LECTURE 11

Visualization, Part II

KDEs, Visualization Theory, and Transformations

KDE Mechanics

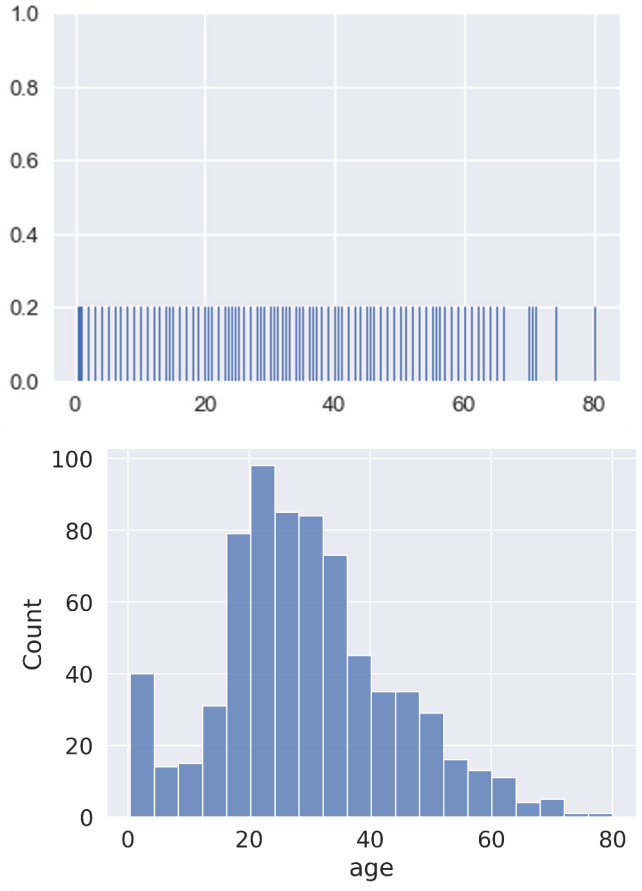
- **Kernel Density Functions**
  - **KDE Mechanics**
  - Kernel Functions and Bandwidth
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context
- Transformations

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.

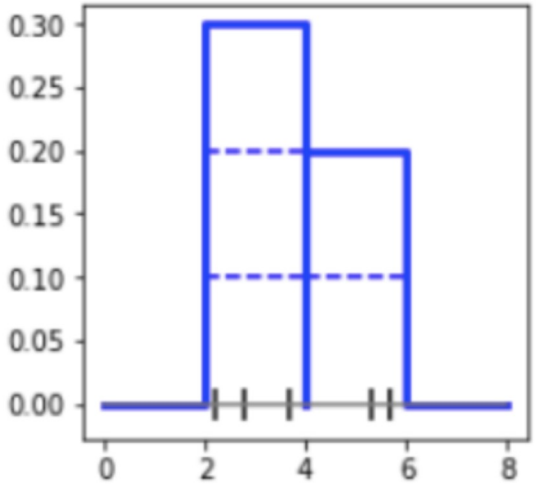


Spreading Proportion Uniformly

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 ⅕ proportion of the sample.
- 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.

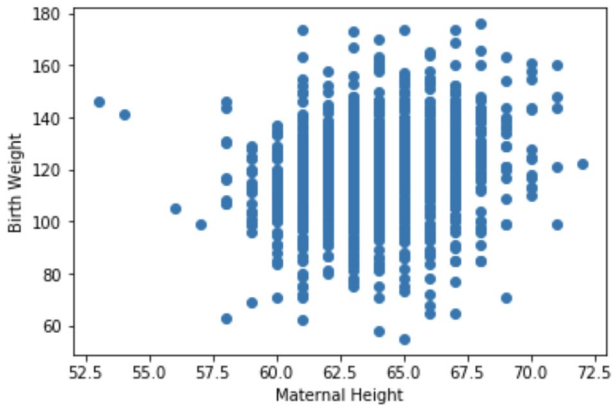


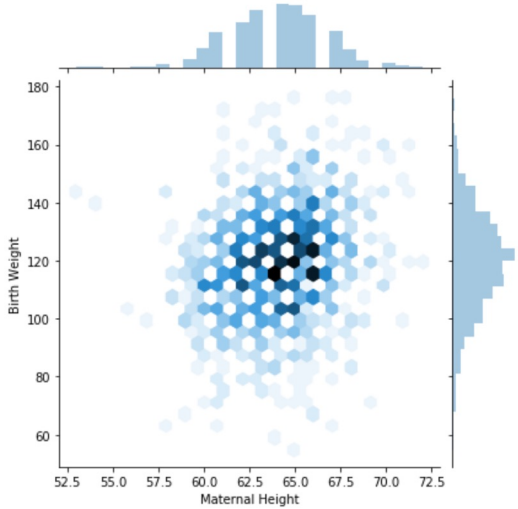
In each bin, add a rectangle with area 1/5 for each point in that bin.

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".

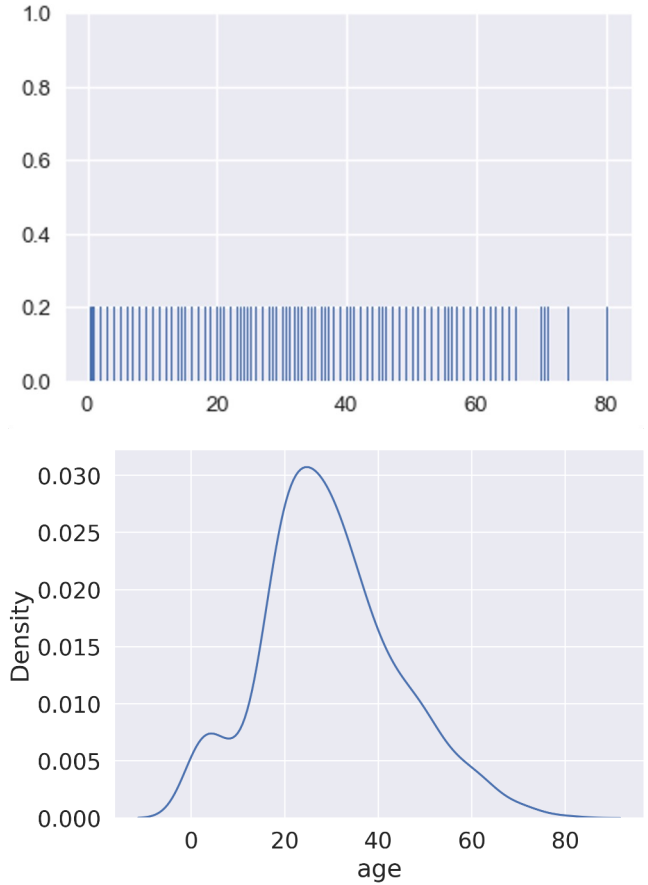




Smoothing in 1D (KDEs)

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 object.



Note: The x-axes for our KDE are wider than the rug plot axes (e.g. goes below zero).

Kernel density estimation (KDE)

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

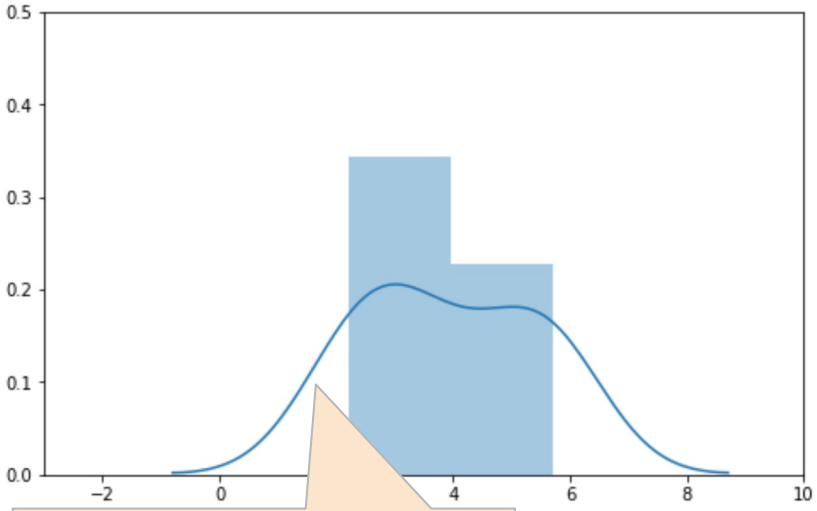
- Just like a histogram, a **density curve**'s total area must sum to 1.

To create a KDE:

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

To generate a curve we need to choose a **kernel** and **bandwidth.**

We will formally define "**probability density function**" later in the class.

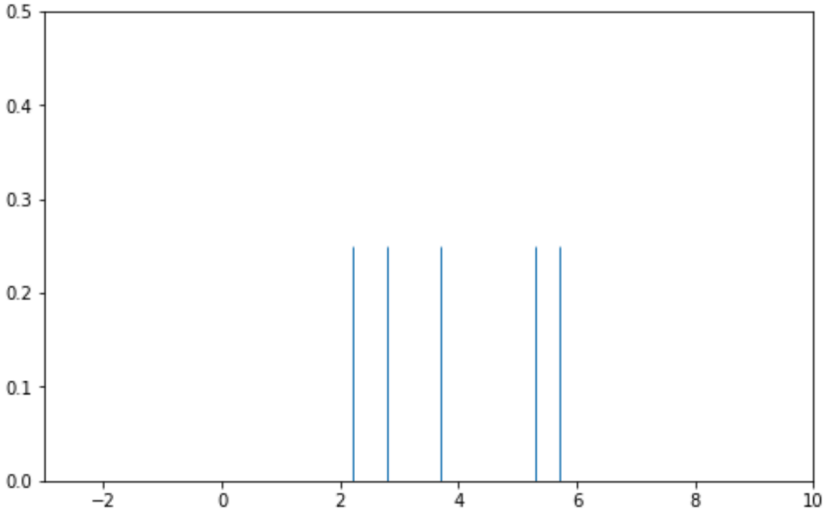


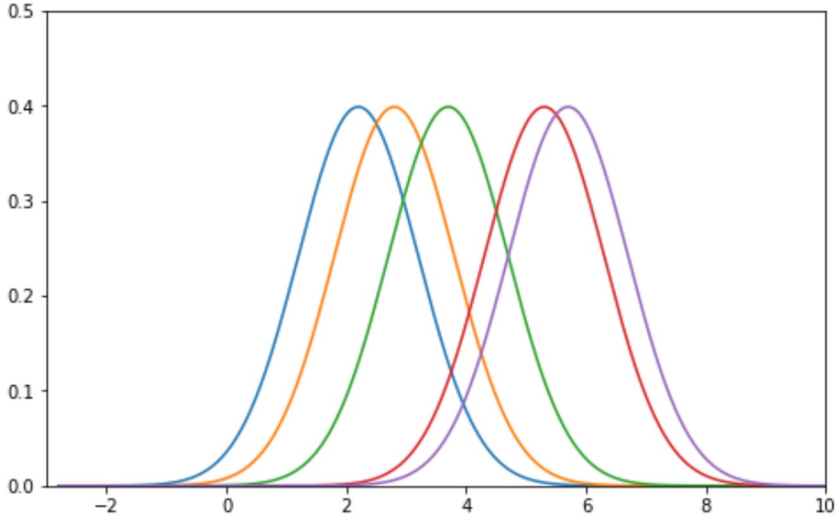
Our goal is to recreate this smooth curve ourselves.

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 **bandwidth** of **alpha = 1**. The idea is that there is a higher density near the points we've already seen.

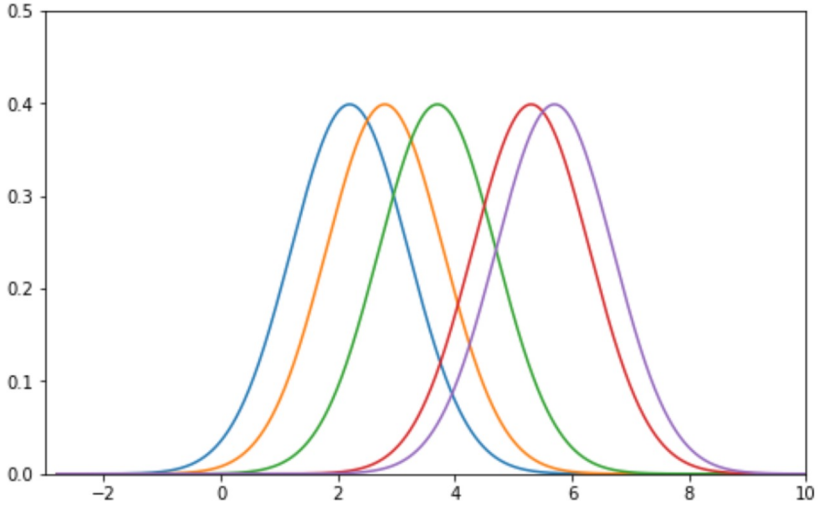
- We will precisely define a **Gaussian kernel** and **alpha** in a few slides.

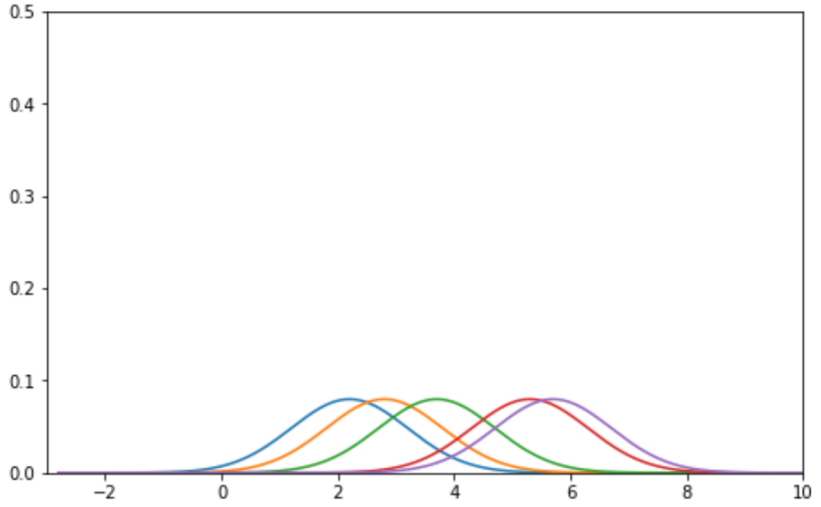




Step 2 – normalize kernels

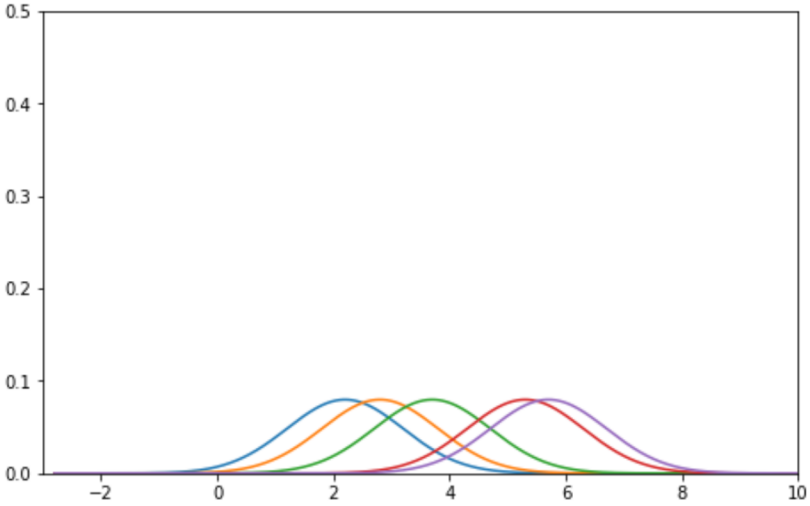
In Step 3, we will be summing each of these **kernels**. We want the result to be a valid density, that has area 1. Right now, we have 5 different **kernels**, each with an area 1. So, we **multiply each kernel by 1/5**.

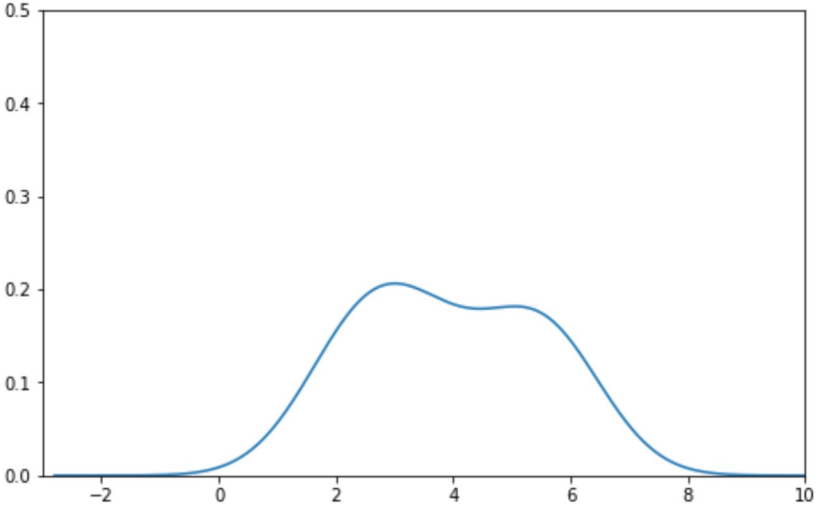




Step 3 – sum kernels

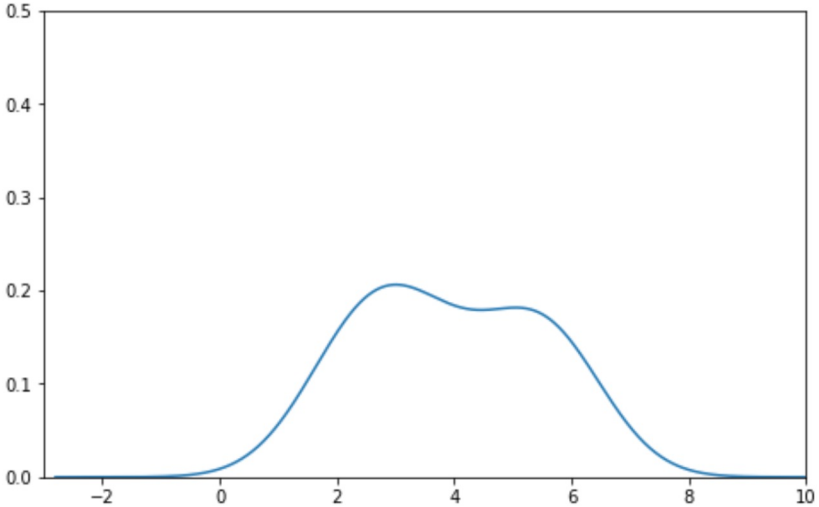
Our **kernel density estimate (KDE)** is the **sum of the normalized kernels at each point.** It is depicted below on the right.

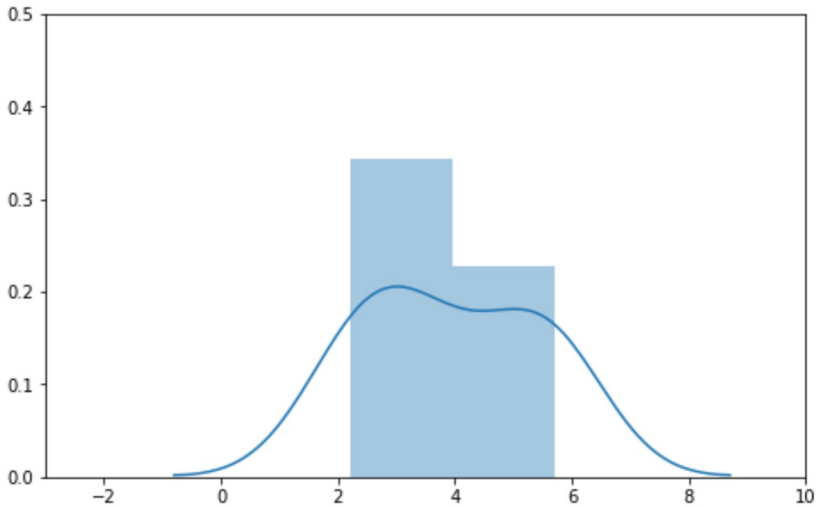




Kernel density estimates

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





Kernel Functions and Bandwidth

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

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.

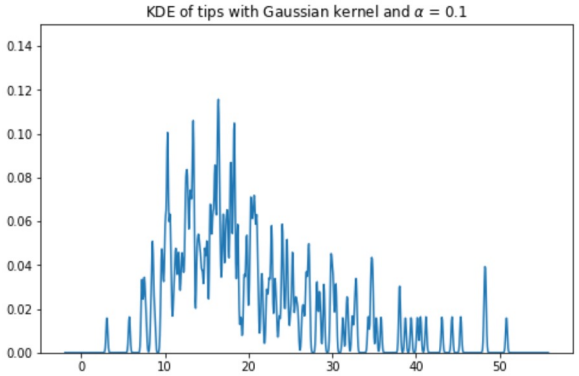
The most common **kernel** is the **Gaussian kernel**.

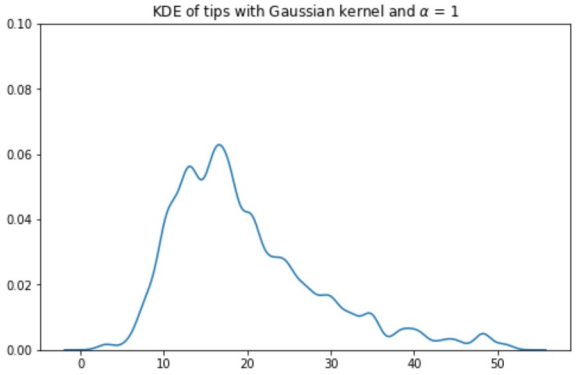
- Here, x represents any input, and x_i represents the ith observed value. The kernels are centered on our observed values (and so the mean of this distribution is x_i).
- **α** is the **bandwidth parameter**. It controls the smoothness of our KDE. Here, it is also the standard deviation of the Gaussian.

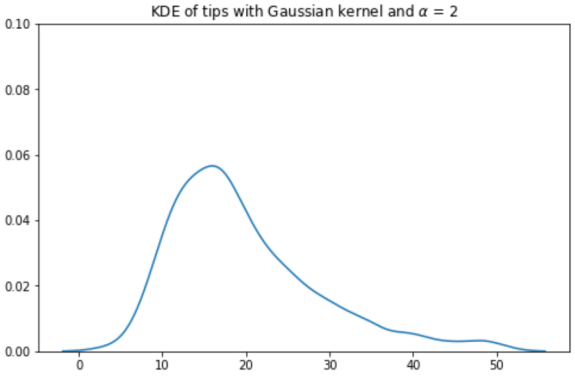
$$K_\alpha(x, x_i) = \frac{1}{\sqrt{2\pi\alpha^2}} e^{-\frac{(x-x_i)^2}{2\alpha^2}}$$

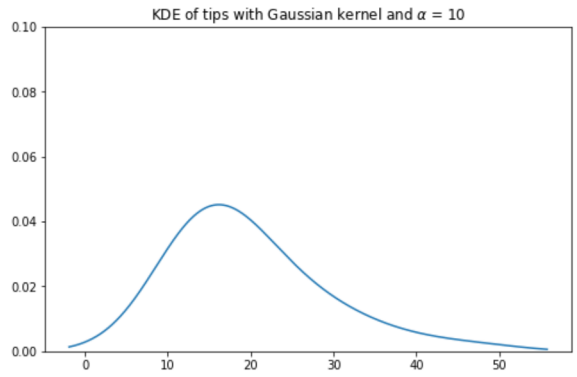
If you've taken a probability class, the mean of the kernel for the ith observed value is x_i, and the standard deviation of the kernel is **α.**

Effect of bandwidth on KDEs



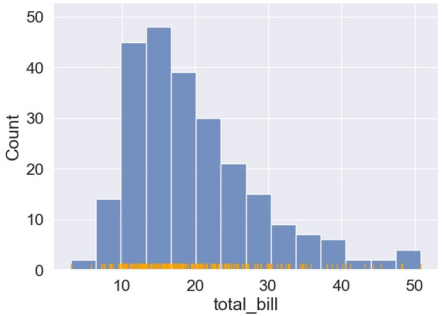






Bandwidth is analogous to the width of each bin in a histogram.

- 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).



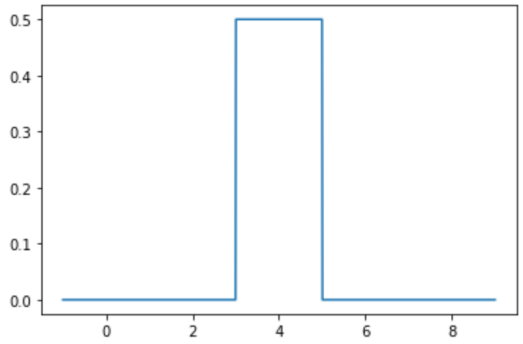
Kernels

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

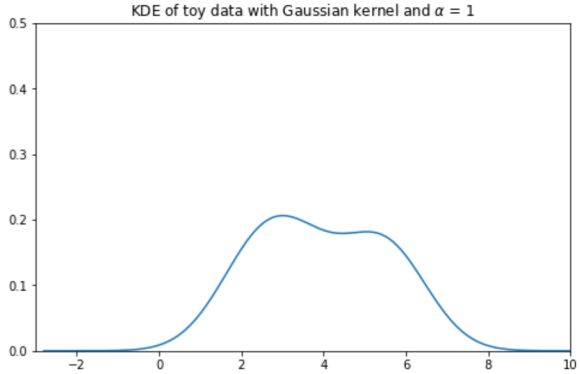
- It assigns uniform density to points within a "window" of the observation, and 0 elsewhere.
- Resembles a histogram… sort of.

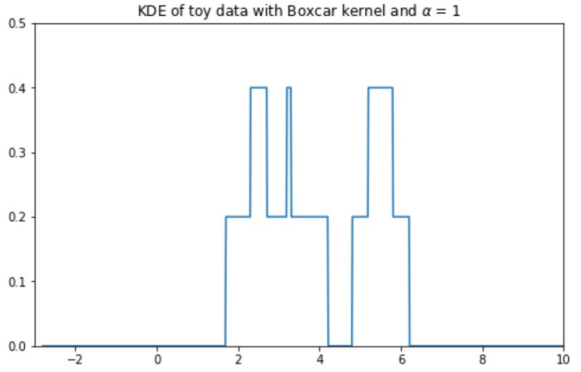
$$K_\alpha(x, x_i) = \begin{cases} \frac{1}{\alpha}, & |x - x_i| \leq \frac{\alpha}{2} \\ 0, & \text{else} \end{cases}$$

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



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





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*

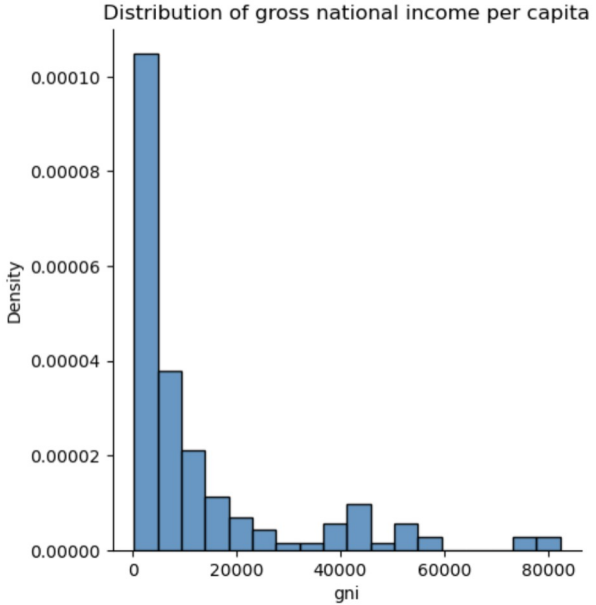
Function that defines the kernel.

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

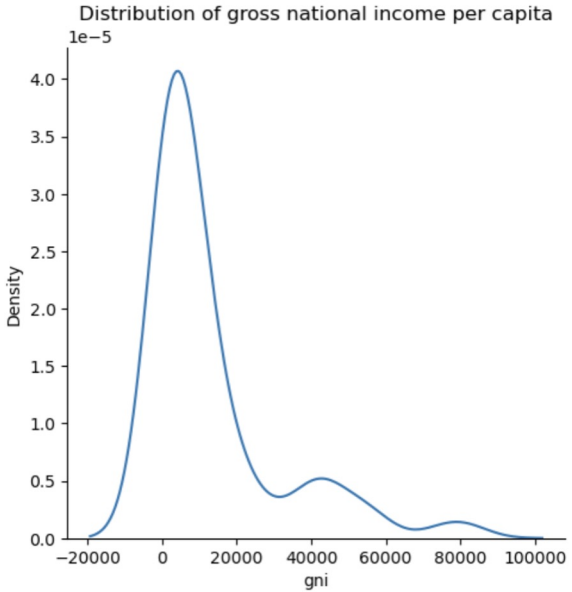
displot

displot is a wrapper for **histplot**, **kdeplot**, and **ecdfplot** to plot distributions.

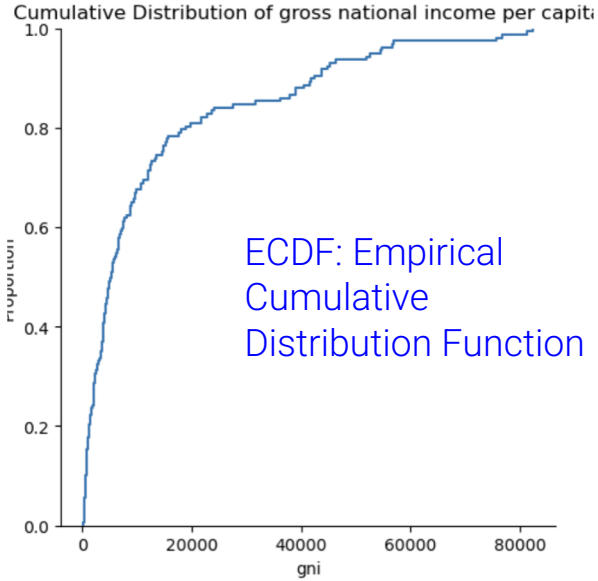
```
sns.displot(data=wb,
            x="gni",
            kind="hist",
            stat="density")
```




```
sns.displot(data=wb,
            x="gni",
            kind="kde")
```




```
sns.displot(data=wb,
            x="gni",
            kind="ecdf")
```



ECDF: Empirical Cumulative Distribution Function

Summary of KDE

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

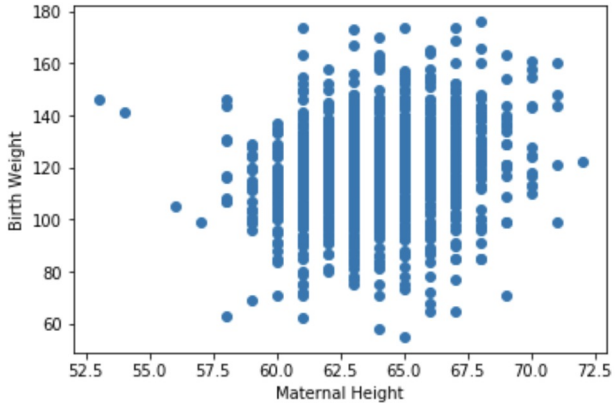
A general “KDE formula” function is given above. We will not cover this in lecture.

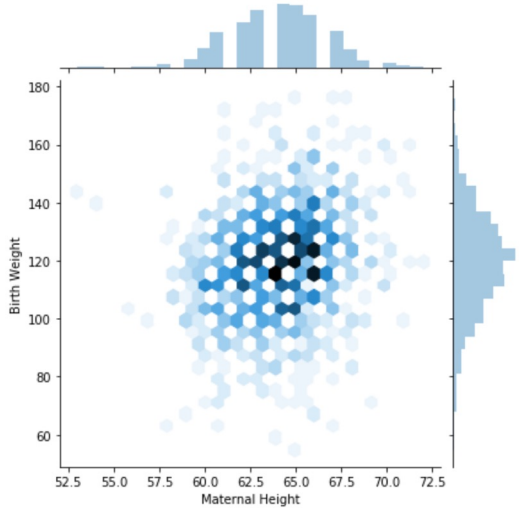
- $\mathbf{x}$ represents any number on the number line. It is the input to our function.
- $\mathbf{n}$ is the number of observed data points that we have.
- Each $\mathbf{x}_i$ ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$) represents an observed data point. These are what we use to create our 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 $\mathbf{1/n}$ so that the total area is still 1.

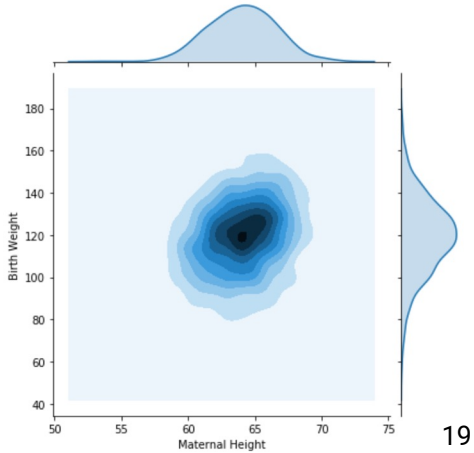
Generalizing to 2D (Contour Map)

We can also do the same thing analogously in 2D.

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





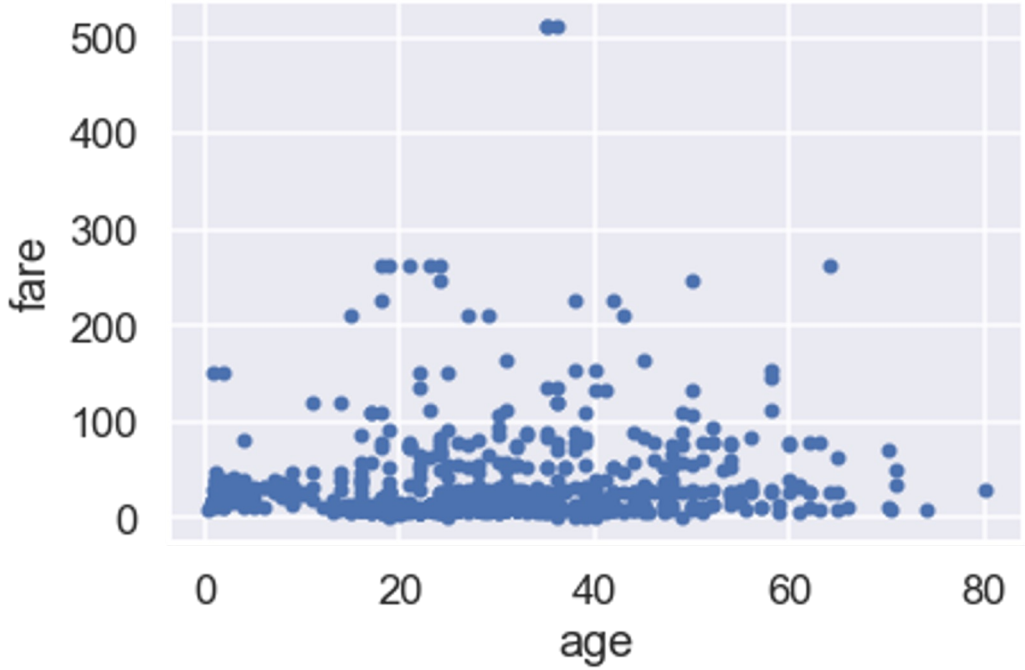


Visualization Theory

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

Visualizations Are For Humans

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

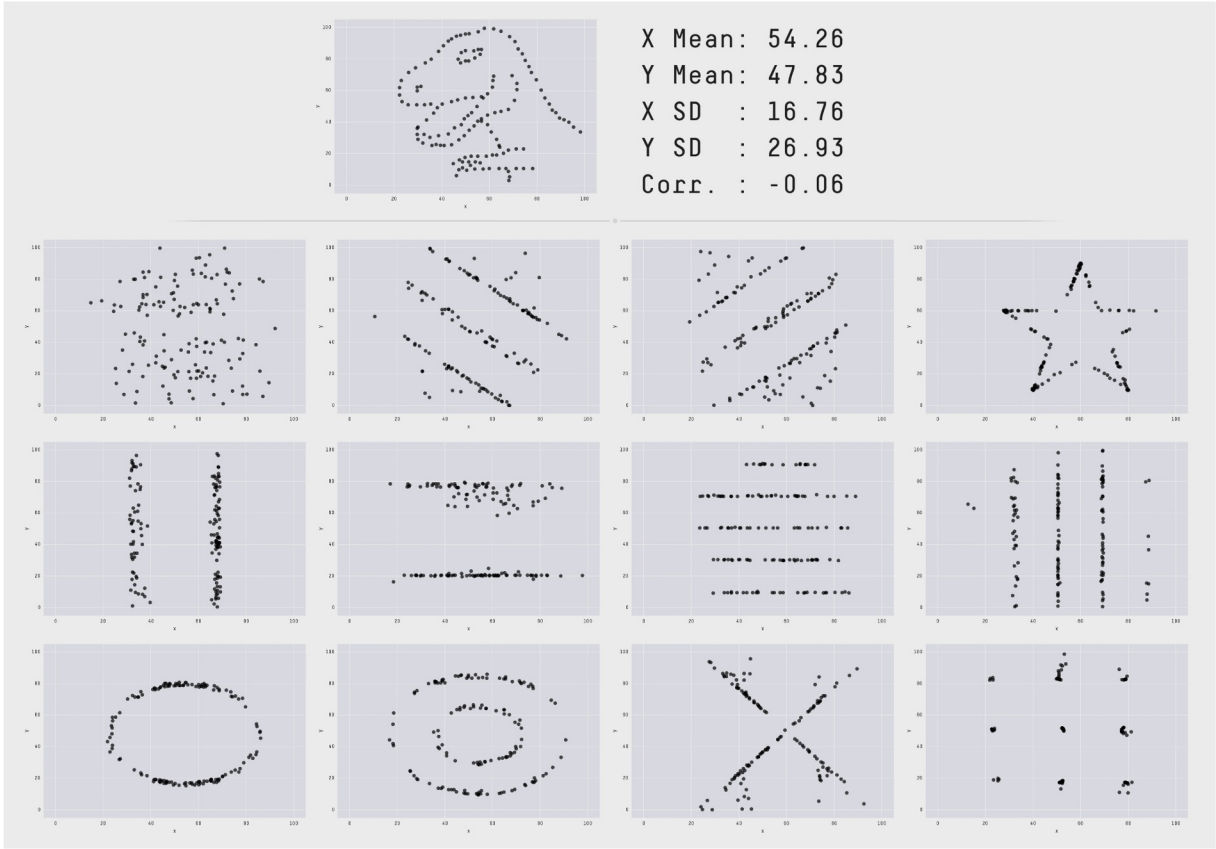


Note: A histogram or KDE would give stronger evidence than a scatter plot.

Visualizations Are More Expressive than Summary Statistics

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

Visualization complements statistics.



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

Information Channels

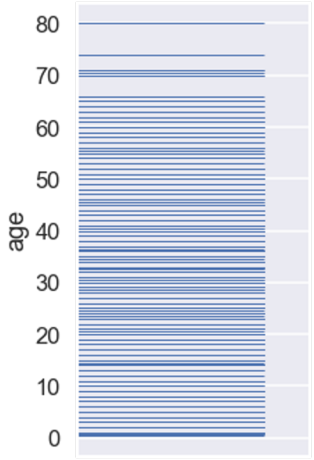
- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Visualization Theory
  - **Information Channels**
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context
- Transformations

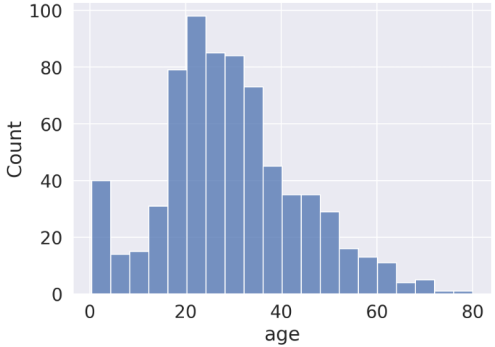
Take advantage of the human visual perception system

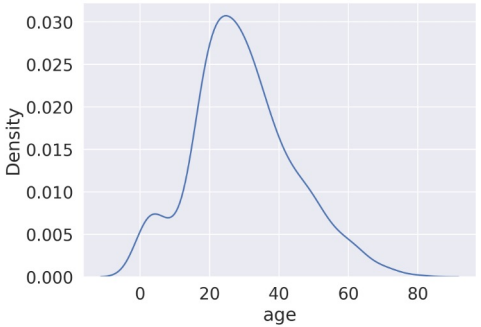
Data can be visualized in many ways!

- 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

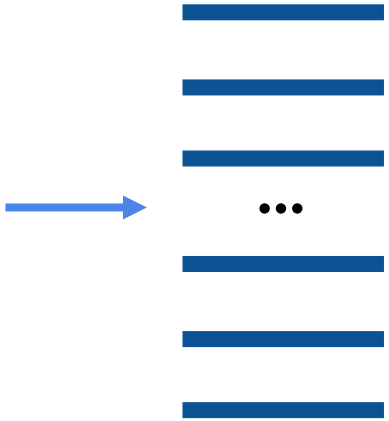




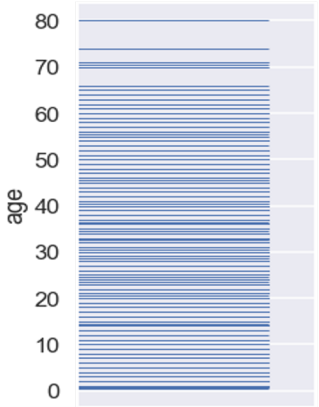


Encoding

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



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

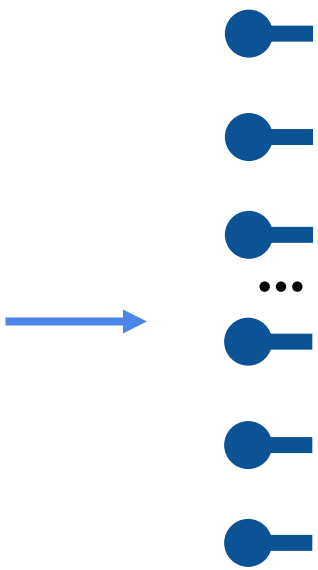


Mark
(Represents a datum)

Encoding
(Maps datum to visual position)

Encoding

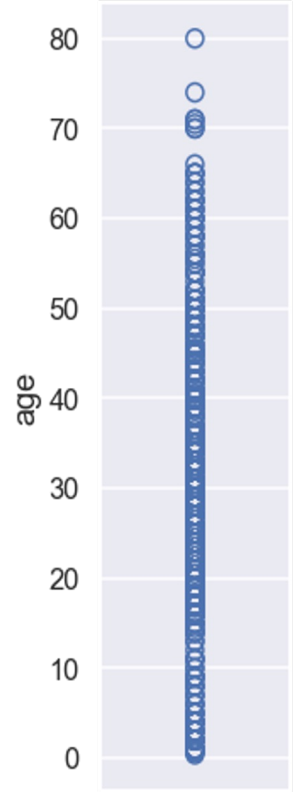
	age
0	22.0
1	38.0
2	26.0
...	...
888	NaN
889	26.0
890	32.0



Mark
(Represents a datum)

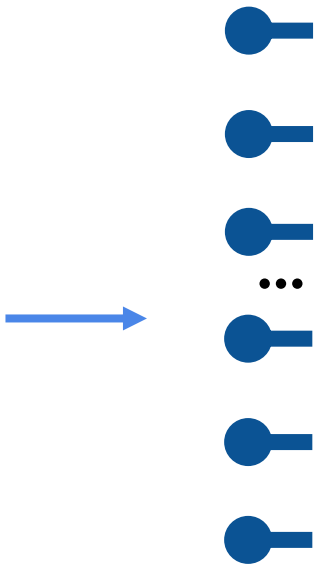
10px
16px
11px
...
NONE
11px
15px

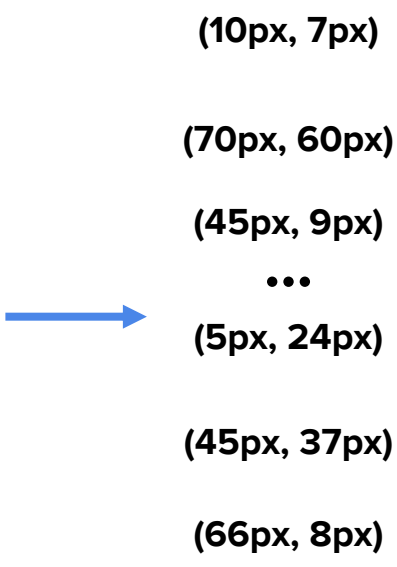
Encoding
(Maps datum to visual position)

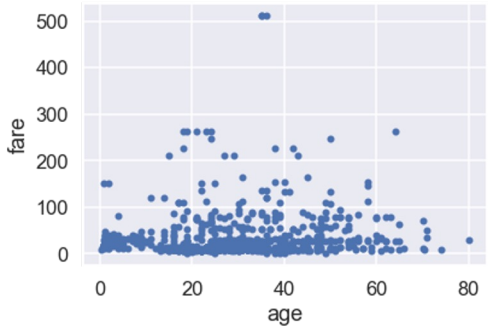


Encoding

	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







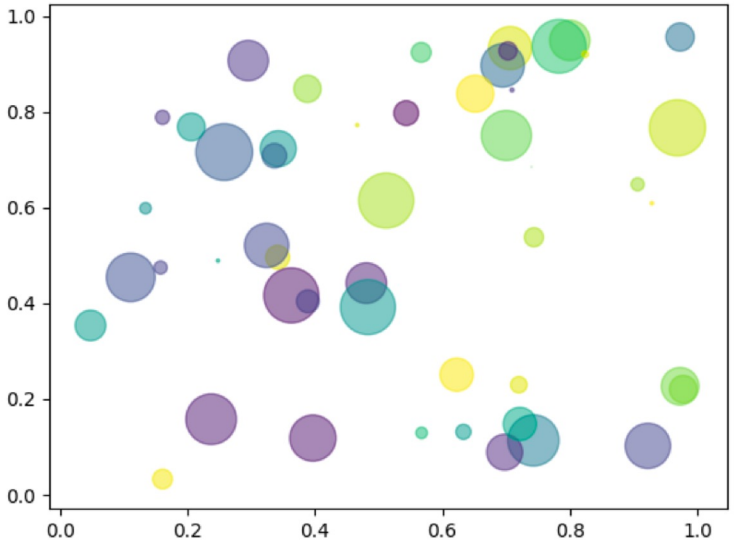
Mark
(Represents a datum)

Encoding
(Maps datum to visual position)

Going Beyond 3D

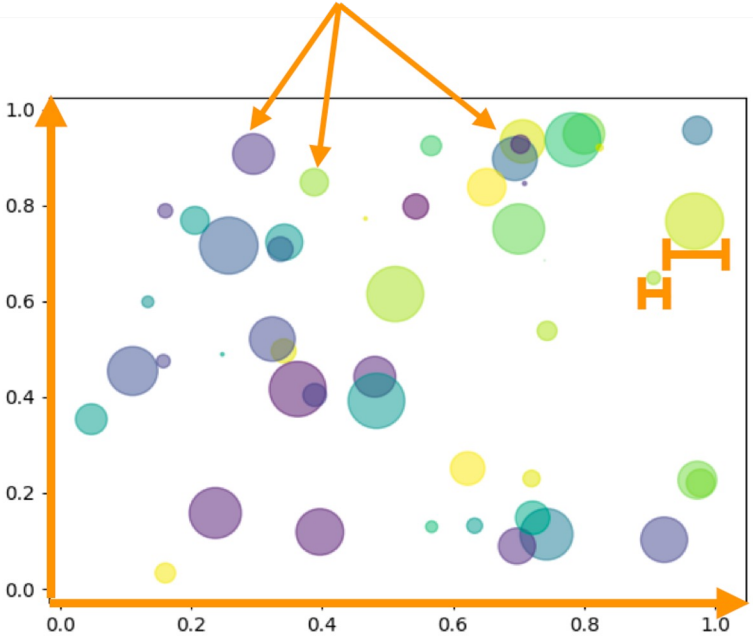
How many variables are we encoding here?

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



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 Length

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

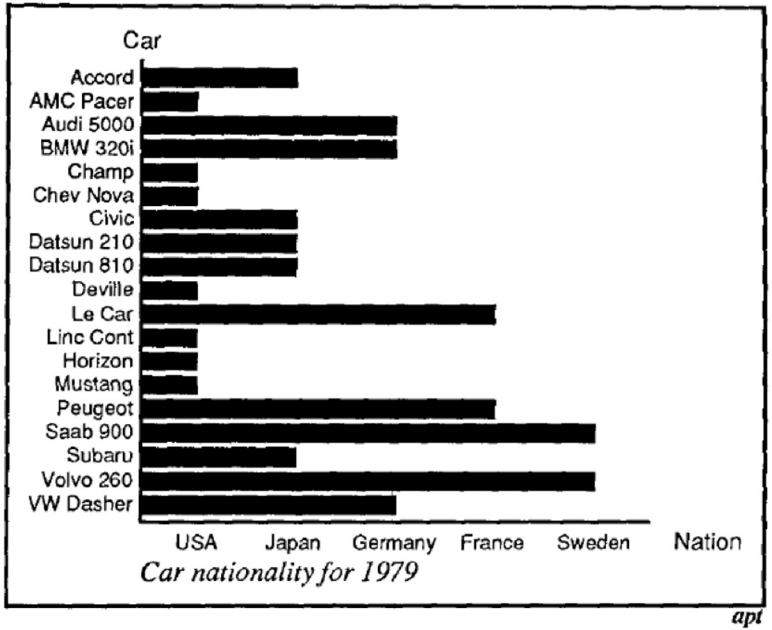


Fig. 11. Incorrect use of a bar chart for the *Nation* relation. The lengths of the bars suggest an ordering on the vertical axis, as if the USA cars were longer or better than the other cars, which is not true for the *Nation* relation.

Link

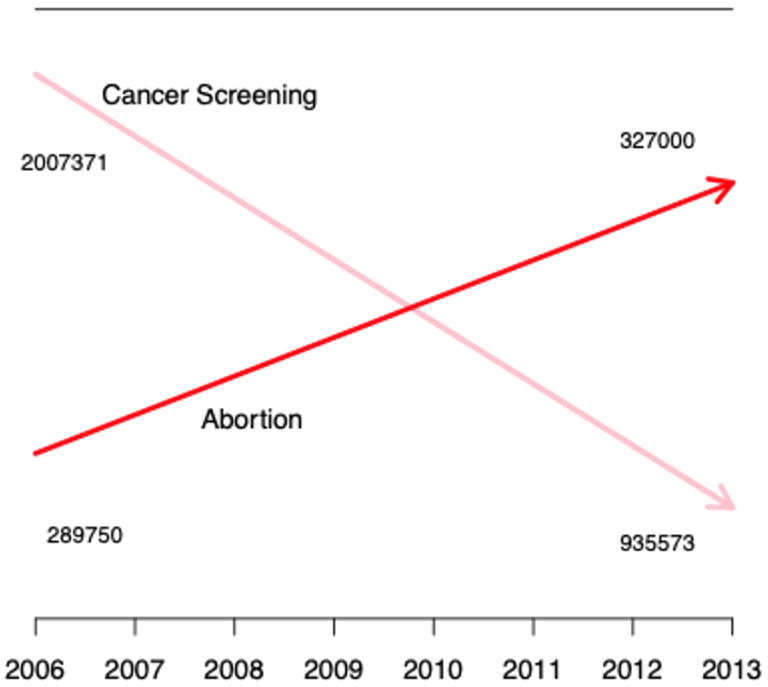
For the next huge chunk of today's lecture, we'll dive into ways to properly use other aspects of a visualization:

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

Harnessing X/Y

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

Case Study: Planned Parenthood Hearing

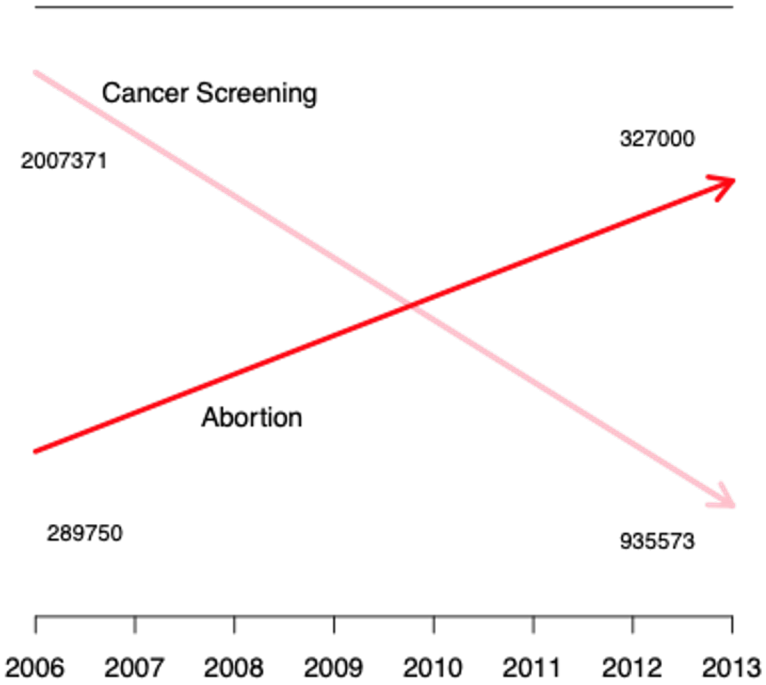


In 2015, Planned Parenthood was accused of selling aborted fetal tissue for profit.

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



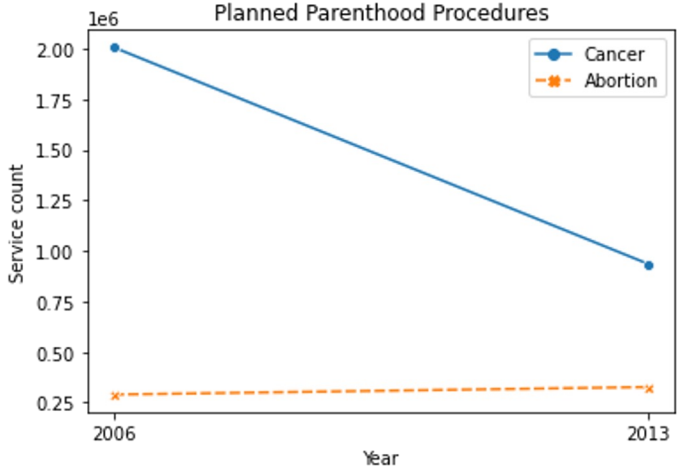
The scales for the two lines are completely different!

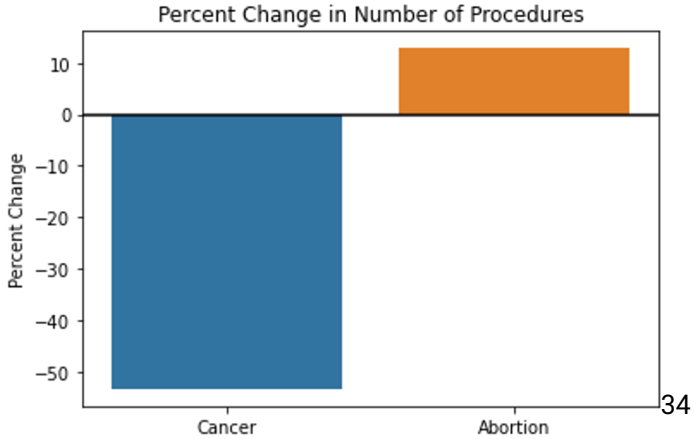
- 327000 is smaller than 935573, but appears to be way bigger.
- **Do not use two different scales for the same axis!**

Consider scale of the data

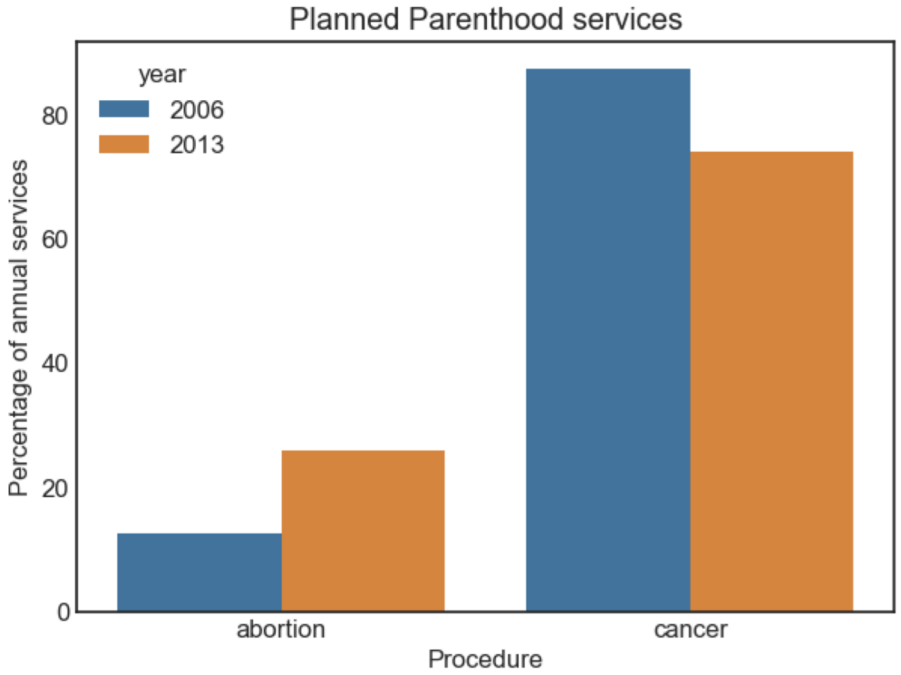
The top plot draws all of the data on the same scale.

- 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



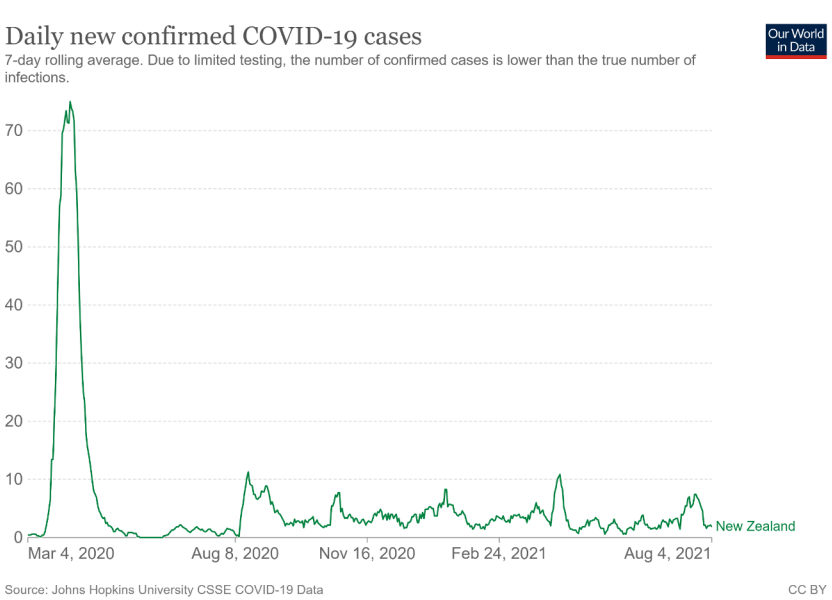
We could also visualize abortions and cancer screenings as a percentage of total procedures.

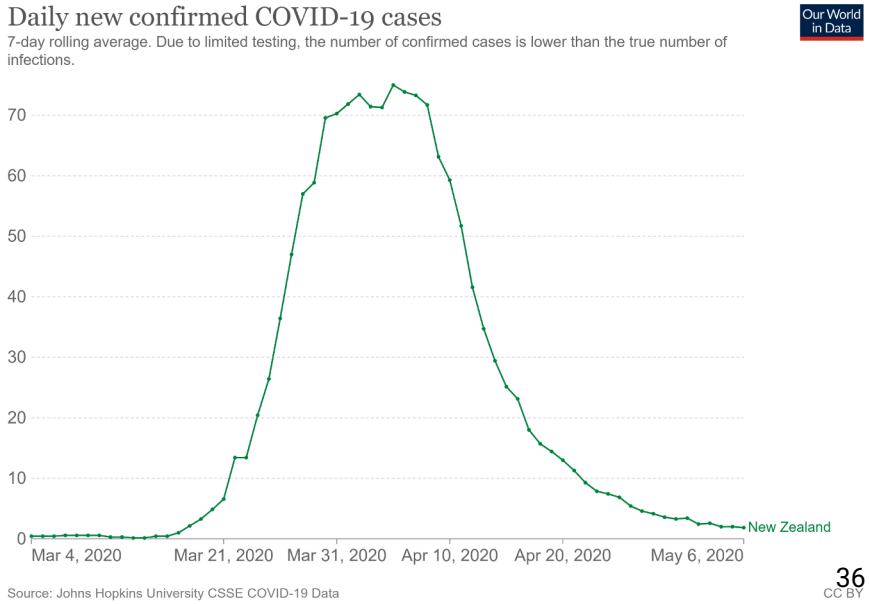
- Abortions increased from 13% to 26% of total procedures.

Reveal the Data

Recommendations:

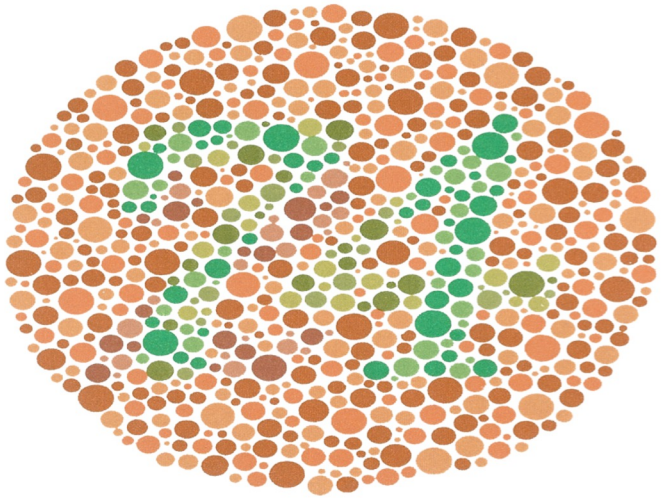
- Choose axis limits to fill the visualization.
- **You don't have to visualize all of the data at once:**
  - Zoom in on the bulk of the data (it's ok to not include 0!) if only one part matters.
  - Can also create multiple plots to show different regions of interest.





Harnessing Color

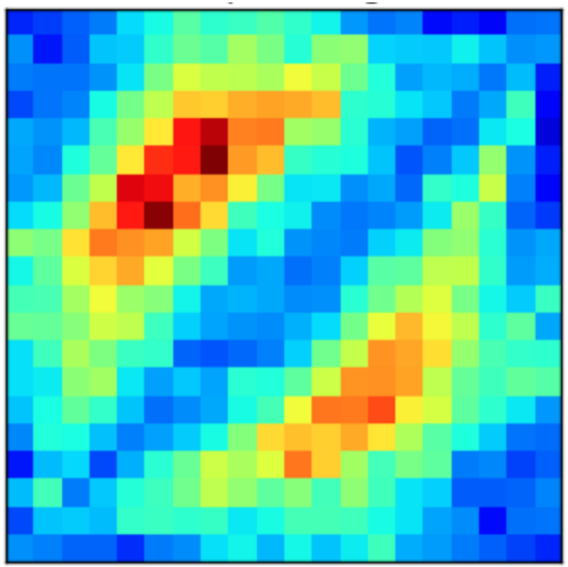
- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - **Harnessing Color**
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context
- Transformations



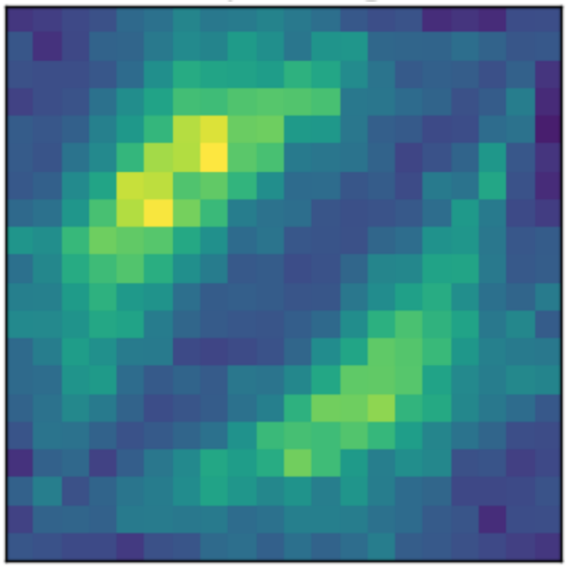
Choosing a set of colors which work together is a challenging task!

Perception of Color

Colormaps

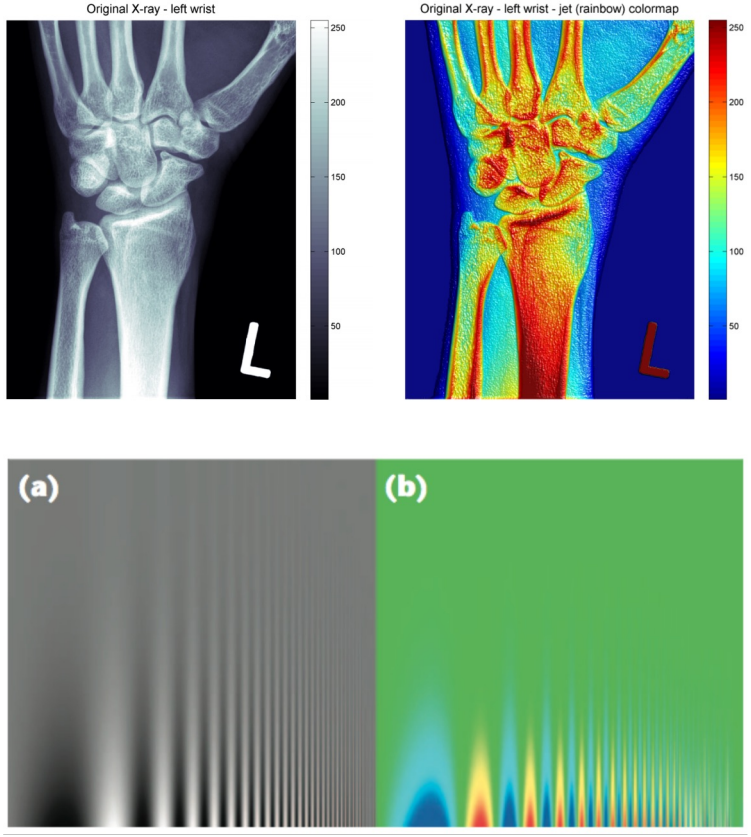


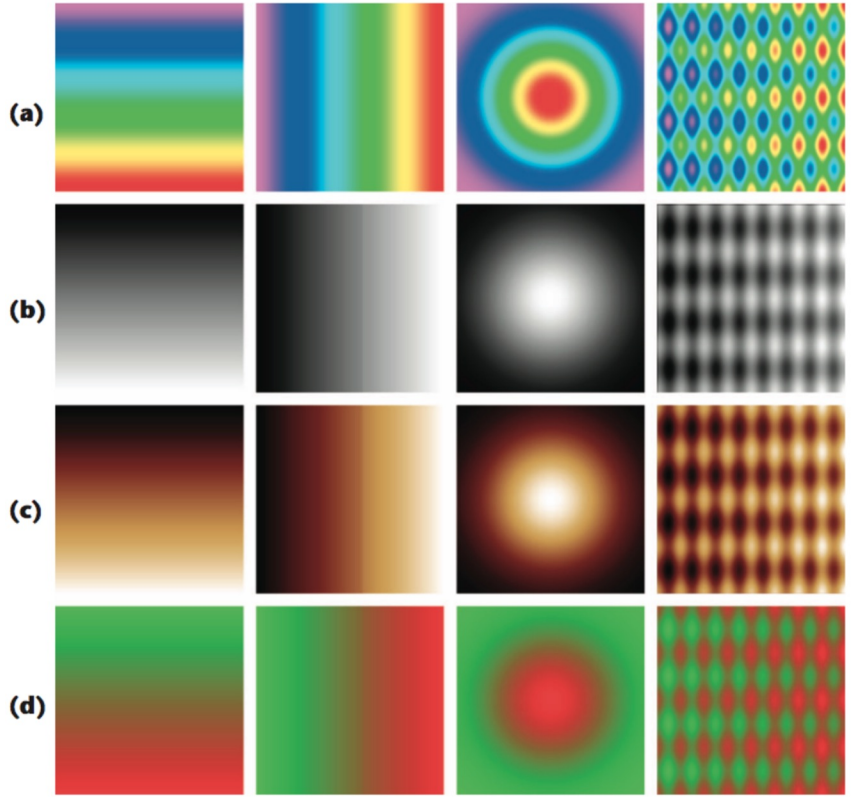
Jet



Viridis

The jet/rainbow colormap actively misleads

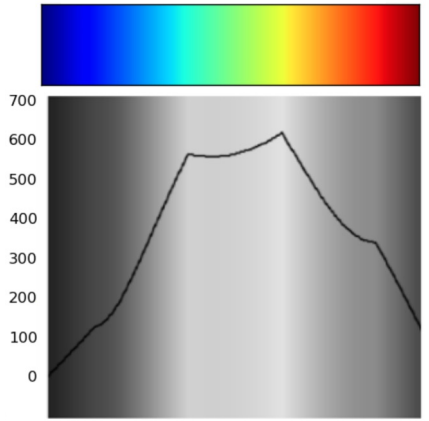




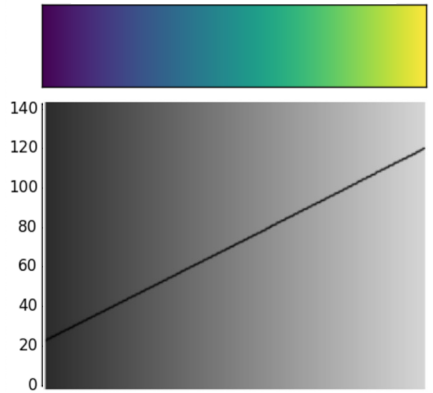
"Rainbow Colormap (Still) Considered Harmful", Borland and Taylor, 2007.

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.
  - https://bids.github.io/colormap/
- Avoid combinations of red and green, due to red-green color blindness.



Bounces all over



Slope is constant

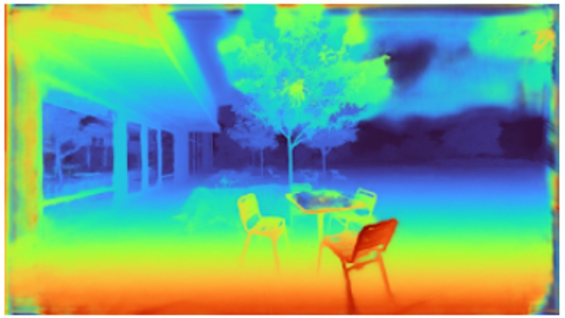
x-axis is color, y-axis is "lightness"

Except when not :) The Google Turbo Colormap

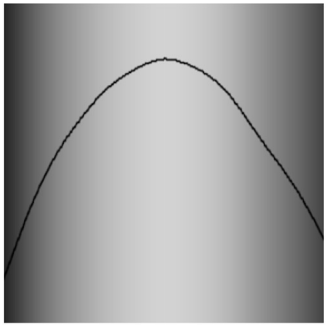
Turbo

Jet

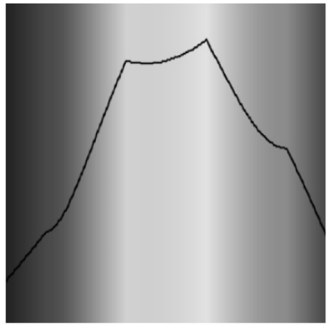
Inferno



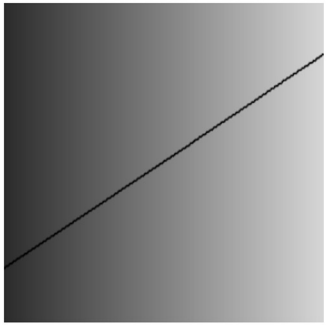
Turbo



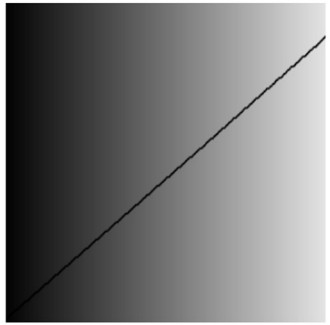
Turbo



Jet



Viridis



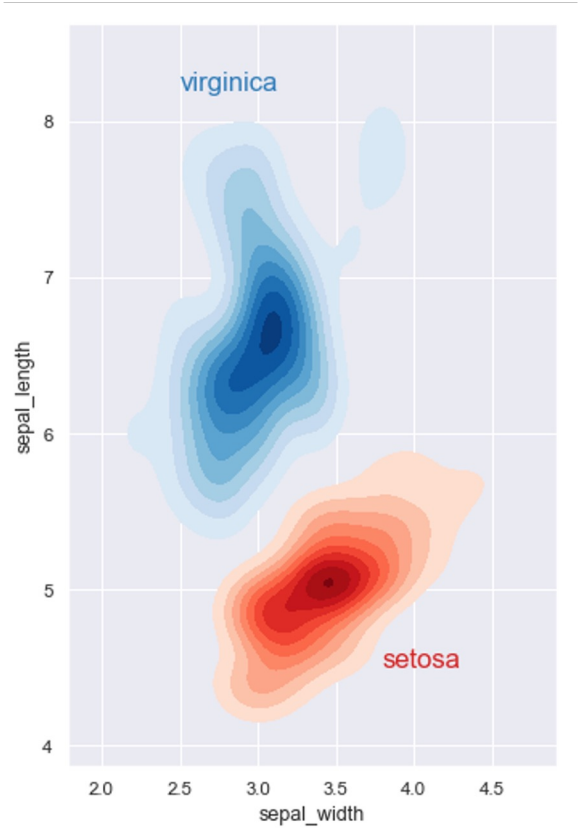
Inferno

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

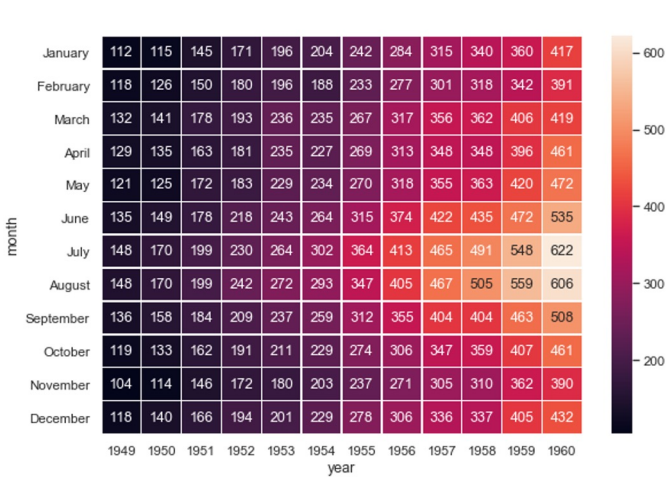
https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html

Use color to highlight data type

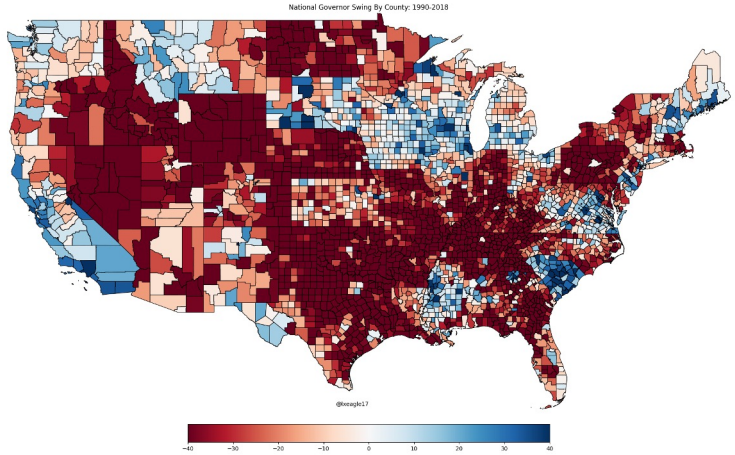
- **Qualitative**: Choose a qualitative scheme that makes it easy to distinguish between categories.
  - One category isn't "higher" or "lower" than another.
- **Quantitative**: Choose a color scheme that implies magnitude.
  - More on this in the next slide.
- The plot on the right has both!



Sequential vs. diverging colormaps for quantitative data

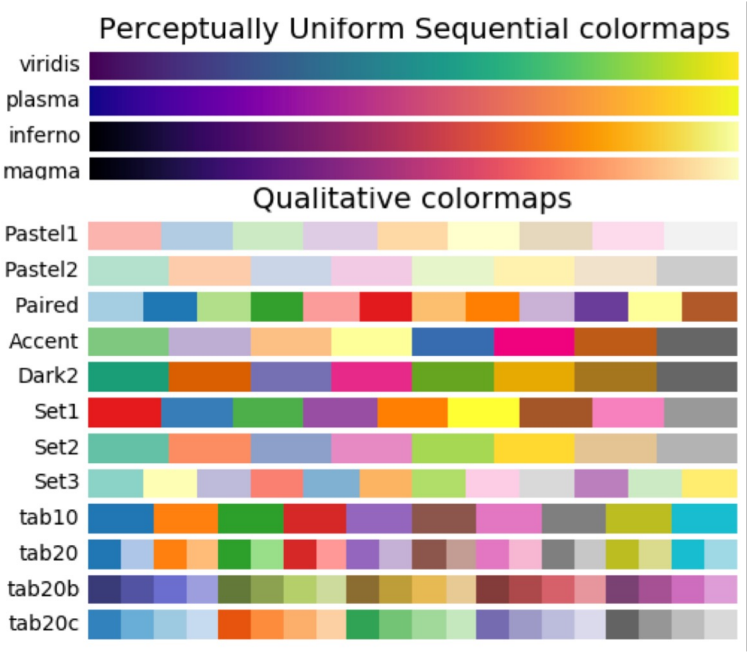


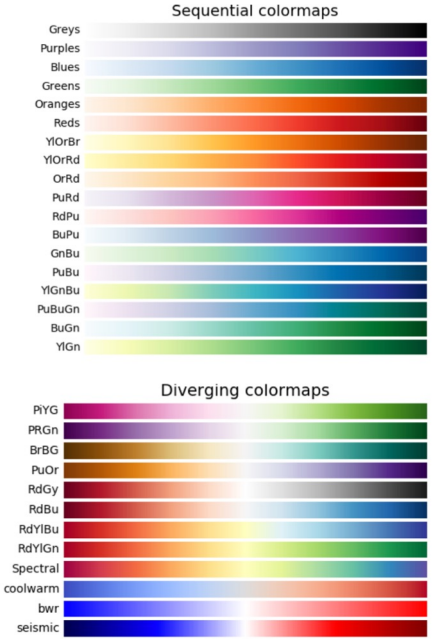
If the data progresses from low to high, use a **sequential** scheme where lighter colors are for more extreme values.



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.

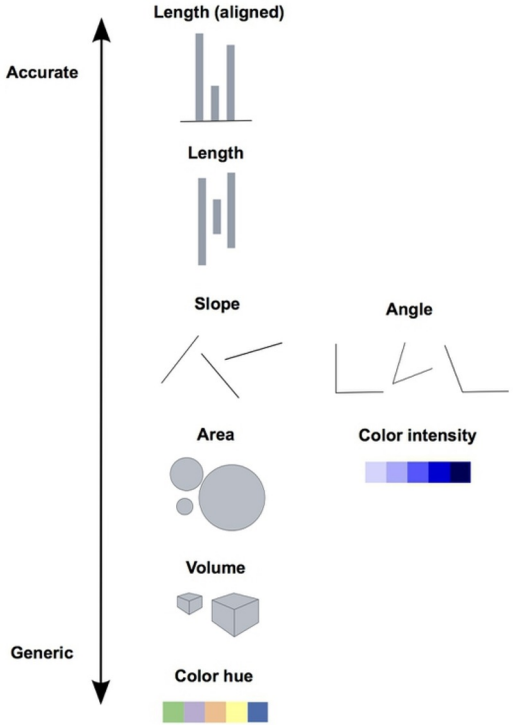
Extra reading

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

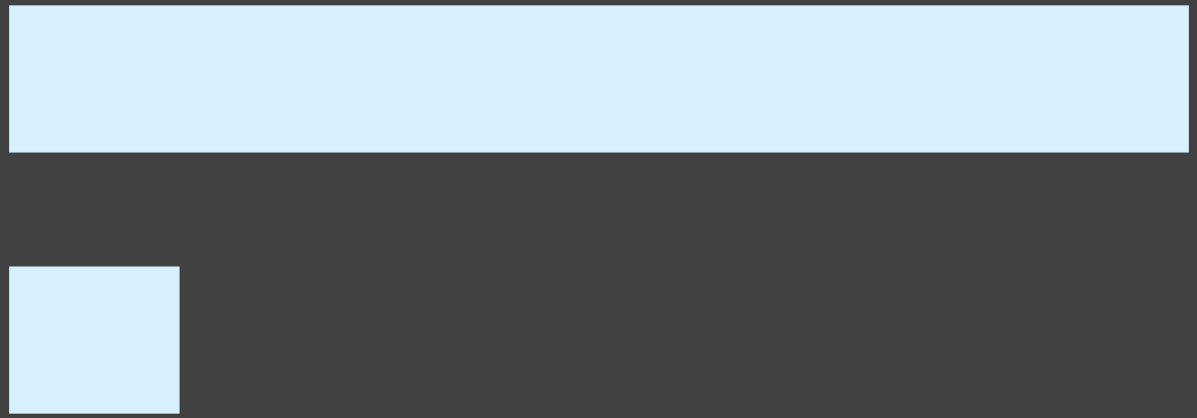
Harnessing Markings

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

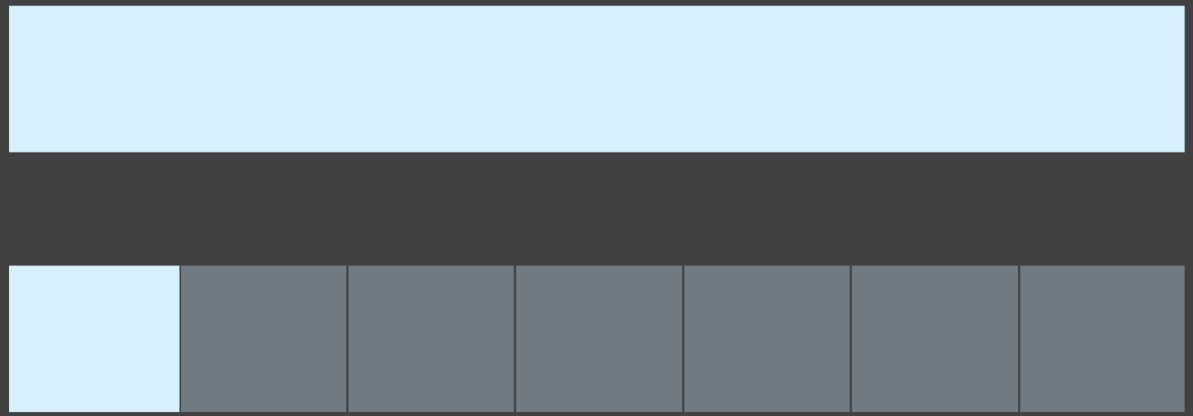


The accuracy of our judgements depend on the type of marking.

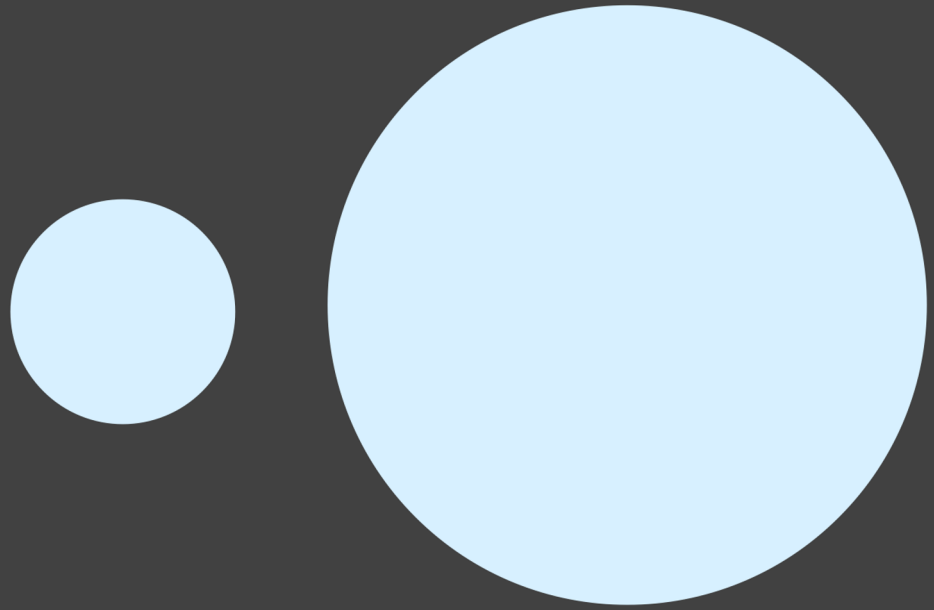
Perception of Markings



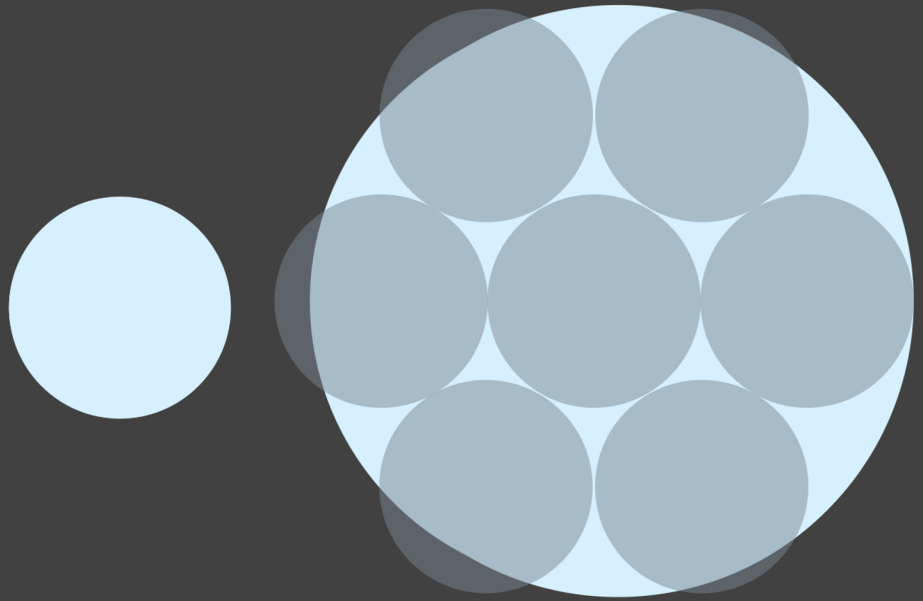
How much longer is the top bar?



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

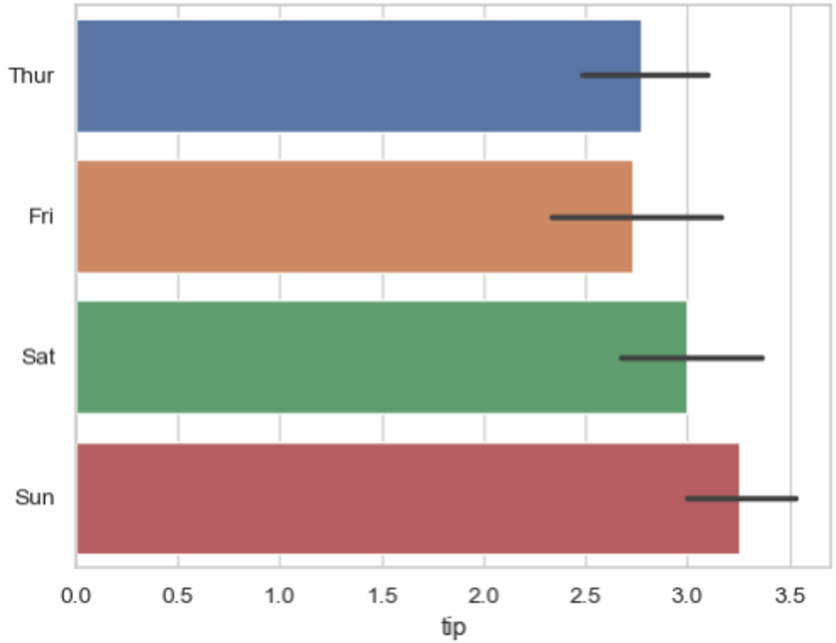


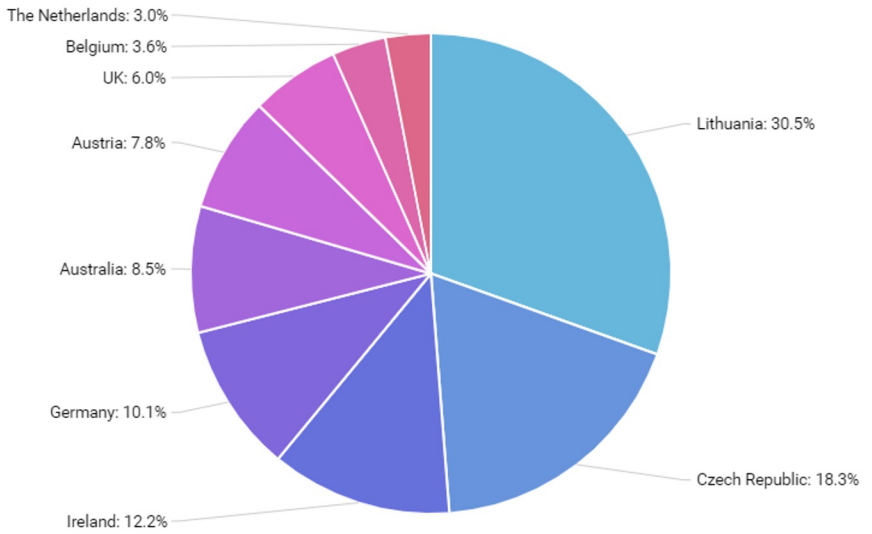
How much bigger is the big circle?



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

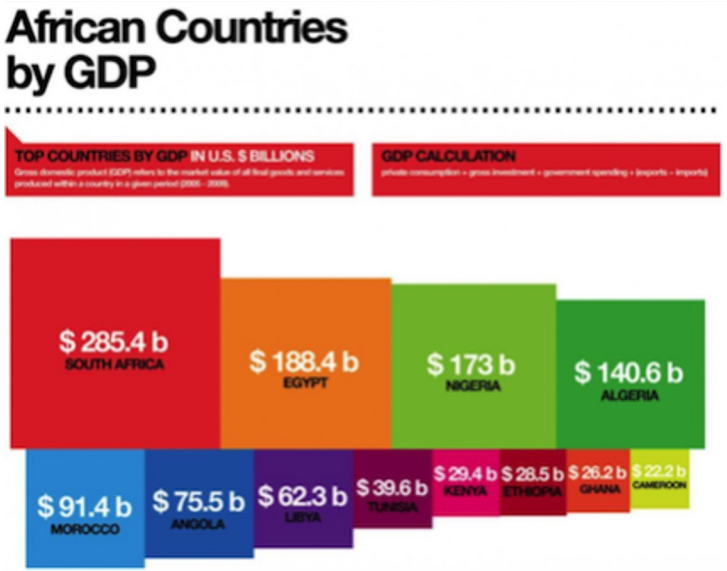
Lengths are easy to distinguish; angles are hard

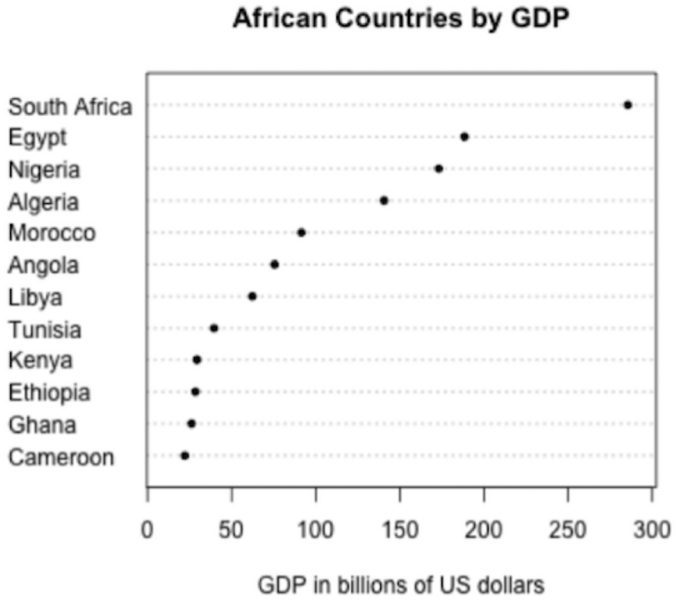




Don’t use pie charts! Angle judgements are inaccurate.

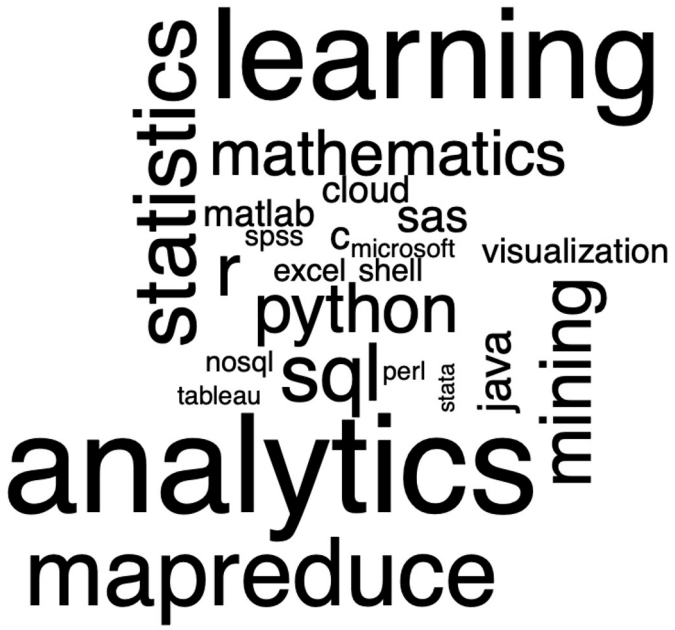
Areas are hard to distinguish

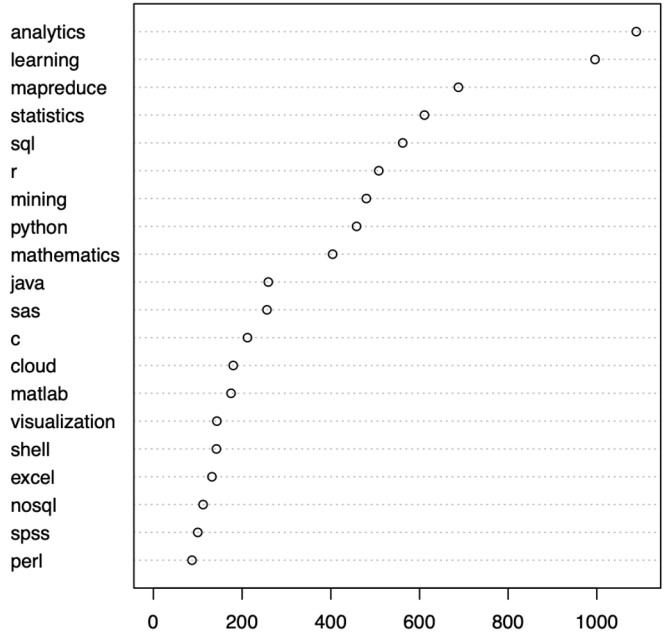




Avoid area charts! Area judgements are inaccurate. (For instance, South Africa has twice the GDP of Algeria, but that isn't clear from the areas.)

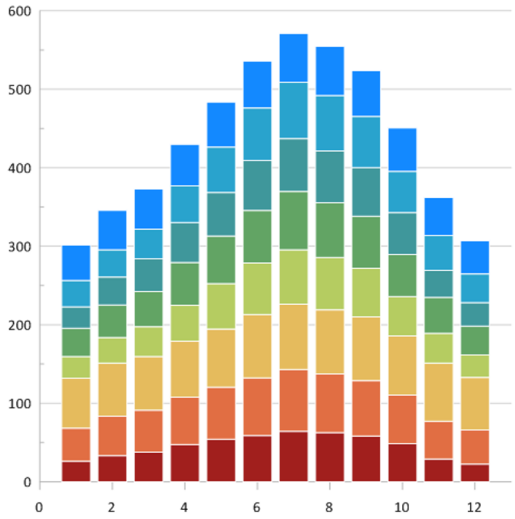
Areas are hard to distinguish





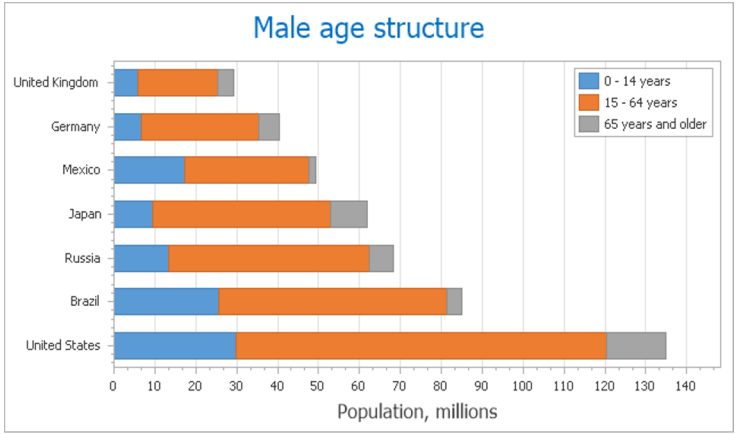
Avoid word clouds too! It's hard to tell the area taken up by a word.

Avoid jiggling the baseline



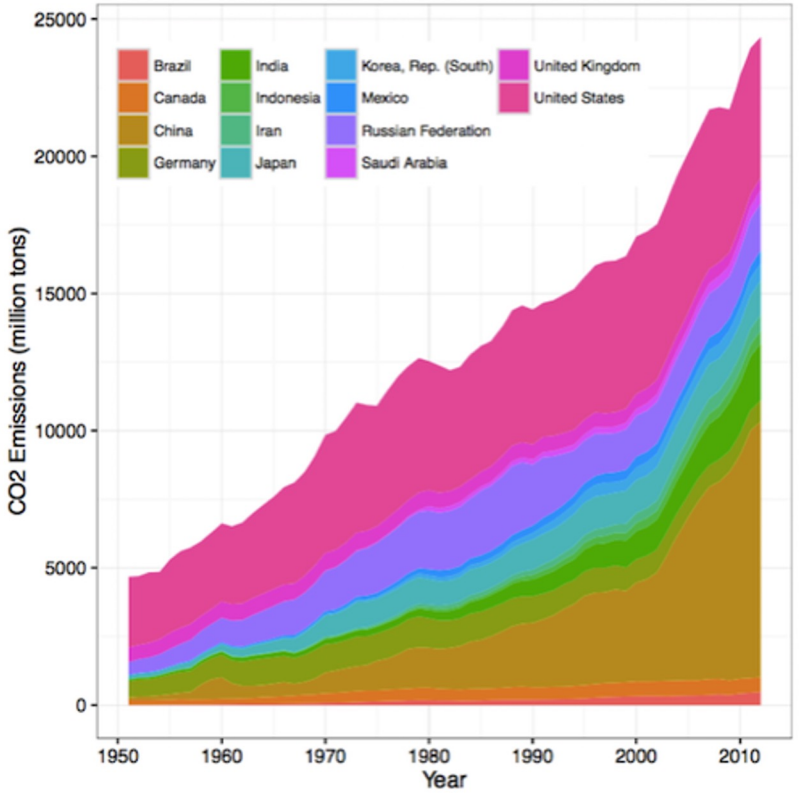
Stacked bar charts, histograms, and area charts are hard to read because the baseline moves.

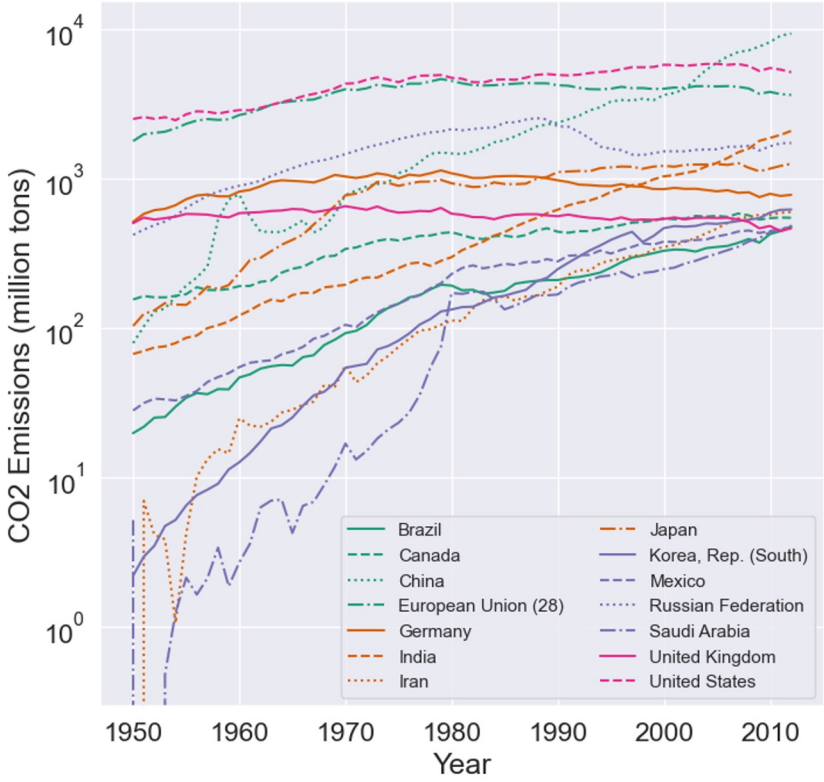
- In the first plot, the top blue bars are all roughly of the same length. But that's not immediately obvious!
- In the second plot, comparing the number of 15-64 year old males in Germany and Mexico is difficult.



Avoid jiggling the baseline

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

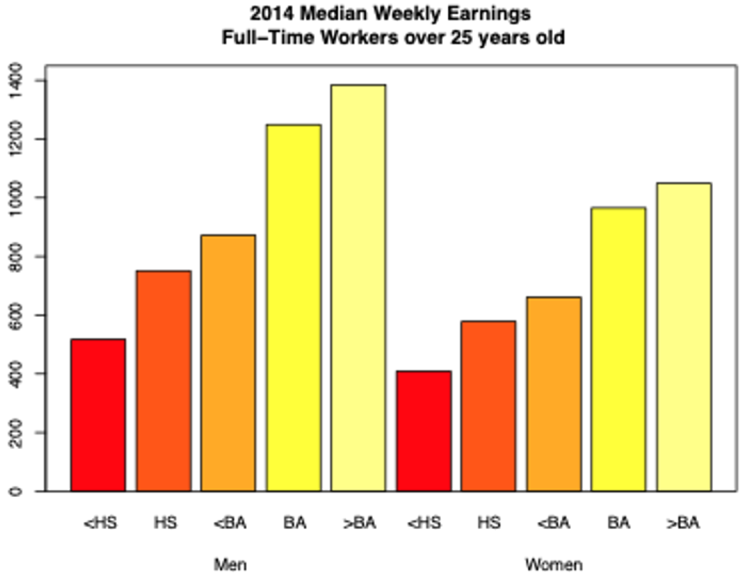




Harnessing Conditioning

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

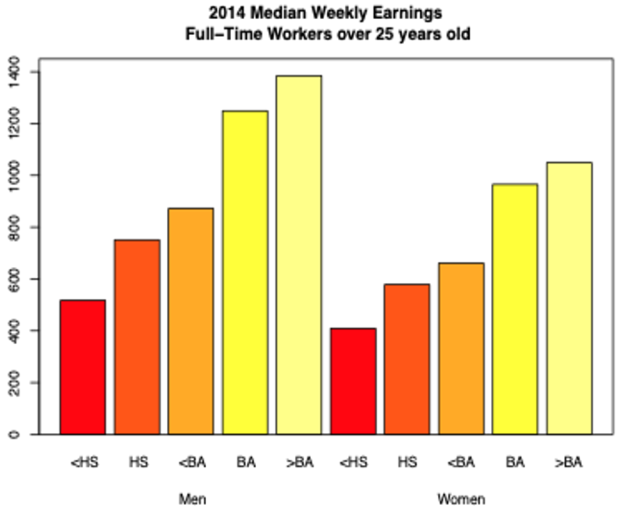
Case Study: Median Weekly Earnings



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?

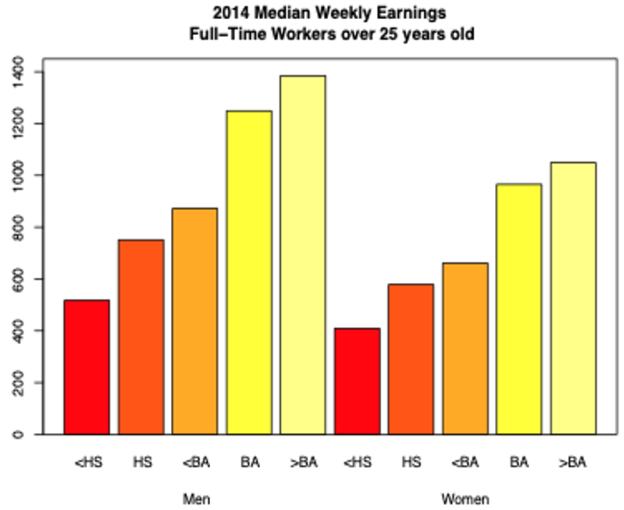
Use conditioning to aid comparison

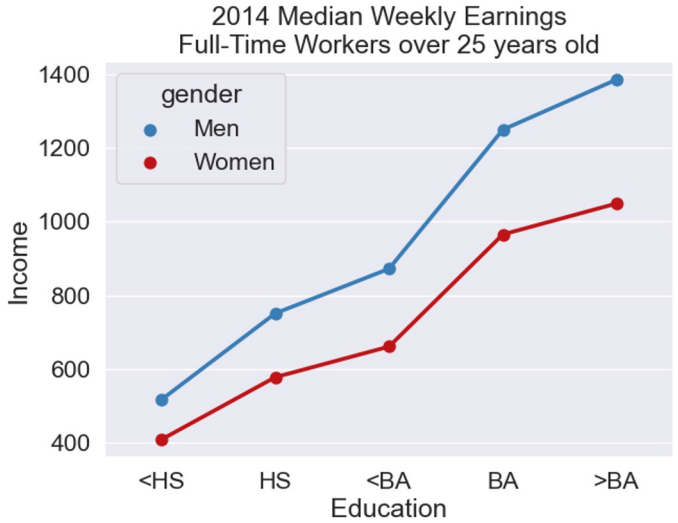


Easy to see the effect of education on earnings. But hard to compare between the two genders in the dataset.

- How could we more easily make this difficult comparison?

Use conditioning to aid comparison

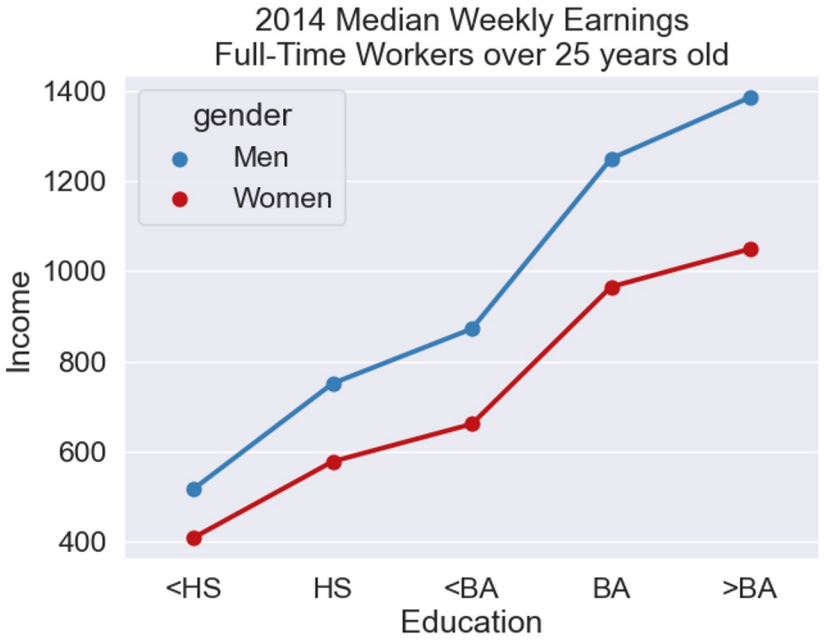


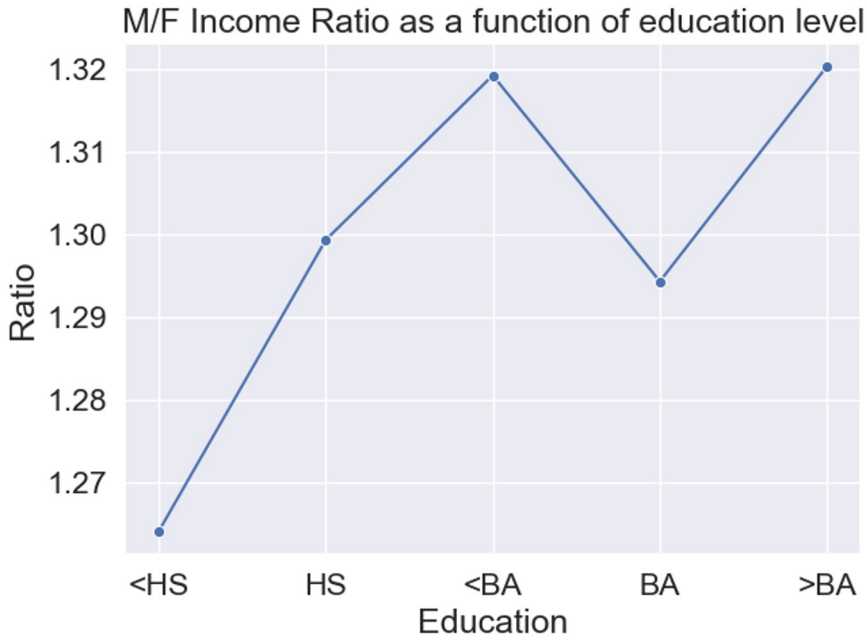


Left figure: Easy to see the effect of education on earnings. But hard to compare between the two genders in the dataset.

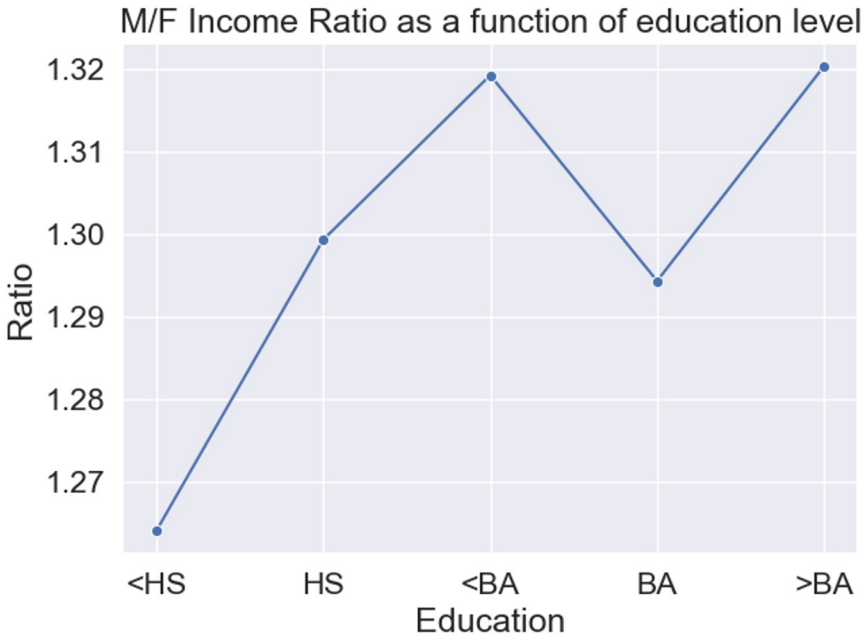
Right figure: Having two separate lines makes clear the wage difference between men and women.

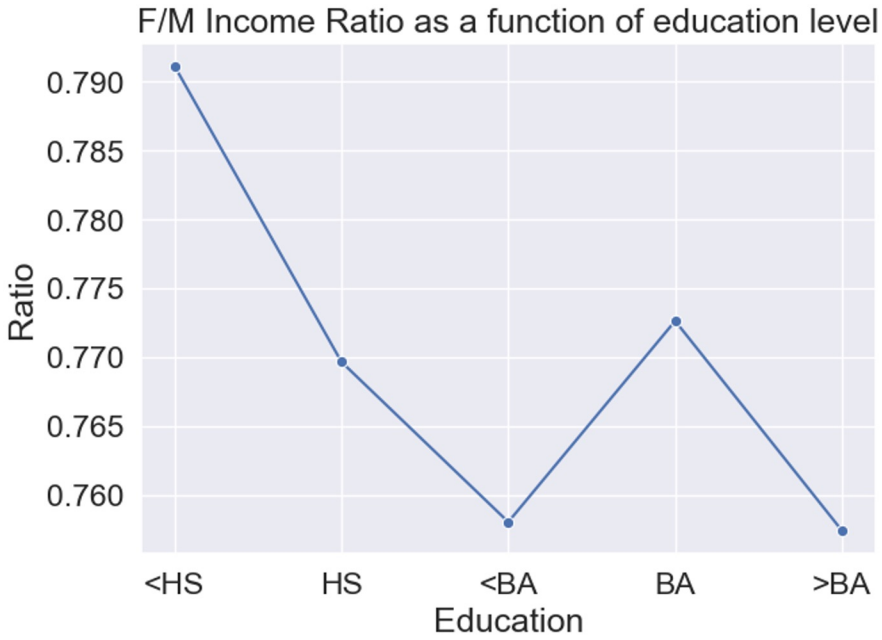
How does the income gap increase with education?



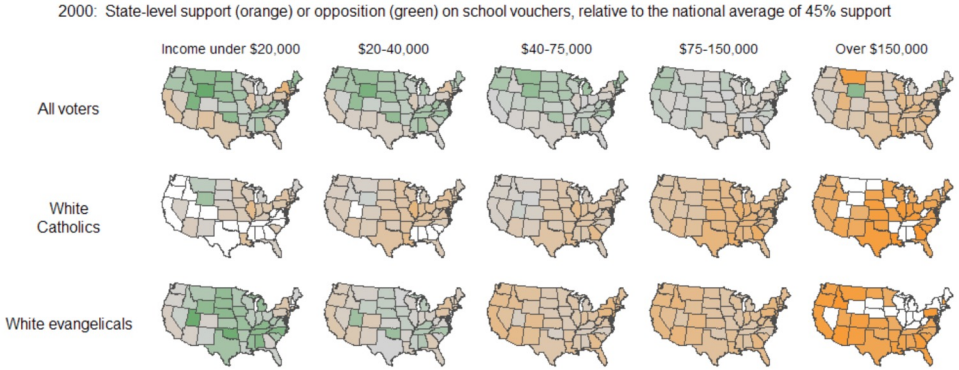


But… which ratio should we pick? M/F or F/M?





Superposition vs. Juxtaposition



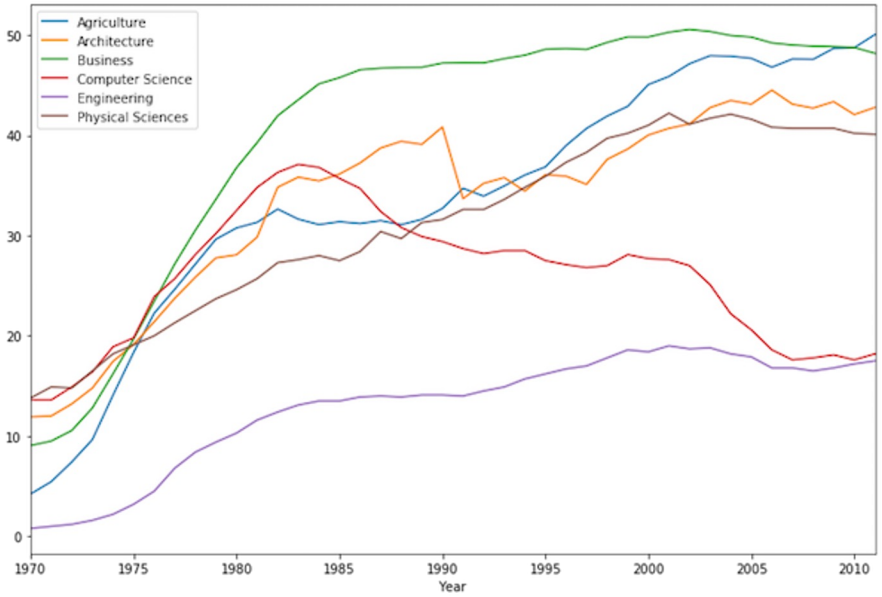
An example of **small multiples**.

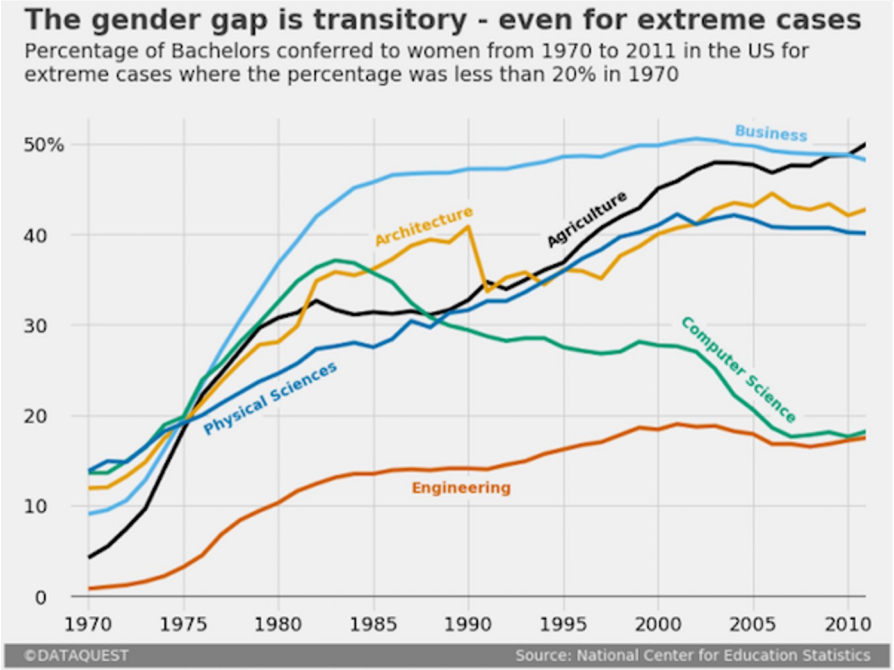
Superposition: placing multiple density curves, scatter plots on top of each other (what we've usually been doing)

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

Harnessing Context

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


Agriculture
Architecture
Business
Computer Science
Engineering
Physical Sciences
50
40
30
20
10
0
1970
1975
1980
1985
1990
1995
2000
2005
2010
Year


The gender gap is transitory - even for extreme cases
Percentage of Bachelors conferred to women from 1970 to 2011 in the US for extreme cases where the percentage was less than 20% in 1970
50%
40
30
20
10
0
1970
1975
1980
1985
1990
1995
2000
2005
2010
Business
Architecture
Agriculture
Computer Science
Physical Sciences
Engineering
©DATAQUEST
Source: National Center for Education Statistics

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.

Transformations

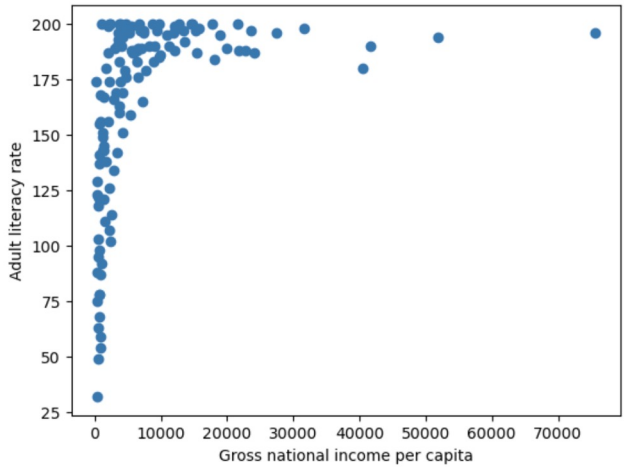
- Kernel Density Functions
  - KDE Mechanics
  - Kernel Functions and Bandwidth
- Visualization Theory
  - Information Channels
  - Harnessing X/Y
  - Harnessing Color
  - Harnessing Markings
  - Harnessing Conditioning
  - Harnessing Context
- **Transformations**

Visualization Theory

Remember our goals of visualization:

1. To help your own understanding of your data/results.
2. To communicate results/conclusions to others.

These are influenced by our choice of visualization and our choices in *how to prepare data for visualization*.



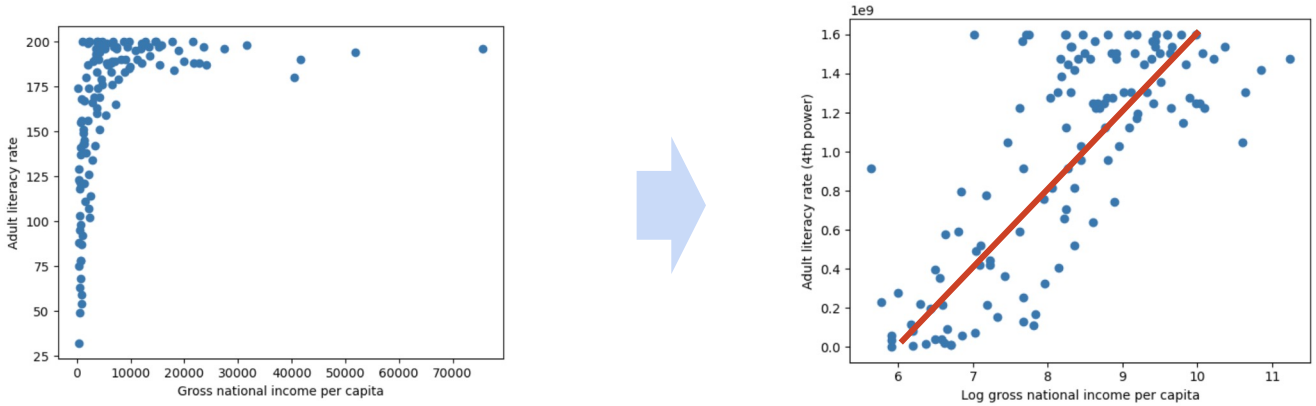
What problems are there here?

- Data is "smushed" – hard to interpret, even if we jittered.
- Difficult to generalize a clear relationship between the variables.

We often **transform** a dataset to help prepare it for being visualized.

Linearization

When applying transformations, we often want to **linearize** the data – rescale the data so the x and y variables share a linear relationship.

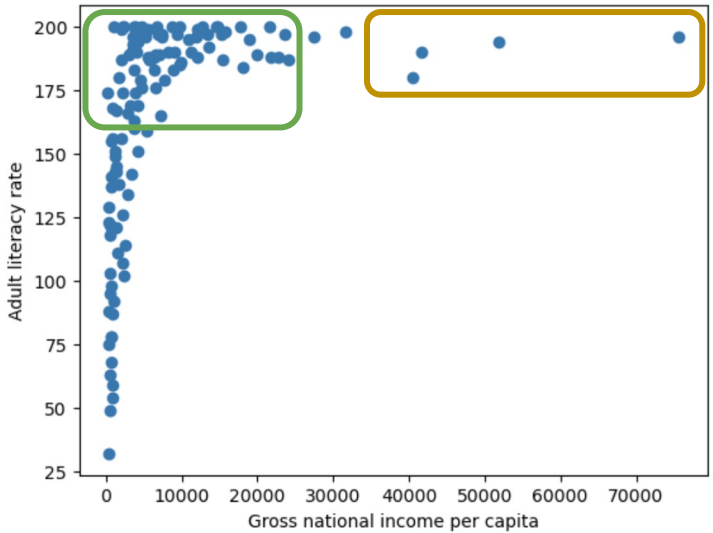


Why?

- Linear relationships are simple to interpret – we know how to work with slopes and intercepts to understand how two variables are related.
- Starting next week, we will start building linear models – these are more effective with linearized data.

Applying Transformations

What makes this plot non-linear?



1. A few large outlying x values are distorting the horizontal axis.
2. Many large y values are all clumped together, compressing the vertical axis.

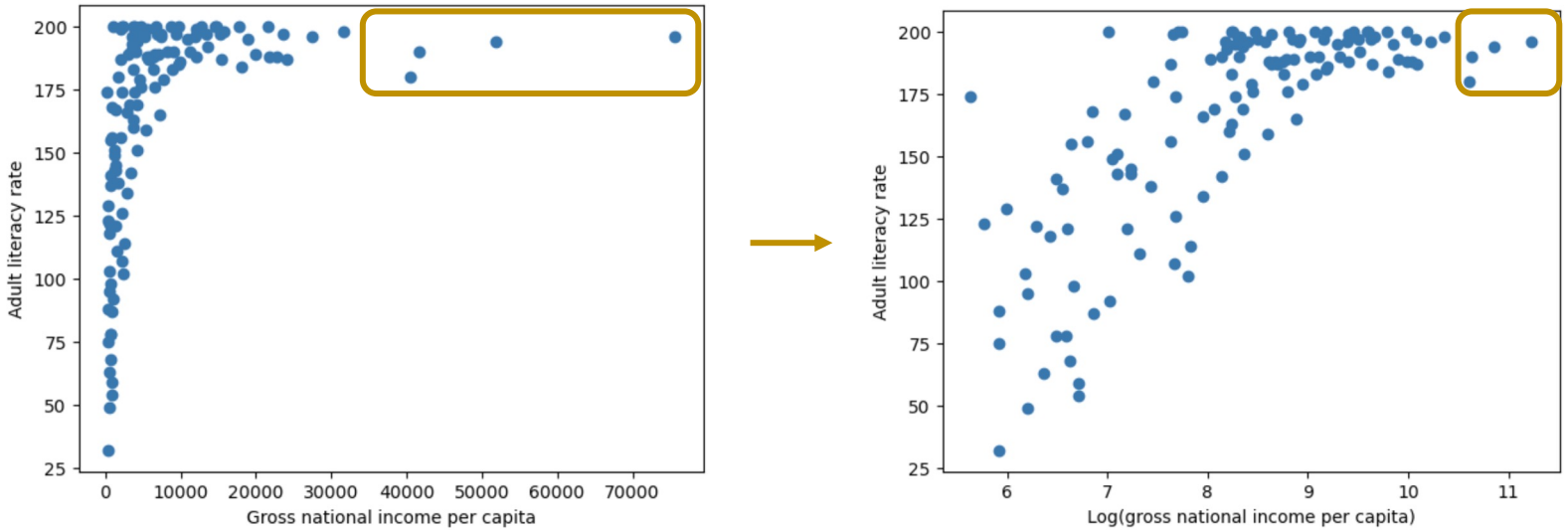
Applying Transformations

What makes this plot non-linear?

1. A few large outlying x values are distorting the horizontal axis.

Resolve by log-transforming the x data:

- Taking the log of a large number decreases its value significantly.
- Taking the log of a small number does not change its value as significantly.



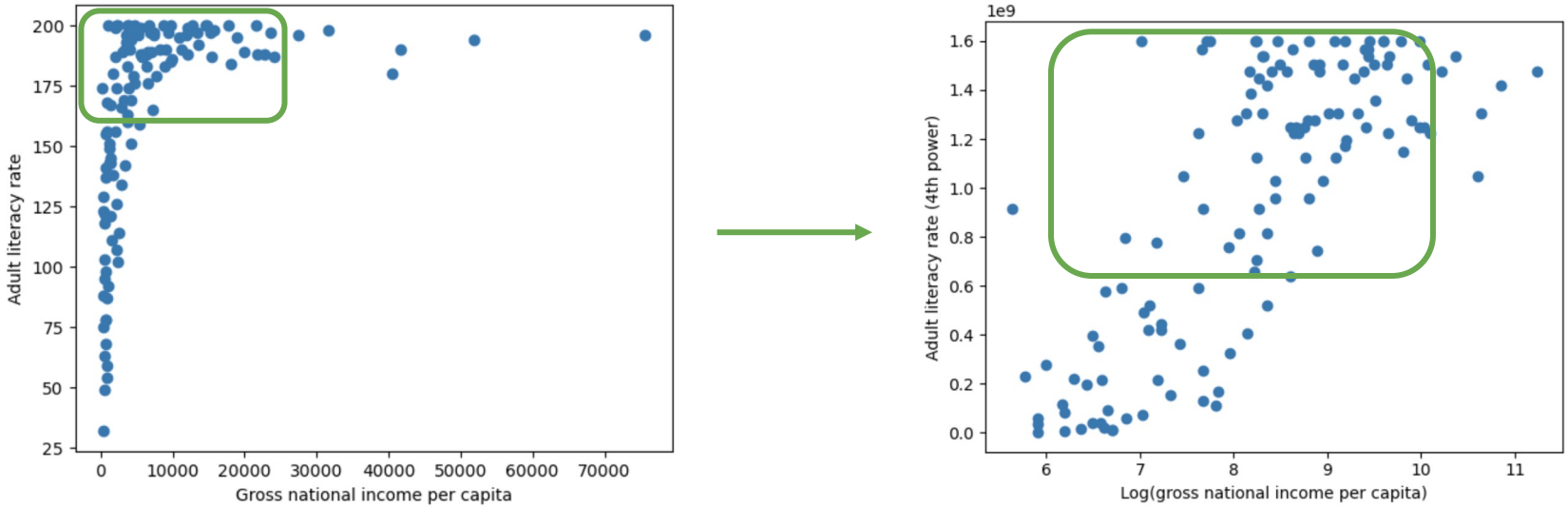
Applying Transformations

What makes this plot non-linear?

2. Many large y values are all clumped together, compressing the vertical axis.

Resolve by power-transforming the x data:

- Raising a large number to a power increases its value significantly.
- Raising a small number to a power does not change its value as significantly.

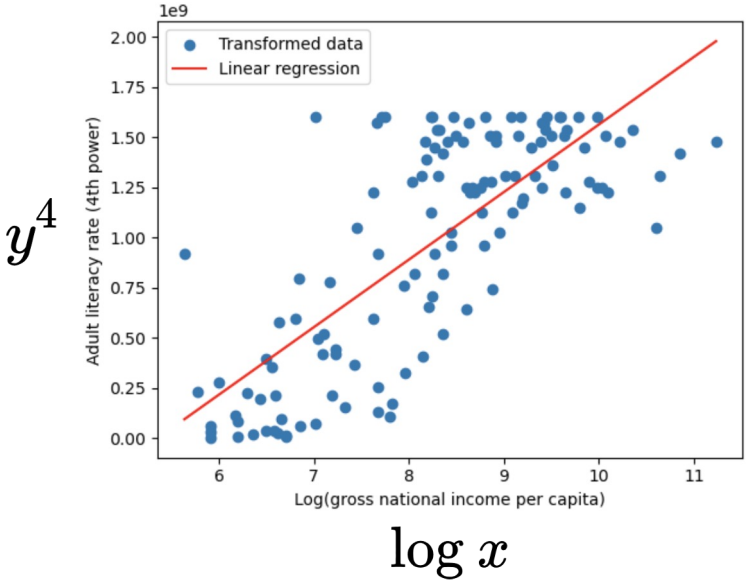


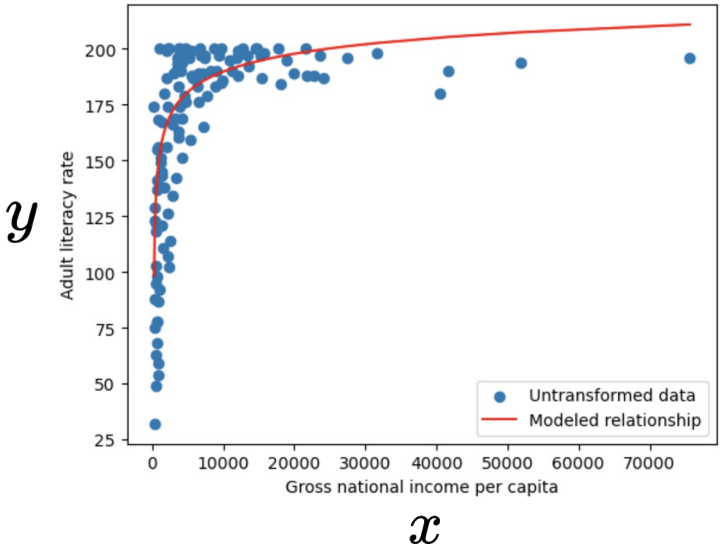
Interpreting Transformed Data

Now, we see a linear relationship between the transformed variables.

This tells us about the underlying relationship between the *original* x and y!

$$y^4 = m(\log x) + b \quad \longrightarrow \quad y = [m(\log x) + b]^{1/4}$$



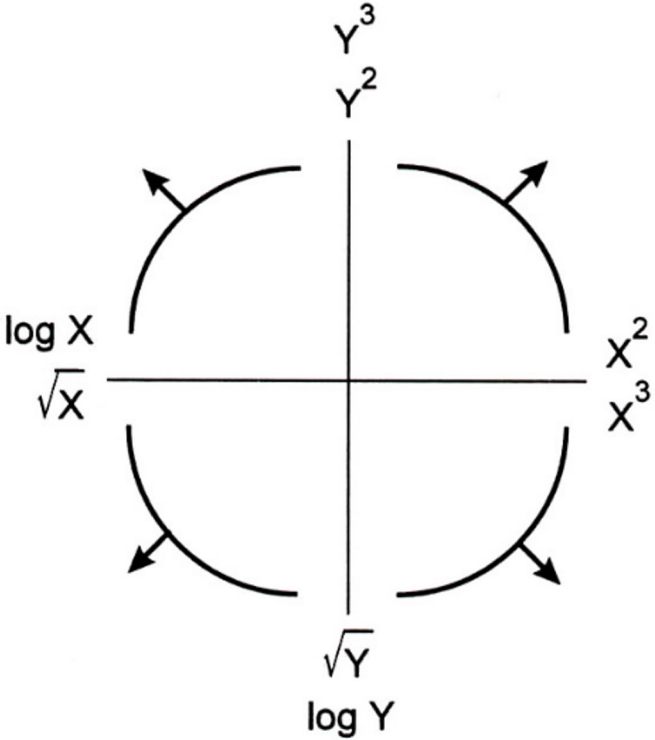


Tukey-Mosteller Bulge Diagram

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

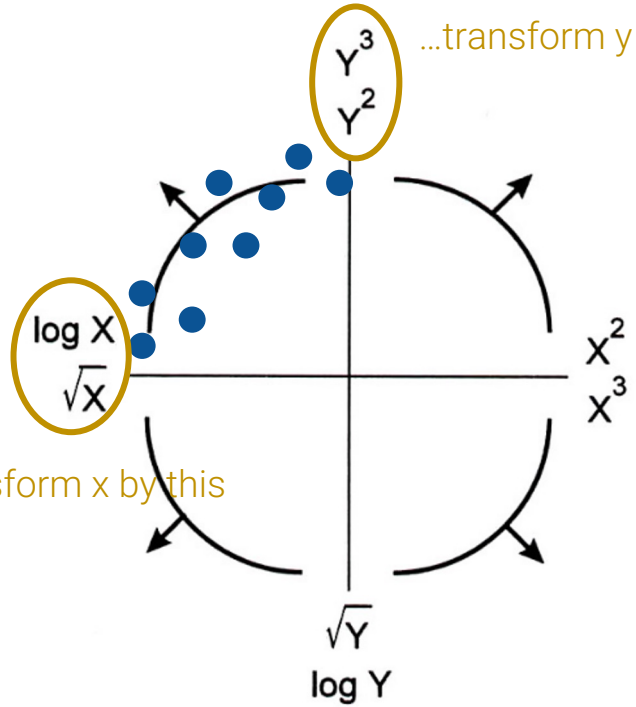
- A visual summary of the reasoning we just worked through.
- sqrt and log make a value "smaller".
- Raising to a value to a power makes it "bigger".
- There are multiple solutions. Some will fit better than others.

You should still understand the *logic* we just worked through to decide how to transform the data. The bulge diagram is just a summary.



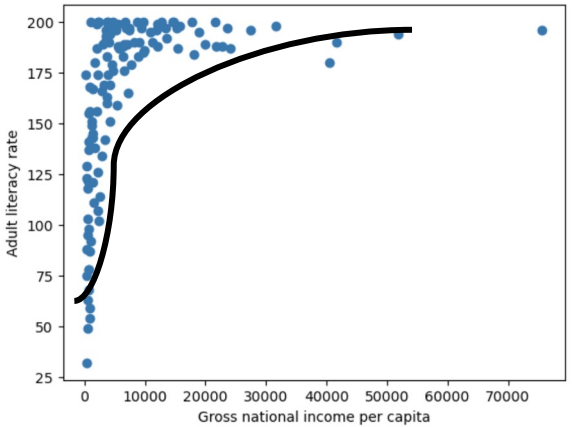
Tukey-Mosteller Bulge Diagram

If the data bulges like this…



Applying to the data from before:

Could have transformed y by y^2, y^3



Could have transformed x by log(x), sqrt(x)

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 – "chart junk."
    - It's fine to plot the same thing multiple ways, if it helps fit the narrative better.