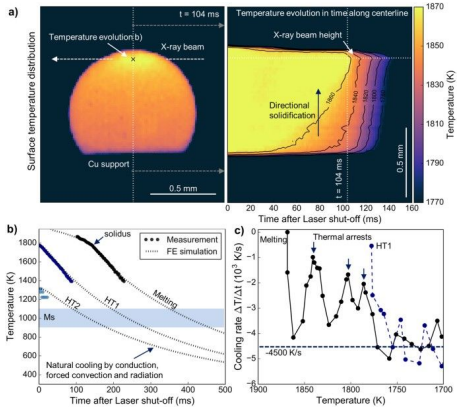




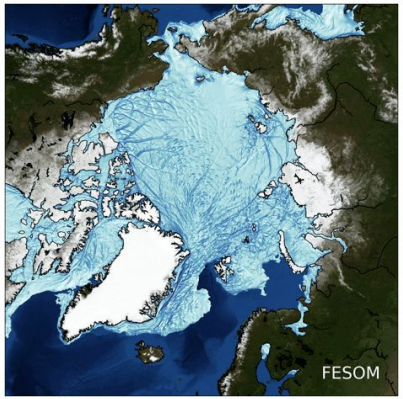
Taken from matplotlib documentation.

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

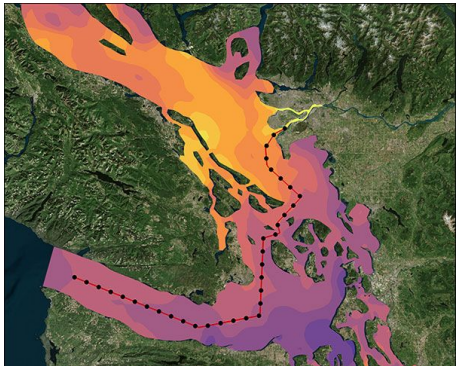


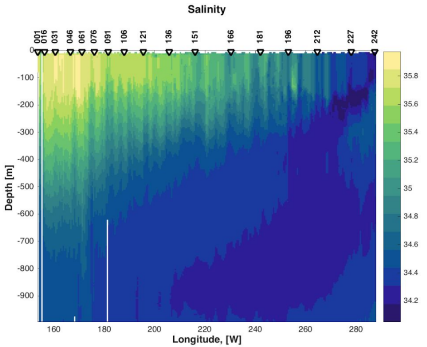


Thermal



Ice





Haline

Extra reading

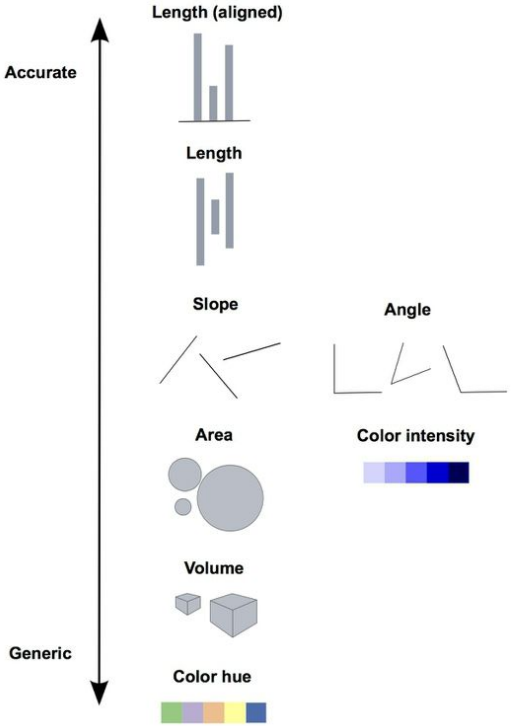
You may want to refer to these articles, which also discuss colormaps.

- 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

Harnessing Markings

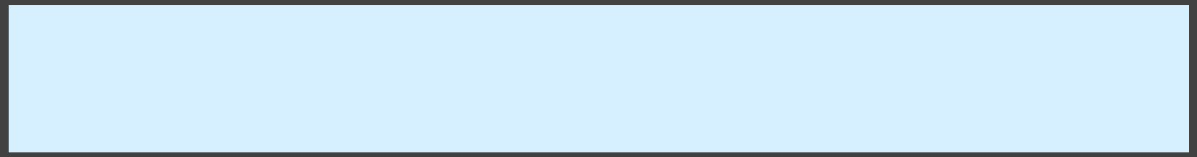
Lecture 08, Data 100 Spring 2023

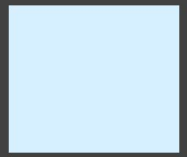
- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - Transformations
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - **Harnessing Markings**
  - Harnessing Conditioning
  - Harnessing Context



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

Perception of Markings



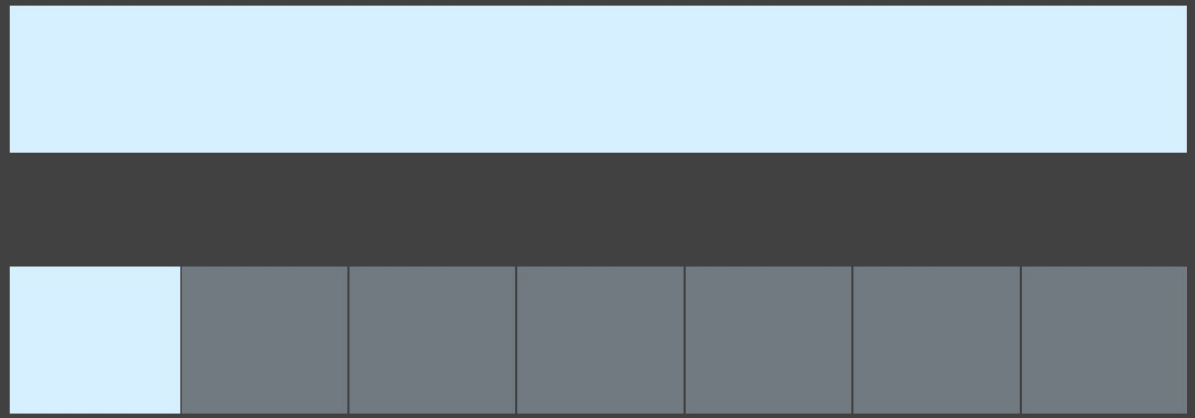


How much longer is the long bar?

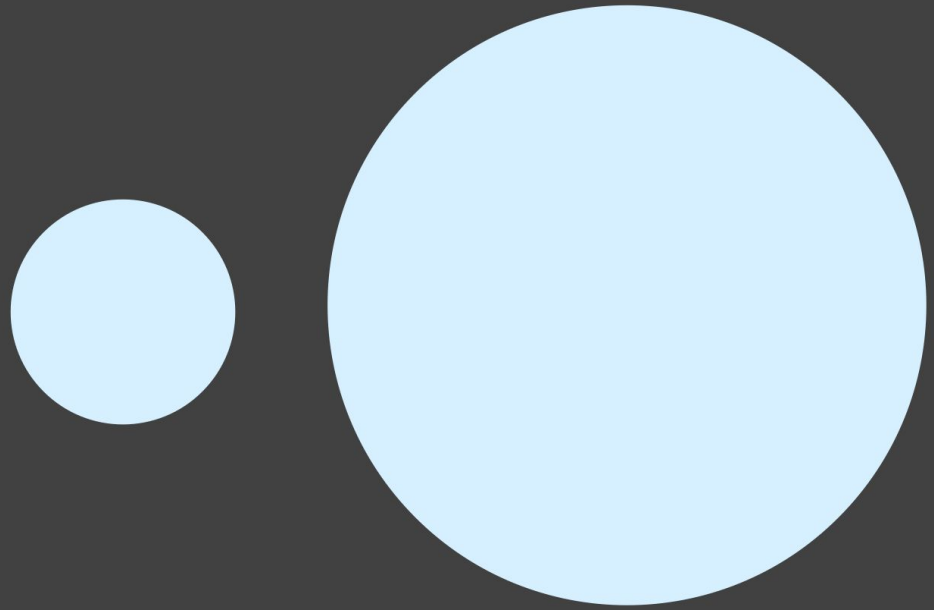
slido

How much longer is the long bar?

ⓘ Start presenting to display the poll results on this slide.



The long bar is 7 times longer than the short bar.

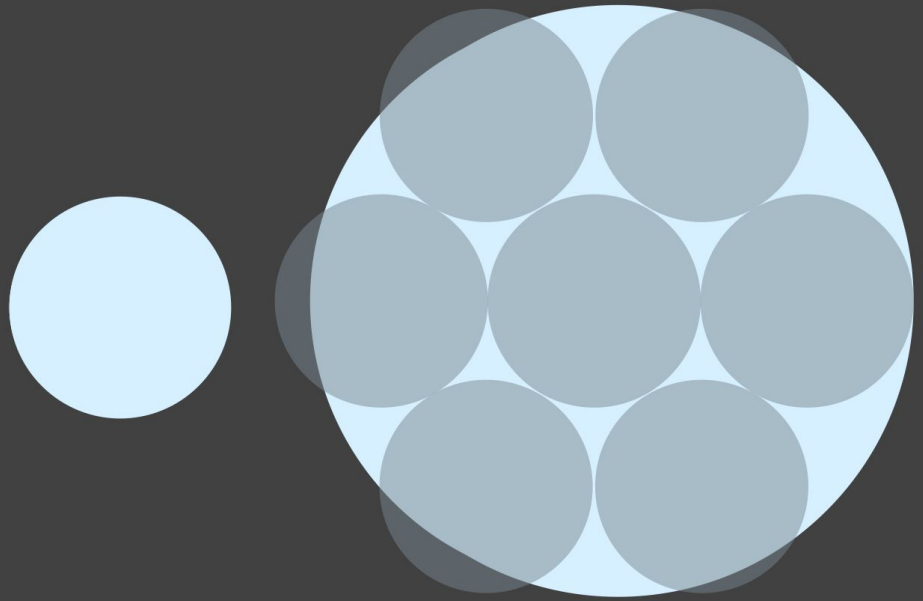


How much bigger is the big circle?

slido

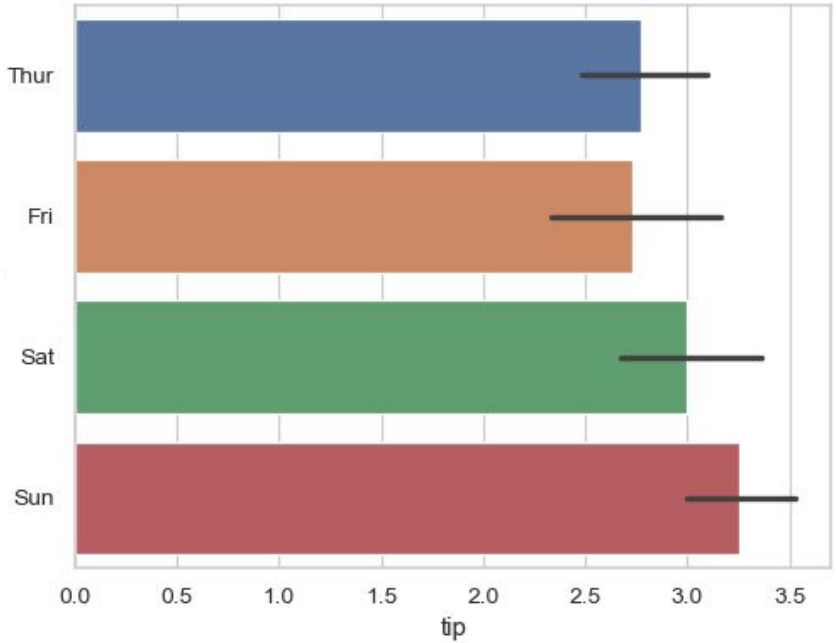
How much bigger is the big circle?

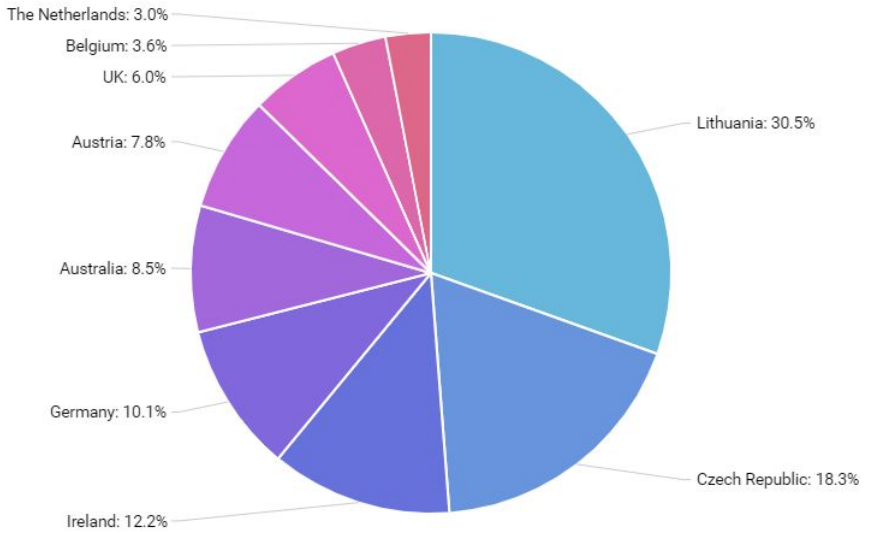
ⓘ Start presenting to display the poll results on this slide.



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

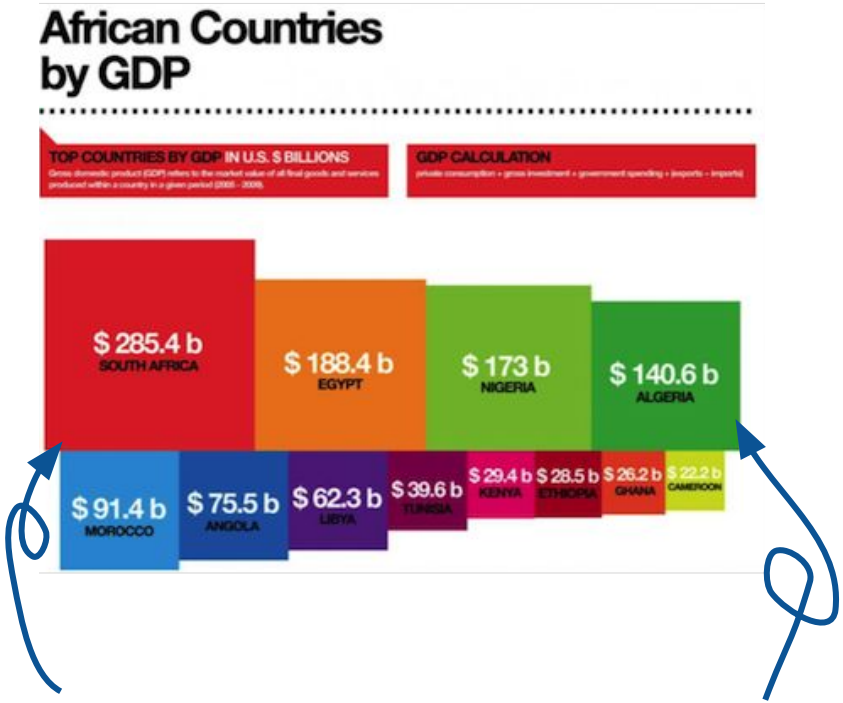
Lengths are easy to distinguish. Others, like angles, are hard.



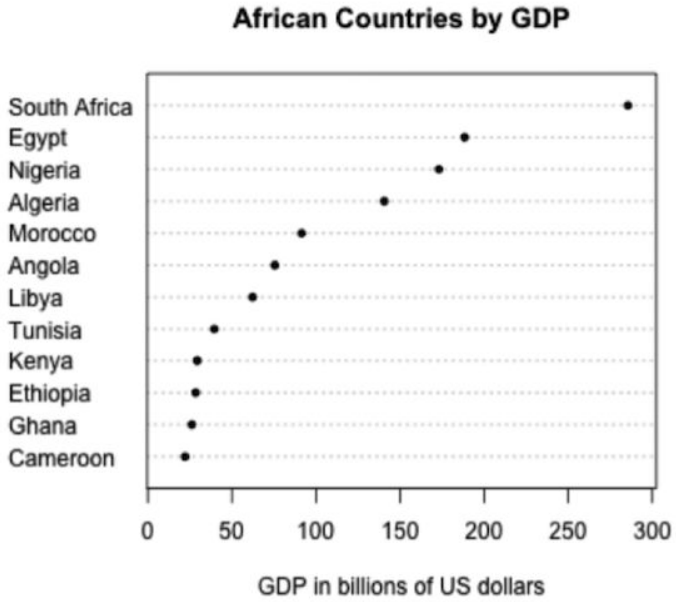


Don't use pie charts! Visual angle judgments are inaccurate.

Areas are hard to distinguish

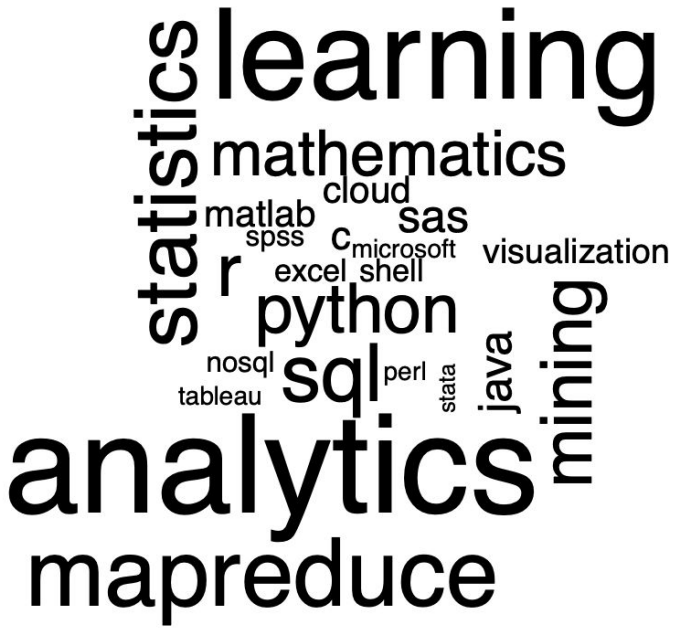


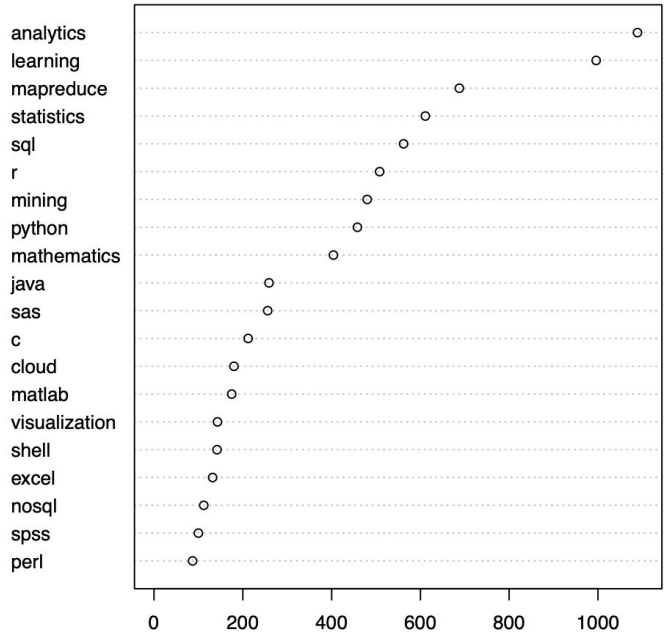
(South Africa has twice the GDP of Algeria, but that isn't clear from the areas.)



Avoid area charts!
Visual area judgments are inaccurate.

Areas are hard to distinguish





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

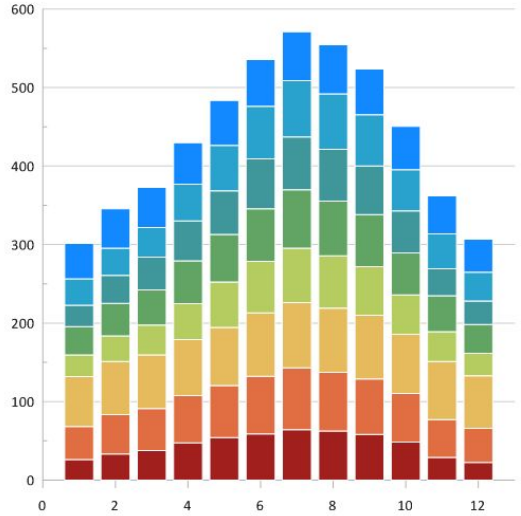
…that being said, if you are not trying to make quantifiable comparisons, then word clouds are useful for "the idea."

Avoid “jiggling” the baseline!

Stacked bar charts, histograms, and area charts are hard to read because the baseline moves (“jiggles”).

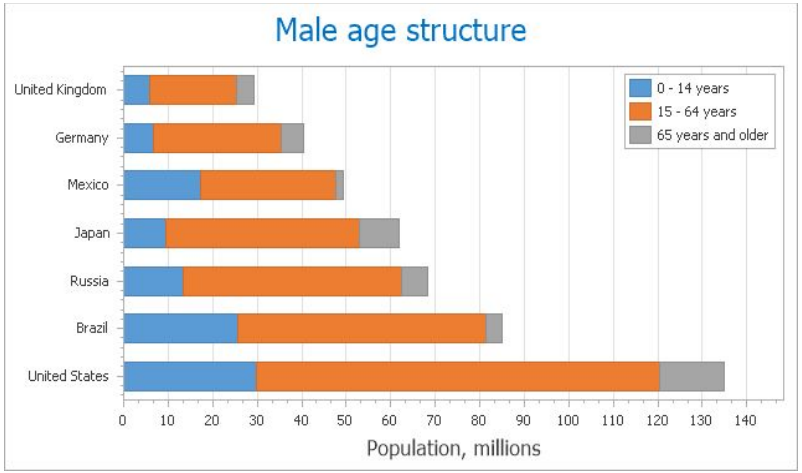
In the first plot:

- The top blue bars are all roughly of the same length.
- Not immediately obvious!



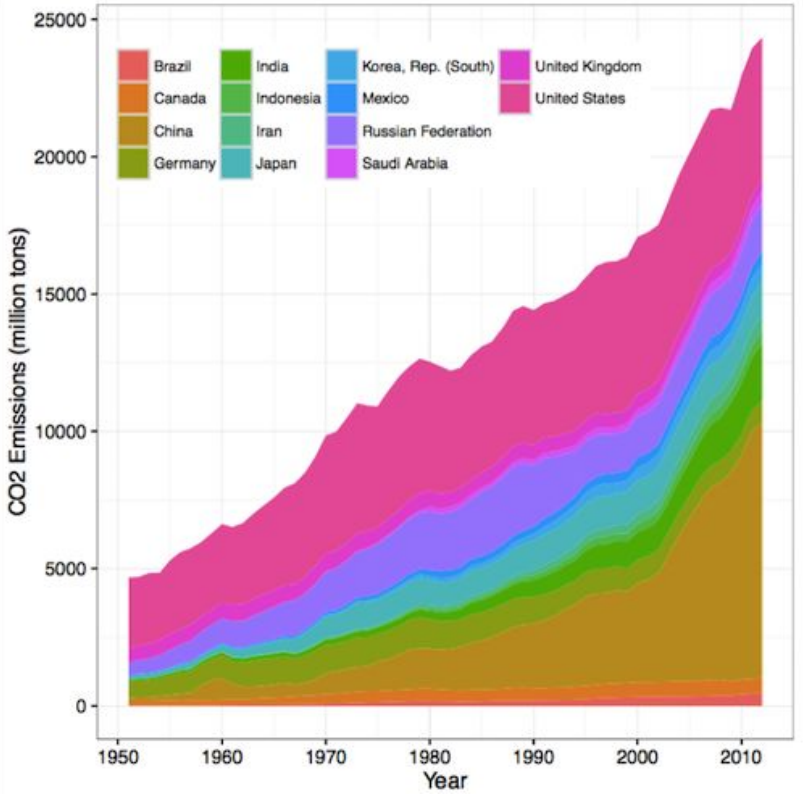
In the second plot:

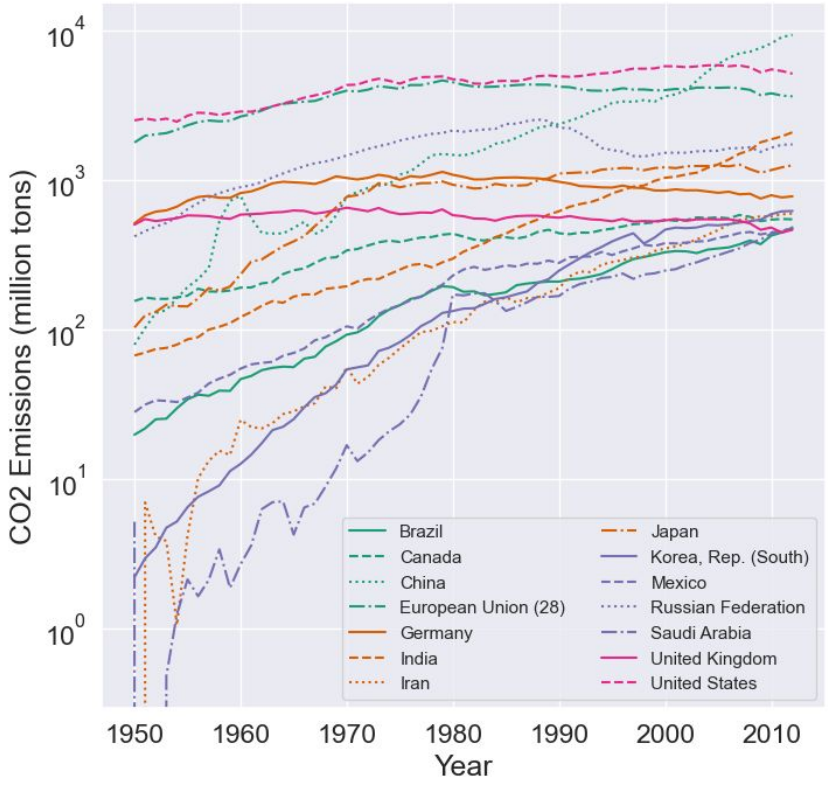
- Comparing the number of 15-64 year old males in Germany and Mexico is difficult.



Avoid jiggling the baseline

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



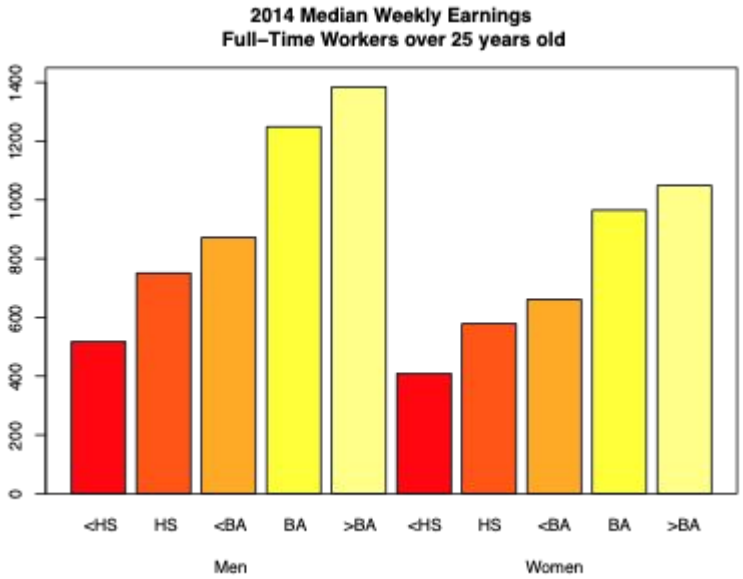


Harnessing Conditioning

Lecture 08, Data 100 Spring 2023

- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - Transformations
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - **Harnessing Conditioning**
  - Harnessing Context

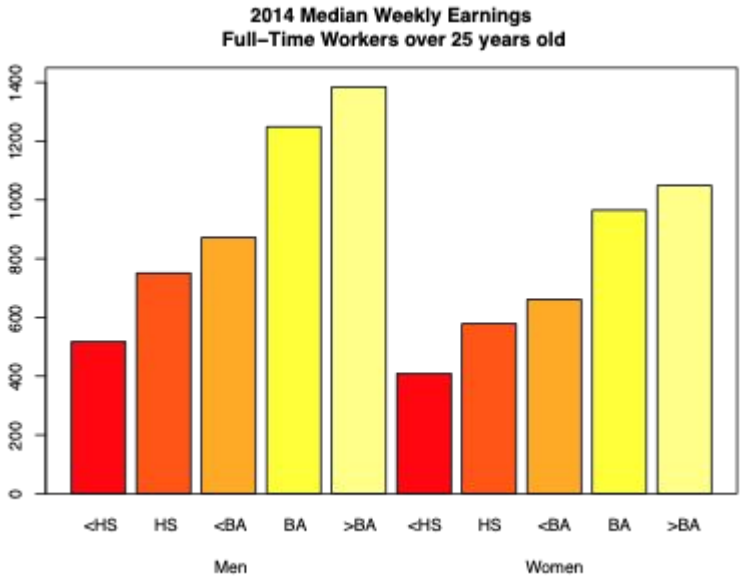
Use conditioning to aid comparison



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



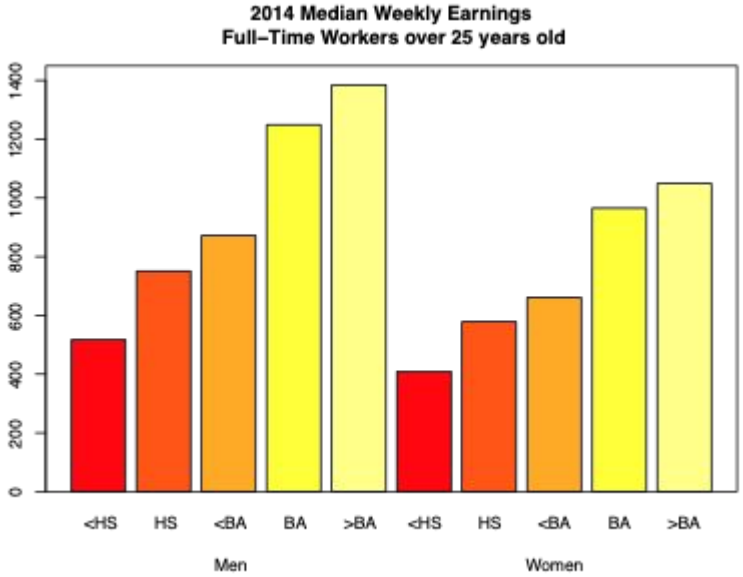
This data comes from the Bureau of Labor Statistics, who oversees surveys regarding the economic health of the US. They have plotted median weekly earnings for men and women by education level.

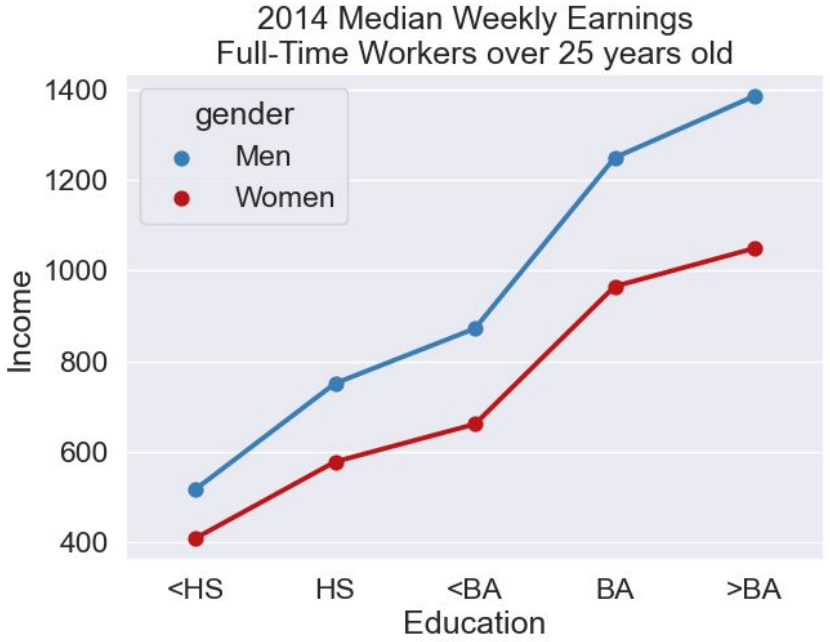
- What comparisons are made easily with this plot?
- What comparisons are most interesting and important?

- Easy to see the effect of education on earnings.
- Hard to compare between the two genders in the dataset.

How could we more easily make this difficult comparison?

Use conditioning to aid comparison

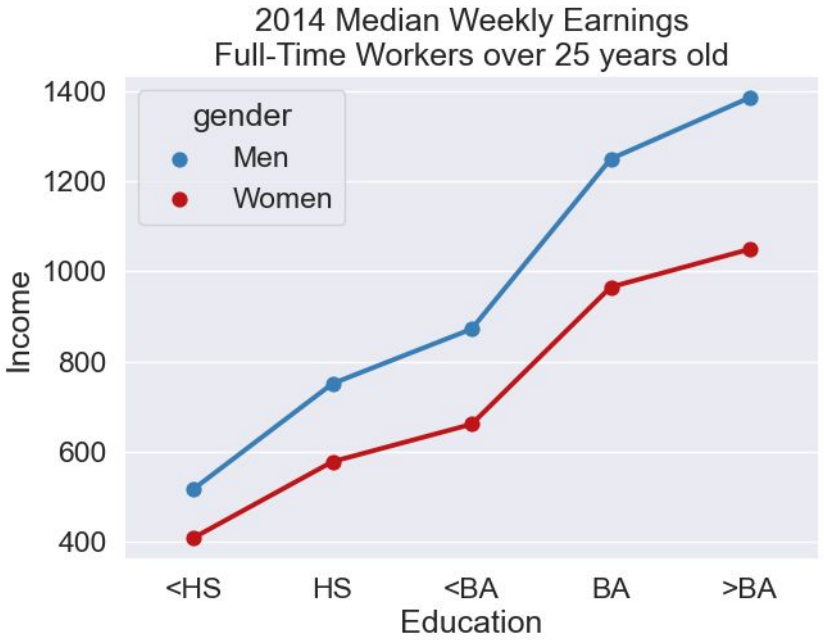


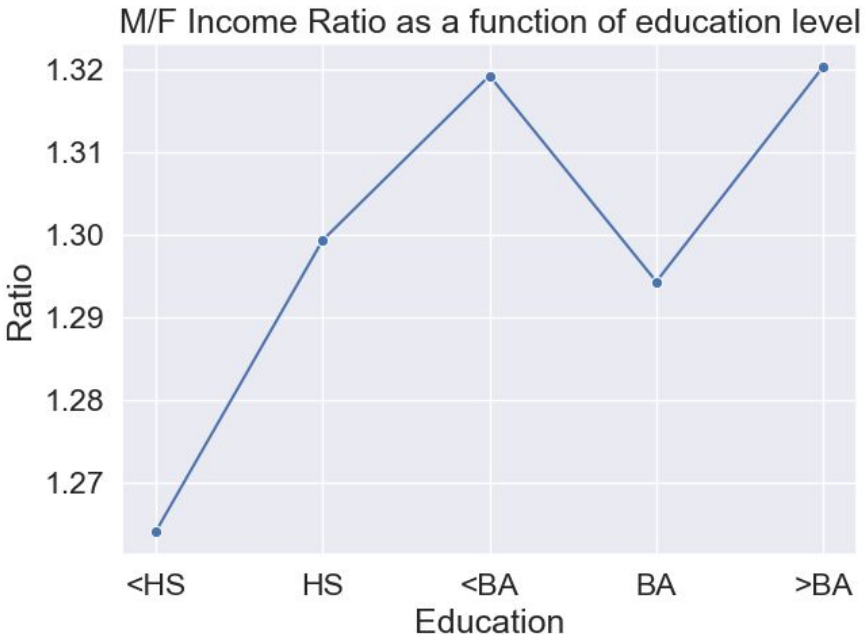


- Easy to see the effect of education on earnings.
- Hard to compare between the two genders in the dataset.

Having two separate lines makes clear the wage difference between men and women.

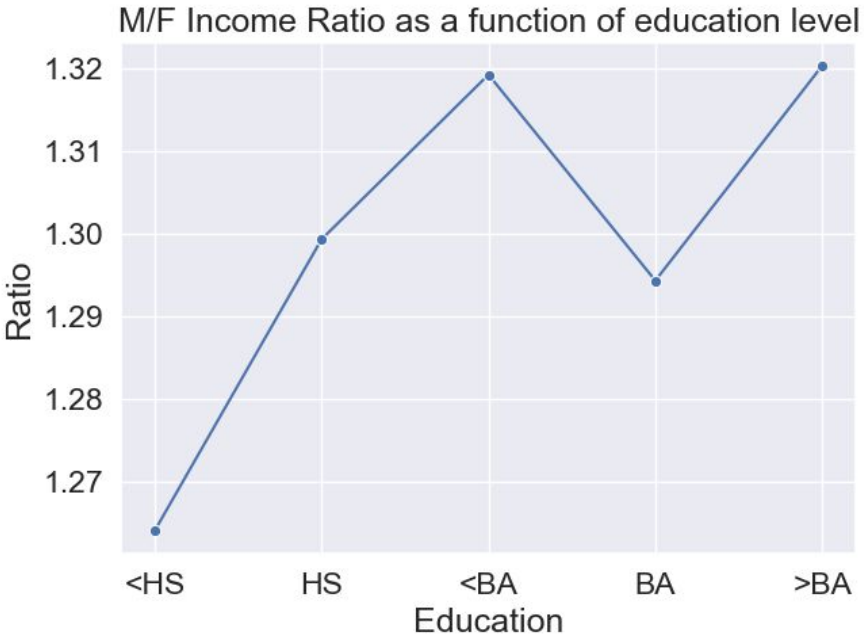
How does the income gap increase with education?

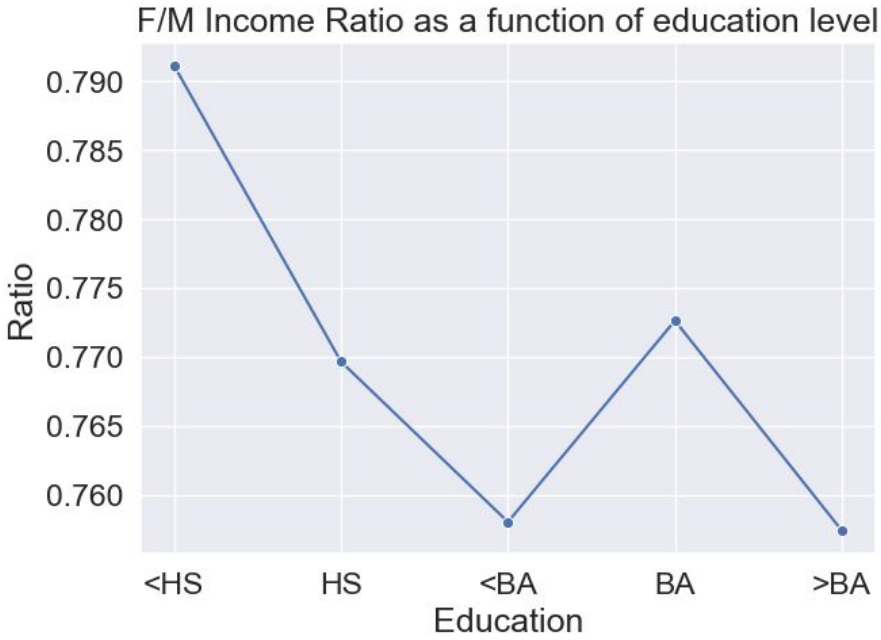




See notebook for how to get this figure with groupby!

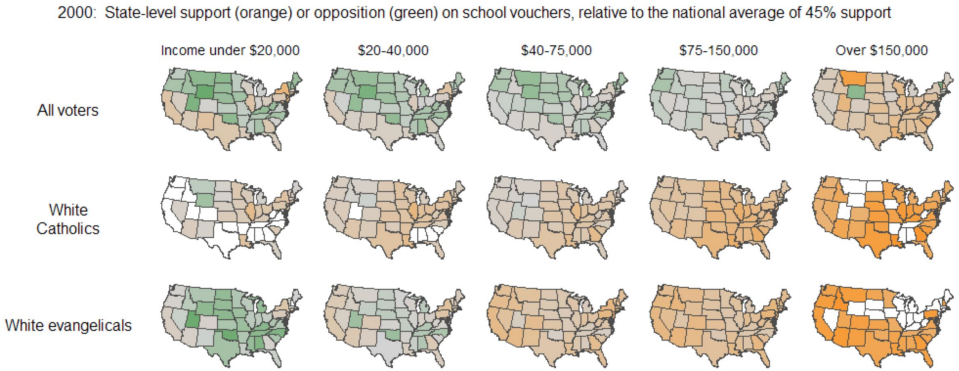
But… which ratio should we pick? M/F or F/M?





The visual metric you choose tells your story.

Other notes: Superposition vs. Juxtaposition



An example of **small multiples**.

Superposition: placing multiple density curves, scatter plots on top of each other (what we've usually been doing)

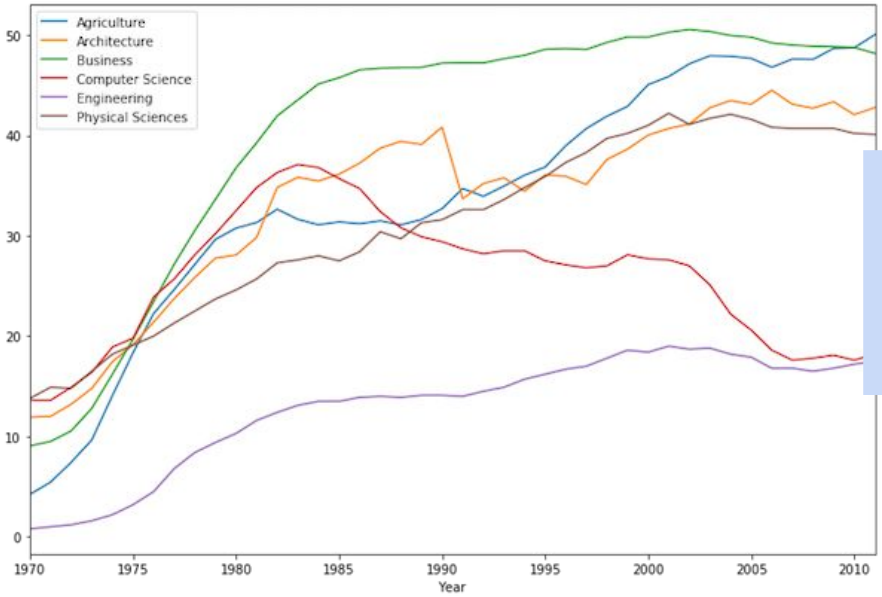
Juxtaposition: placing multiple plots side by side, with the same scale (called "small multiples") (see left).

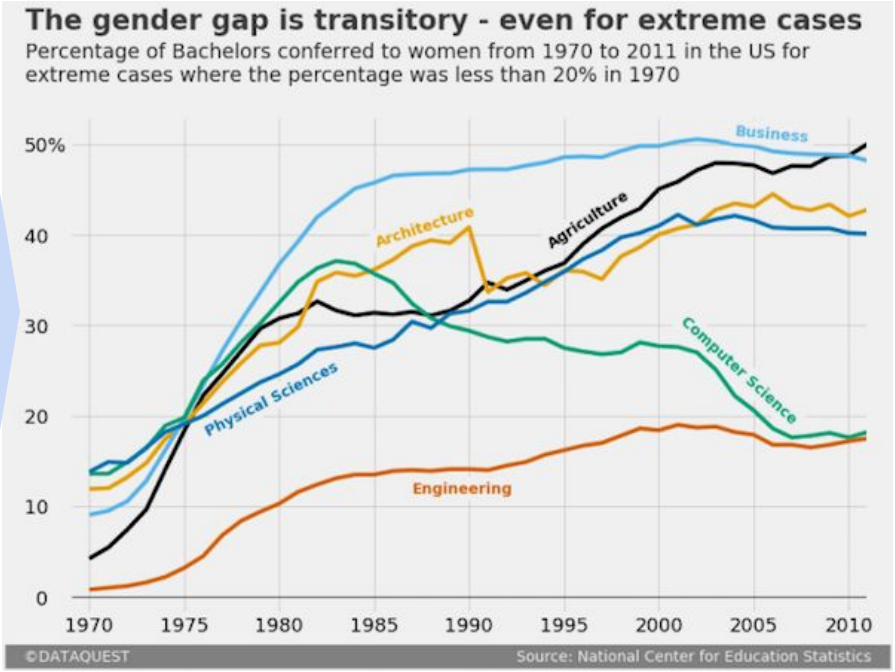
Harnessing Context (for Publication)

Lecture 08, Data 100 Spring 2023

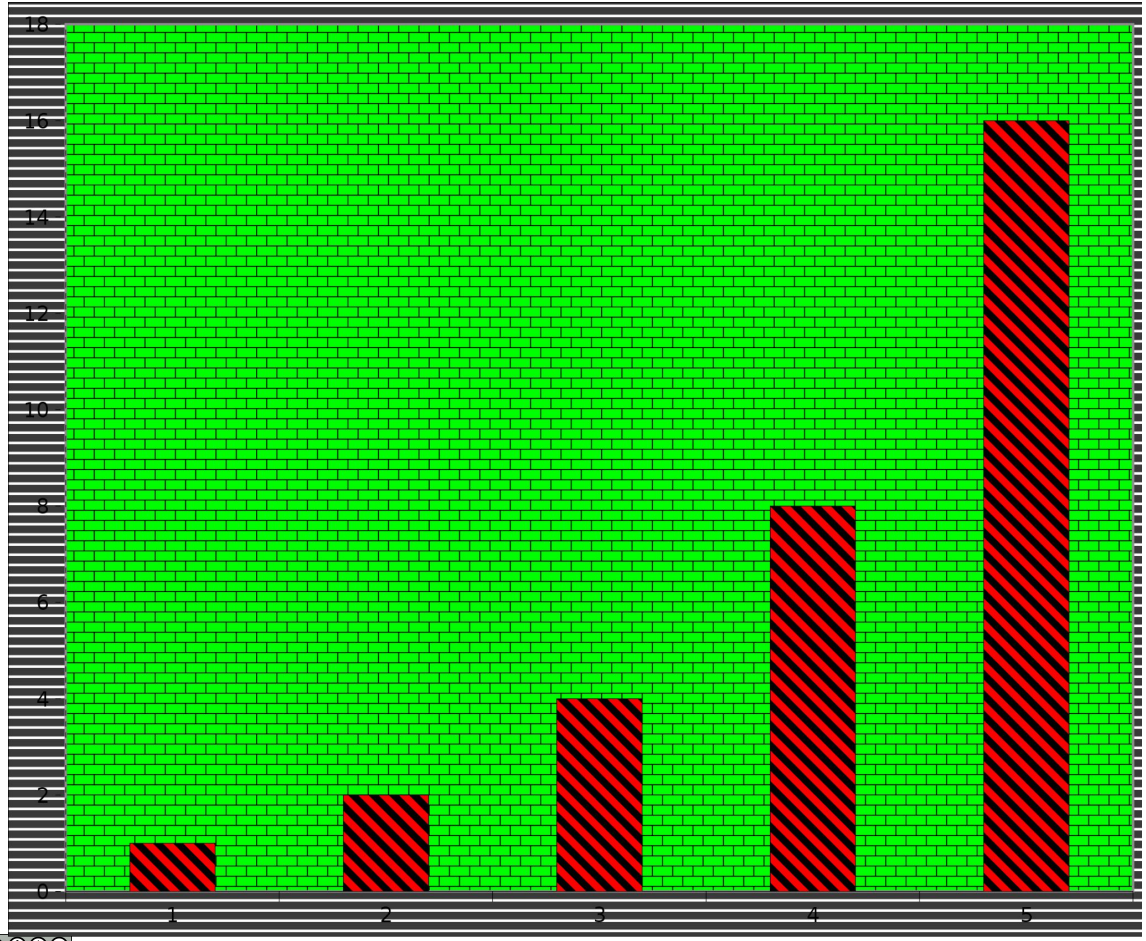
- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Relationships between Quantitative Variables
  - Transformations
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - **Harnessing Context**

Getting Ready for Publication





Going overboard: Chartjunk



From chartjunk Wikipedia:

> An example of a chart containing gratuitous chartjunk. This chart uses a large area and much "ink" (many symbols and lines) to show only five hard-to-read numbers, 1, 2, 4, 8, and 16.

Publication-ready: 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 **axis labels**.

Publication-ready: Captions

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 **axis labels**.

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:

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

A captioned, publication-ready (famous) figure

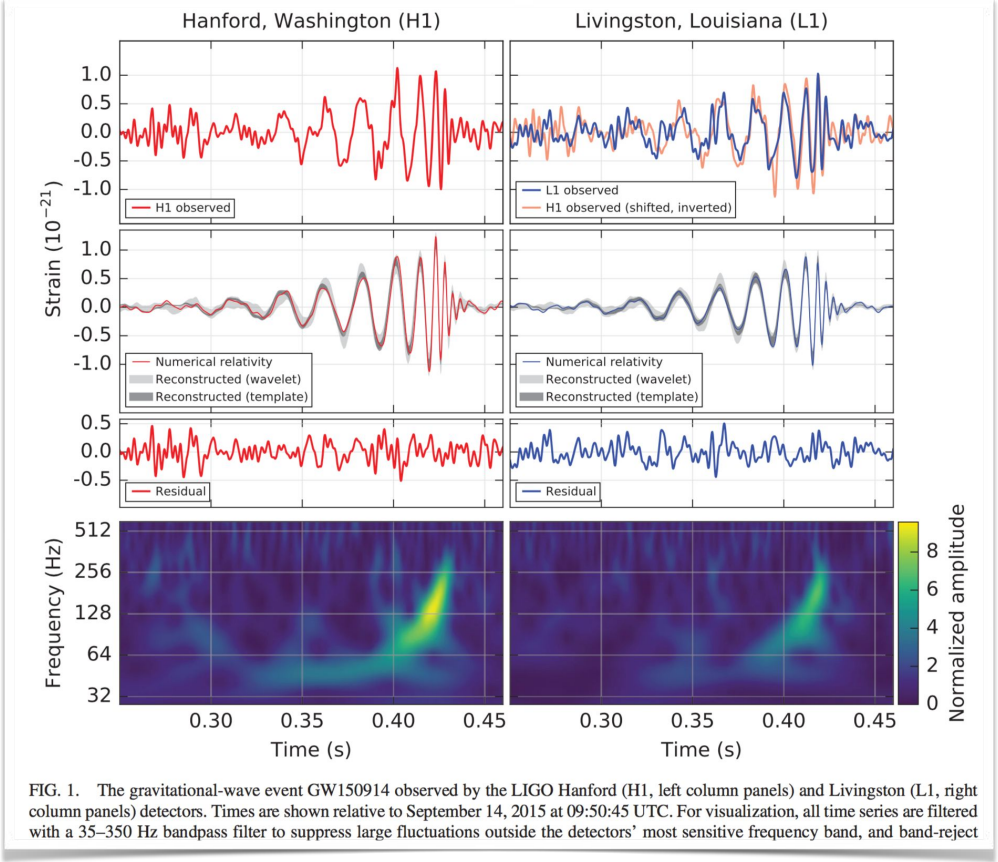


FIG. 1. The gravitational-wave event GW150914 observed by the LIGO Hanford (H1, left column panels) and Livingston (L1, right column panels) detectors. Times are shown relative to September 14, 2015 at 09:50:45 UTC. For visualization, all time series are filtered with a 35–350 Hz bandpass filter to suppress large fluctuations outside the detectors' most sensitive frequency band, and band-reject

Figure 1

The gravitational-wave event GW150914 observed by the LIGO Hanford (H1, left column panels) and Livingston (L1, right column panels) detectors. Times are shown relative to September 14, 2015 at 09:50:45 UTC. For visualization, all time series are filtered with a 35–350 Hz bandpass filter to suppress large fluctuations outside the detectors' most sensitive frequency band, and band-reject filters to remove the strong instrumental spectral lines seen in the Fig. 3 spectra. *Top row, left:* H1 strain. *Top row, right:* L1 strain. GW150914 arrived first at L1 and $6.9^{+0.5}_{-0.4}$ ms later at H1; for a visual comparison, the H1 data are also shown, shifted in time by this amount and inverted (to account for the detectors' relative orientations). *Second row:* Gravitational-wave strain projected

"Observation of Gravitational Waves from a Binary Black Hole Merger" - 2017 Nobel Prize in Physics.

https://www.gw-openscience.org/tutorials

Summary

Some key ideas from today:

- KDEs are not magic! They're just copies of a Gaussian curve added together.
- Choose appropriate scales.
- Choose colors and markings that are easy to interpret correctly.
- Condition in order to make comparisons more natural.
- Add context and captions that help tell the story.
- 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 – "**chartjunk**."
  - It's fine to plot the same thing multiple ways, if it helps fit the narrative better.