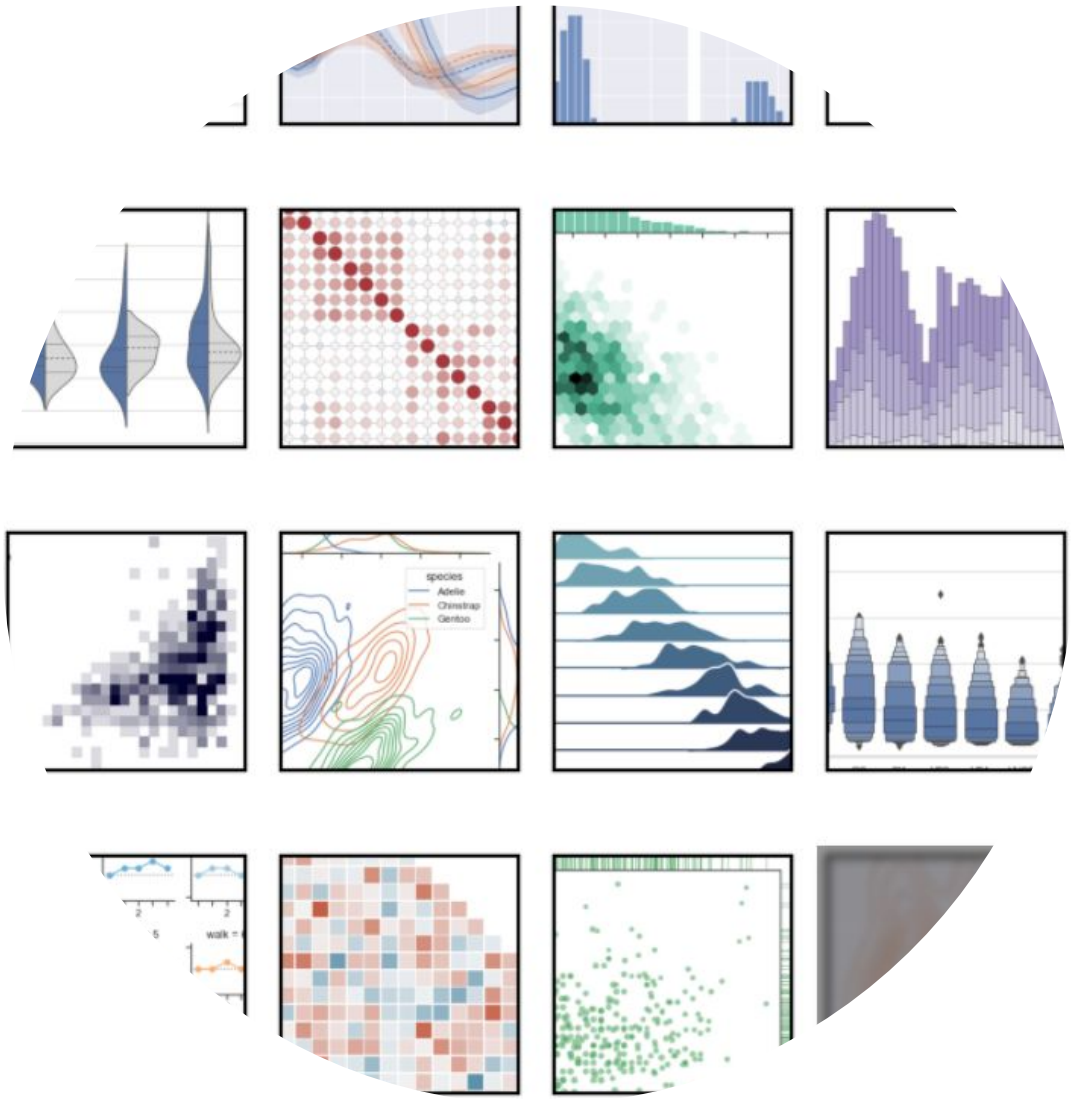
Data Science (COSE471) Spring 2021

Visualizations



Dept. of Computer Science and Engineering

Korea University

Announcements

- Homework #2 is out. (**Due: Apr 9**)
  - Exploratory Data Analysis
  - + Pandas and Regular Expressions

Outline

- Introduction
  - Encoding
  - Distribution
- Types of Visualizations
  - Bar plots
  - Rug plots, histograms, density curves
  - Describing quantitative distributions
  - Box plots and violin plots
- Principles of Visualization
  - Scale
  - Conditioning
  - Perception
  - Context
  - Smoothing
  - Transformation

What is visualization?

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

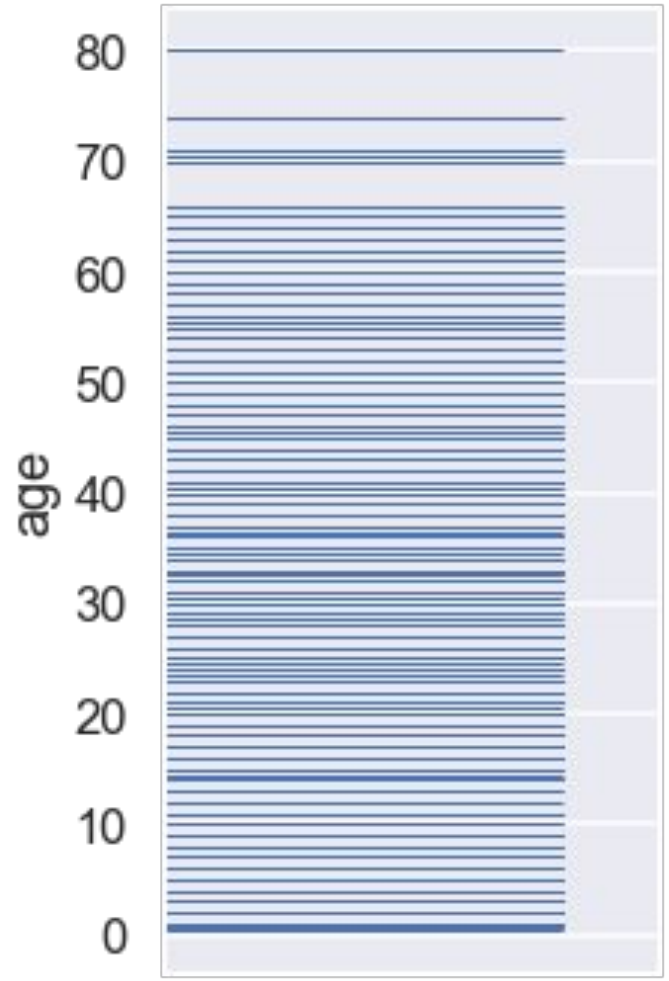
Card, Mackinlay, & Shneiderman 1999

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

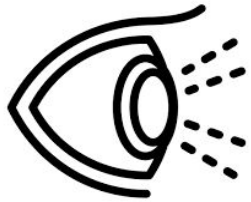
[Bertin 1967]

Take advantage of the human visual perception system

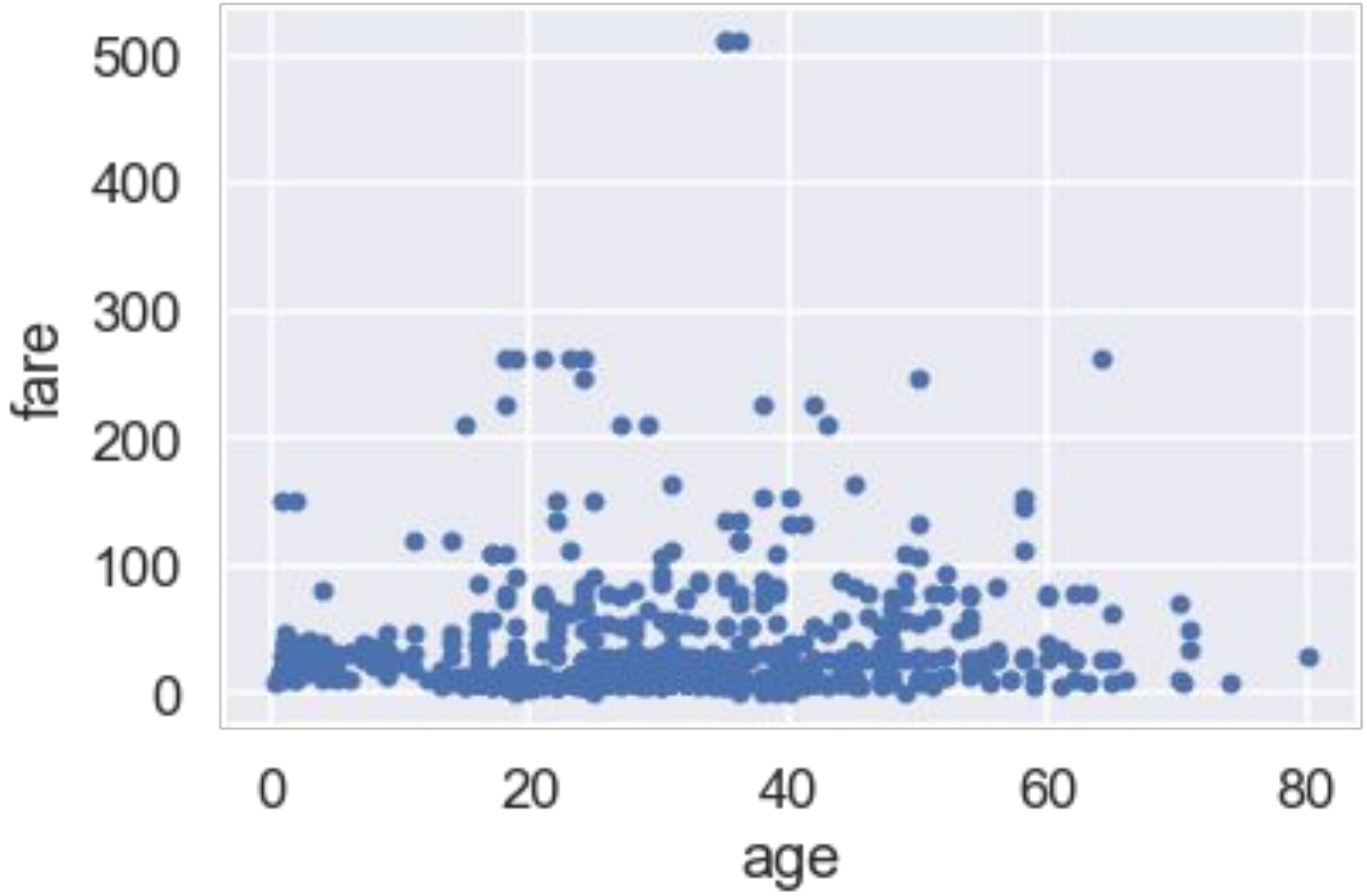
	age
0	22.0
1	38.0
2	26.0
...	...
888	NaN
889	26.0
890	32.0



Visualizations are for humans



“Looks like older people didn’t spend more than younger people.”



Visualize, then quantify!

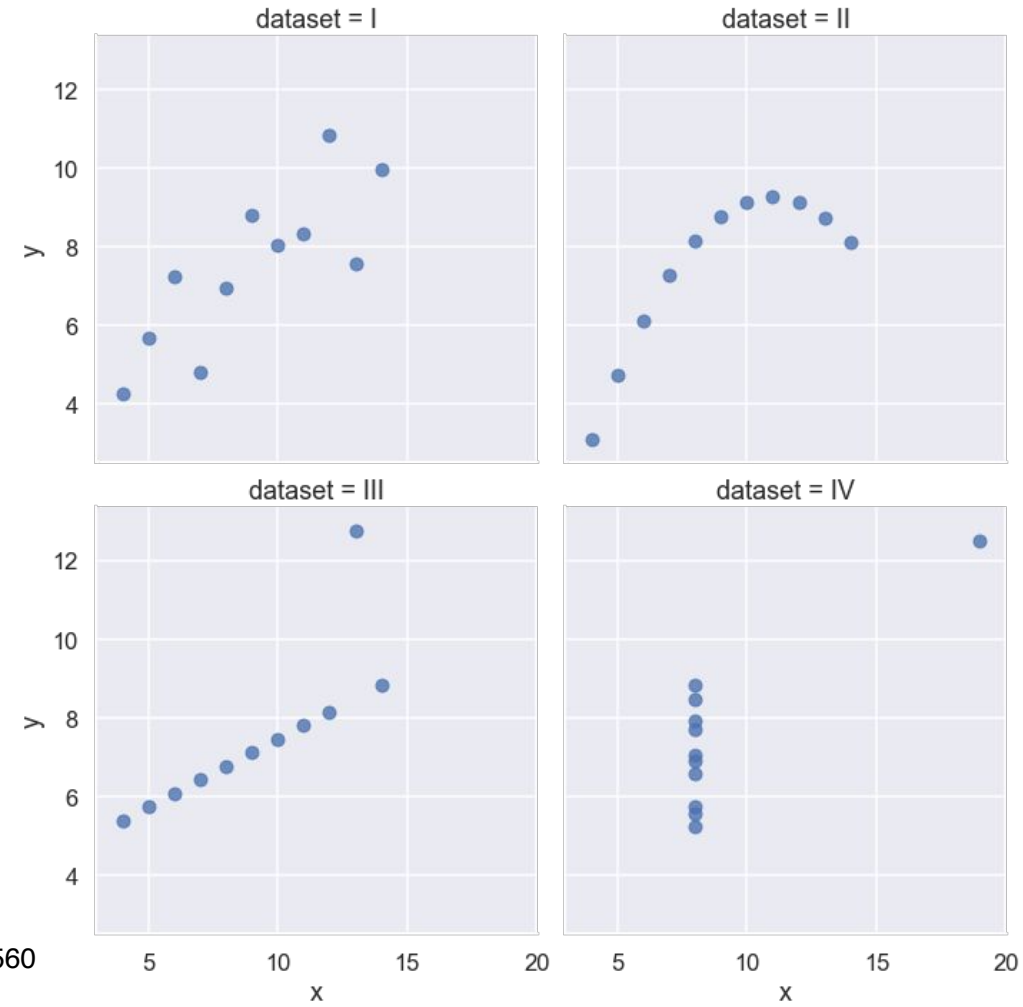
수량화하다

x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Each of these datasets has the same means, standard deviations, and correlation. As we will see in a few lectures, this means they have the same regression line.

https://john-analyst.medium.com/%ED%9A%8C%EA%B7%80-regression-%EB%9E%80-398c548e1560

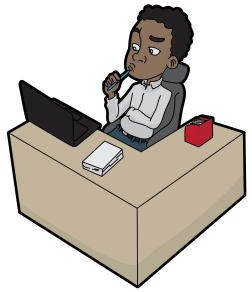
Visualization complements statistics.



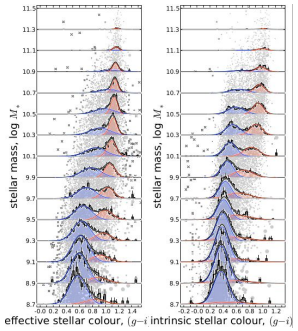
Anscombe's Quartet

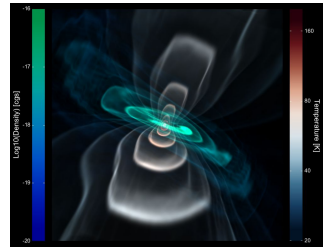
Goals of data visualization

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.



1. 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.





The John Hunter Excellence in Plotting Contest
https://jhepc.github.io/gallery.html

A constant tool across the lifecycle of data science

Why data visualization?

- One goal of data science is to inform human decisions.
  - Excellent plots **directly** address this goal.
  - Sometimes the most useful results from data analysis are the visualizations!
- Data visualization isn't as simple as calling `plot()`.
  - Many plots are possible, but only a few are useful!
  - Every visualization has tradeoffs.

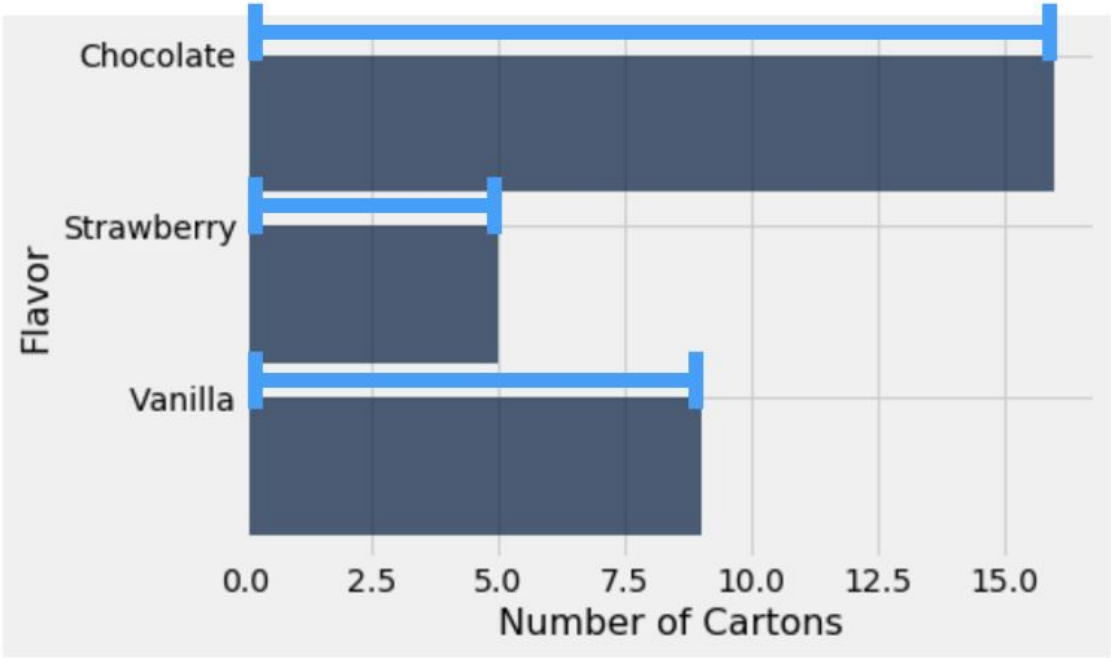
Roadmap:

- Establish when to use certain types of visualizations.
- Discuss various principles of visualization, along with kernel density estimation and transformation.

Encoding

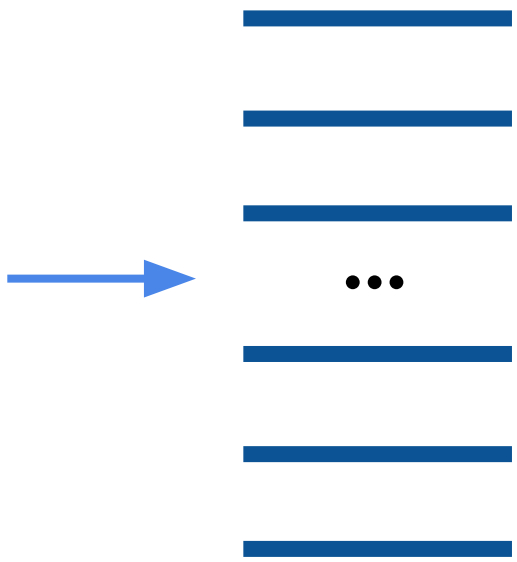
Encoding

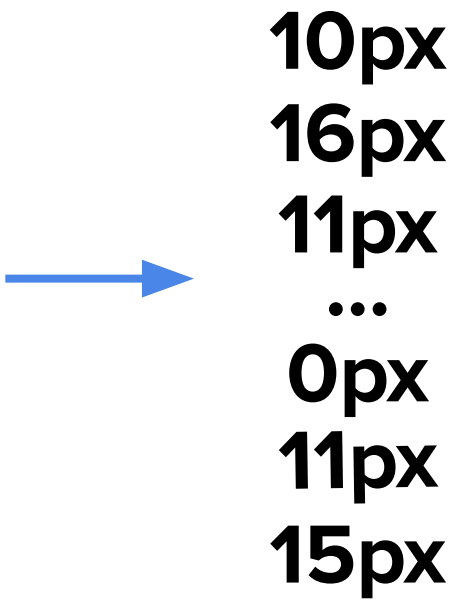
An **encoding** is a mapping from a variable to a visual element.

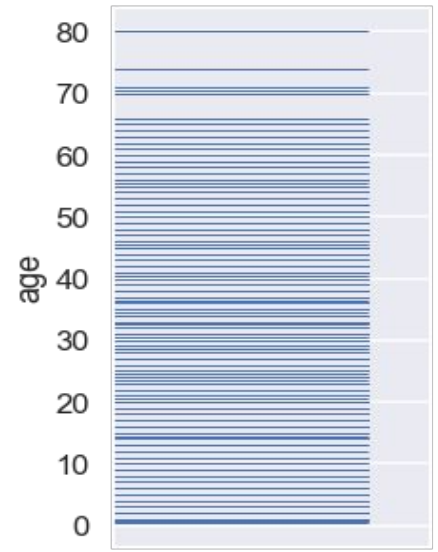


Encoding

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







Mark

(Represents a datum)

datum -> 자료

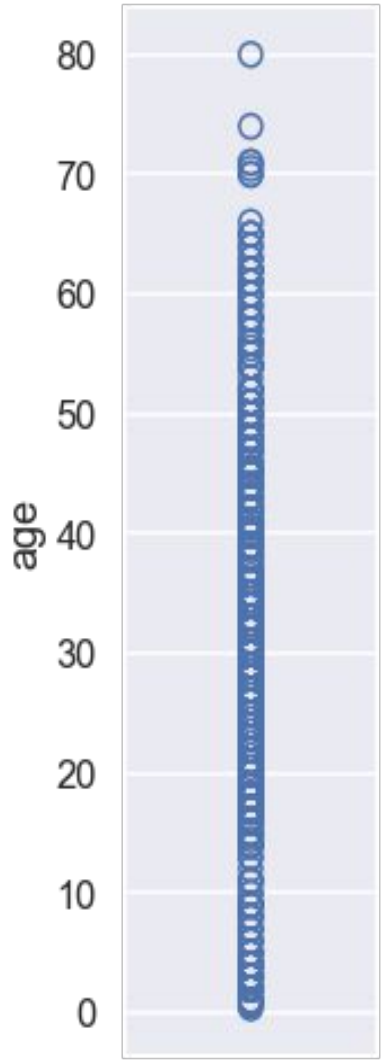
Encoding

(Maps datum to visual position)

Encoding

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

10px
16px
11px
...
0px
11px
15px



Mark

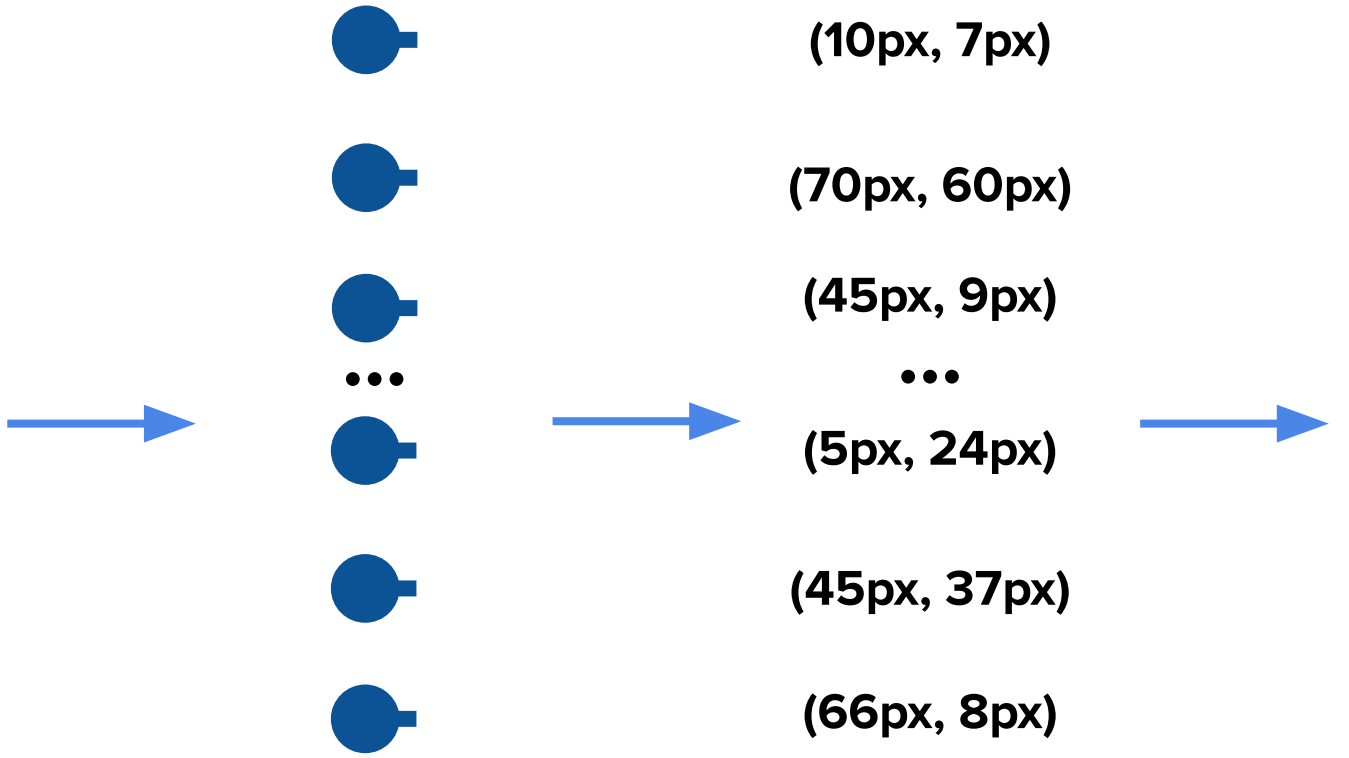
(Represents a datum)

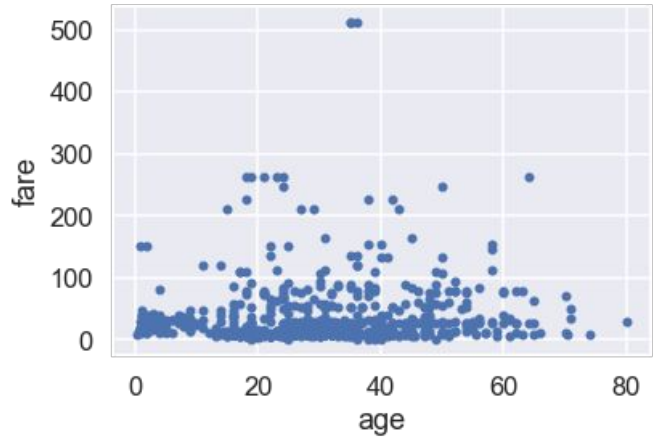
Encoding

(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



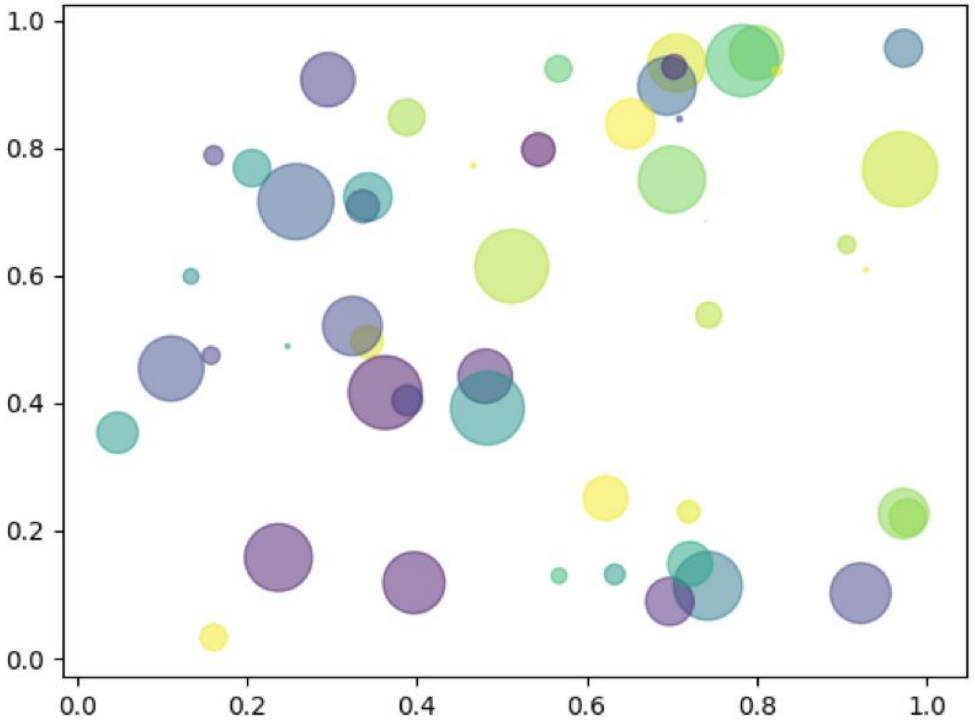


Mark

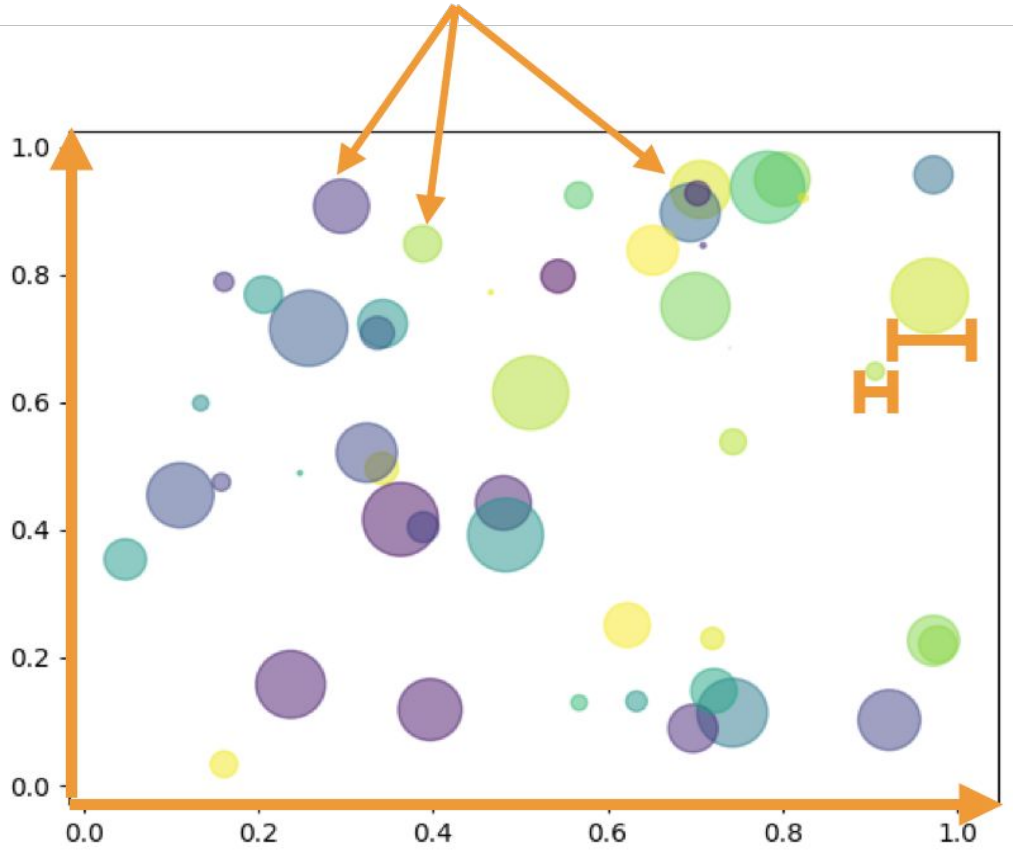
(Represents a datum)

Encoding

(Maps datum to visual position)



How many variables are we encoding here?



How many variables are we encoding here?

Answer: 4.

- x
- y
- area
- color

What's wrong?

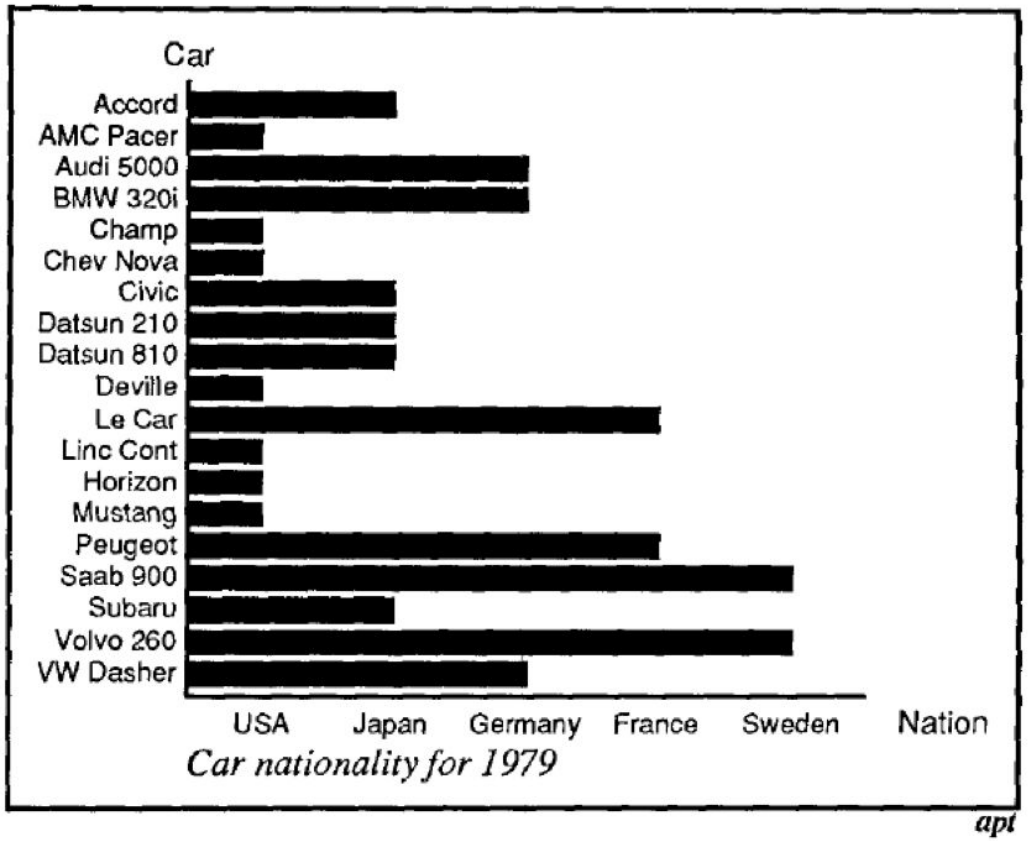


Fig. 11. Incorrect use of a bar chart for the *Nation* relation. The lengths of the bars suggest an ordering on the vertical axis, as if the USA cars were longer or better than the other cars, which is not true for the *Nation* relation.

Not all encoding channels are exchangeable.

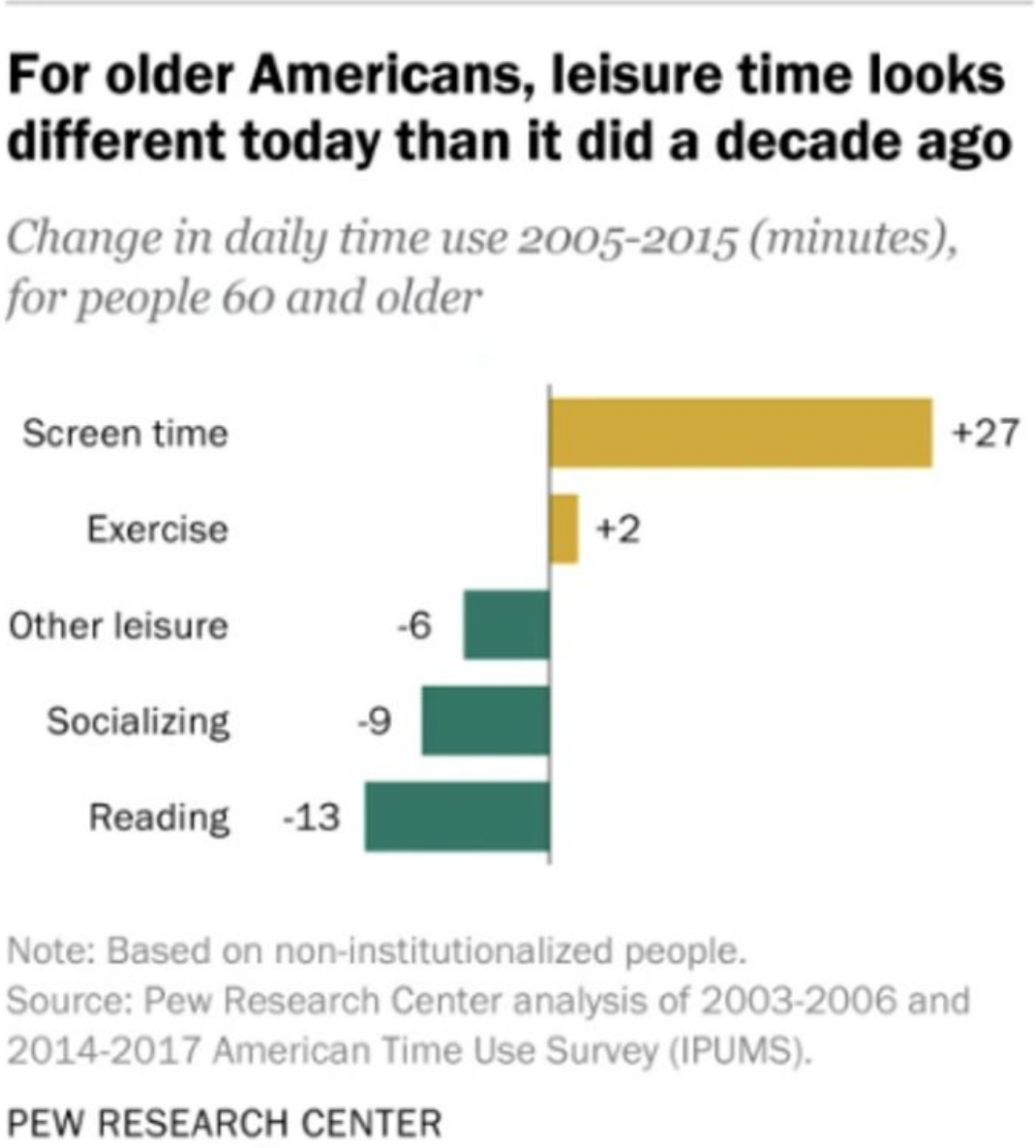
This is quite an extreme example, but watch out for encoding mismatches!

Distributions

What is a distribution?

- A **distribution** describes the frequency at which values of a variable occur.
- All values must be accounted for once, and only once.
- The total frequencies must add up to 100%, or to the number of values that we're observing.

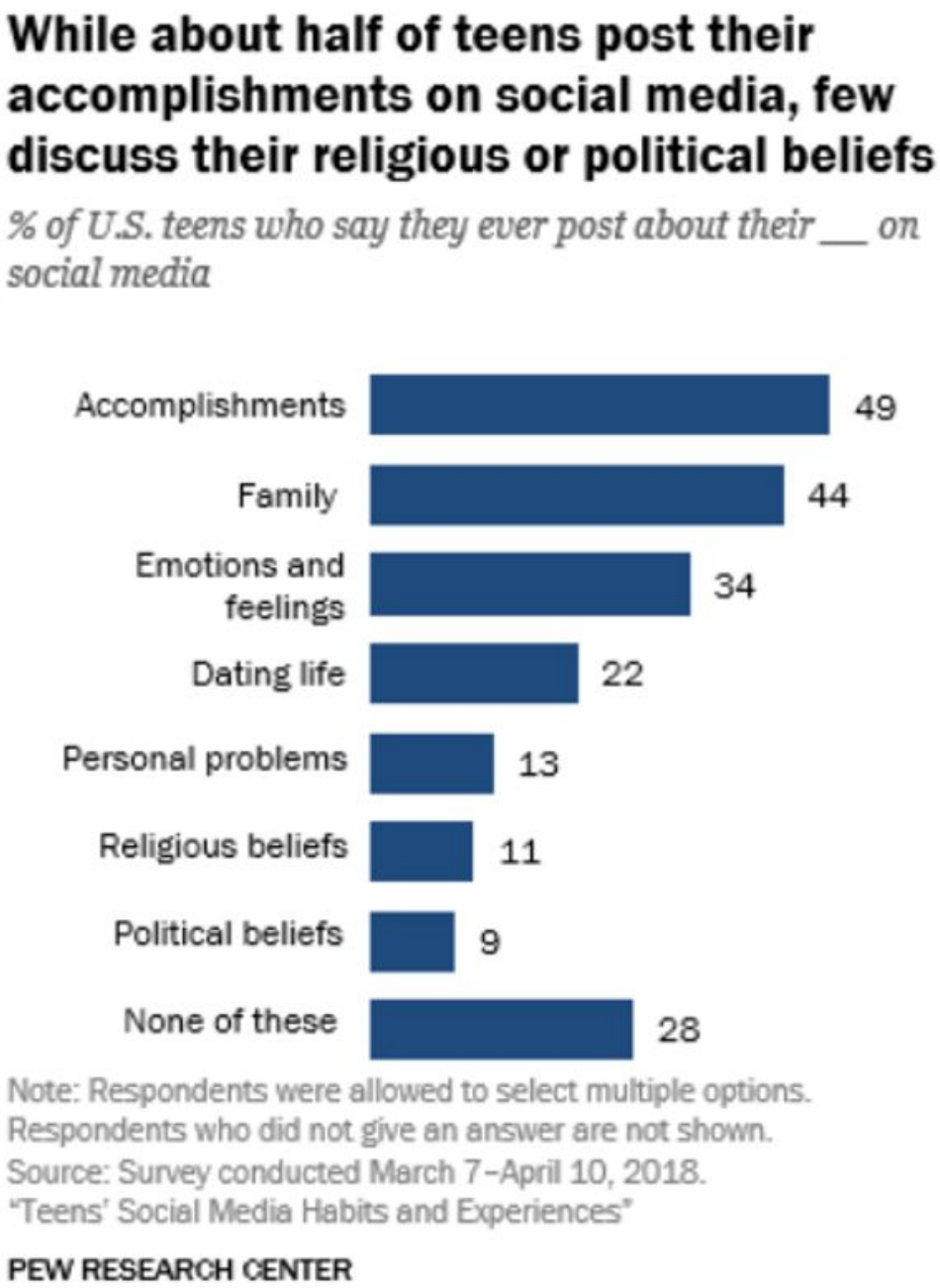
Let's look at some examples.



Does this chart show a distribution?

No.

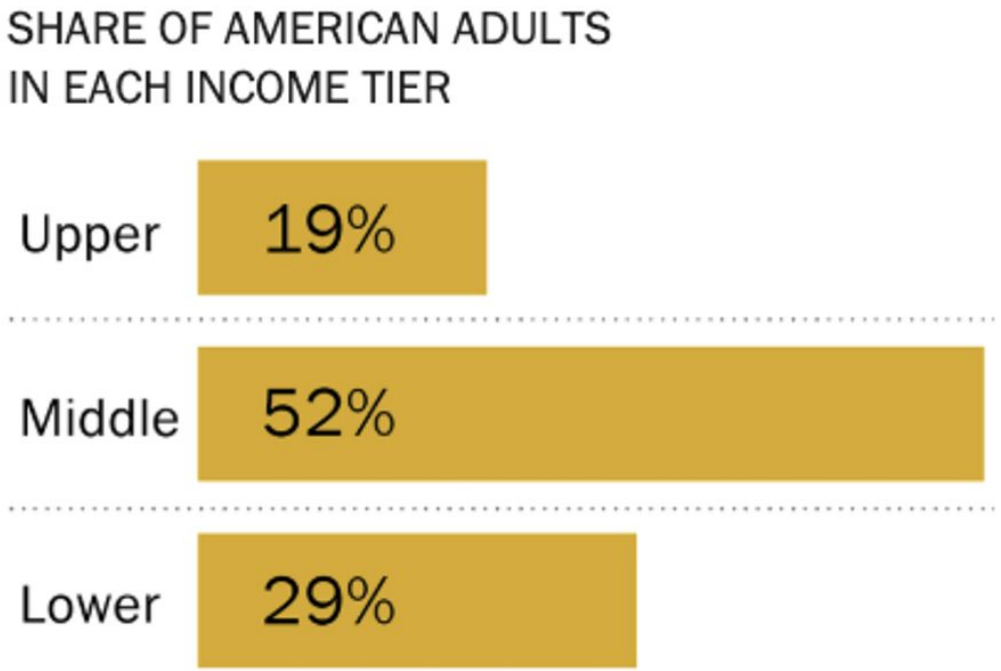
- Individuals can be in more than one category.
- The numbers (and bar lengths) correspond to “time”, not the proportion or number of individuals in the category.



Does this chart show a distribution?

No.

- The chart does show percents of individuals in different categories!
- But, this is not a distribution because individuals can be in more than one category (see the fine print).



Does this chart show a distribution?

Yes!

- This chart shows the distribution of the qualitative ordinal variable “income tier.”
- Each individual is in exactly one category.
- The values we see are the proportions of individuals in that category.
- Everyone is represented, as the total percentage is 100%.

Bar plots

Bar plots

- Bar plots are the most common way of displaying the distribution of a qualitative (**categorical**) variable.
  - For example, the proportion of adults in the upper, middle, and lower classes.

- They are also used to display a numerical variable that has been measured on individuals in different categories.
  - For example, the average GPAs of students in several majors.
  - Not a distribution! But bar plots still make sense.

- Lengths encode values.
  - Widths encode **nothing!**
  - Color could indicate a sub-category (but not necessarily).

Example dataset

We will be using the baby weights dataset for most of our plots today. Here is what that looks like.

```
1 births = pd.read_csv('baby.csv')
```

```
1 births.head()
```

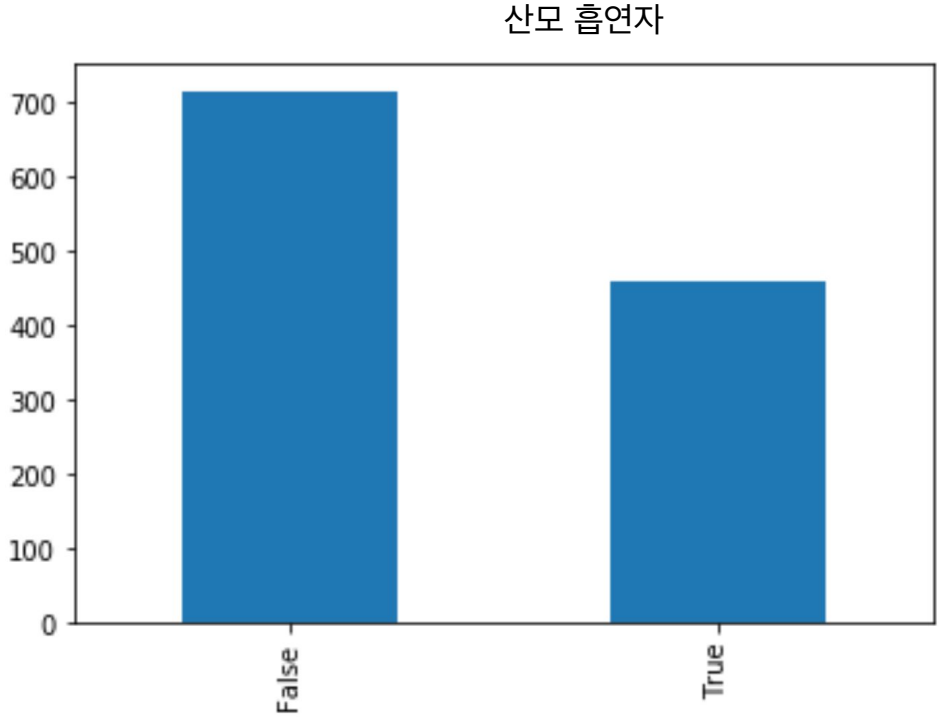
	Birth Weight	Gestational Days	Maternal Age	Maternal Height	Maternal Pregnancy Weight	Maternal Smoker
0	120	284	27	62	100	False
1	113	282	33	64	135	False
2	128	279	28	64	115	True
3	108	282	23	67	125	True
4	136	286	25	62	93	False

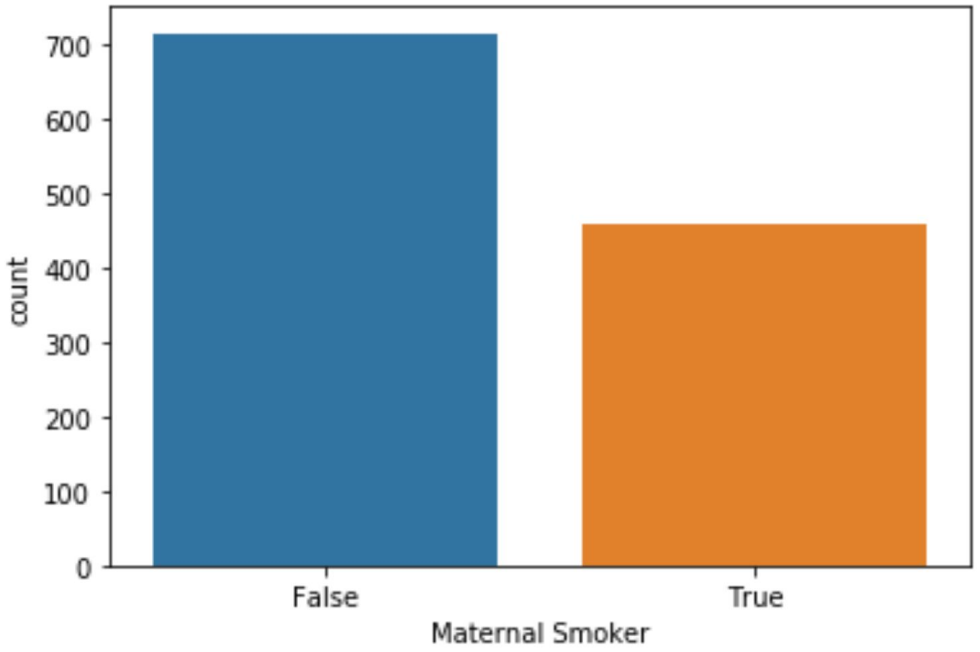
```
1 births.shape
```

```
(1174, 6)
```

Bar plots

Suppose **`births['Maternal Smoker']`** is a series containing True and False. Then:



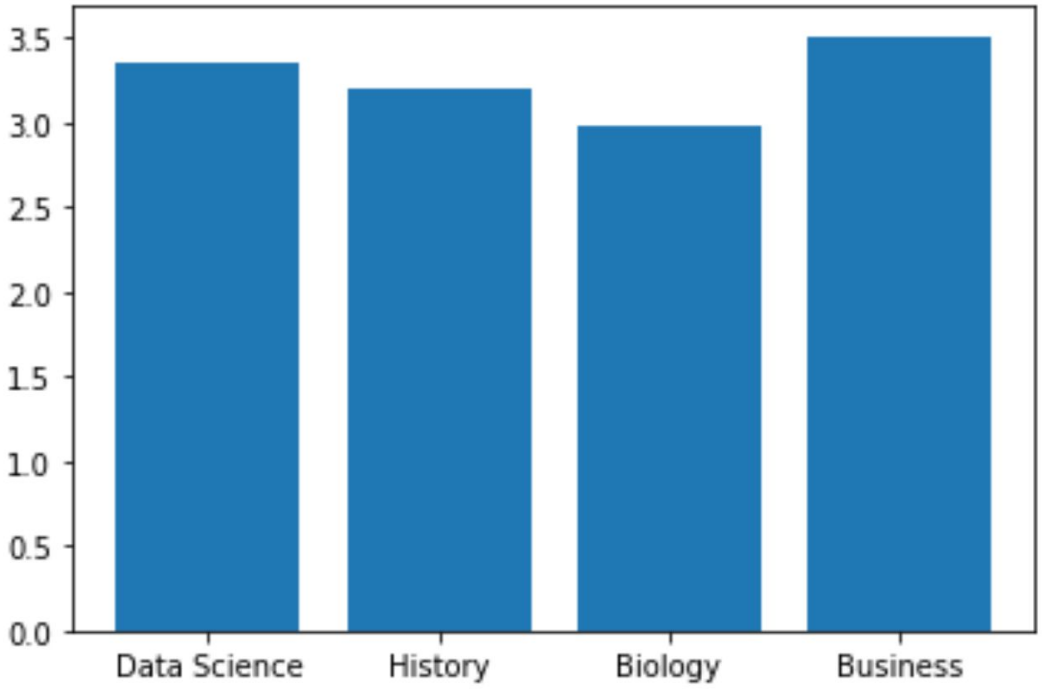



```
births['Maternal Smoker']
.value_counts().plot(kind = 'bar');
```

```
sns.countplot(births['Maternal Smoker'])
```

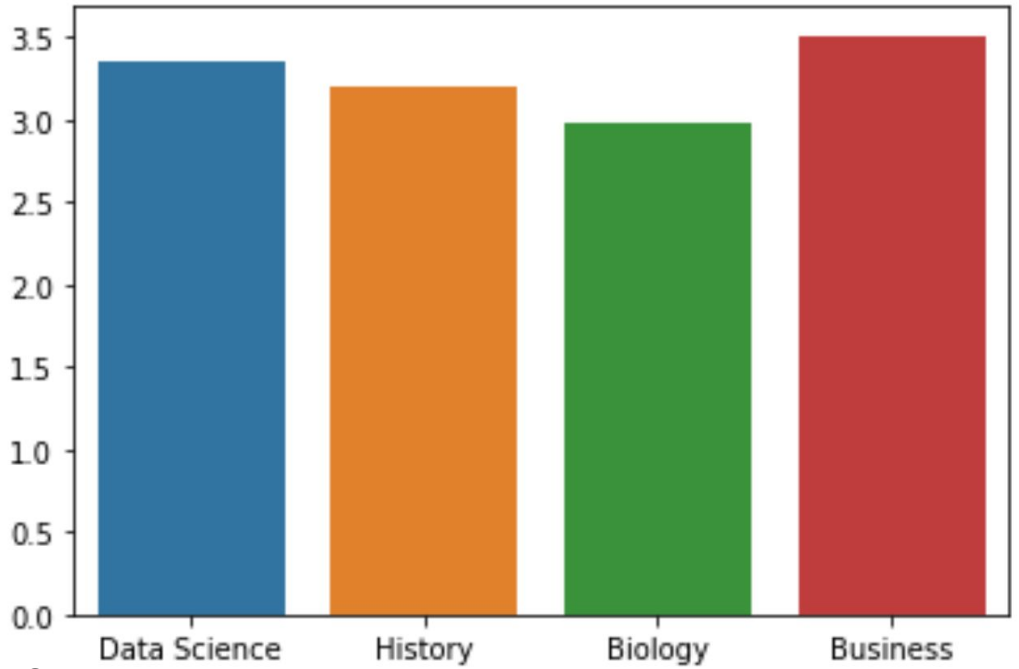
Bar plots

Suppose we have a list of majors and a list of gpas corresponding to those majors. Then:



`plt.bar(majors, gpas)`

To make horizontal: `plt.barh(majors, gpas)`



Color could indicate a sub-category
(여기선 서브 카테고리가 아니기 때문)

`sns.barplot(majors, gpas)`

Note: Here, color is meaningless.

Three ways to plot

- matplotlib (**`plt`**)
  - The underlying plotting library powering all three of these.
- pandas **`.plot()`**
  - Knows how to make some default plots for you!
- seaborn (**`sns`**)
  - Allows us to create sophisticated visualizations quickly.
  - Not just a colorful version of matplotlib!
- There are several other ways, but these are what we'll focus on.

The rich Python plotting ecosystem - this is not all!

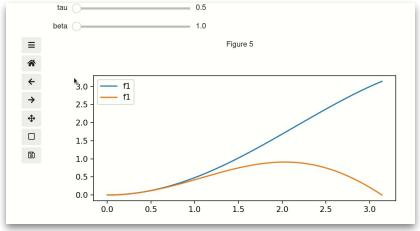
bokeh

Altair

seaborn

cartopy

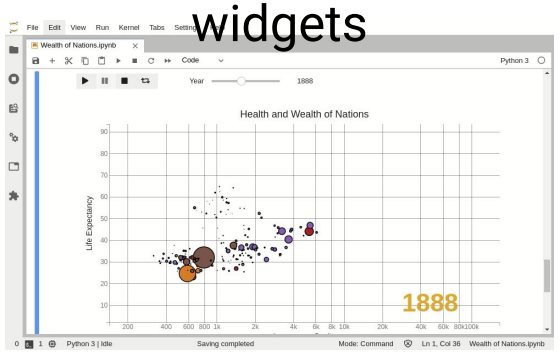
mpl_interactions



plotly

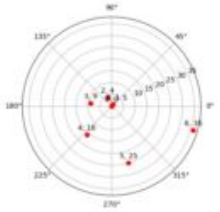
matplotlib

bqplot - Jupyter widgets

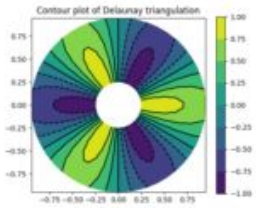


matplotlib

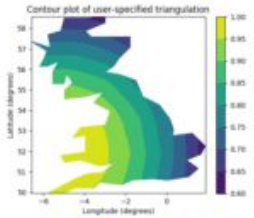
https://matplotlib.org/gallery.html



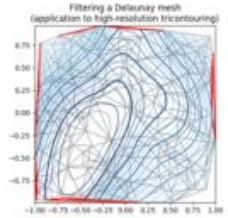
transoffset



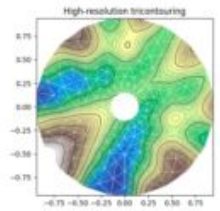
tricontour_demo



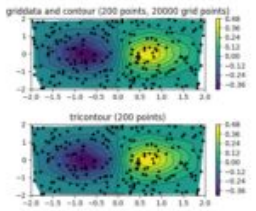
tricontour_demo



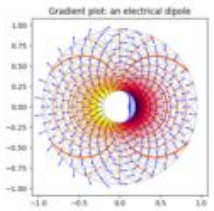
tricontour_smooth_delaunay



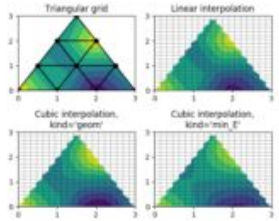
tricontour_smooth_user



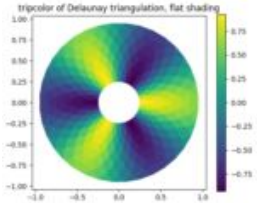
tricontour_vs_griddata



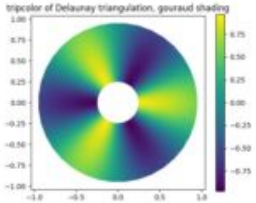
trigradient_demo



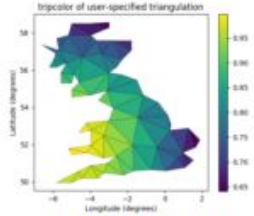
triinterp_demo



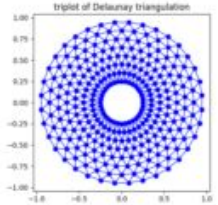
tripcolor_demo



tripcolor_demo



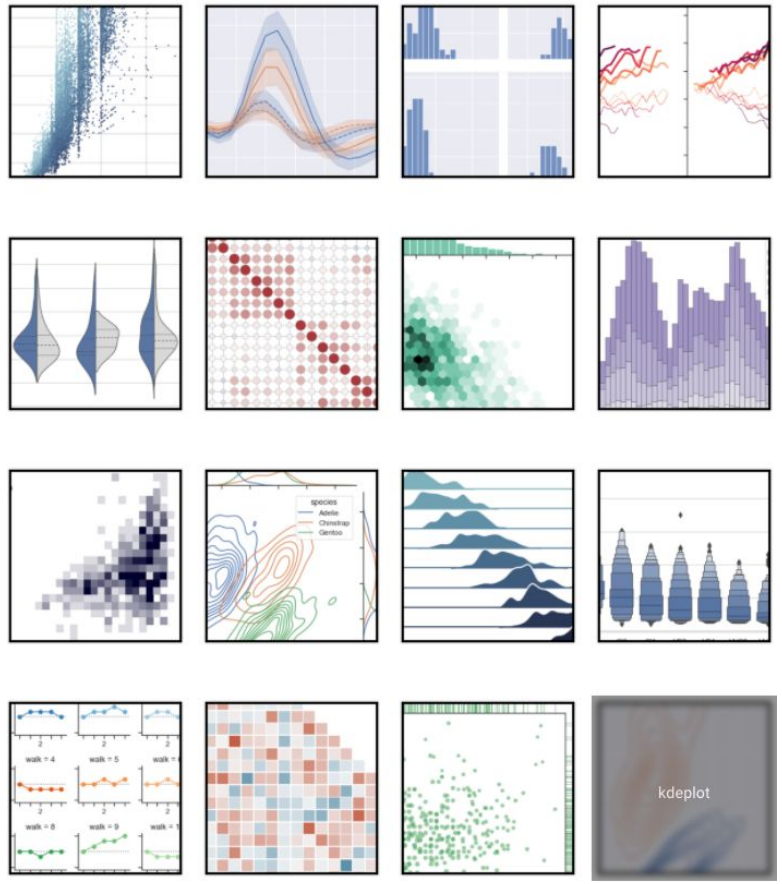
tripcolor_demo



triplot_demo

seaborn

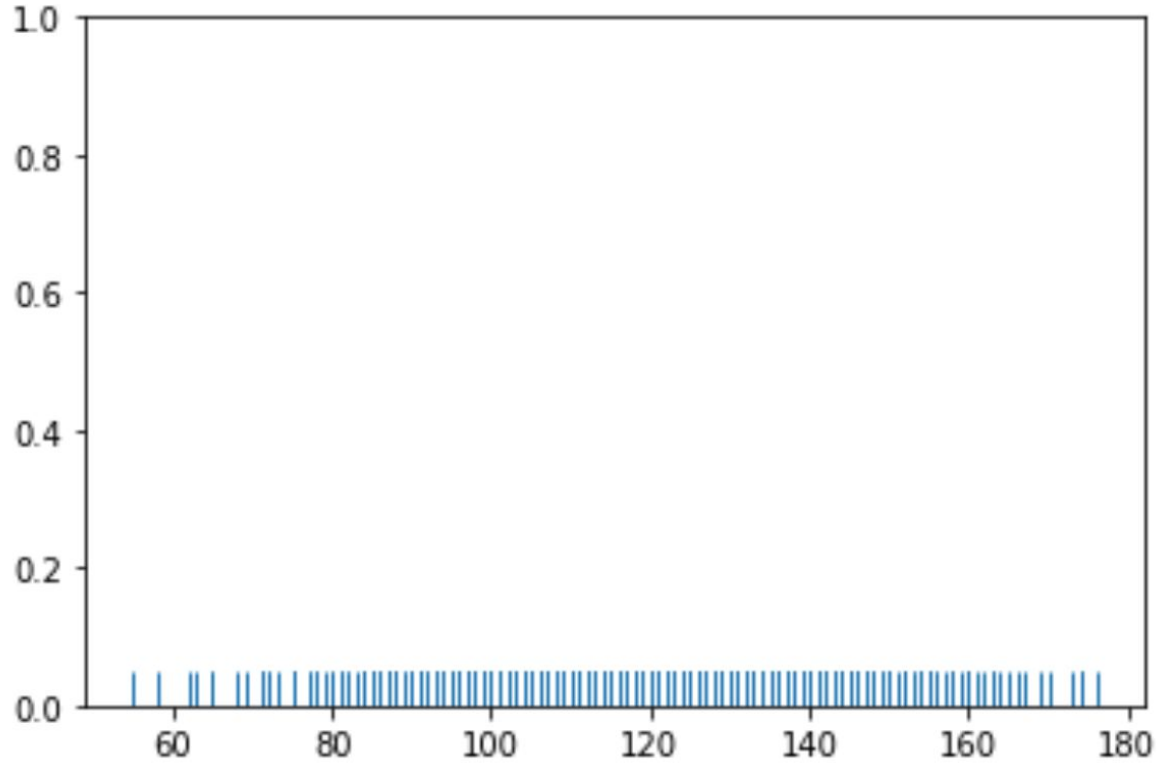
https://seaborn.pydata.org/examples/index.html



Rug plots, histograms, density curves

Rug plot

- Rug plots are used to show the distribution of a single quantitative (**numerical**) variable.
- They show us each and every value!
- Issues with rug plots:
  - Too much detail.
  - Hard to see the bigger picture.
  - **Overplotting**.
    - How many birth weights were at 120?
    - Can't tell – they're all on top of each other.




```
sns.rugplot(bweights)
```

Histograms

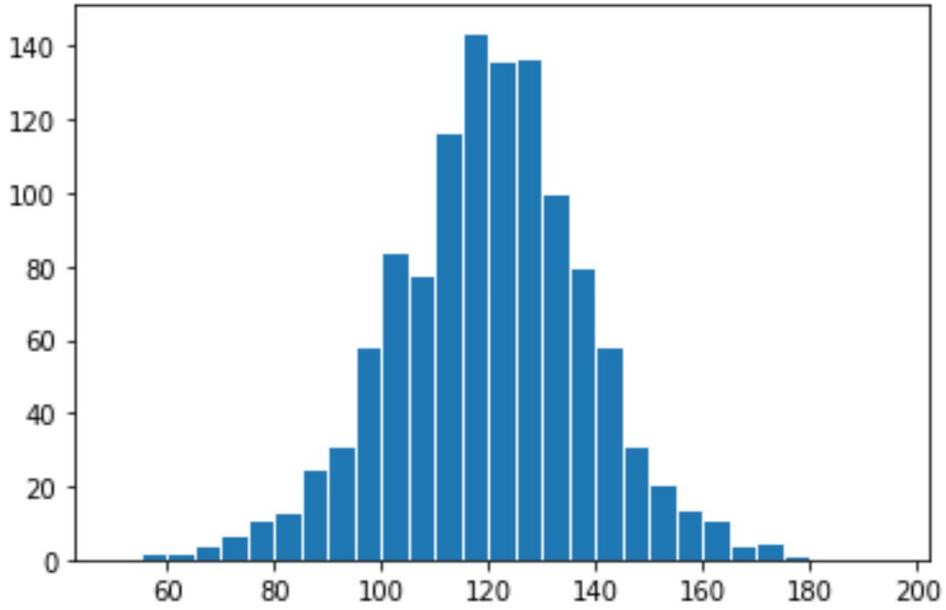
연속된 숫자를 표현할 때 사용한다.

- Histograms can be thought of as a smoothed version of a rug plot.
  - Lose granularity, but gain interpretability.
- Horizontal axis: the number line, divided into **bins**.
- **Areas represent proportions**!
  - Total area = 1 (or 100%).
- Units of height: proportion per unit on the x-axis.
  - Can be seen by dividing the above equation by “width of bin”.

$$\text{proportion in bin} = \text{width of bin} \cdot \text{height of bar}$$

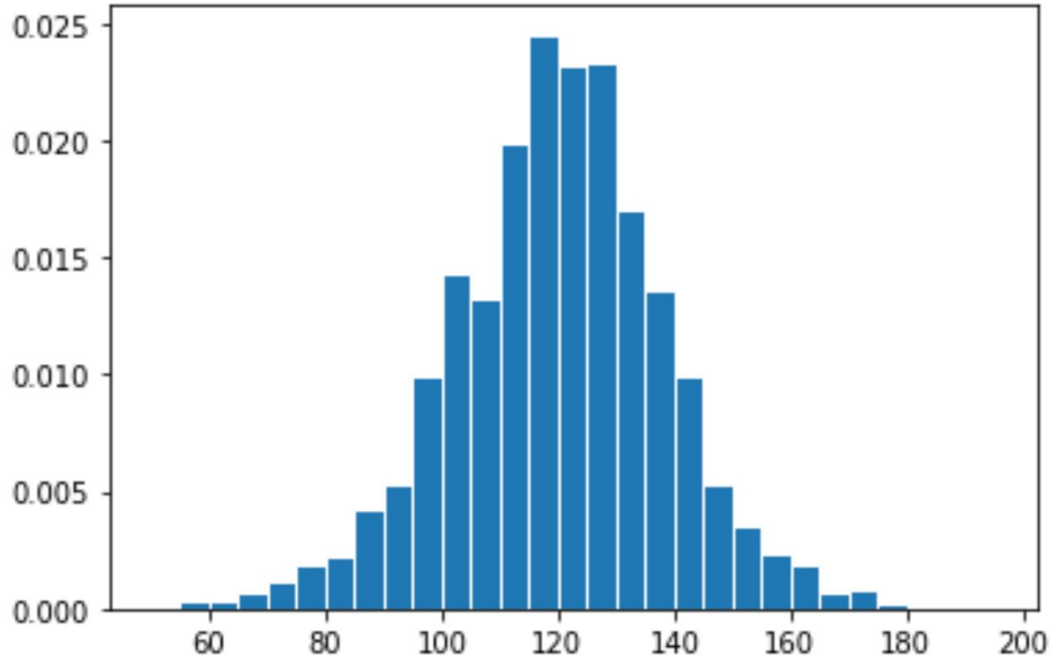
Histograms

By default, **`matplotlib`** histograms show *counts* on the y-axis, *not proportions* per unit.



`plt.hist(bweights, bins=bw_bins, ec='w')`
where `bw_bins = range(50, 200, 5)`

We use the optional **`density`** parameter to fix the y-axis. After doing this, the total area sums to 1.

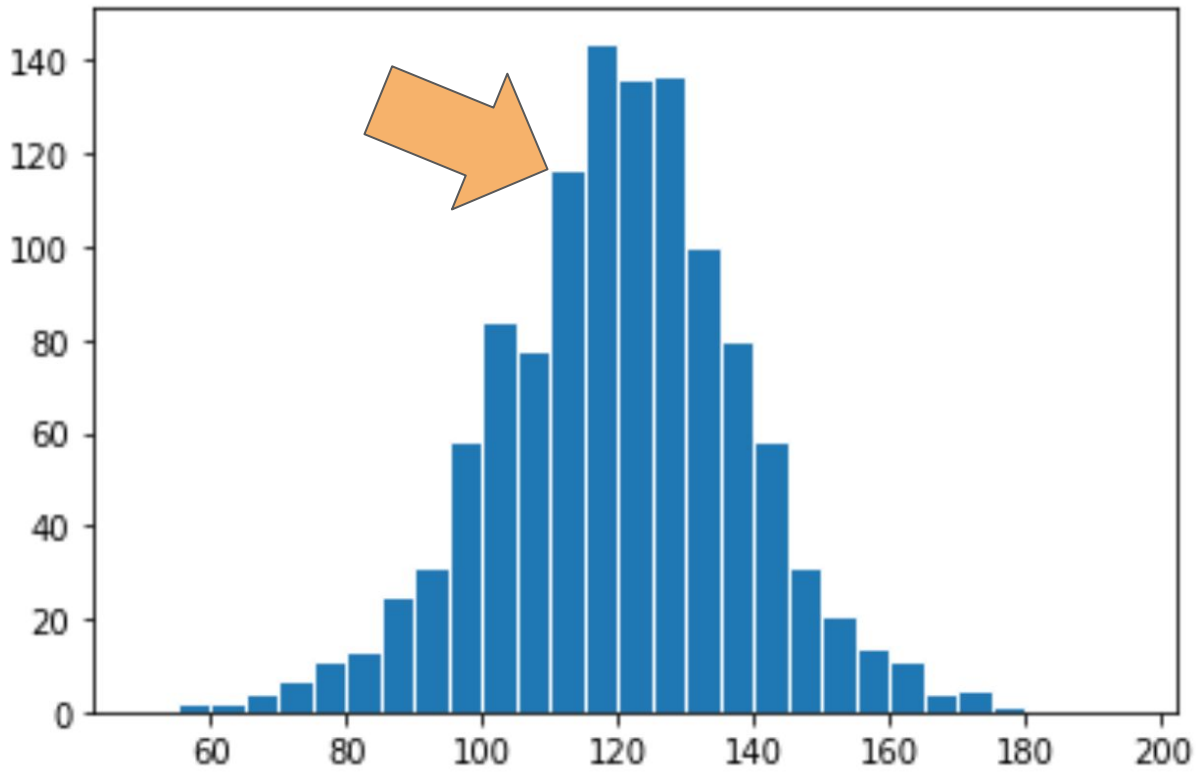


`plt.hist(bweights, density=True, bins=bw_bins, ec='w')`

Example calculation

Approximately ~120 babies were born with a weight between 110 and 115.

There are 1174 observations total.



Example calculation

There are 1174 observations total.

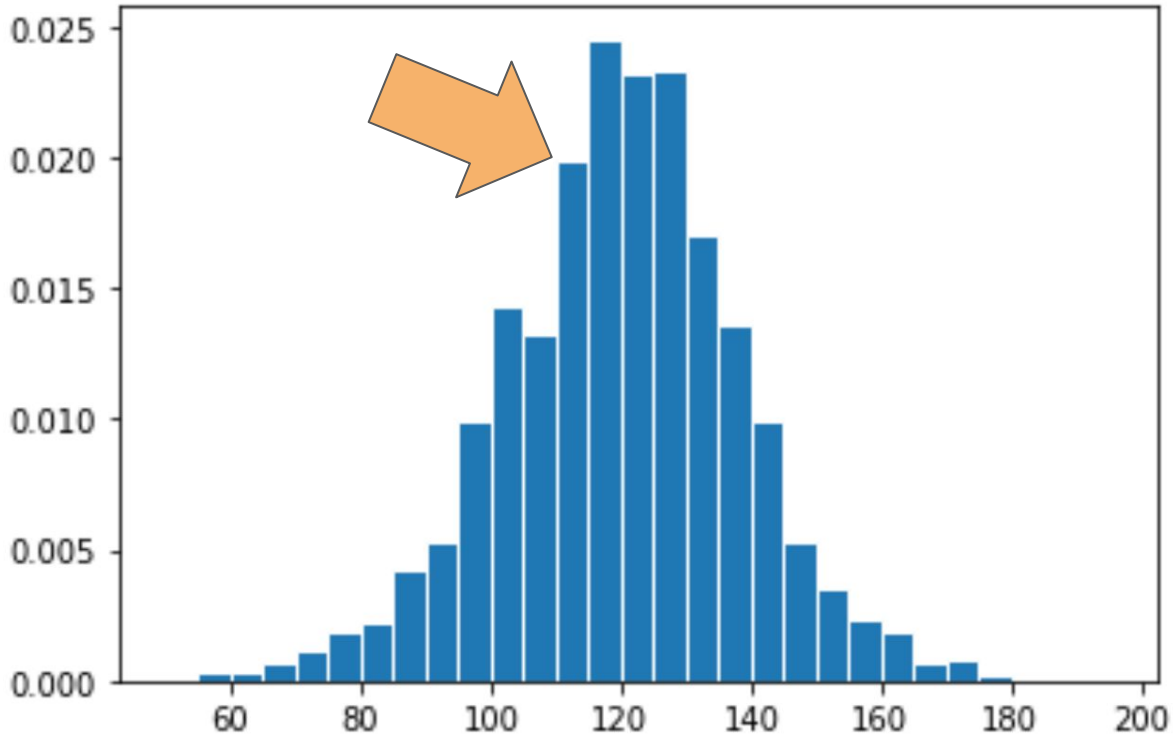
Width of bin [110, 115): 5

Height of bar [110, 115): 0.02

Proportion in bin = 5 * 0.02 = 0.1

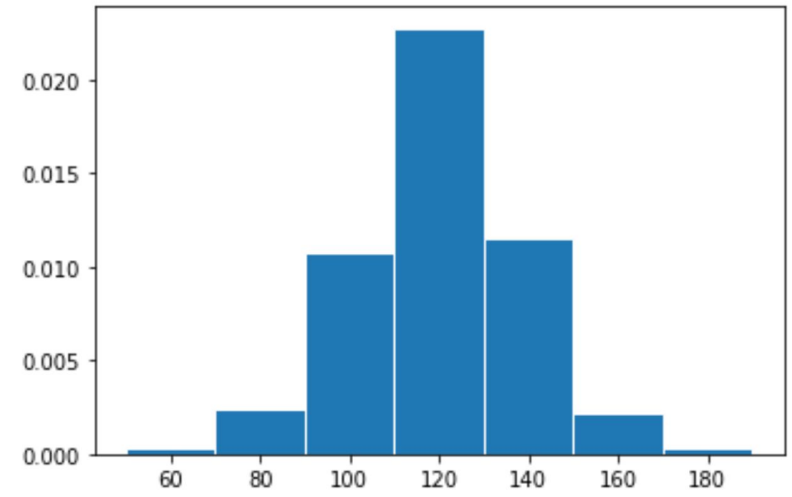
Number in bin = 0.1 * 1174 = **117.4**

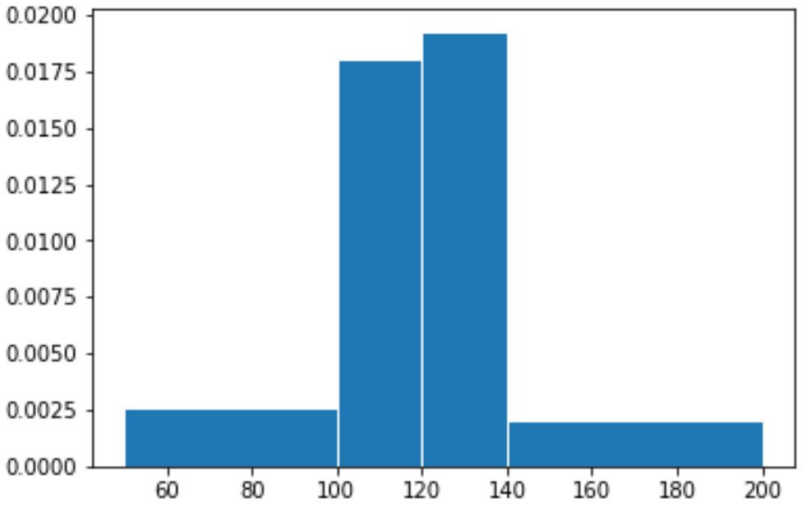
This is roughly the number we got before (120)!



정규분포(Gaussian Distribution)

Histograms





Beware of drawing strong conclusions from the looks of a histogram. **The number of bins influences its appearance!**

bins : 히스토그램의 한 구간

The Freedman-Diaconis rule:

$$\text{Bin width} = 2\,\frac{\text{IQR}(x)}{\sqrt[3]{n}}$$

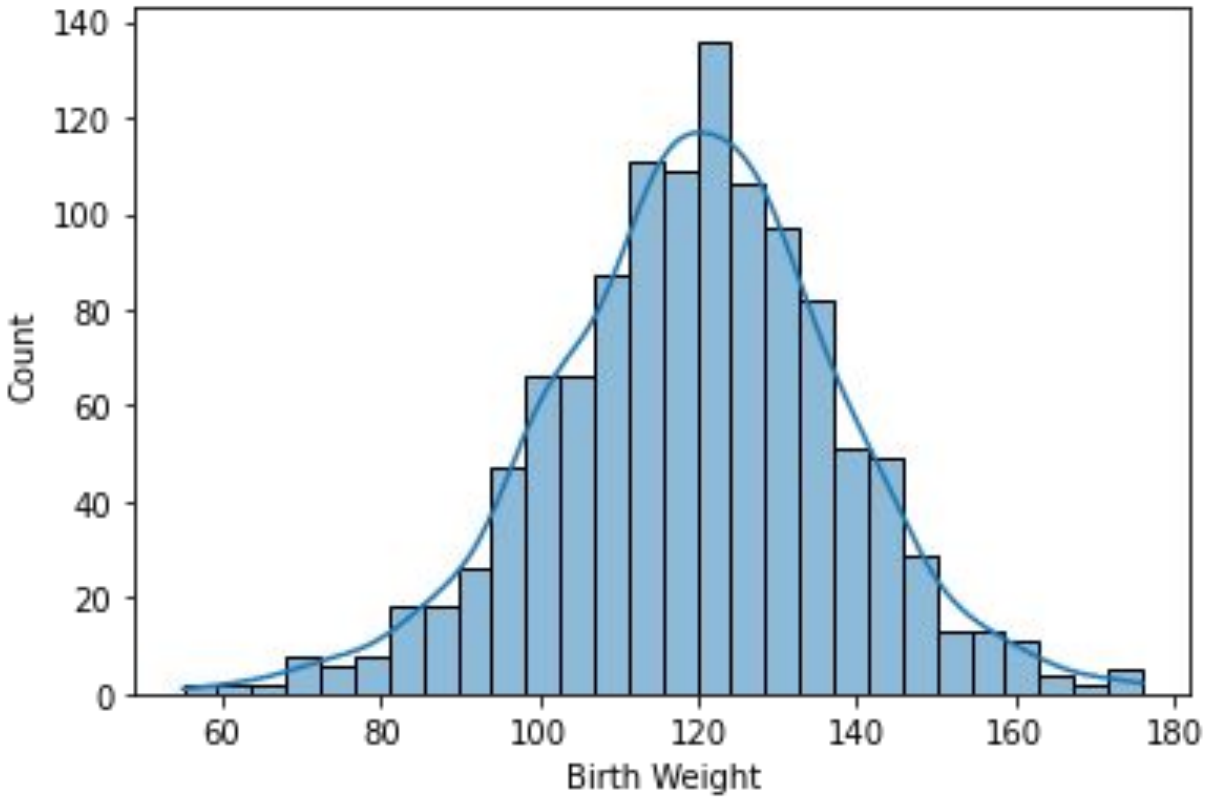
Freedman–Diaconis 규칙을 사용하여 히스토그램에 사용할 빈의 너비를 선택

```
plt.hist(bweights, bins = np.arange(50, 200, 20),
density=True, ec='w')
```

Bins don’t need to have the same width! When they don’t, it’s especially crucial to think of proportions as areas.

```
plt.hist(bweights, bins = [50, 100, 120, 140, 200],
density=True, ec='w');
```

Density curves

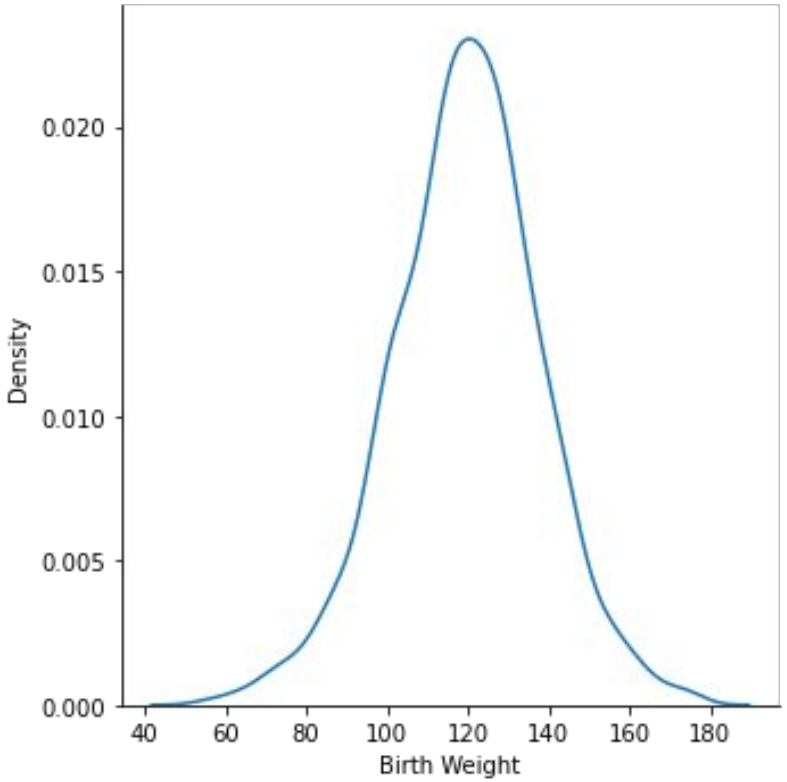


Instead of a discrete histogram, we can visualize what a continuous distribution corresponding to that same histogram could look like...

The smooth curve drawn on top of the histogram here is called a **density curve**.

```
sns.histplot(bweights, kde=True)
```

Density curves



We can also plot a density curve by itself, by appropriately setting the parameters of sns.displot or calling directly sns.kdeplot

Later, we will study how exactly to compute these density curves (using a technique is called Kernel Density Estimation).

커널 밀도 추정

With the appropriate parameter, we can also add a rug plot to our density curve.

```
sns.displot(bweights, kind='kde')
sns.kdeplot(bweights)
```

Describing quantitative distributions

Describing distributions

One of the benefits of a histogram or density curve is that they show us the “bigger picture” of our distribution (something we don’t get with a rug plot).

Some of the terminology we use to describe distributions:

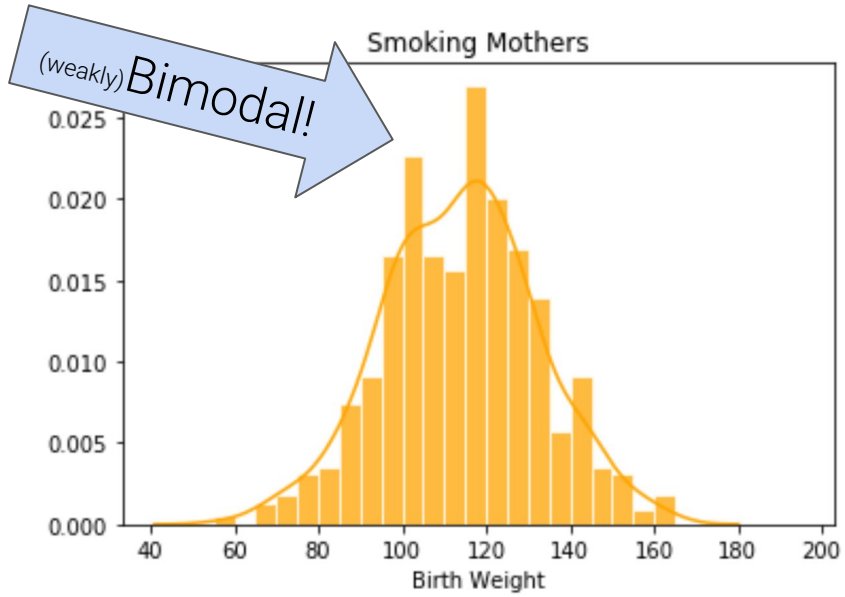
- **Modes.**
- **Skewness.**
  - Skewed left vs skewed right.
- **Tails.**
  - Left tail vs right tail.
- **Outliers.**
  - Define these arbitrarily.
  - Will see one definition in the next section.

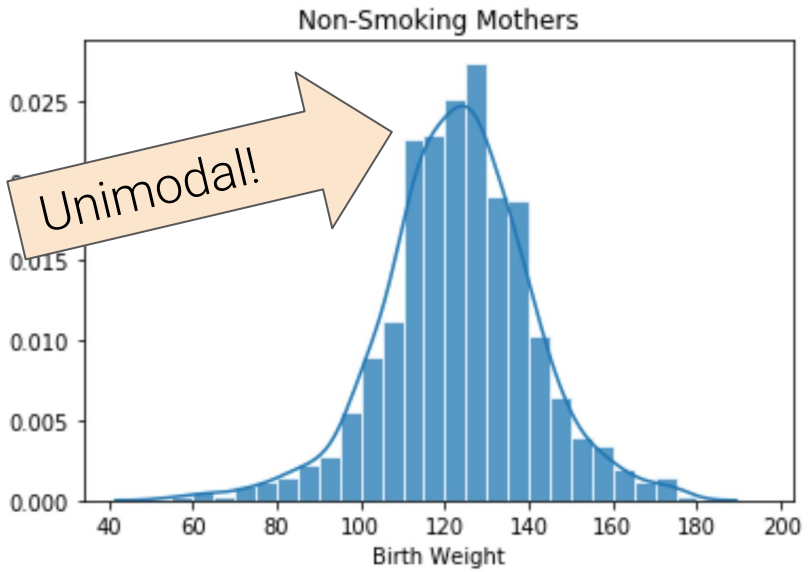
Modes

the mode is the most commonly observed value in a set of data.

A **mode** of a distribution is a local or global maximum.

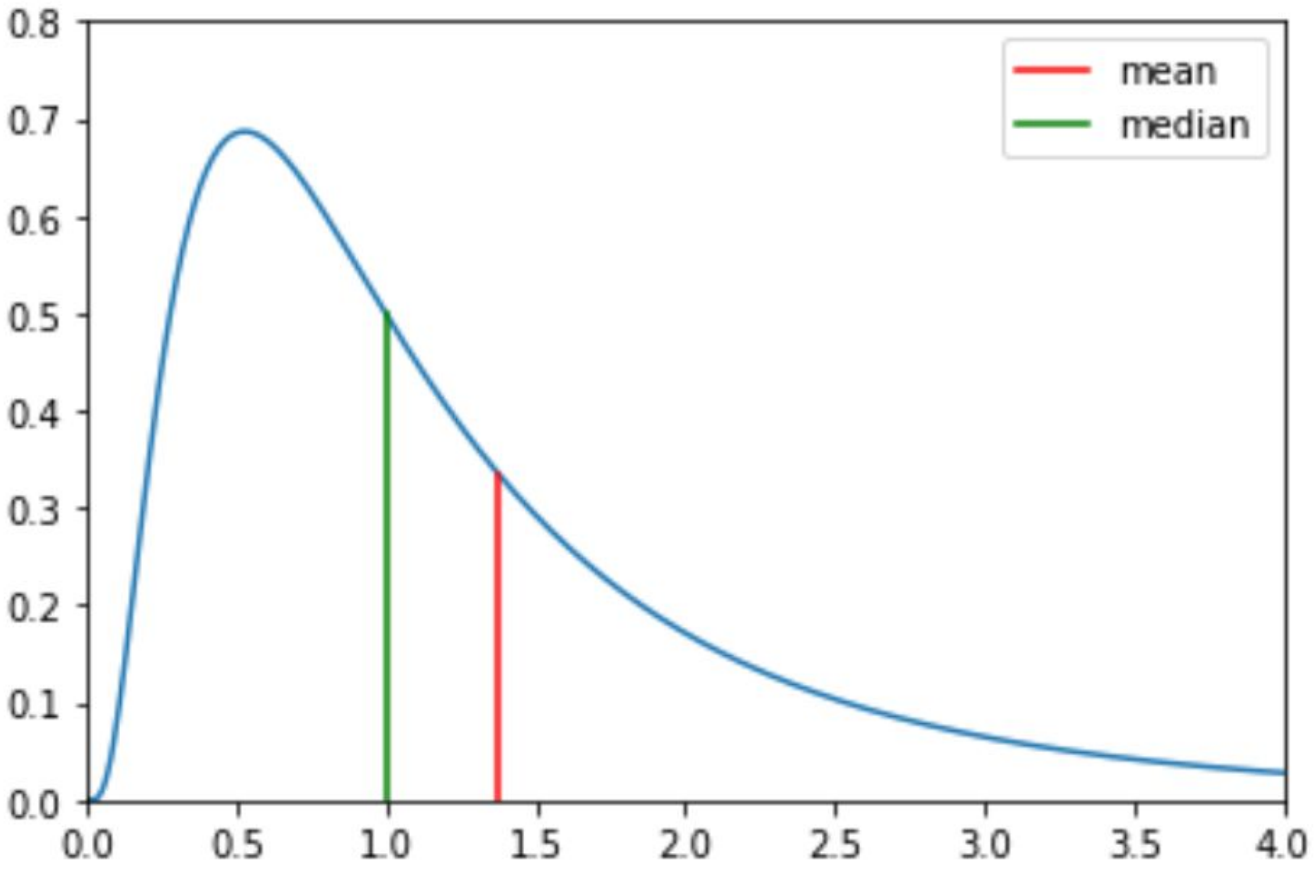
- A distribution with a single clear maximum is called unimodal.
- Distributions with two modes are called bimodal.
    - More than two: multimodal.
- Need to distinguish between modes and random noise.





Skew and tails

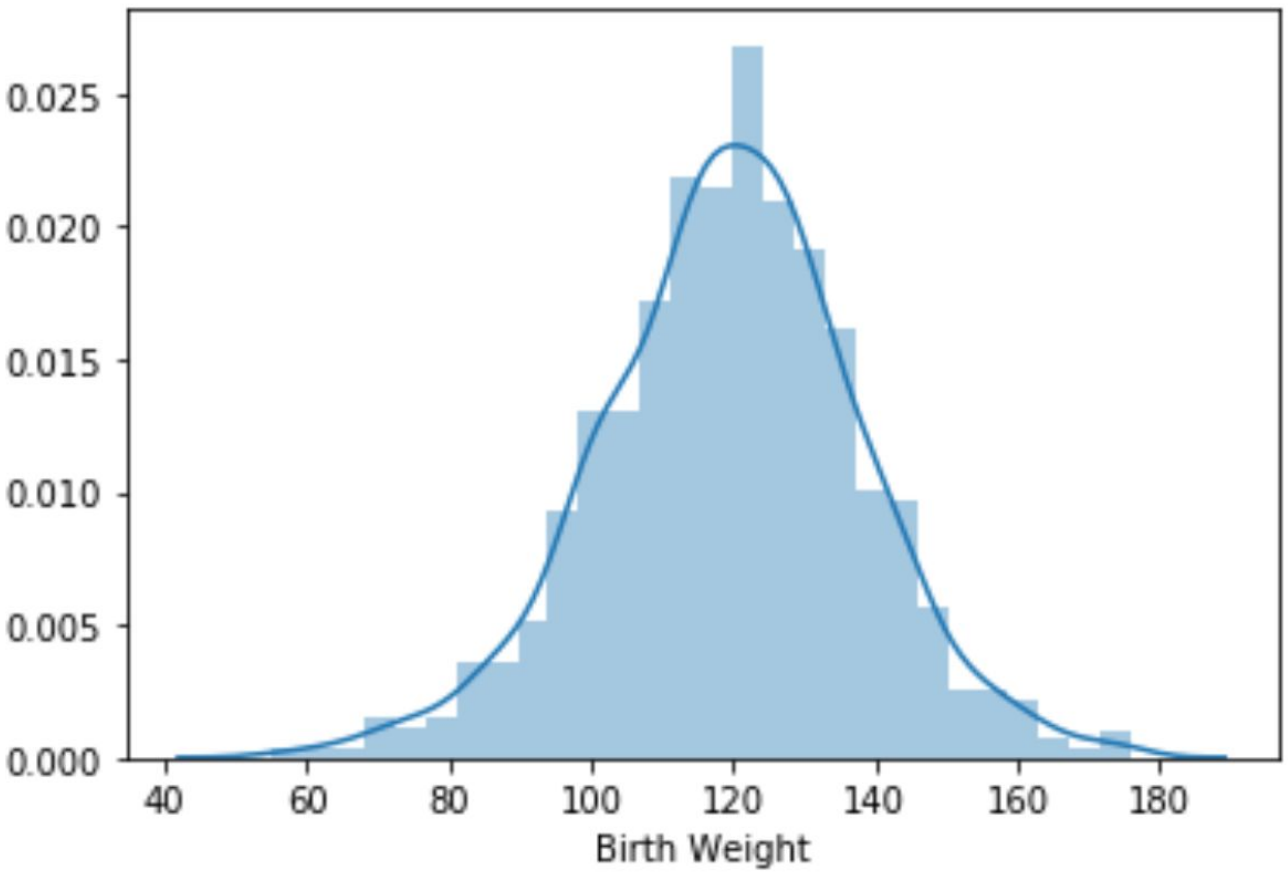
분포 자체가 skew되었다고 보면 되는 것이고, 문제가 있다고 보는 것은 아니다
Gaussian을 전제로 그래프를 작성하는 것을 생각하자



If a distribution has a **long right tail, we call it skewed right.**

- Such an example is on the left.
- In such cases, the mean is typically to the right of the median.
  - Think of the mean as the “balancing point” of the density.
- In the event that the tail is on the left, we say the data is skewed left.
- Our distribution can be symmetric, when both tails are of equal size.

Example



Consider the distribution of birth weights shown to the left. We might describe this as being:

- Unimodal. There is a single clear peak.
- Symmetric. It doesn't appear to be skewed in any direction.
    - Mean is very close to the median.
- Roughly normal.

Box plots and violin plots

Quartiles

현재 이해가 잘 안되는 부분

4분위 수

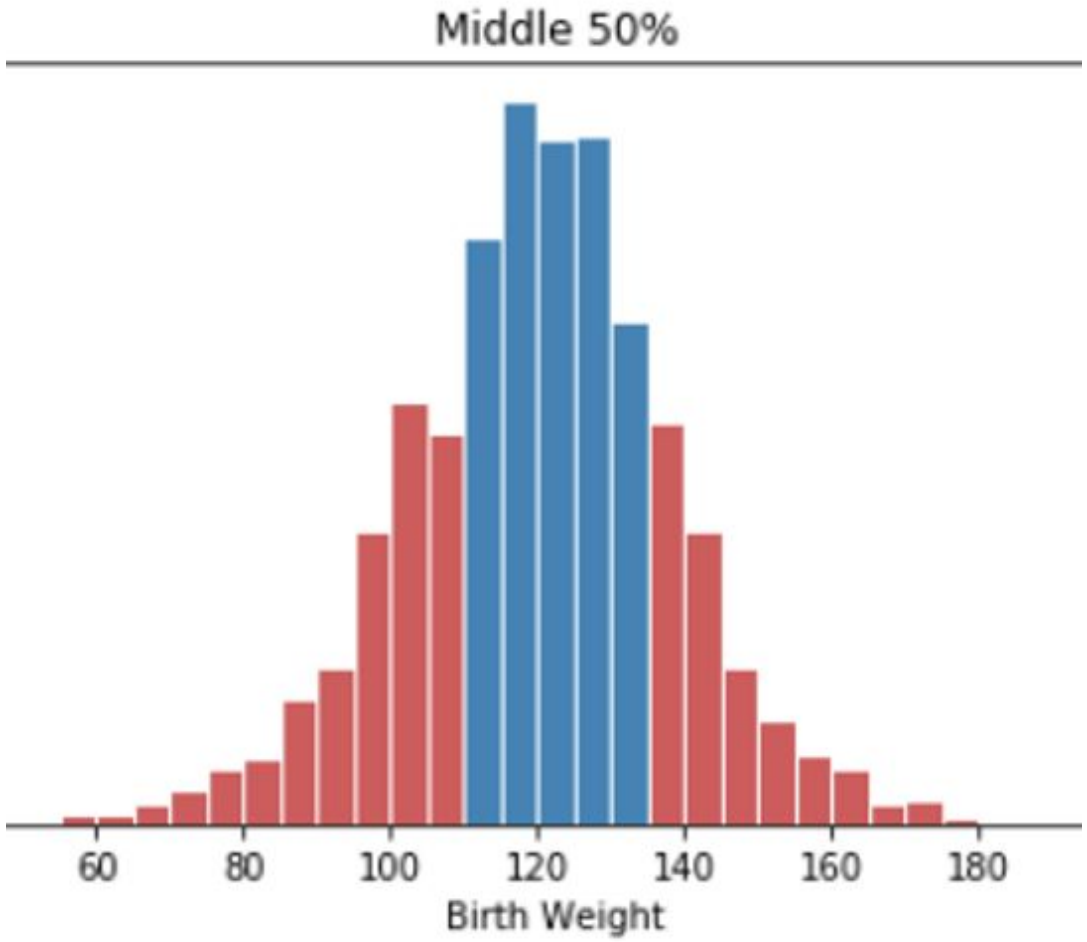
For a quantitative variable:

- First or lower quartile: 25th percentile
- Second quartile: 50th percentile (median)
- Third or upper quartile: 75th percentile

The interval [first quartile, third quartile] contains the “middle 50%" of the data.

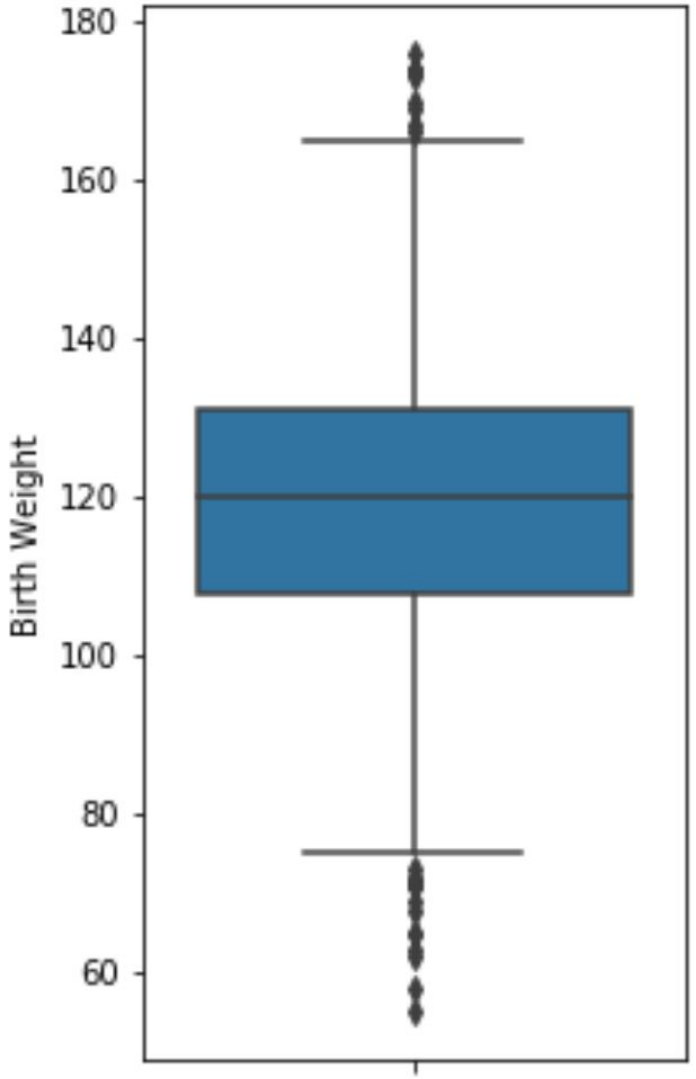
Interquartile range (IQR) measures spread.

- IQR = third quartile – first quartile.



Box plots

현재 이해가 잘 안되는 부분

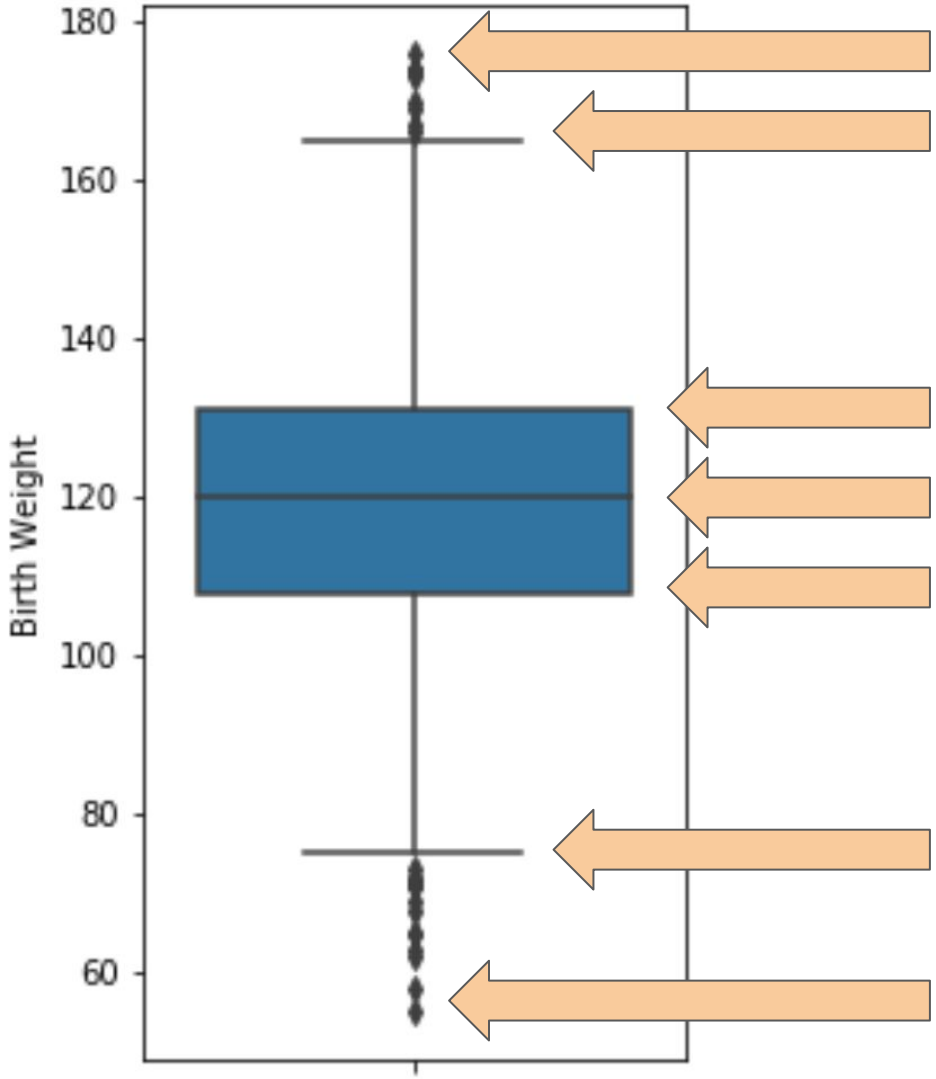


Box plots summarize several characteristics of a numerical distribution. They visualize:

- **Lower quartile**.
- **Median**.
- **Upper quartile**.
- "**Whiskers**", placed at lower quartile minus 1.5*IQR and upper quartile plus 1.5*IQR.
- **Outliers**, which are defined as being further than 1.5*IQR from the extreme quartiles. Arbitrary definition!
- We lose a lot of information, too!

```
sns.boxplot(bweights)
```

Box plots



Outliers

Upper quartile plus 1.5*IQR

Upper quartile

Median

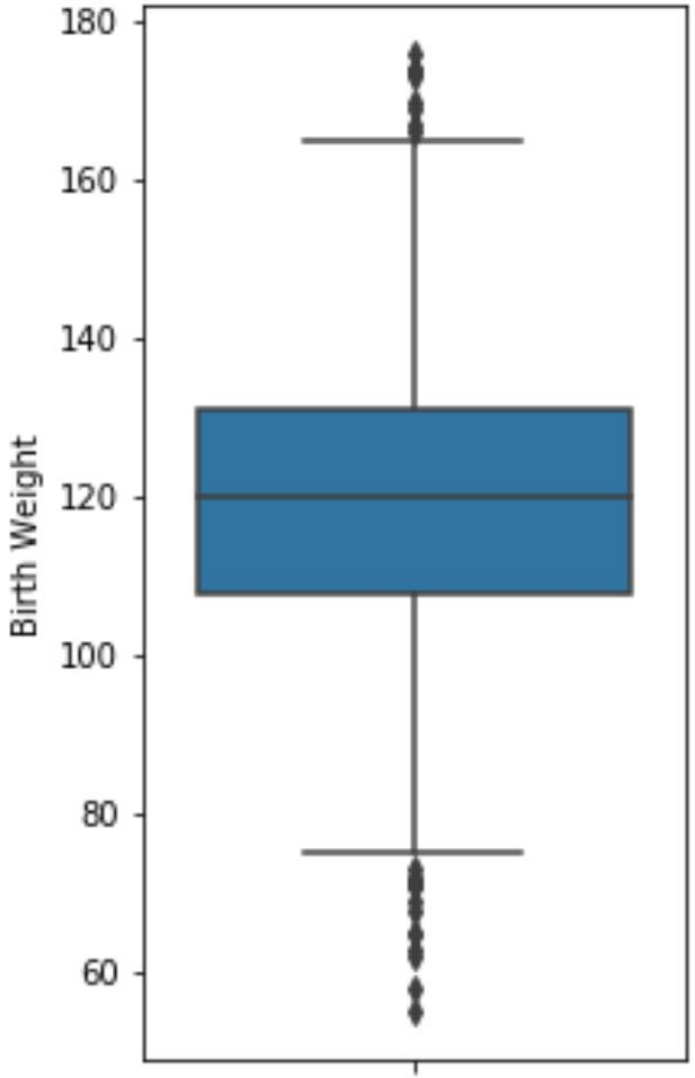
Lower quartile

Lower quartile minus 1.5*IQR

Outliers

Note: The box width is meaningless.

Box plots



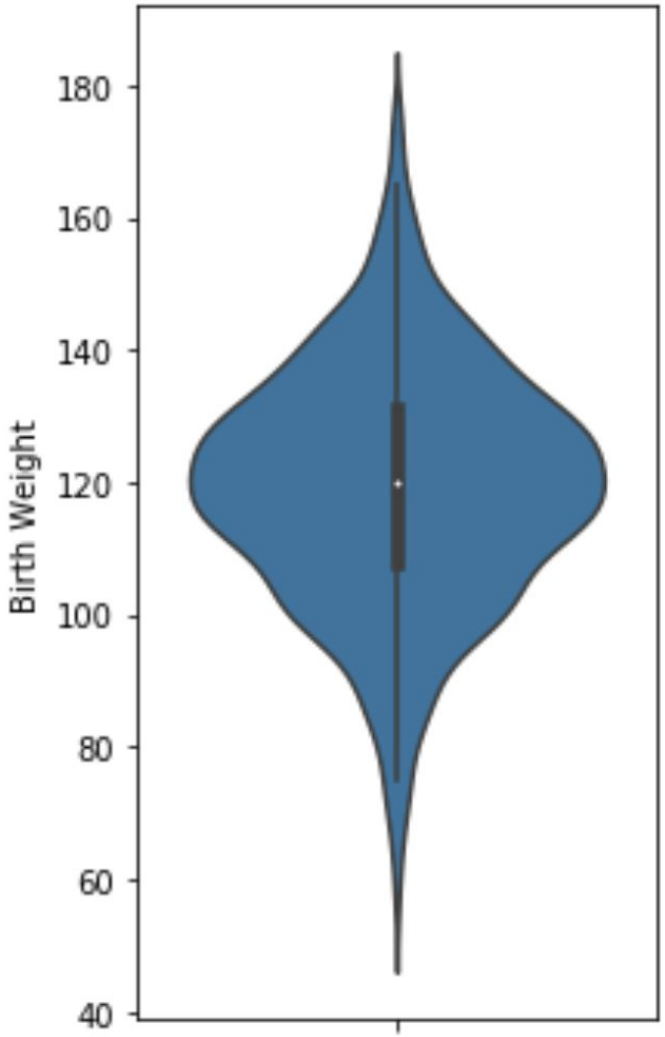

```
1  q1 = np.percentile(bweights, 25)
2  q2 = np.percentile(bweights, 50)
3  q3 = np.percentile(bweights, 75)
4  iqr = q3 - q1
5  whisk1 = q1 - 1.5*iqr
6  whisk2 = q3 + 1.5*iqr
7
8  whisk1, q1, q2, q3, whisk2
```

```
(73.5, 108.0, 120.0, 131.0, 165.5)
```

The five numbers above match what we see on the left.

Violin plots

현재 이해가 잘 안되는 부분

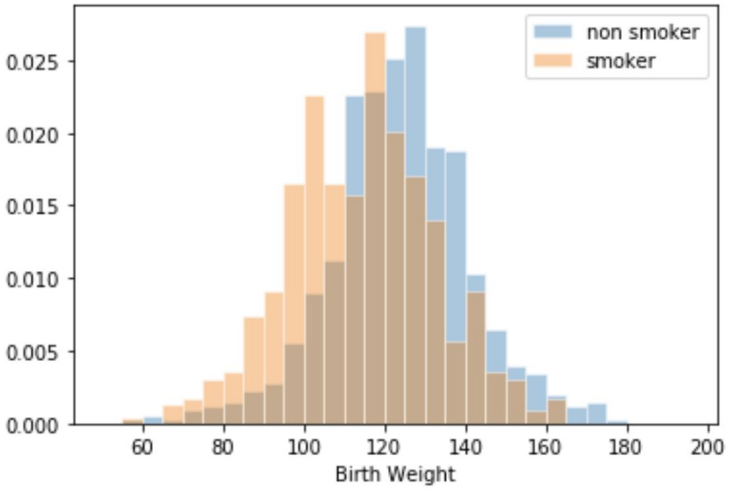


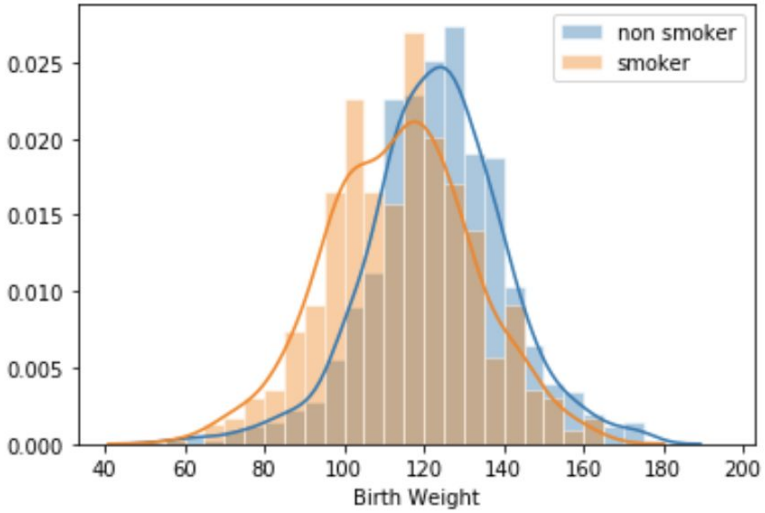
Violin plots are similar to box plots, but also show smoothed density curves.

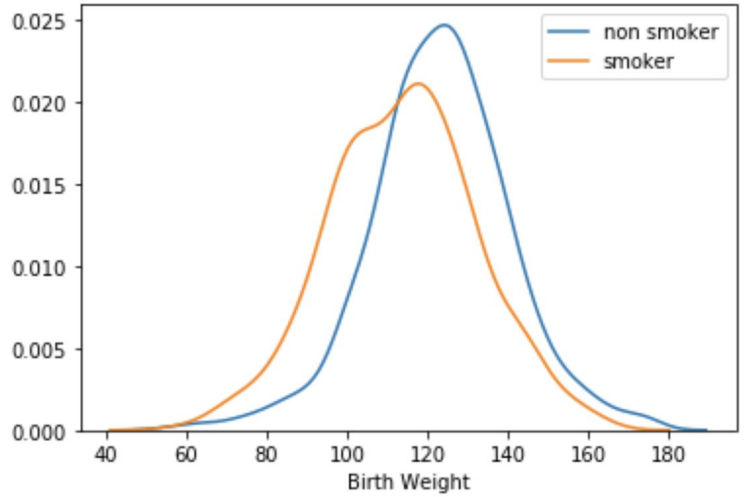
- The “width” of our “box” now has meaning!
- The three quartiles and “whiskers” are still present – look closely.
- Both box plots and violin plots are useful for comparing multiple distributions, which we are about to do.

Comparing quantitative distributions

Overlaid histograms and density curves



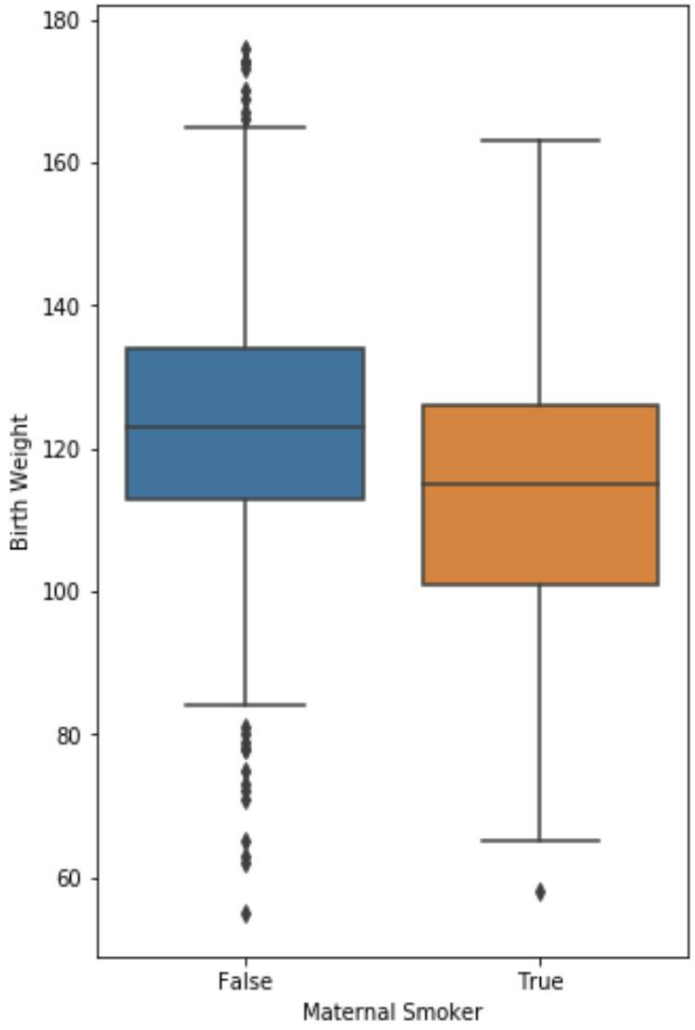


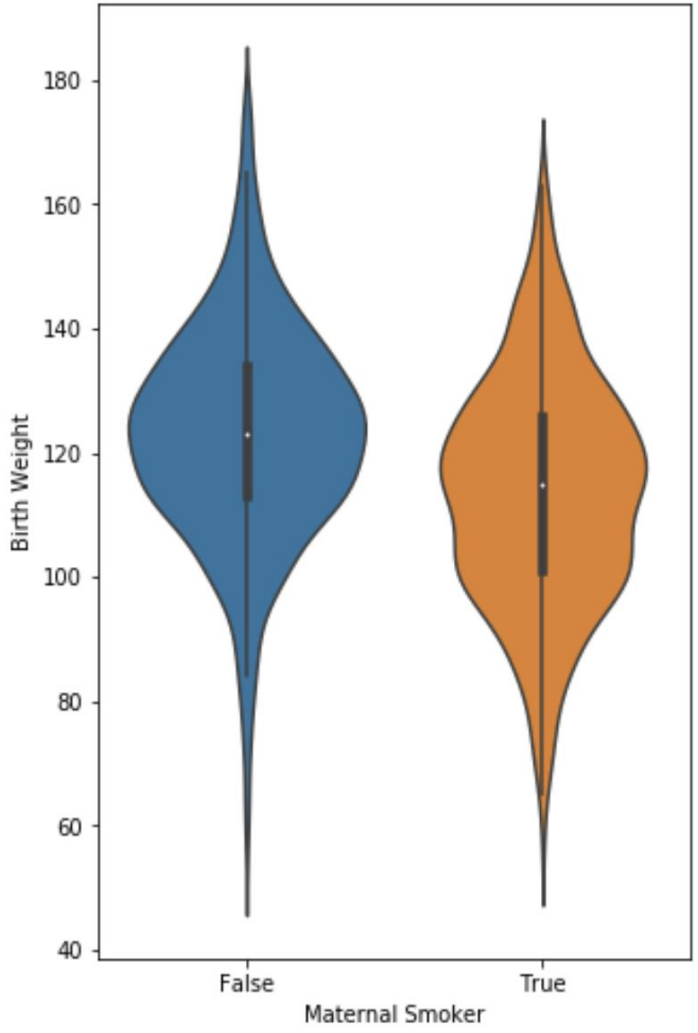


We can overlay multiple histograms and density curves on top of one another.

- First: Not terrible, but looks like three separate histograms.
- Second: Has the most information, but isn't very clear!
- Third: Rough estimate of both distributions, but is the most clear by far.
- Neither will generalize well to three or more categories.

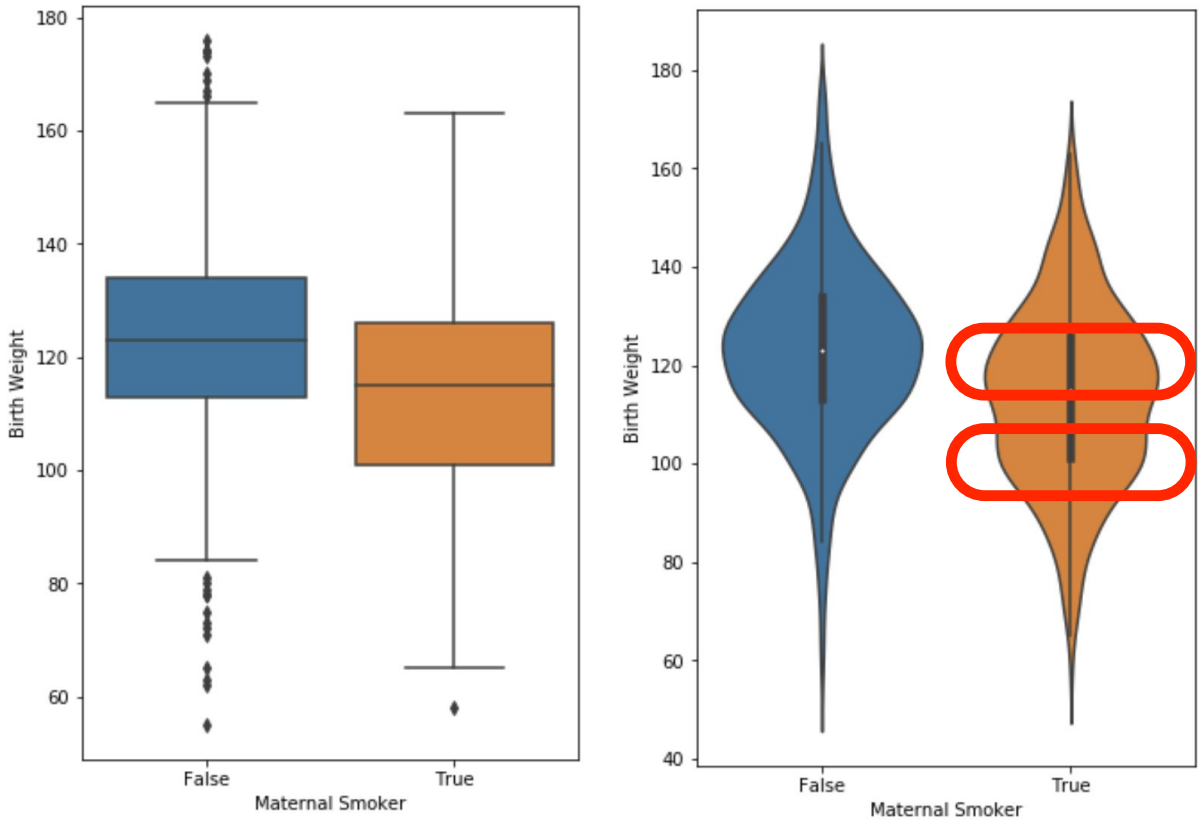
Side by side box plots and violin plots





Side by side box plots and violin plots

모드 : Gaussian이 하나 있으면 Unimodal, 두개면 Bimodal



Distribution을 보여주느냐, 안보여주느냐?

Box plots and violin plots are concise, and thus are well suited to be stacked side by side to compare multiple distributions at once.

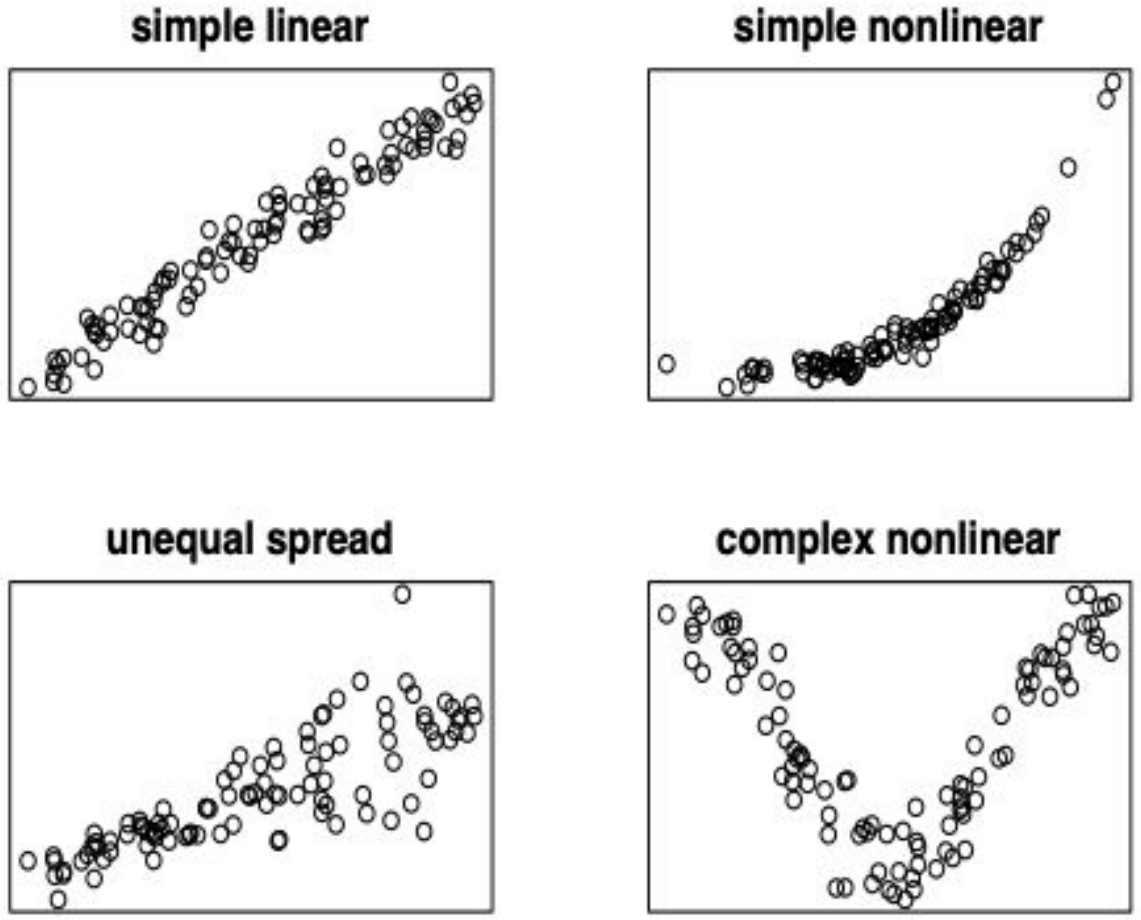
- At a glance, we can tell that the median birth weight is higher for babies whose mothers did not smoke while pregnant (“False”).
- The violin plot shows us the bimodal nature of the “True” category.

Relationships between two quantitative variables

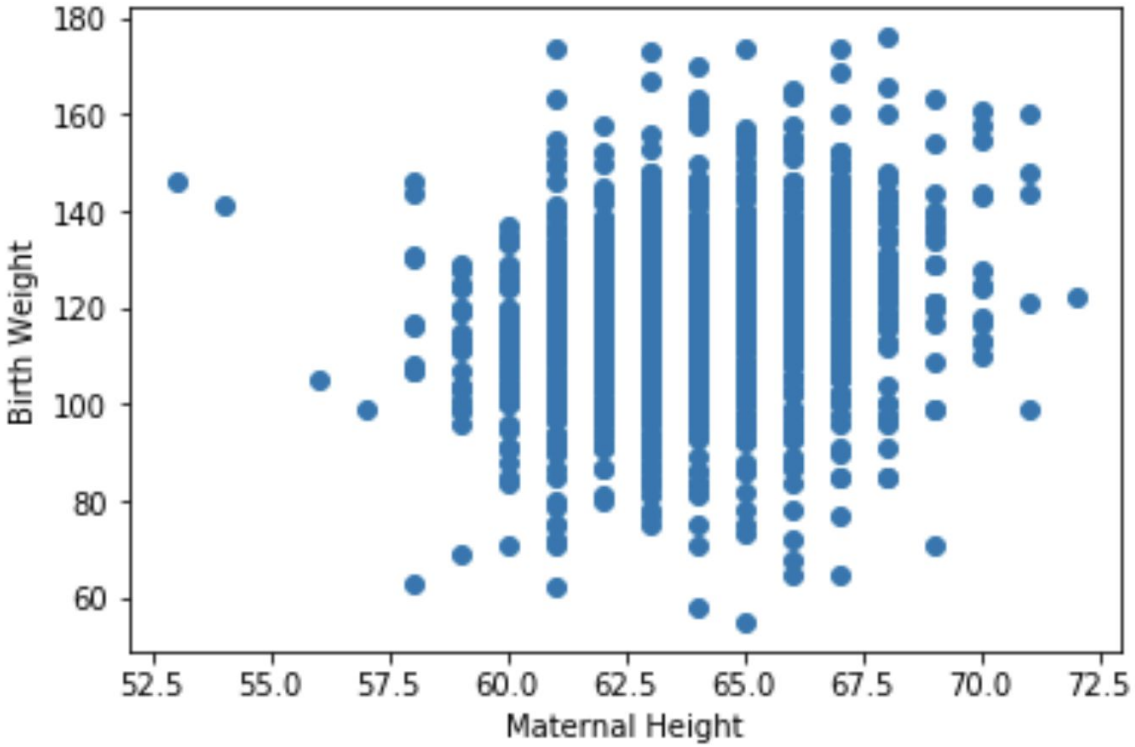
Scatter plots

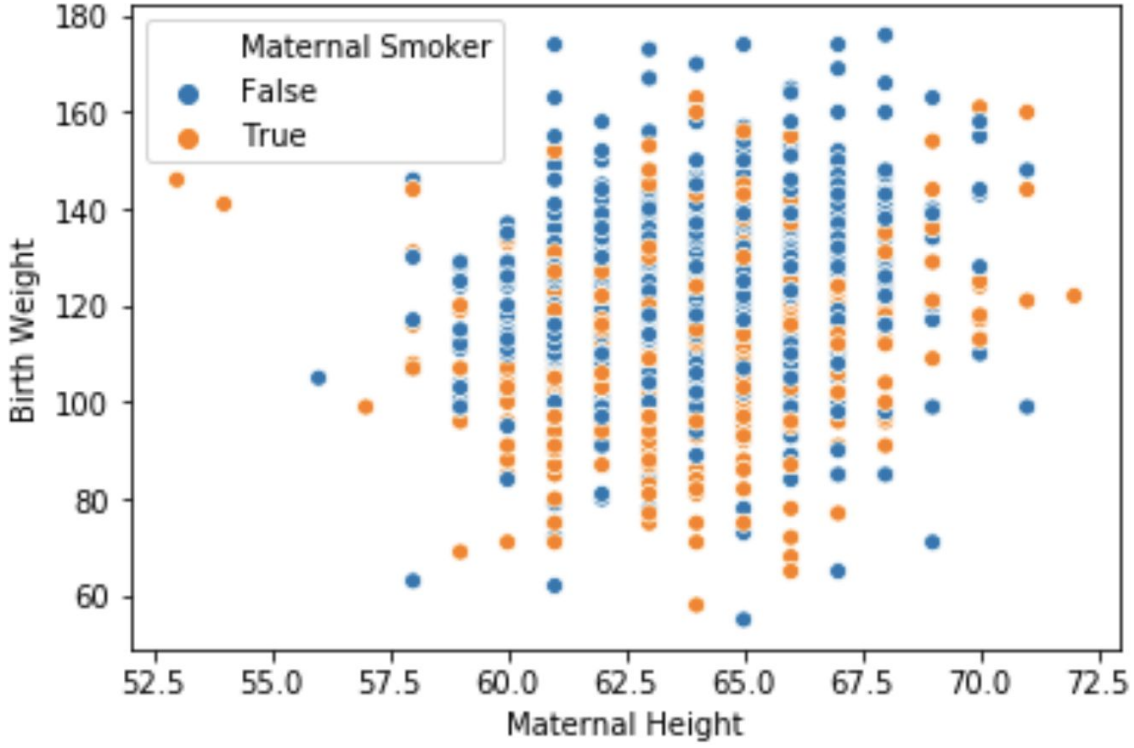
Scatter plots are used to reveal relationships between pairs of numerical variables.

- We often use scatter plots to help inform modeling choices.
- For instance, the simple linear model requires the trend in our data to be roughly linear, and for spread to be roughly equal.



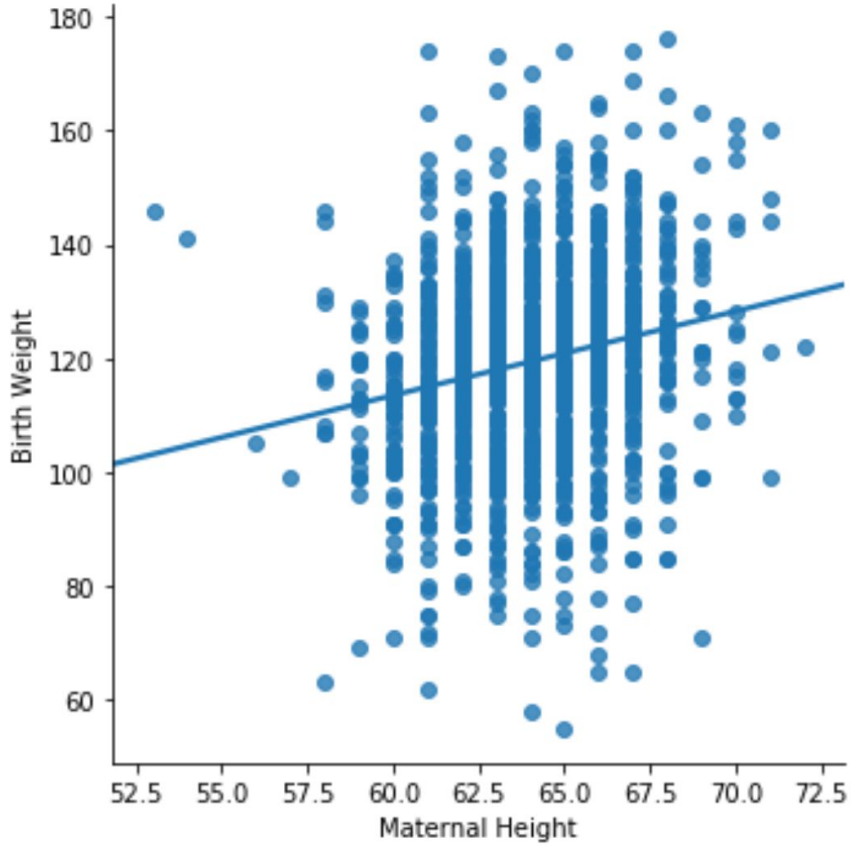
Scatter plots



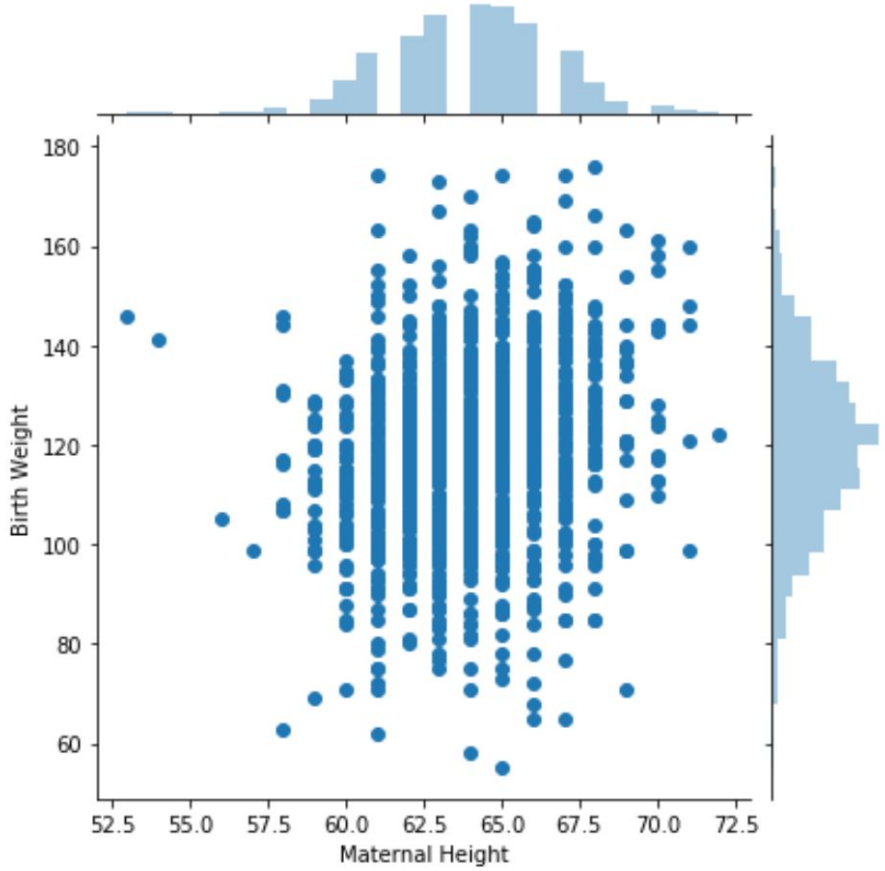


- We can also use color to encode categorical variables.
- These plots suffer from overplotting – many of the points are on top of one another!
  - One solution: add a small amount random noise in both the x and y directions.

Scatter plots




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




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

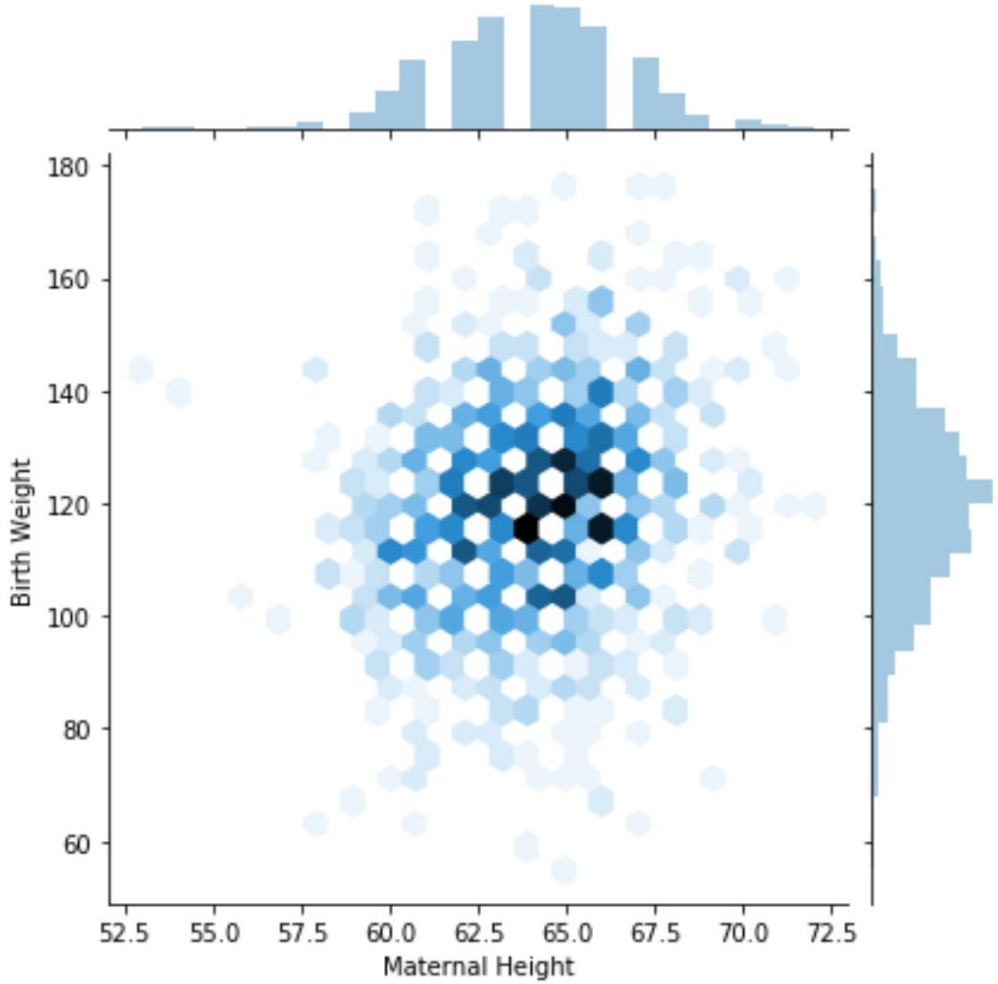
Hex plots

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')
```

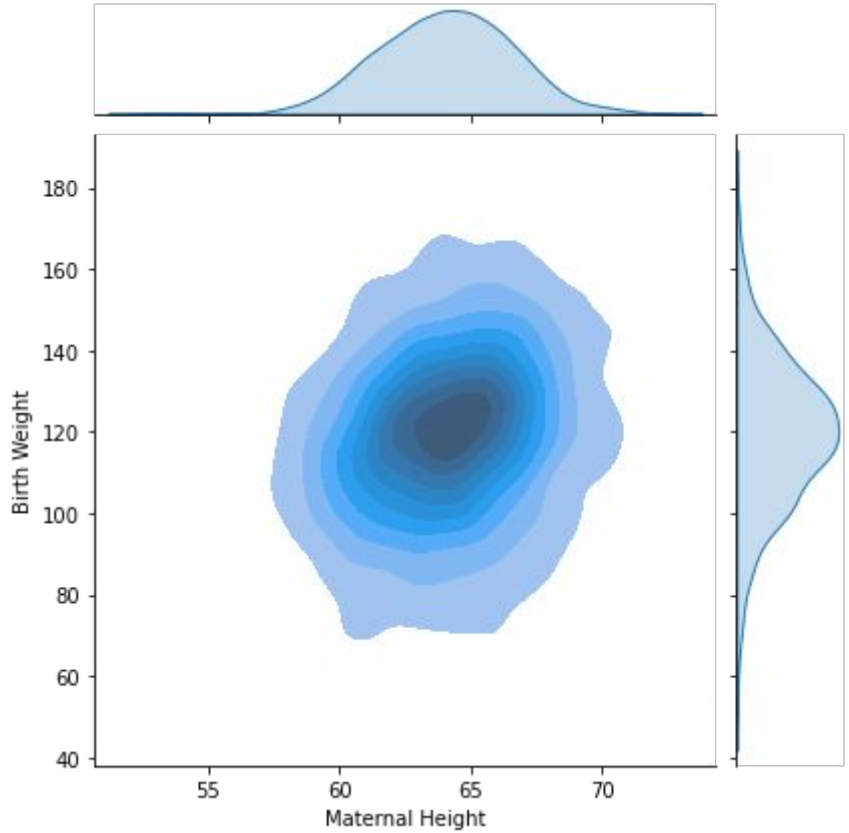
Contour plots

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`**.

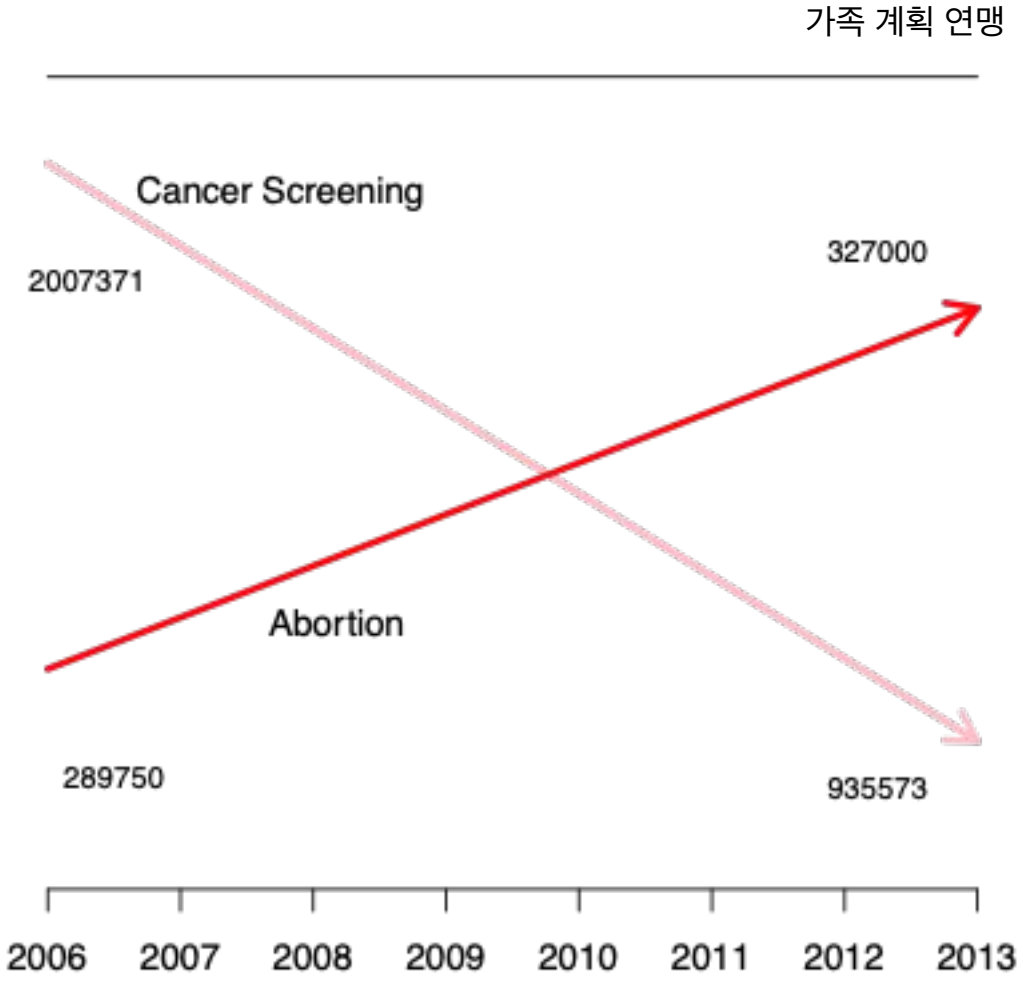
- By default, shows **marginal** distributions on the horizontal and vertical axes.
- These are the histograms/density curves of each variable independently.




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

Principles of Visualizations: Scale

Case Study: Planned Parenthood Hearing

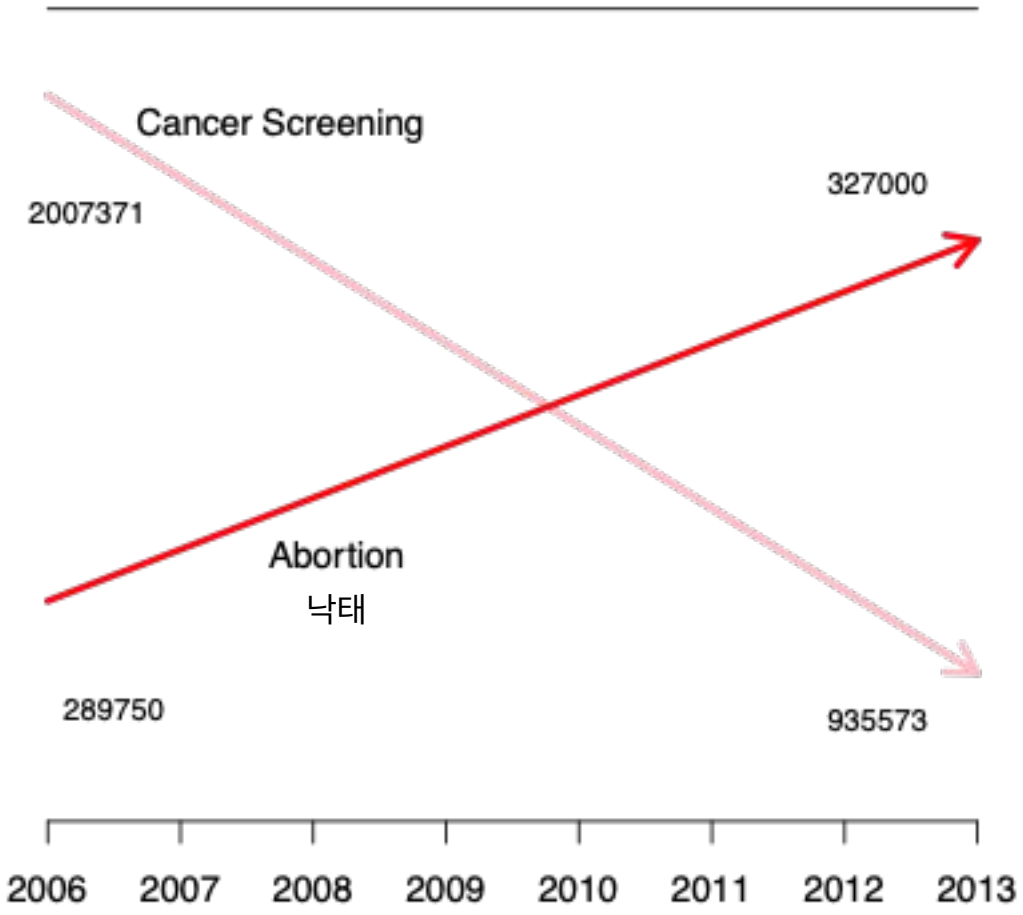


In 2015, Planned Parenthood was accused of selling aborted fetal tissue for profit.

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

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

Keep axis scales consistent



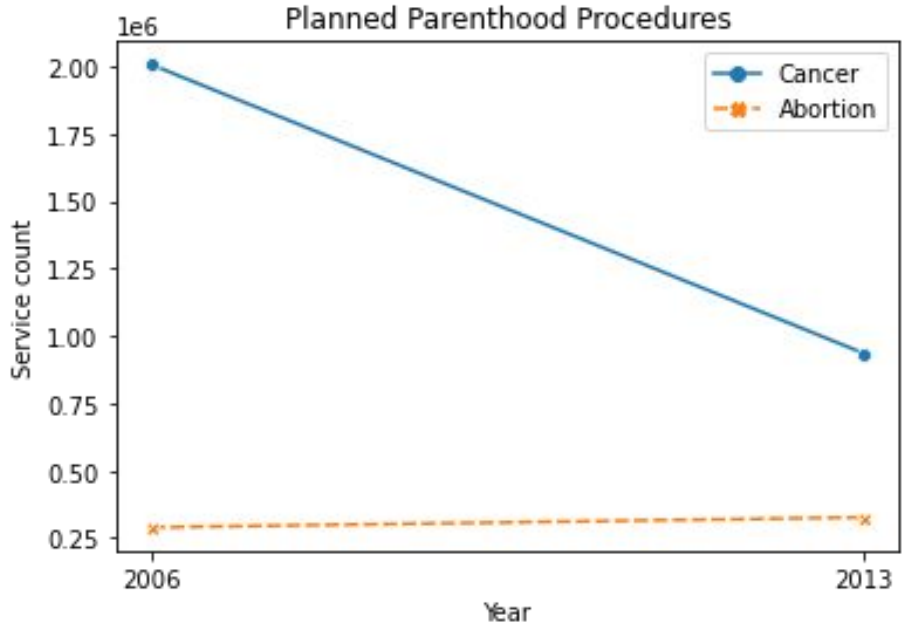
The scales for the two lines are completely different!

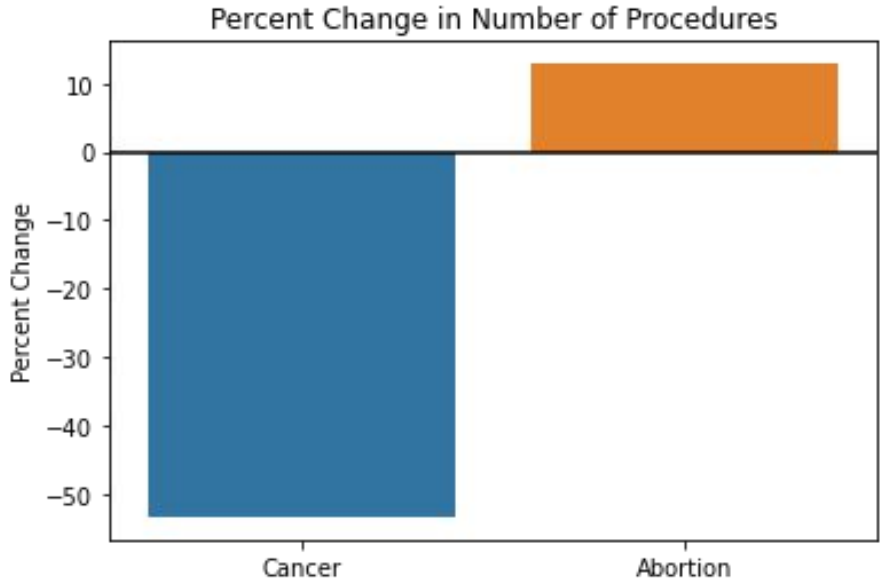
- 327000 is smaller than 935573, but appears to be way bigger.
- **Do not use two different scales for the same axis!**

Consider scale of the data

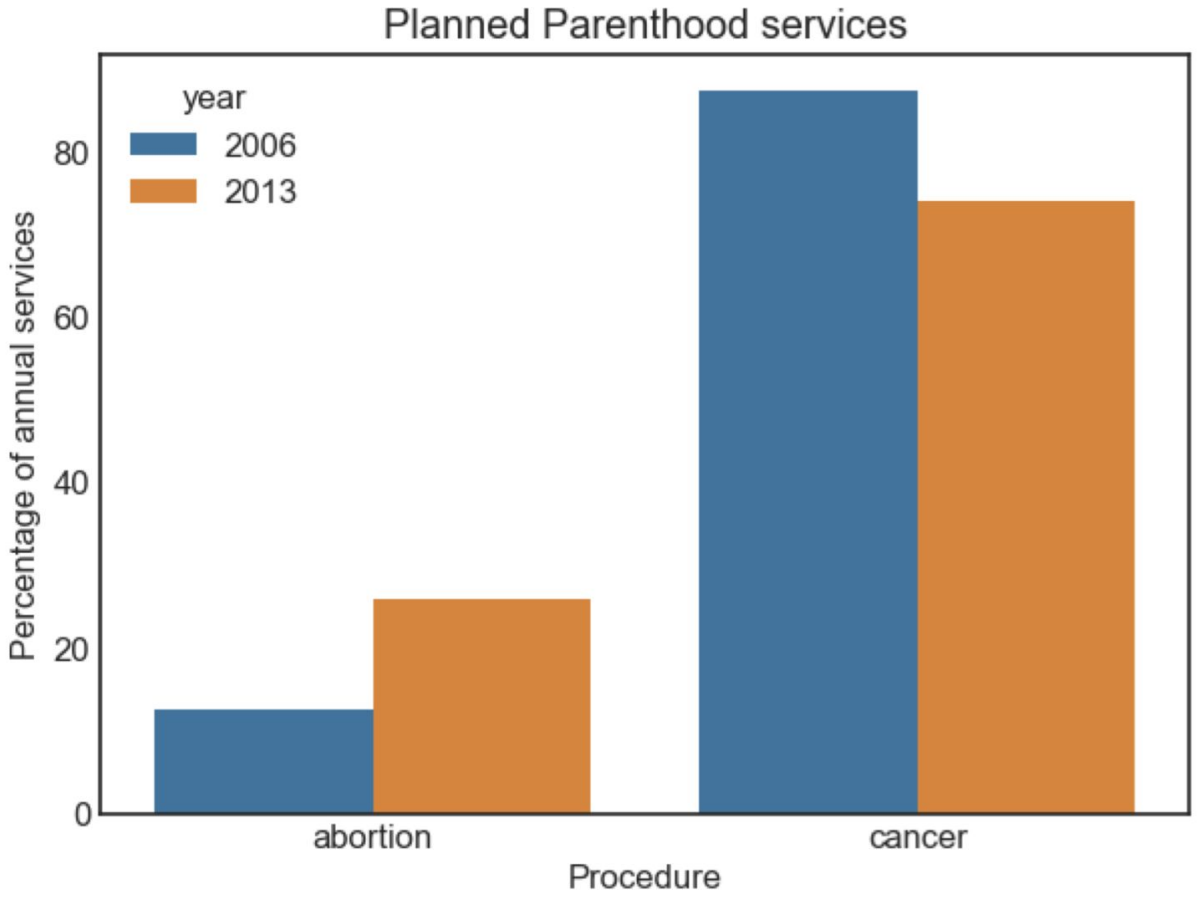
The top plot draws all of the data on the same scale.

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





Consider scale of the data



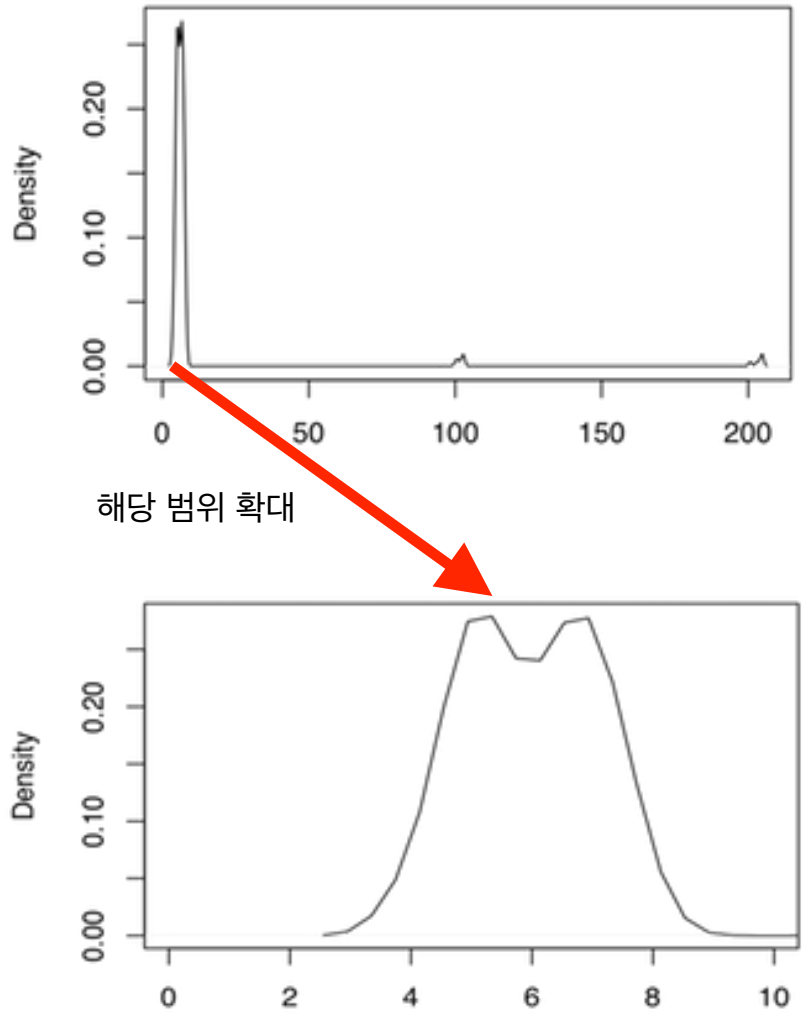
We could also visualize abortions and cancer screenings as a percentage of total procedures.

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

Reveal the data

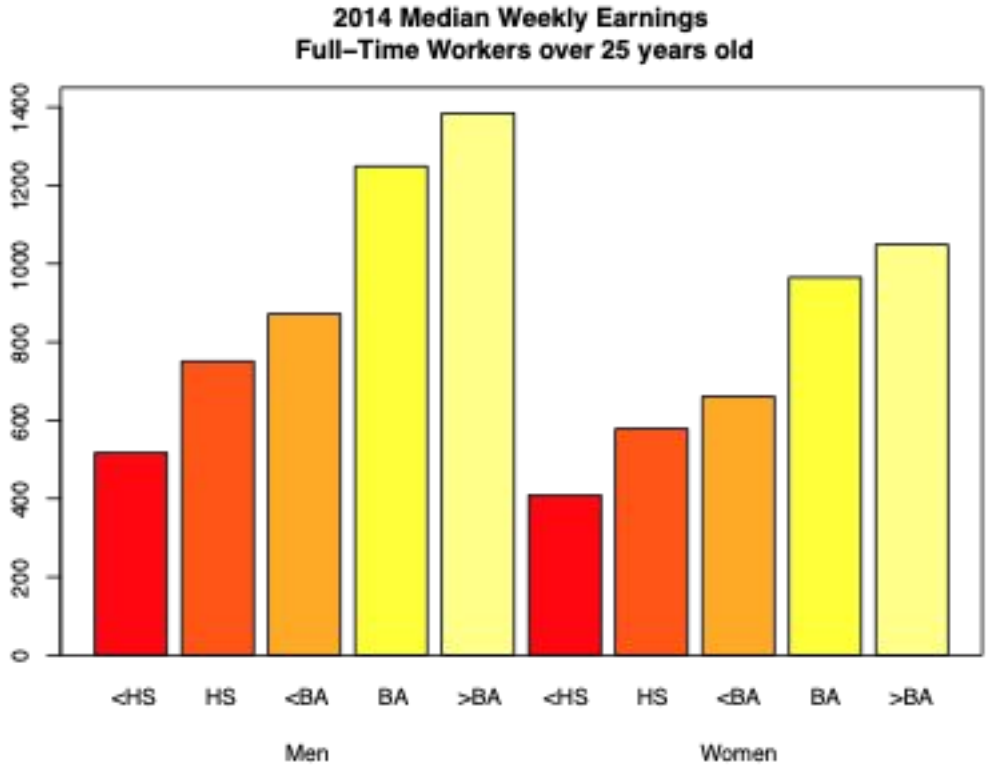
- Choose axis limits to fill the visualization.
- If necessary:
  - Zoom in on the bulk of the data.
  - Create multiple plots to show different regions of interest.

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



Conditioning

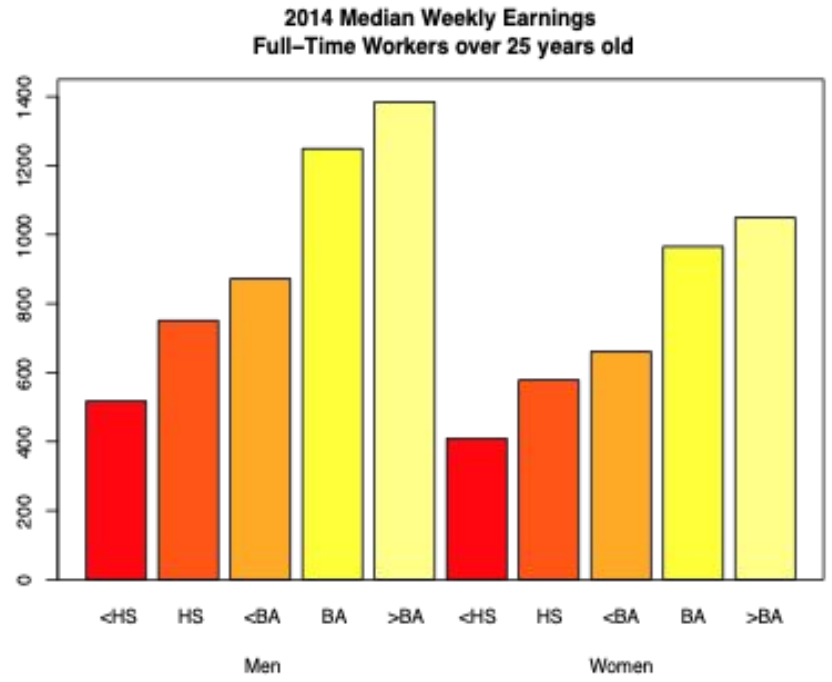
Case Study: Median Weekly Earnings

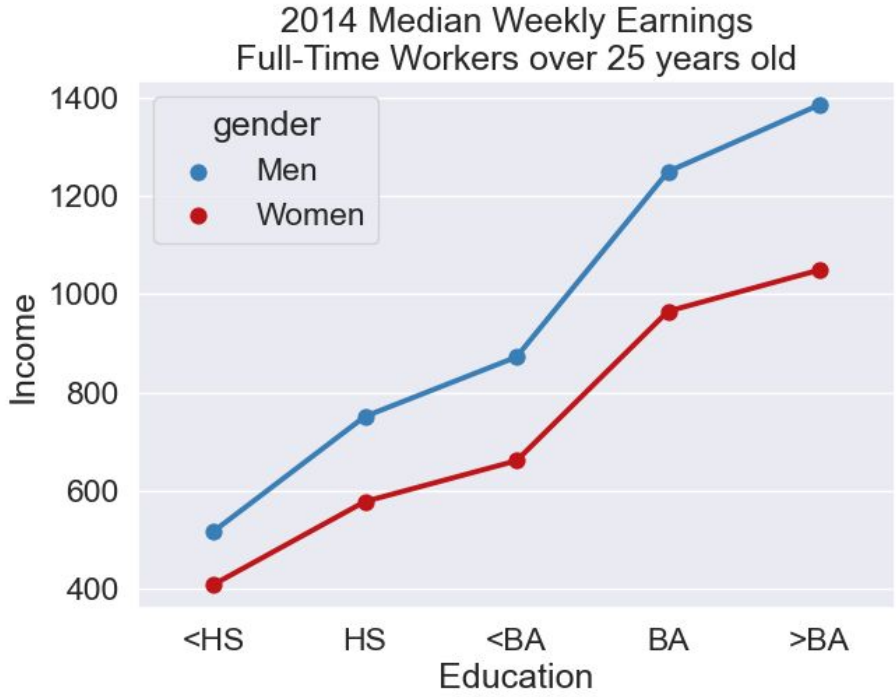


This data comes from the Bureau of Labor Statistics, who oversees surveys regarding the economic health of the US. They have plotted median weekly earnings for men and women by education level.

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

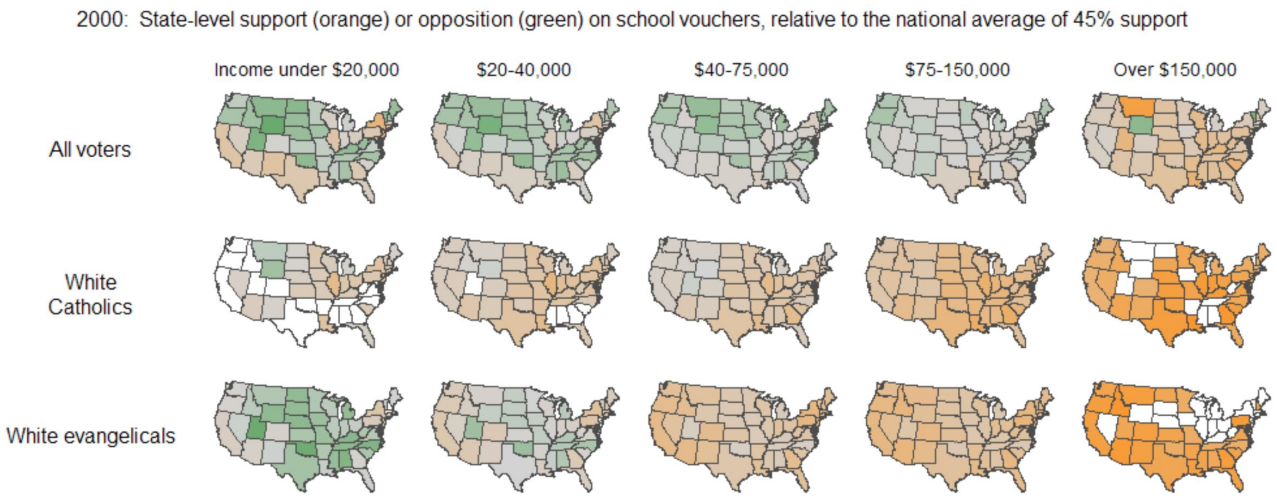
Use conditioning to aid comparison





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

Distributions and relationships in subgroups



An example of **small multiples**.

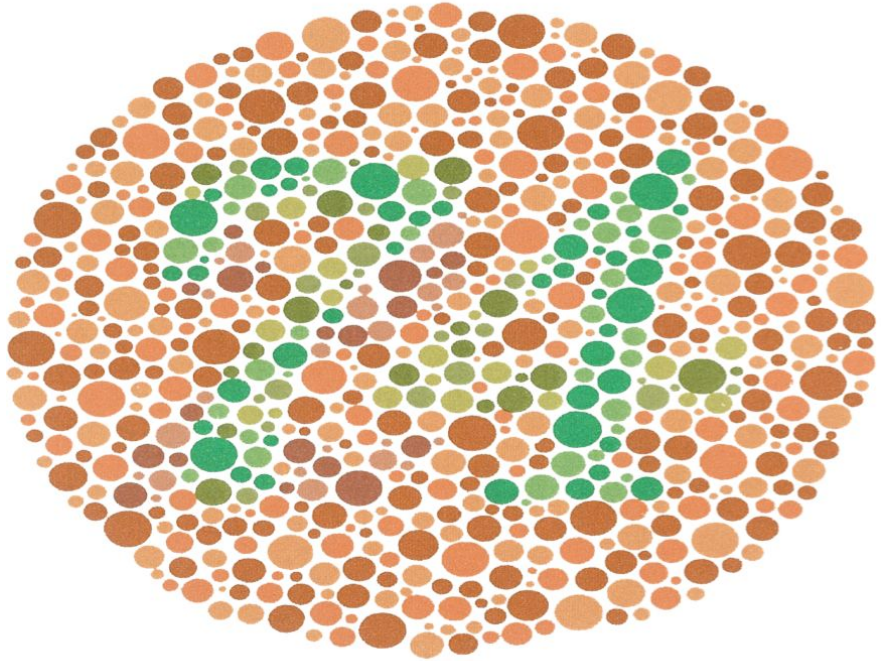
병렬, 병치

- Juxtaposition: placing multiple plots side by side, with the same scale (called “small multiples”).

중첩

- Superposition: placing multiple density curves, scatter plots on top of each other (previous lec)
- Use color and shapes to represent additional variables.
    - See more in discussion.

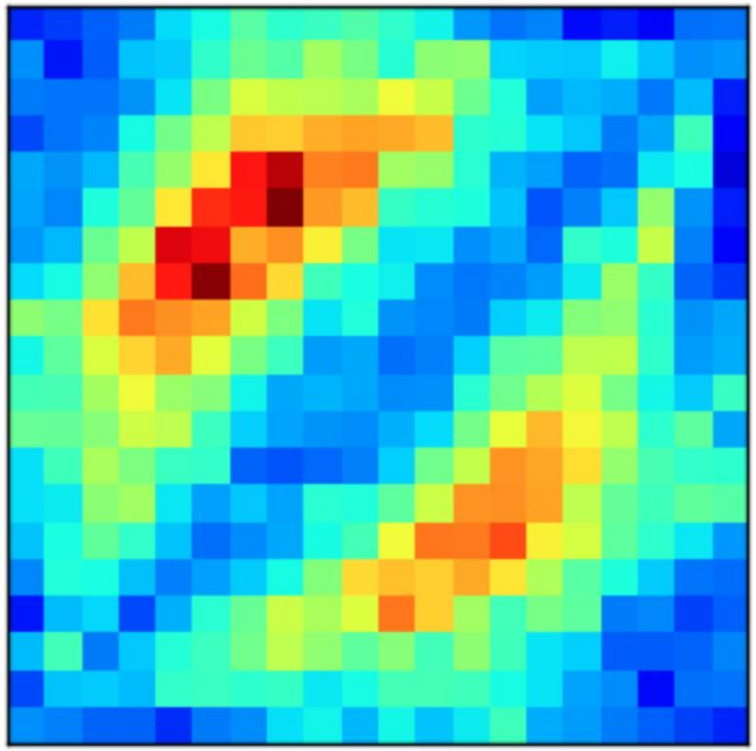
Perception



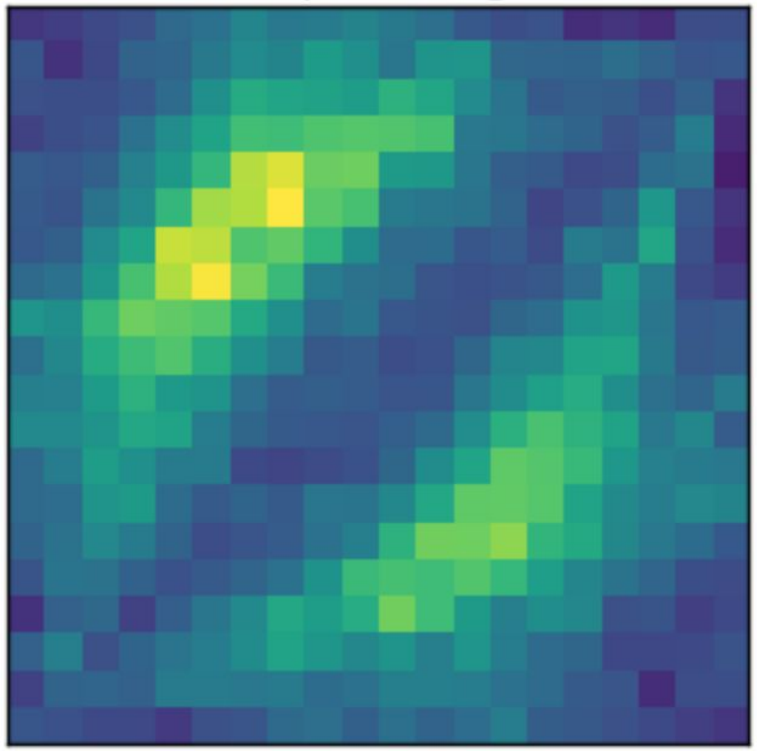
Choosing a set of colors which work together is a challenging task!

Perception of Color

Colormaps



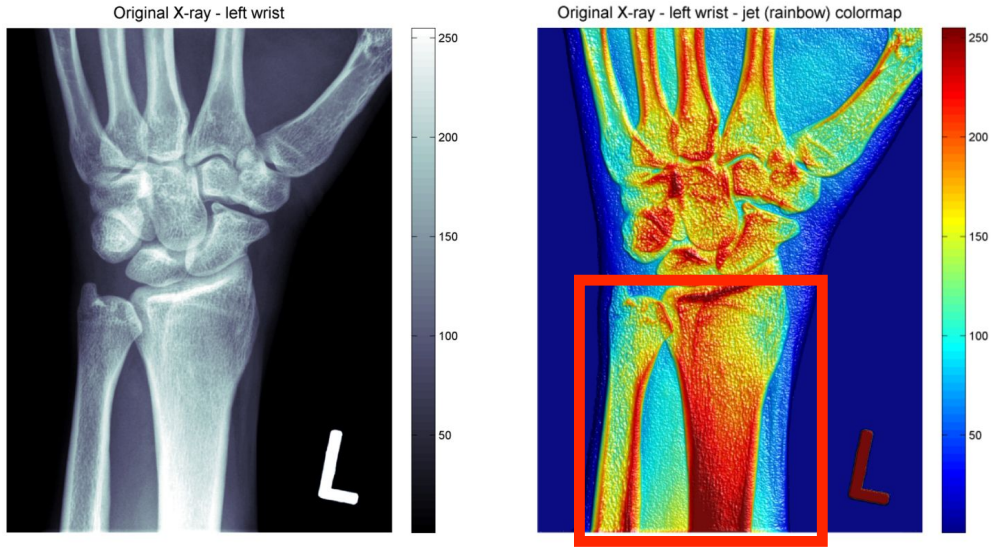
Jet



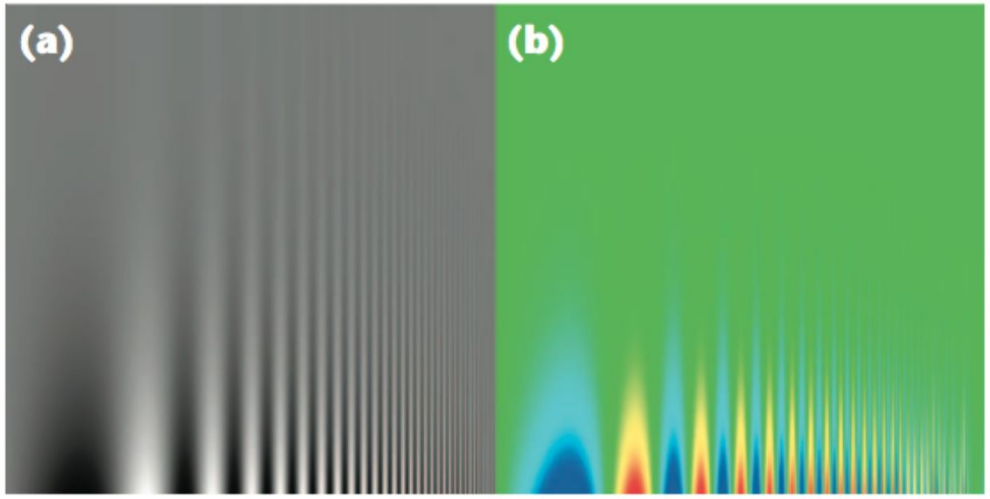
Viridis

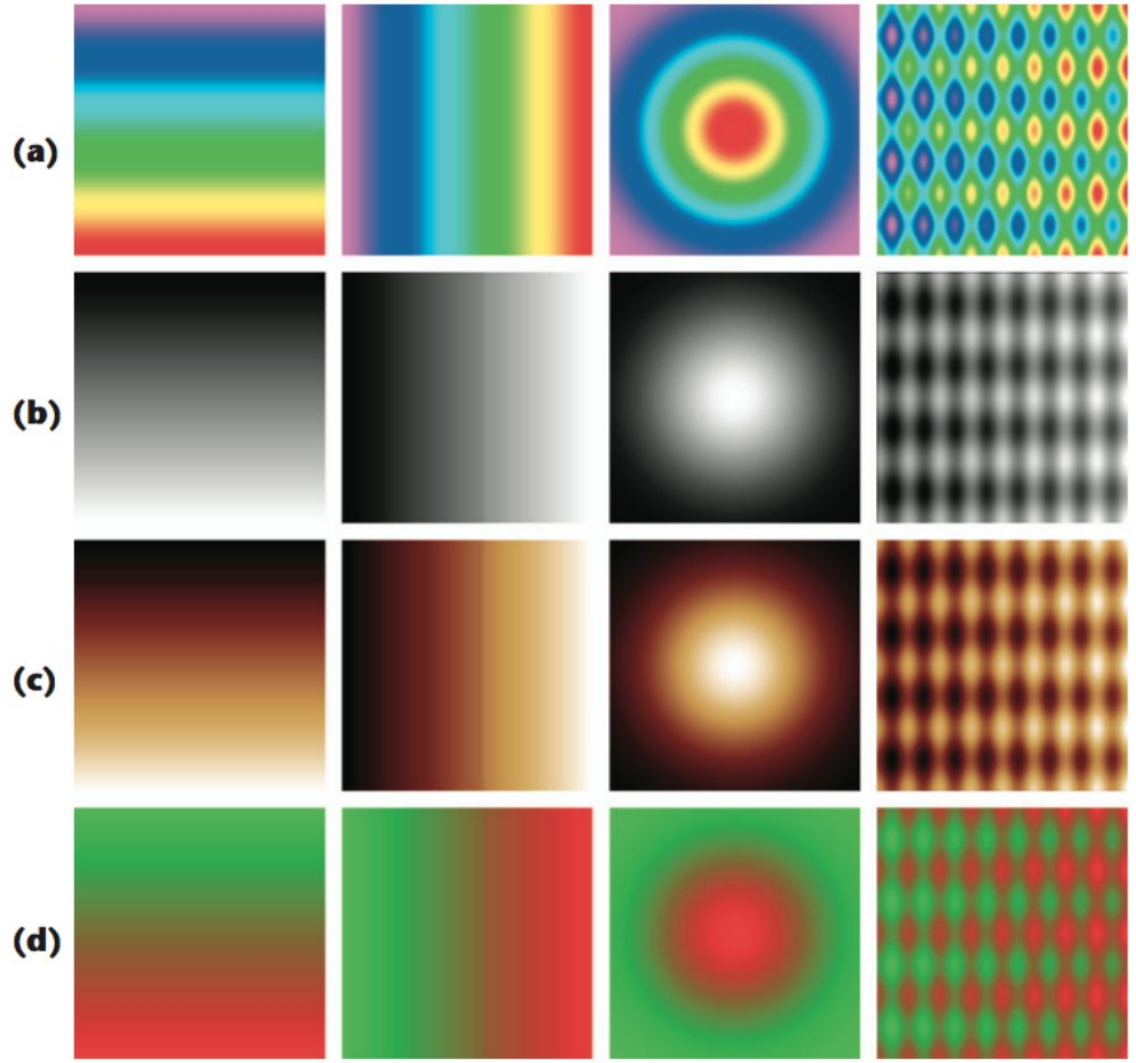
The jet/rainbow colormap actively misleads

적절한 사례



Jet Color Map의 경우 작은 변화를 눈치채기 어려움



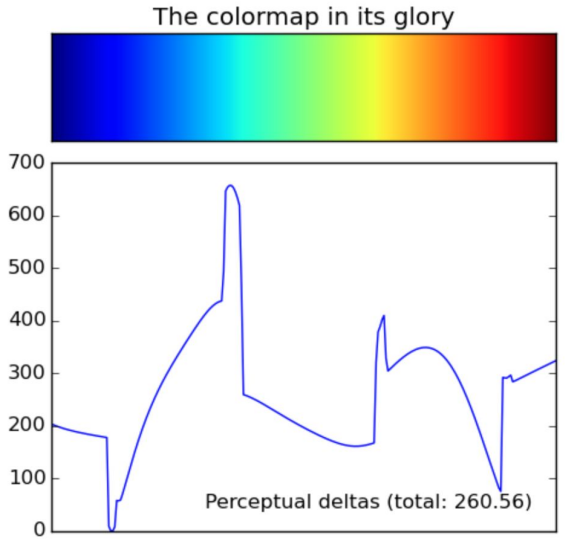


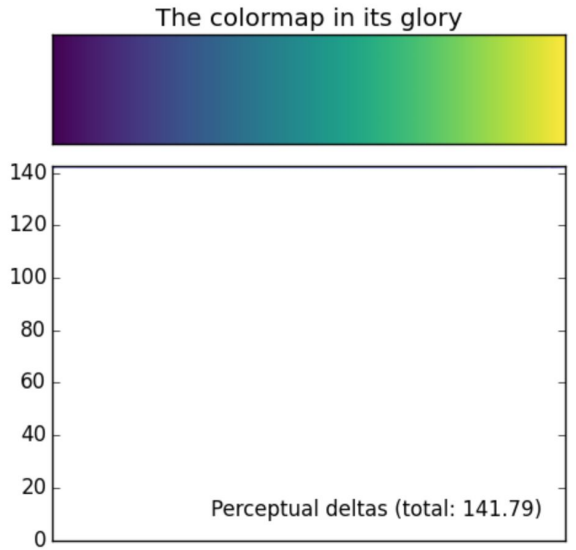
"Rainbow Colormap (Still) Considered Harmful", Borland and Taylor, 2007.

Use a perceptually uniform colormap!

Does not provide Perceptuality

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

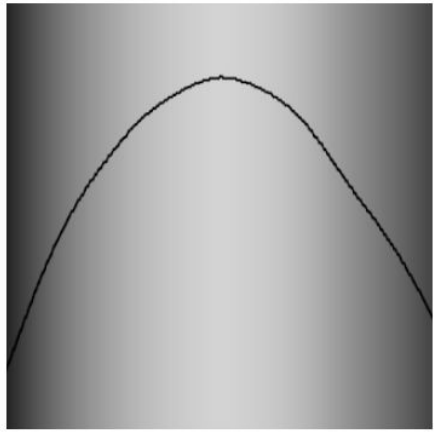




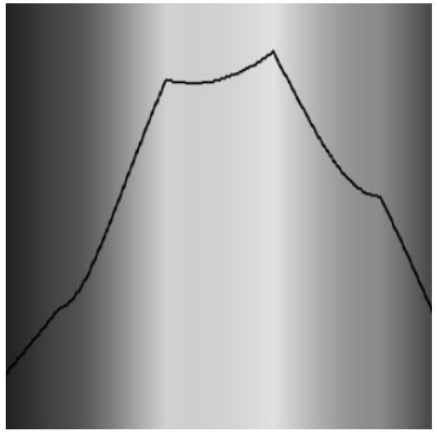
Except when not :) The Google Turbo Colormap

Turbo

Jet

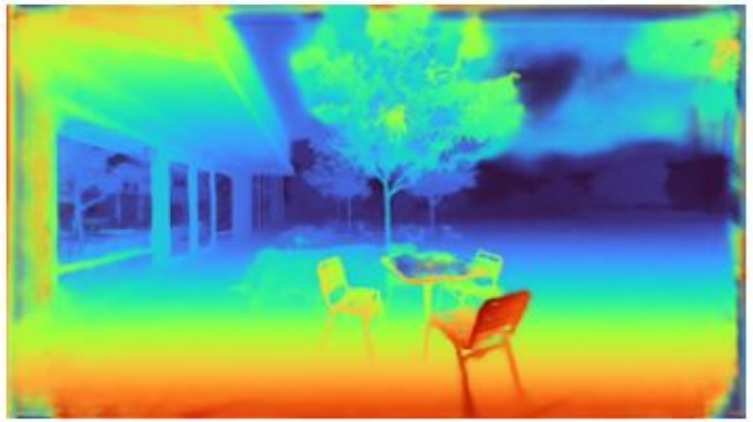


Turbo

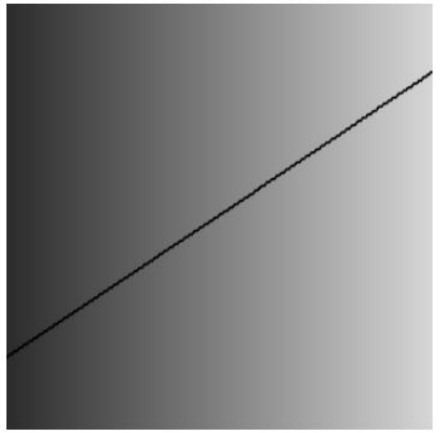


Jet

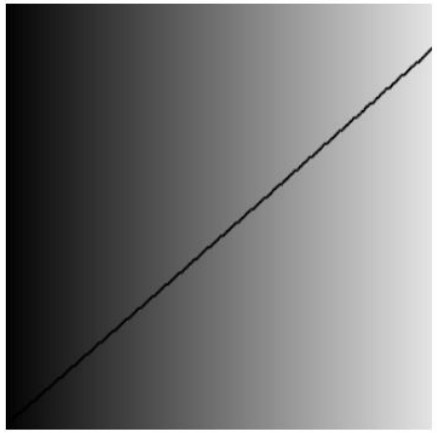
Inferno



Turbo



Viridis

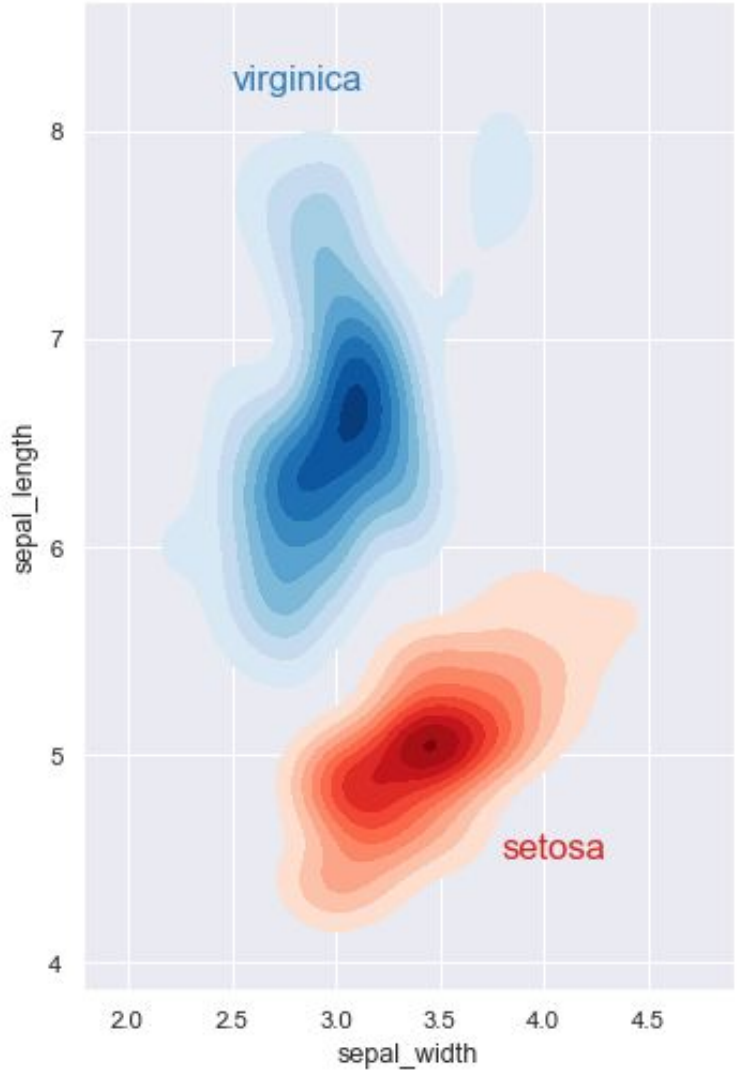


Inferno

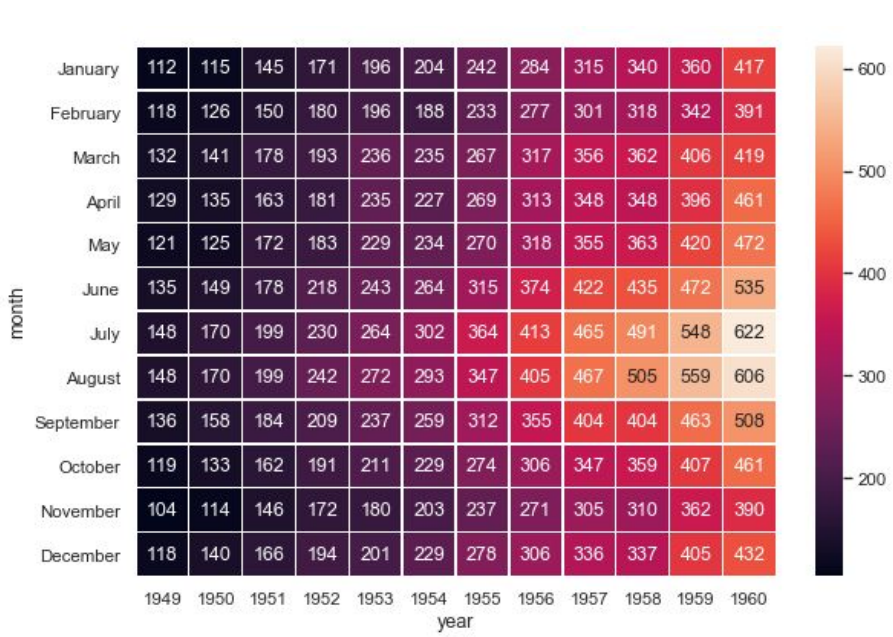
https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html

Use color to highlight data type

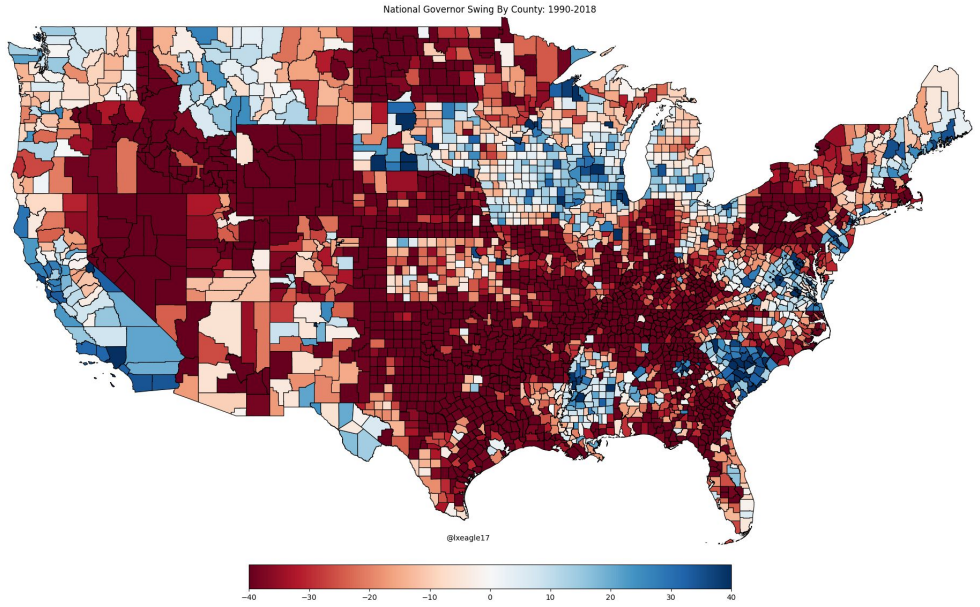
- **Qualitative**: Choose a qualitative scheme that makes it easy to distinguish between categories.
  - One category isn't "higher" or "lower" than another.
- **Quantitative**: Choose a color scheme that implies magnitude.
  - More on this in the next slide.
- The plot on the right has both!



Sequential vs. diverging colormaps for quantitative data

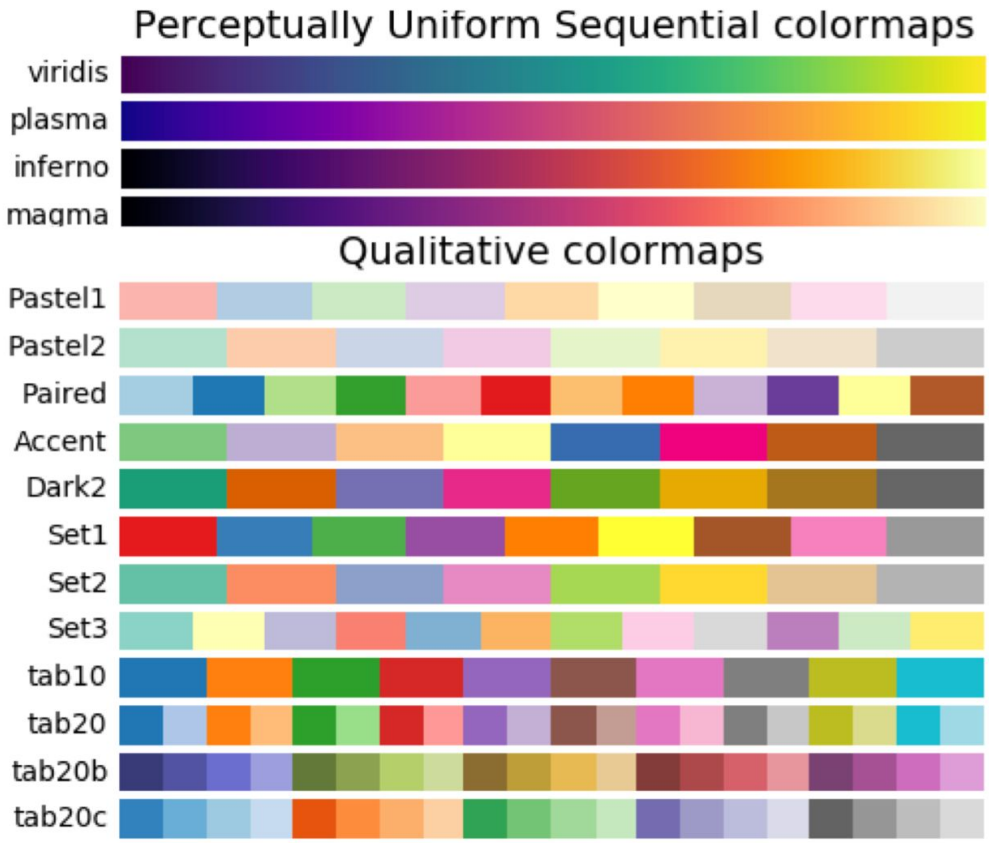


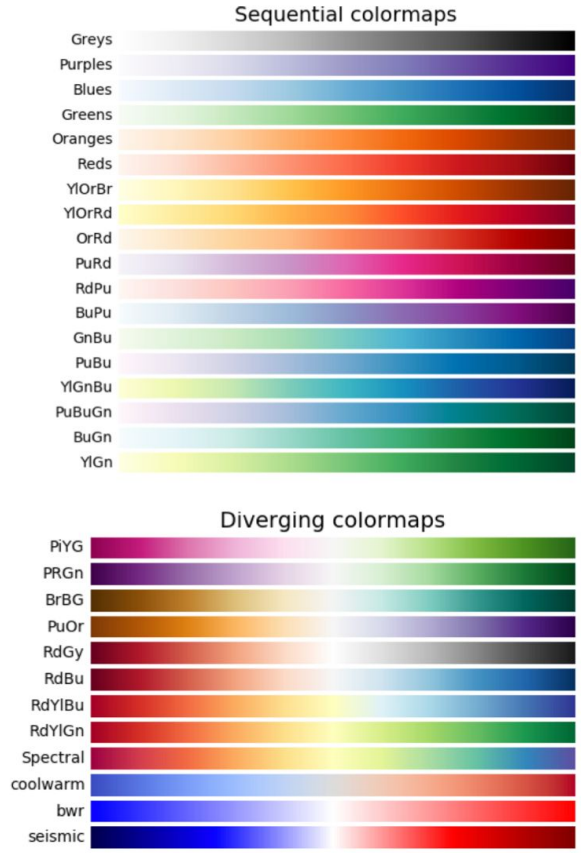
If the data progresses from low to high, use a **sequential** scheme where lighter colors are for more extreme values.



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

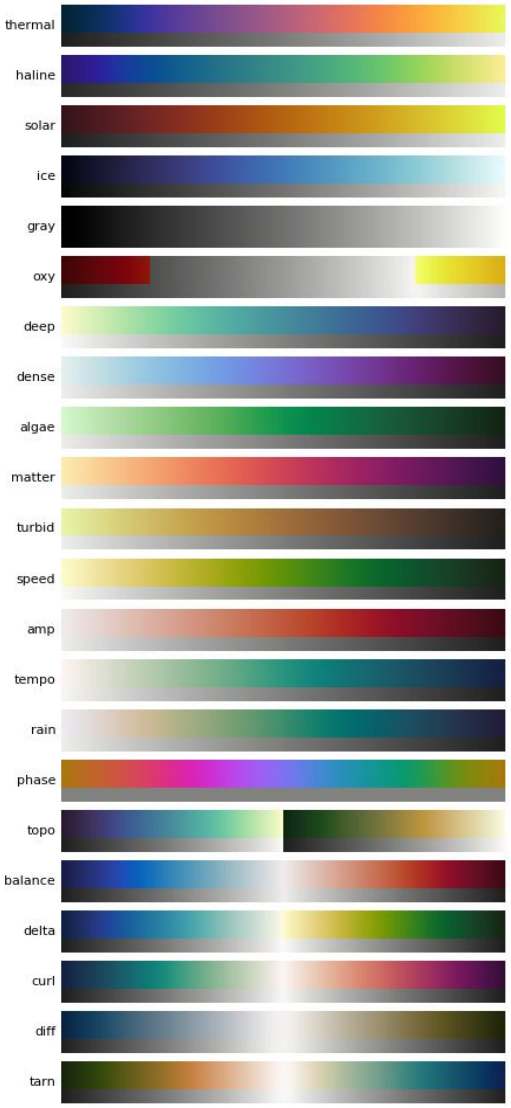
Default matplotlib colormaps

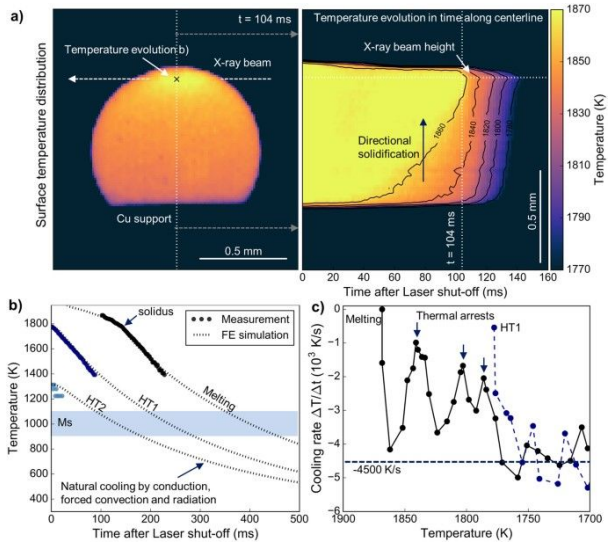




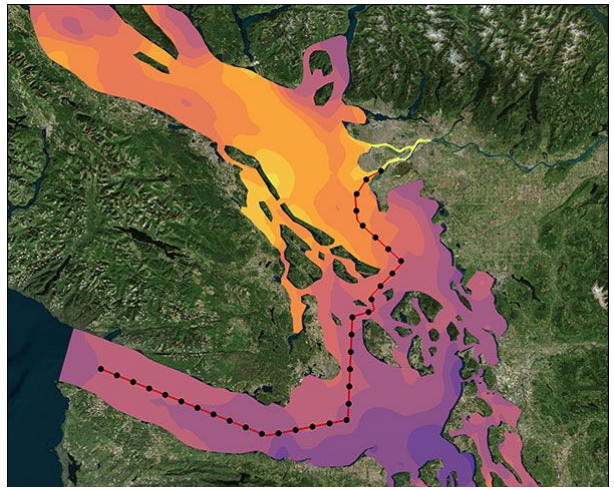
Taken from matplotlib documentation.

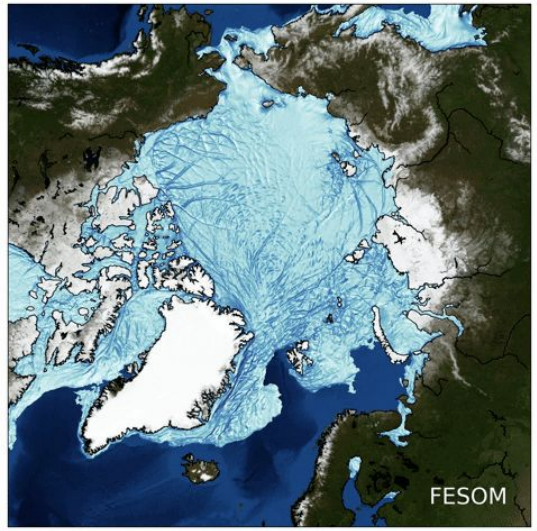
Domain specific colormaps: cmocean
(beautiful colormaps for oceanography, by Kristen Thyng)



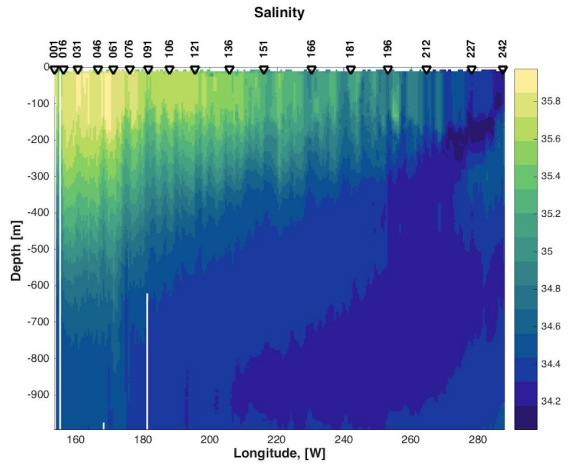


Thermal





Ice

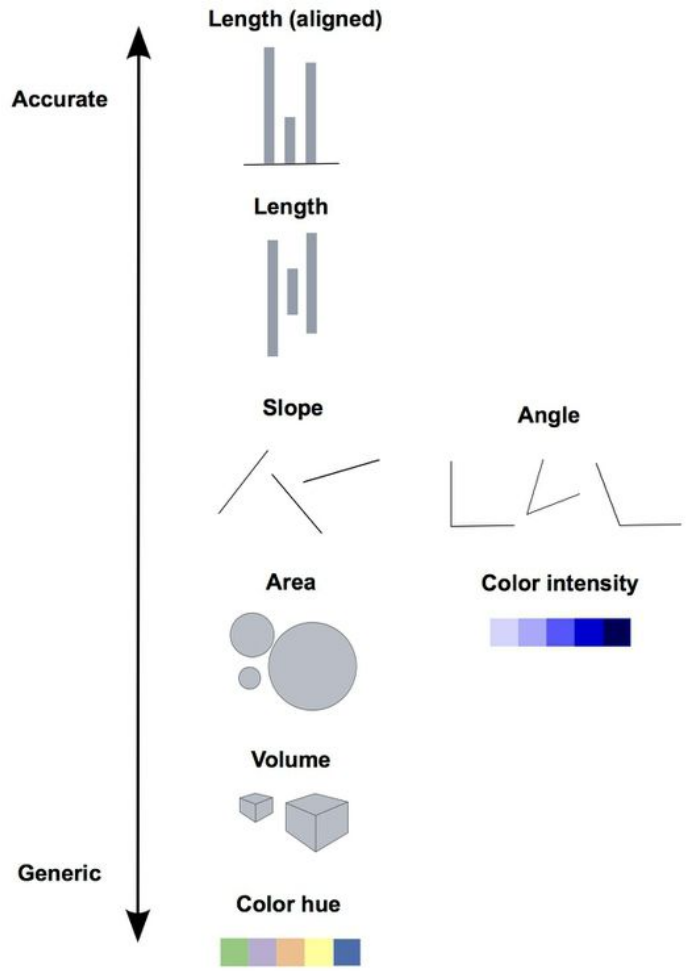


Haline

Extra reading

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

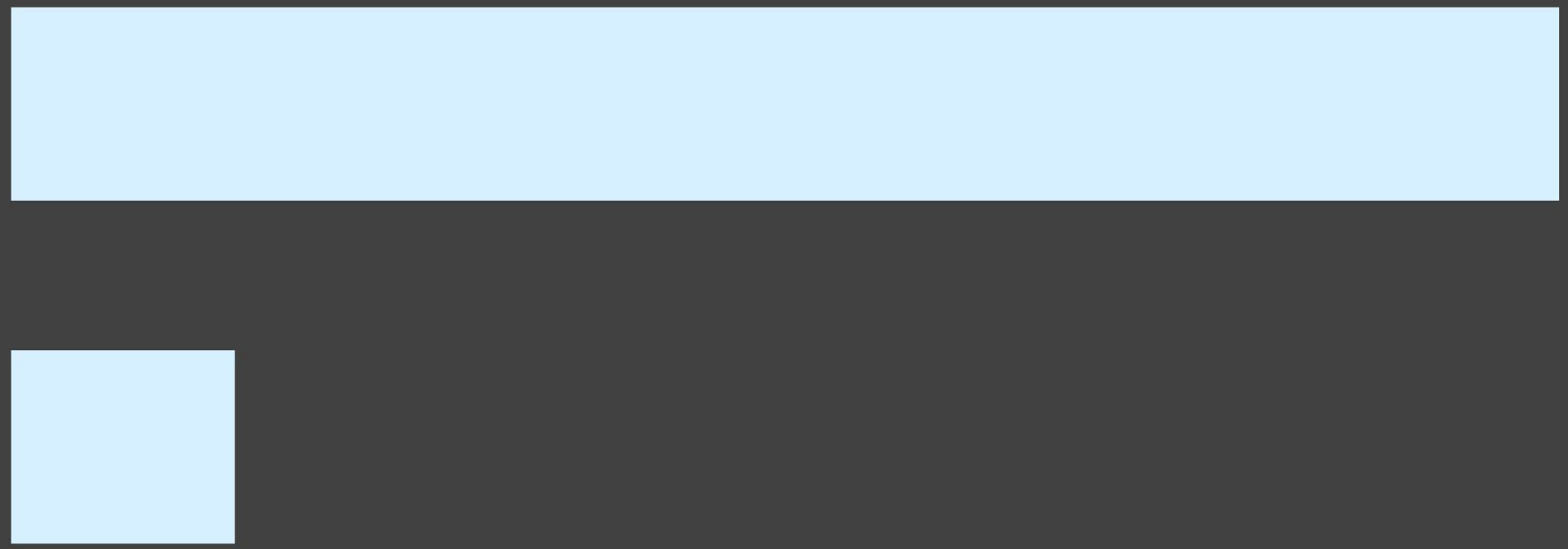
- Rainbow Colormap (Still) Considered Harmful - paper and presentation slides.
- https://eagereyes.org/basics/rainbow-color-map
- https://everydayanalytics.ca/2017/03/when-to-use-sequential-and-diverging-palettes.html
- https://web.natur.cuni.cz/~langhamr/lectures/vtfg1/mapinfo_2/barvy/colors.html



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

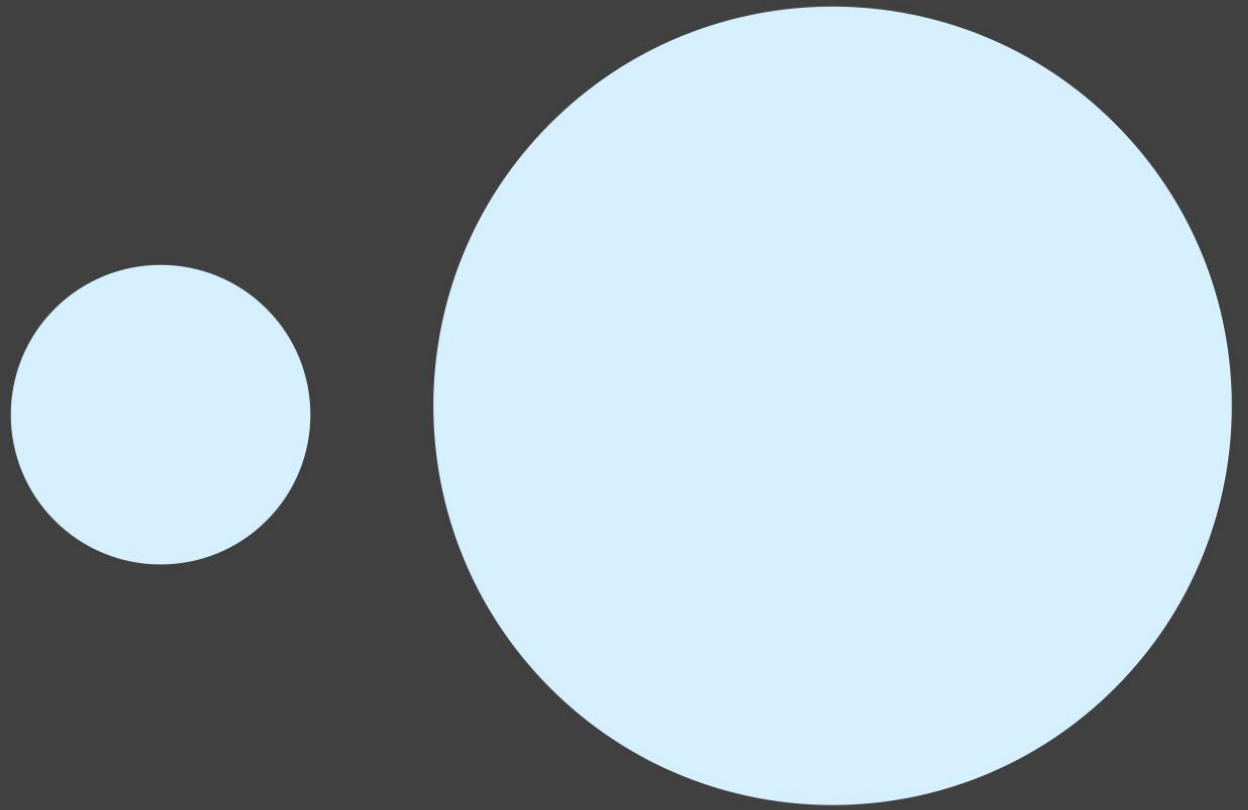
Perception of Markings

The point is that perceptual delta should be uniform where jet/rainbow colormaps do not has such uniformity.

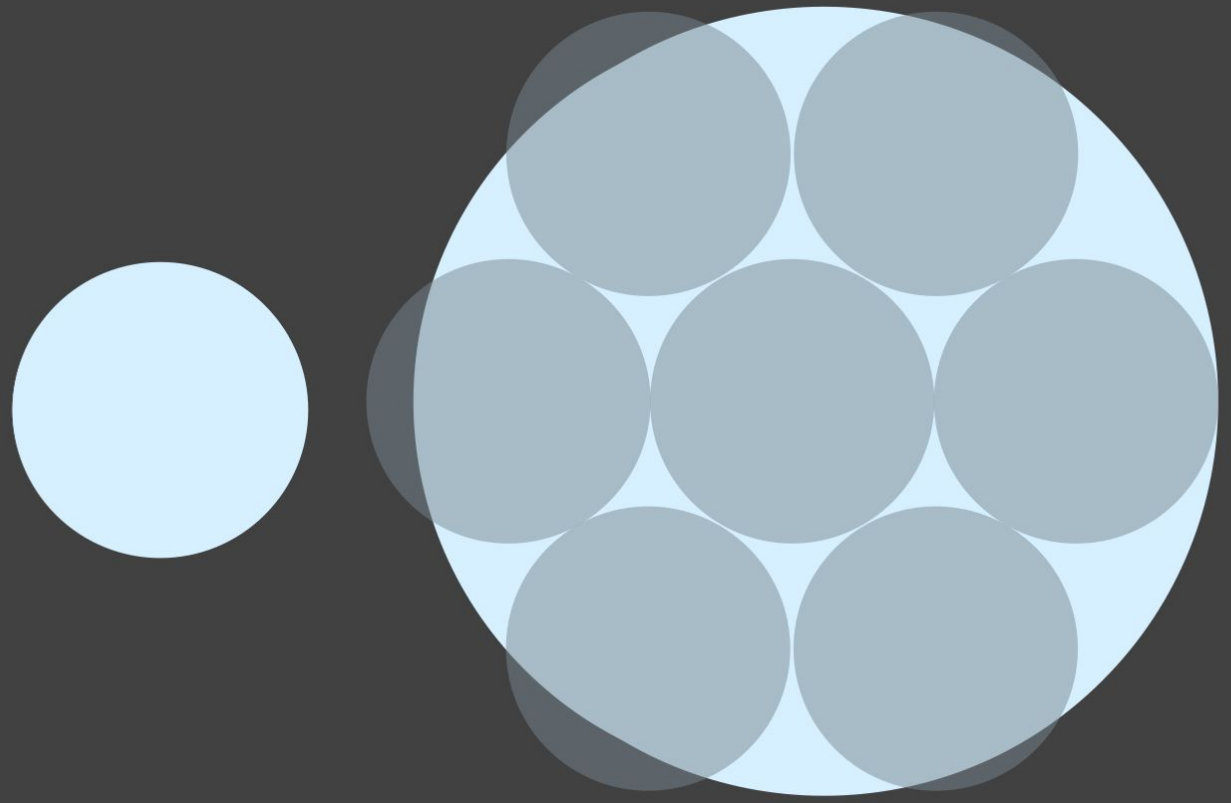


How much longer is the top bar?

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

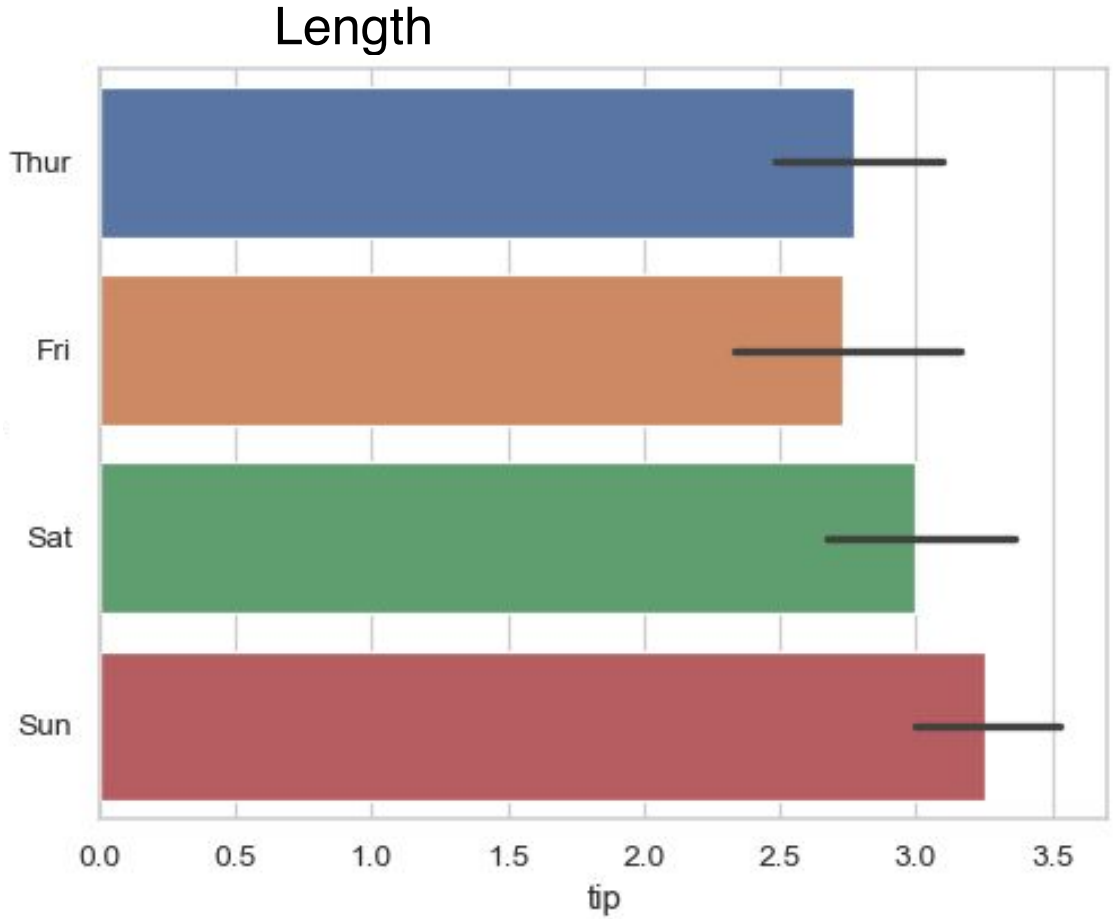


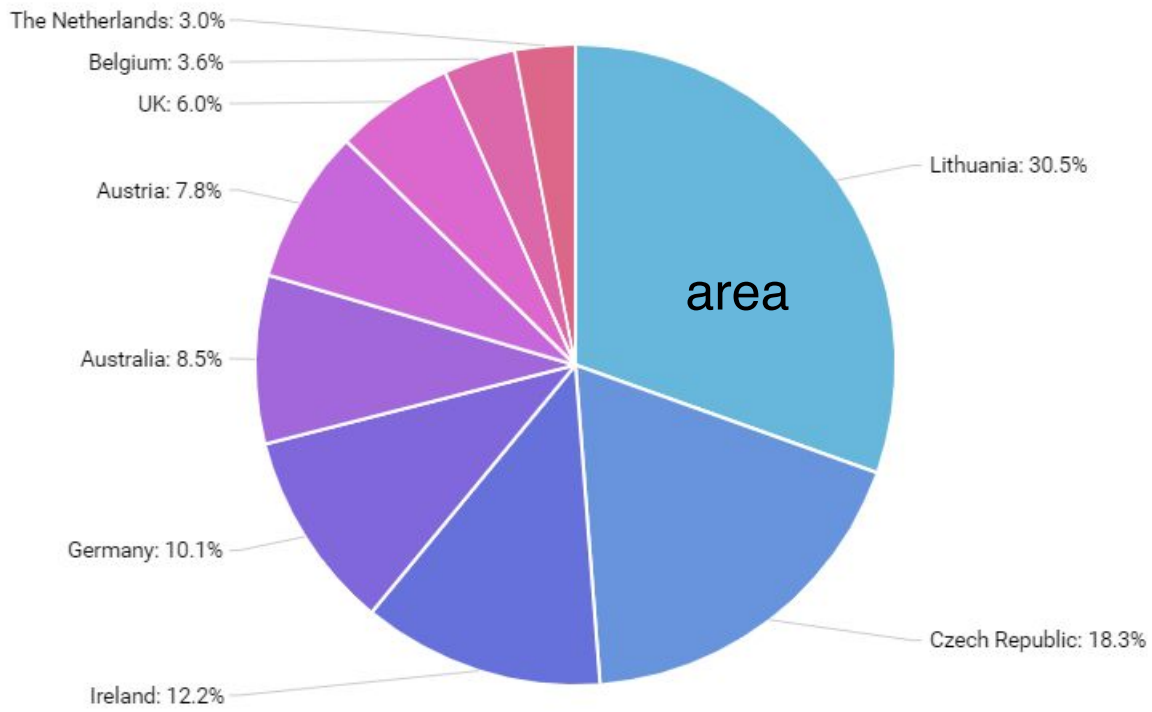
How much bigger is the big circle?



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

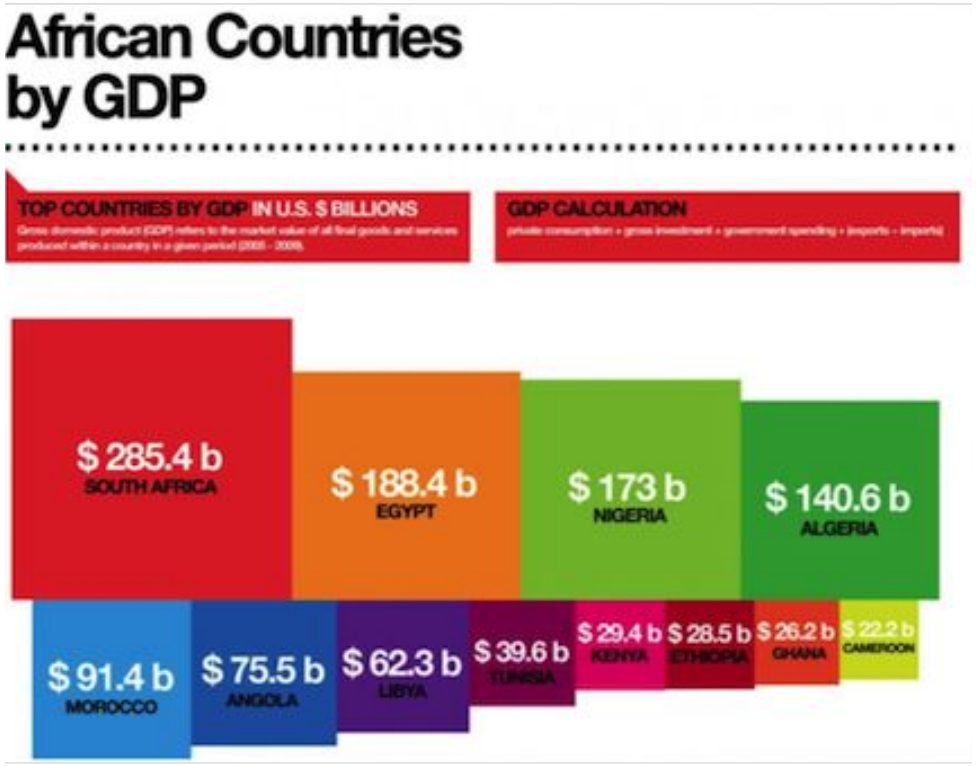
Lengths are easy to distinguish; angles are hard

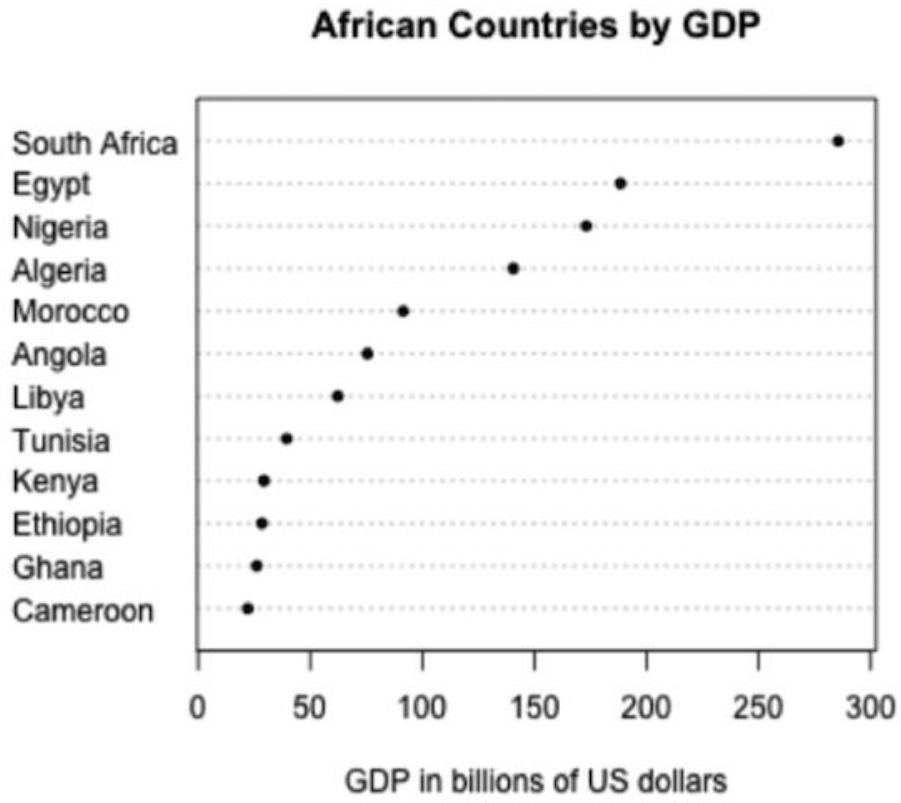




Don’t use pie charts! Angle judgements are inaccurate.

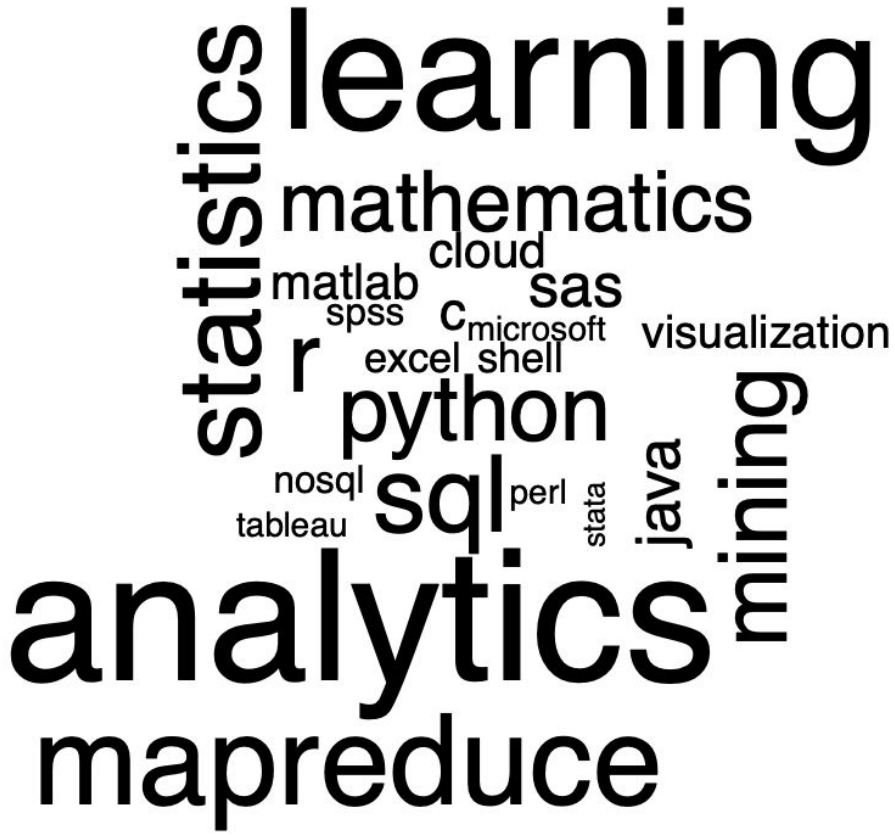
Areas are hard to distinguish



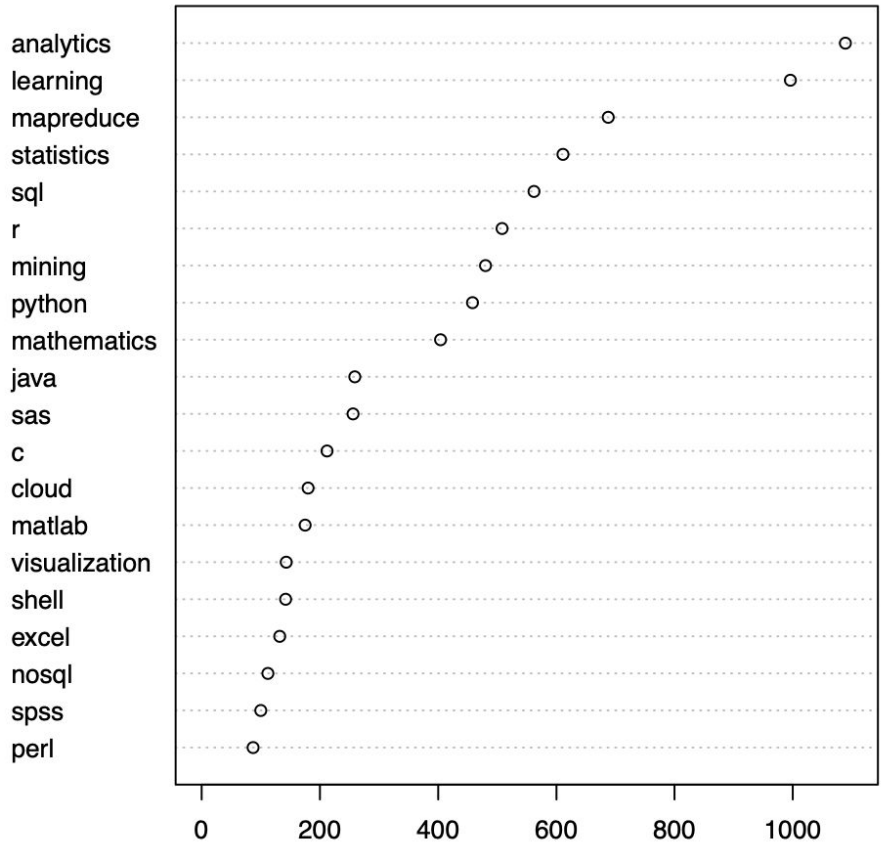


Avoid area charts! Area judgements are inaccurate. (For instance, South Africa has twice the GDP of Algeria, but that isn't clear from the areas.)

Areas are hard to distinguish



Scatter Plot is easy to distingush

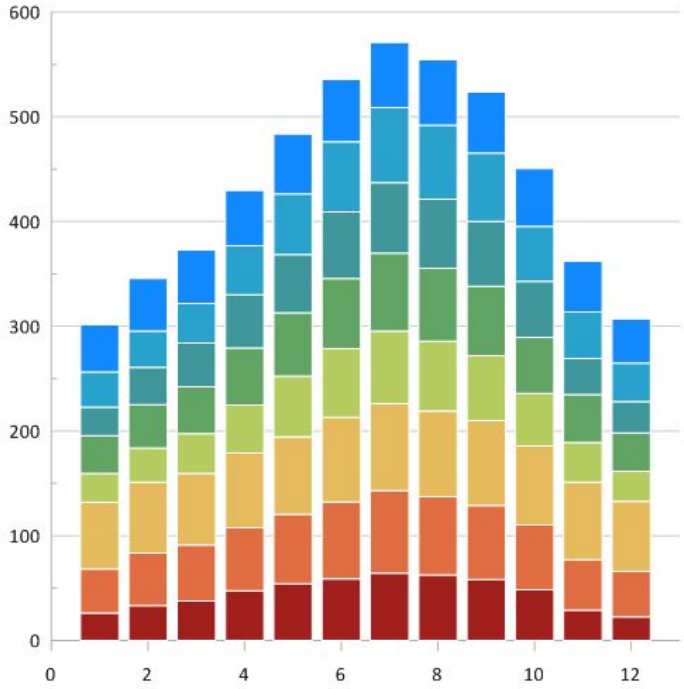


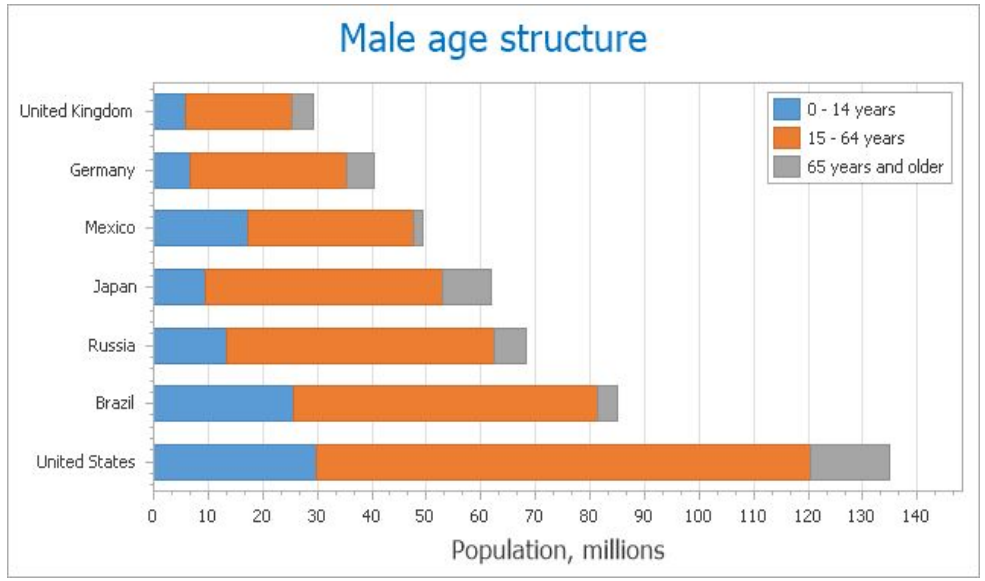
Avoid word clouds too! It's hard to tell the area taken up by a word.

Avoid jiggling the baseline

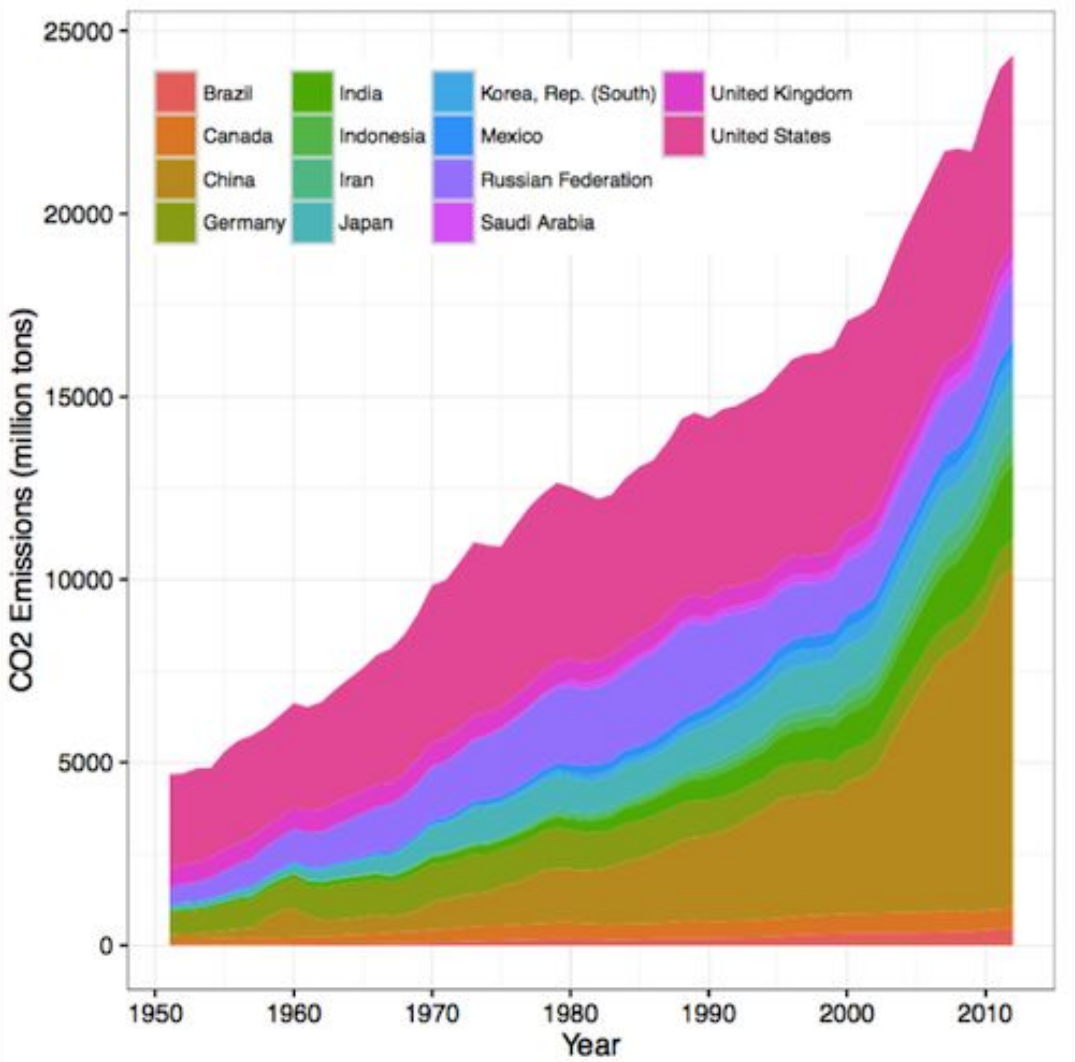
Stacked bar charts, histograms, and area charts are hard to read because the baseline moves.

- 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.

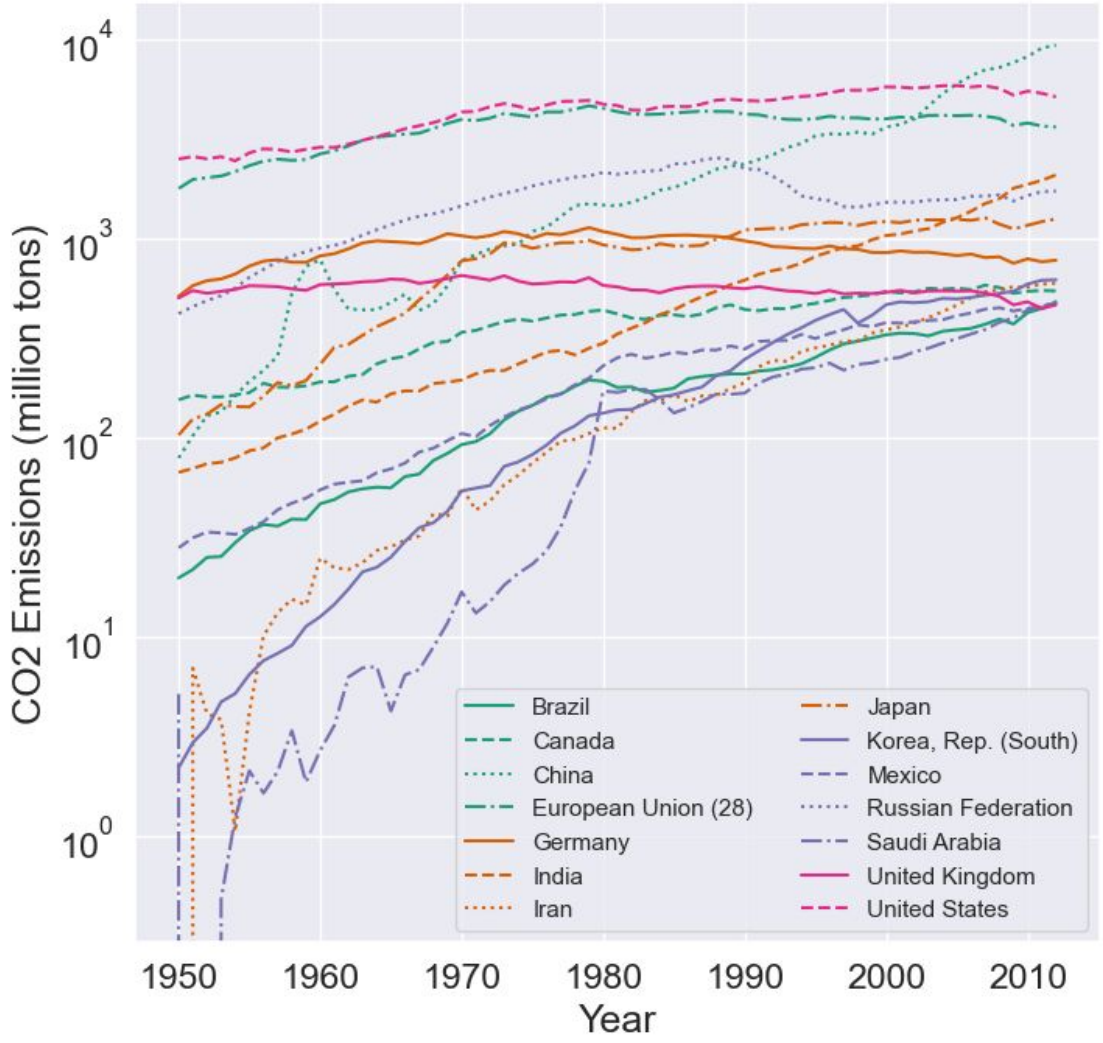




Avoid jiggling the baseline



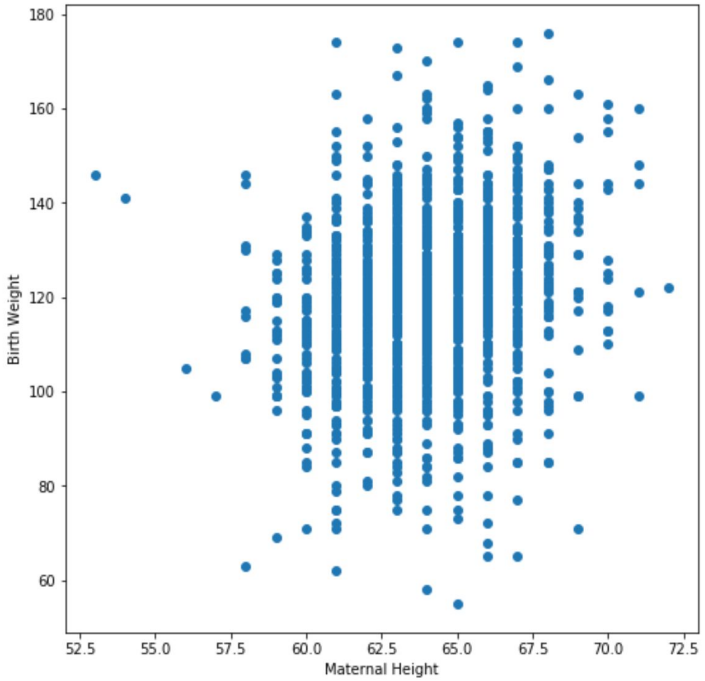
The Line Chart is better

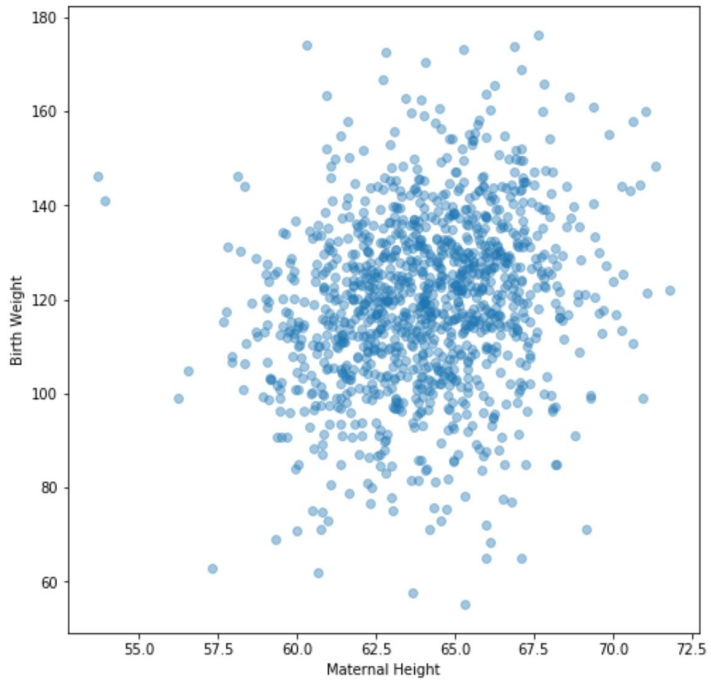


Here, by switching to a line plot, comparisons are made much easier.

Related – overplotting

Undistingushable





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

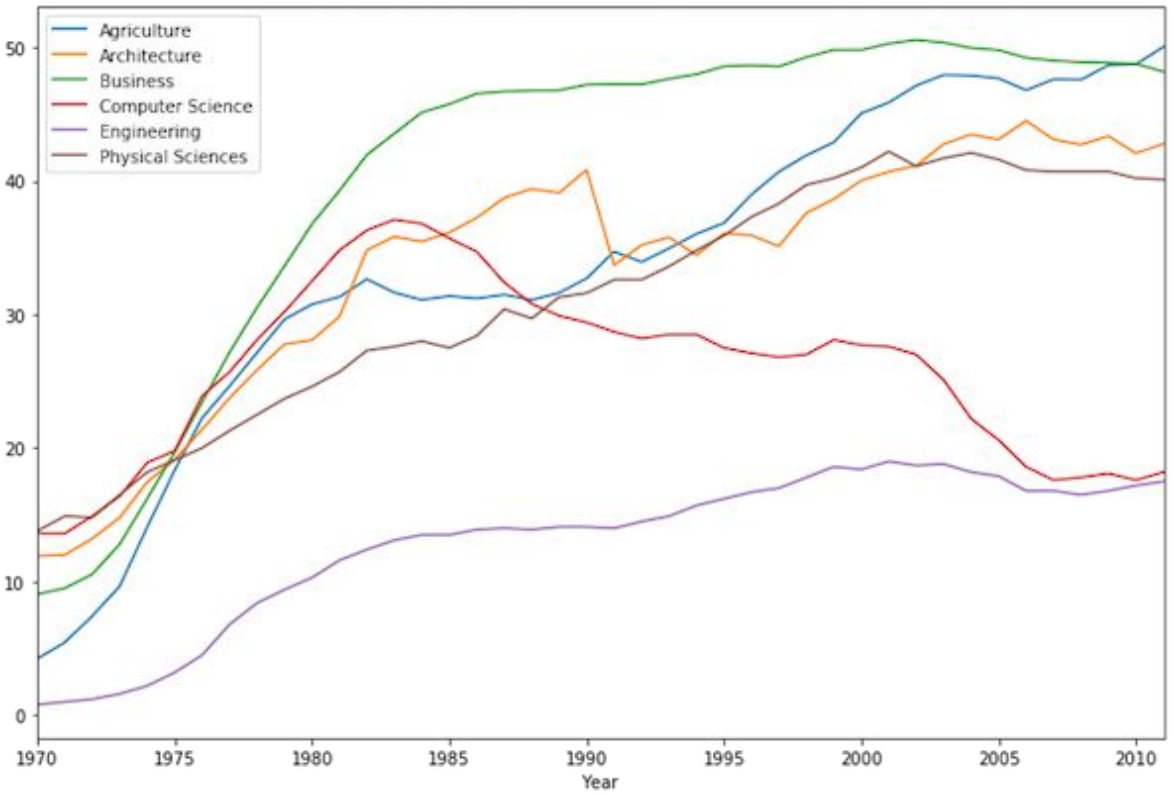
- Many on top of one another.
- Observations only on lattice points.

Some solutions:

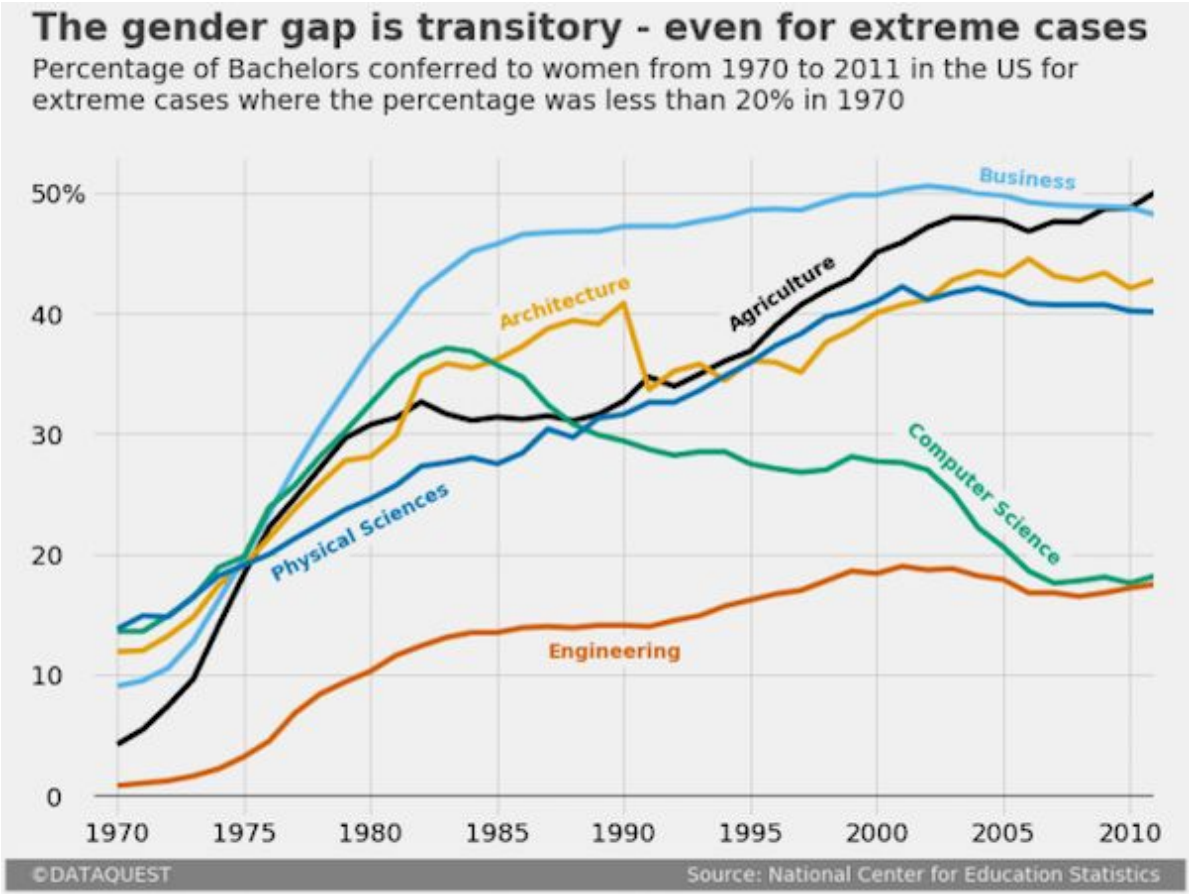
- Add small random noise to both x and y ("jittering").
- Make points smaller (wouldn't help here though).

Context

Y Axis Not Here,
and we don't know what chart is



Only provide X axis



Add context directly to plot

A publication-ready plot needs:

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

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

Captions

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

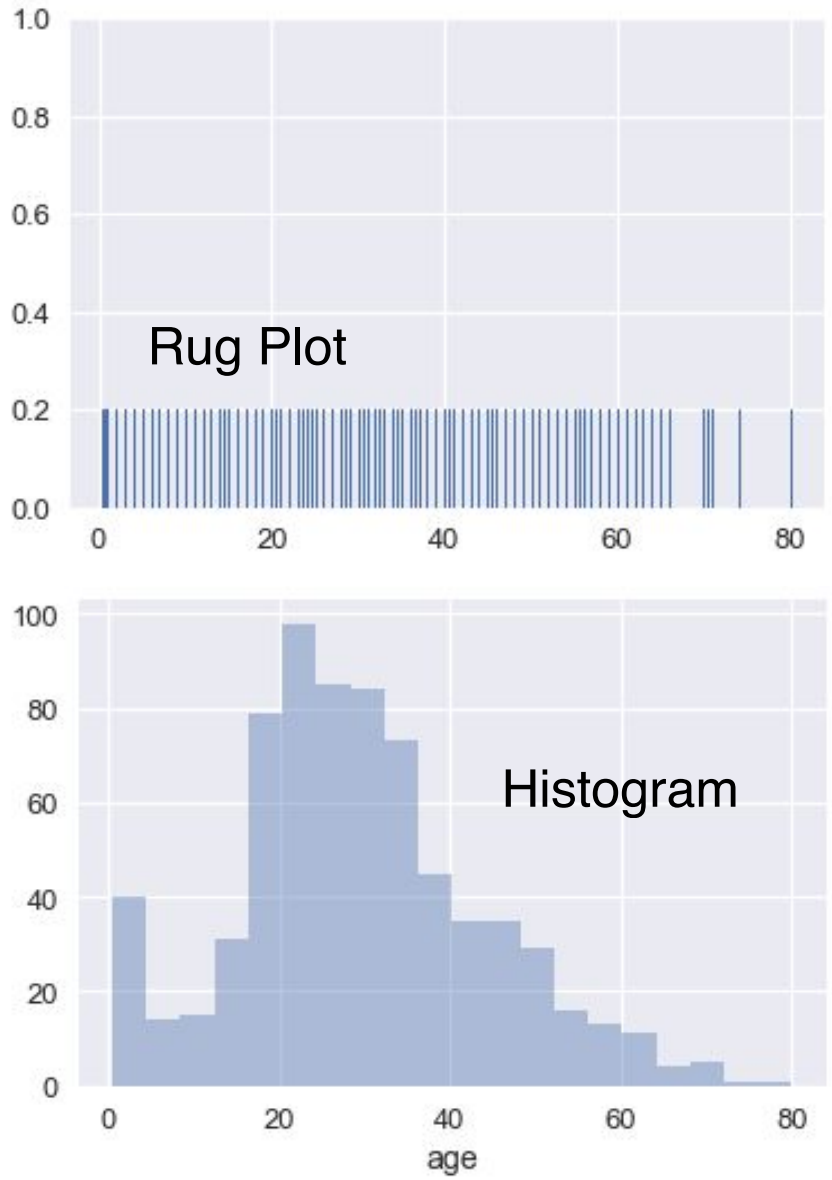
Captions should be:

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

Smoothing

Smoothing

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

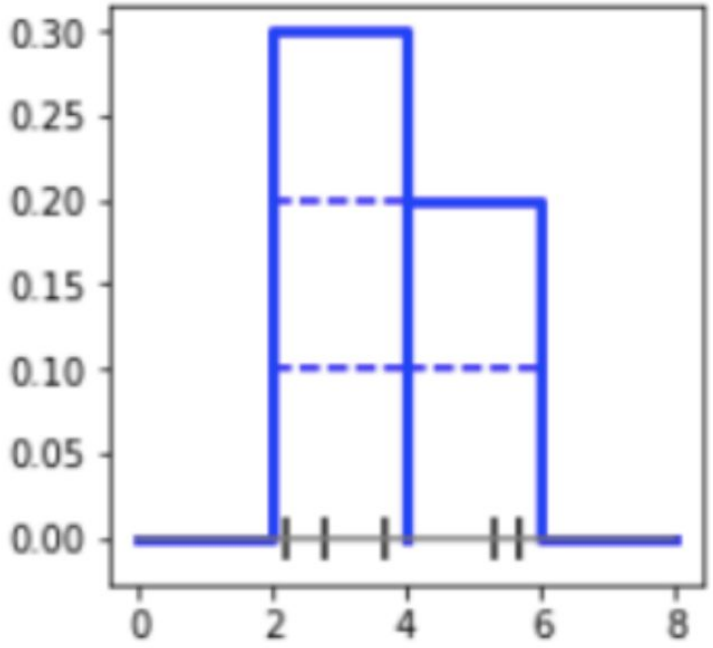


Spreading proportion uniformly

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

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

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



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

Kernel density estimation (KDE)

현재 이해가 잘 안되는 부분

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

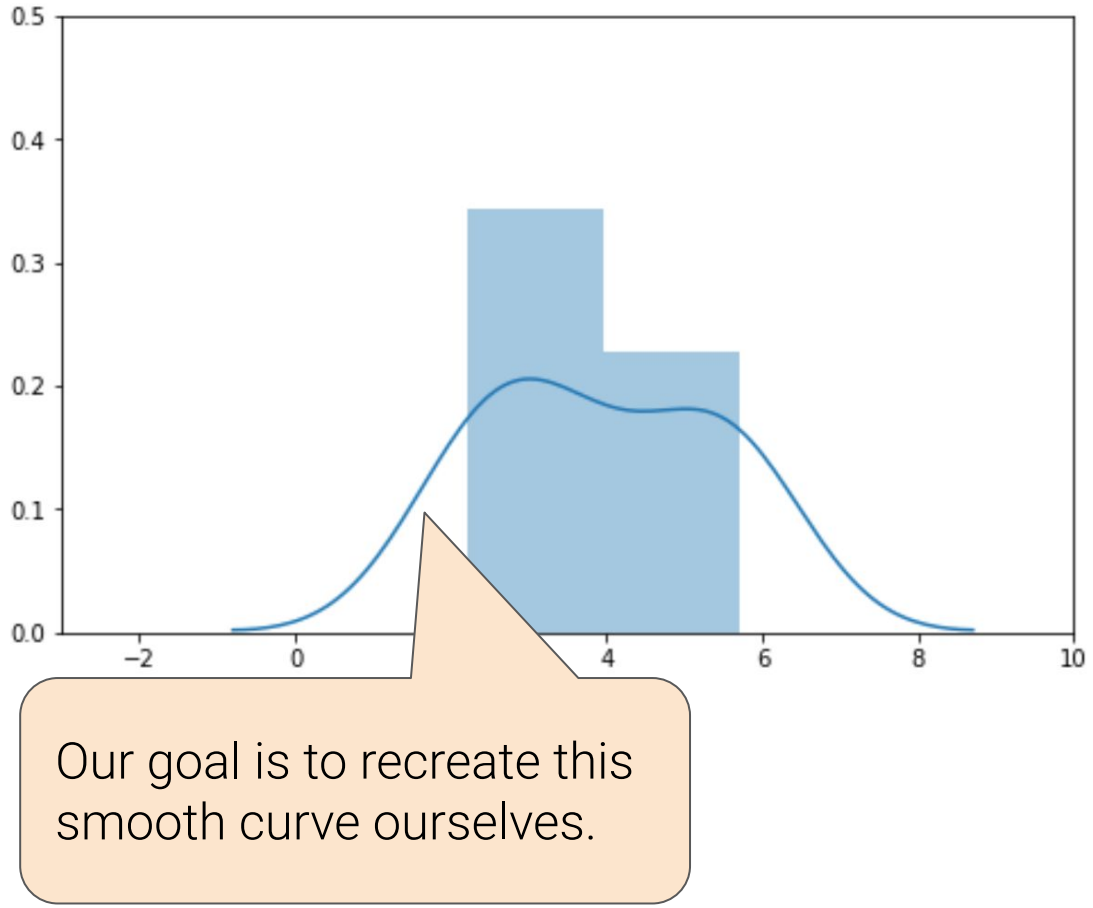
- Just like a histogram, a density function's total area must sum to 1.

To create a KDE:

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

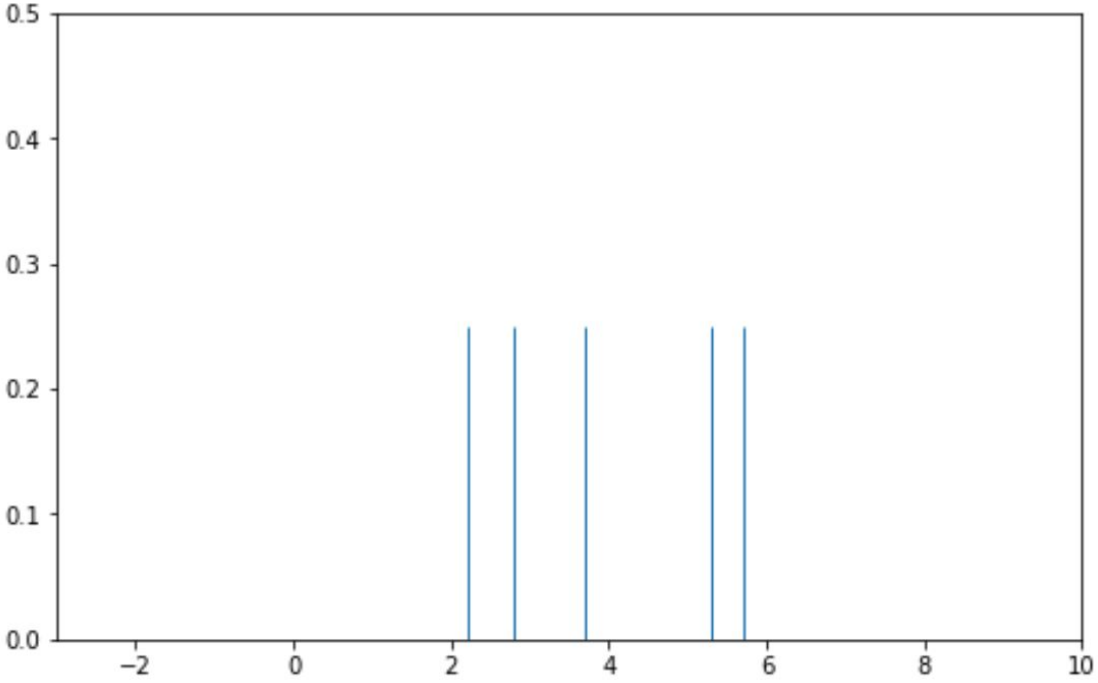
We also need to choose a kernel and **bandwidth.**

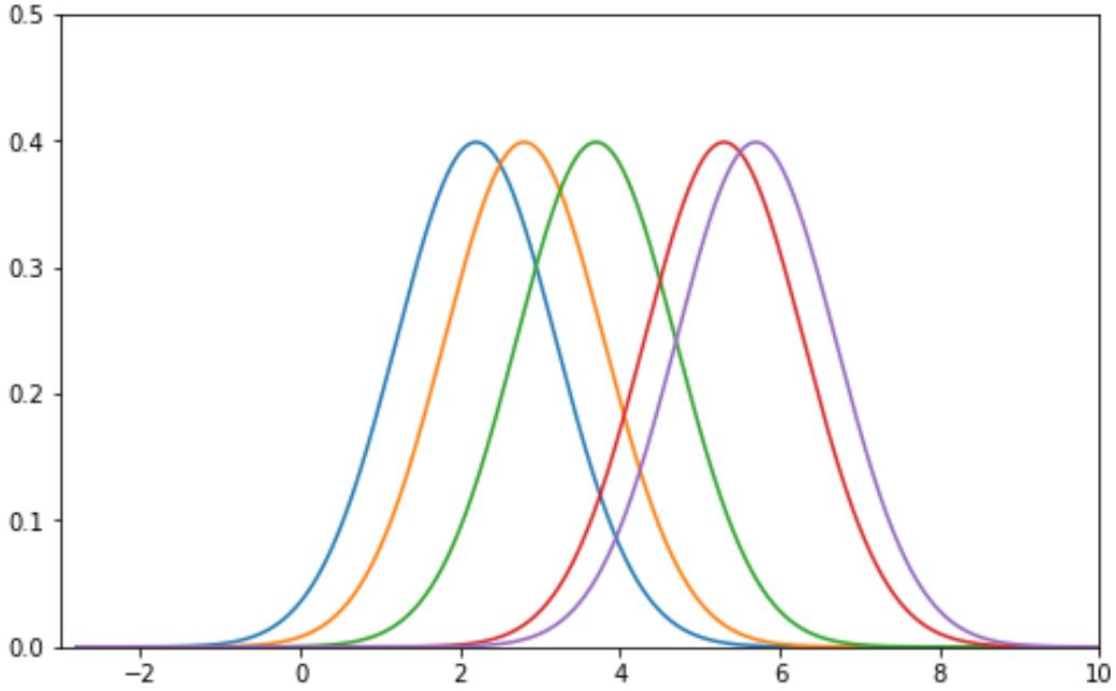
히스토그램에서 만드는 것 아니다
데이터로 만드는 것



Step 1 – place a kernel at each data point

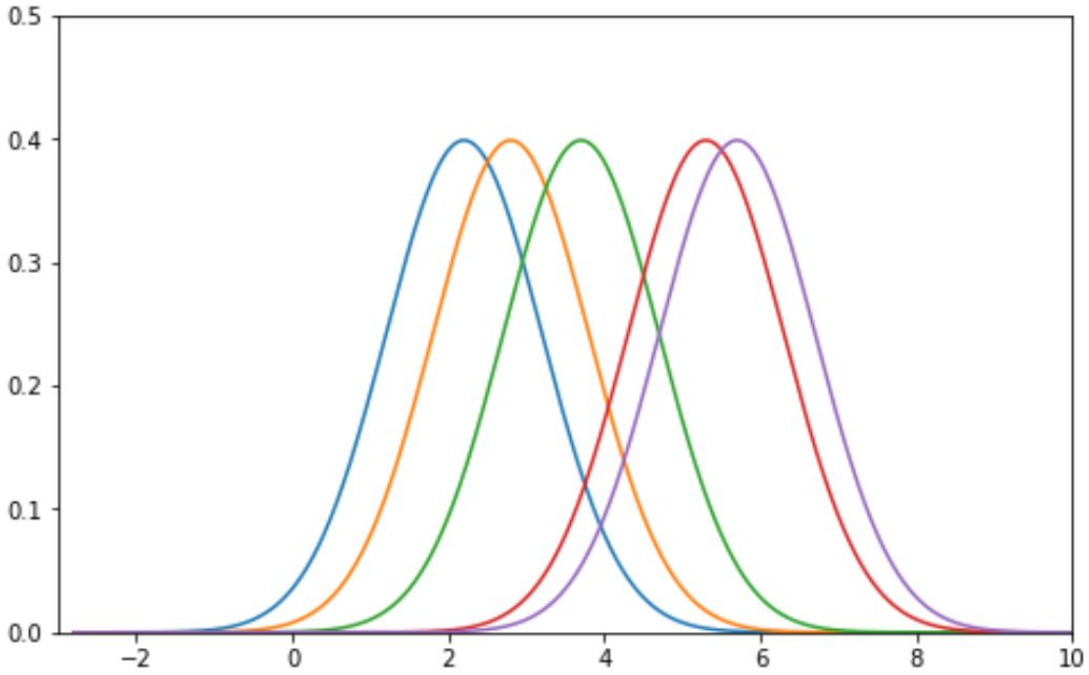
At each of our 5 points (depicted in the rug plot on the left), we've placed a **Gaussian** kernel with **alpha = 1**. The idea is that there is a higher density near the points we've already seen.

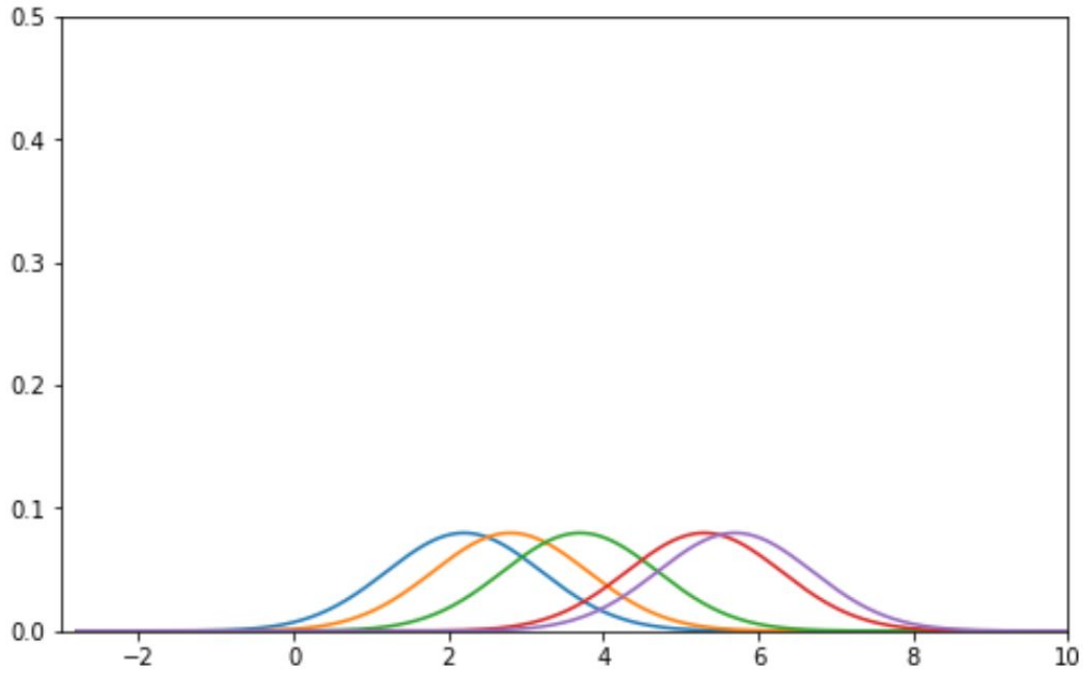




Step 2 – normalize kernels

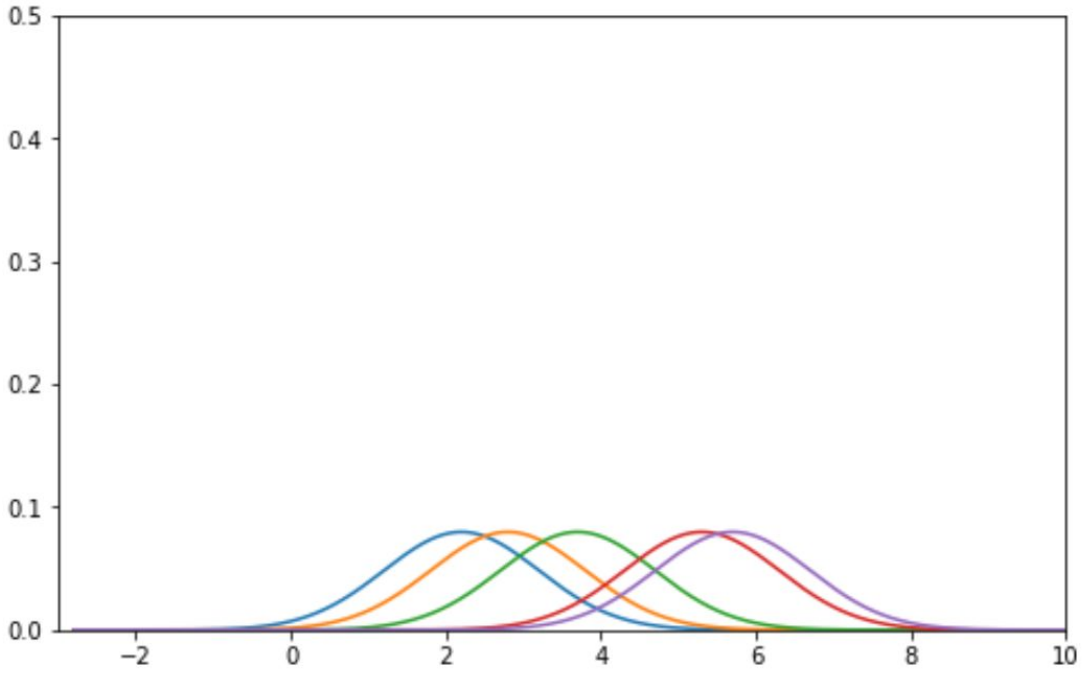
In Step 3, we will be summing each of these kernels. We want the result to be a valid density, that has area 1. Right now, we have 5 different kernels, each with an area 1. So, we **multiply each by 1/5**.

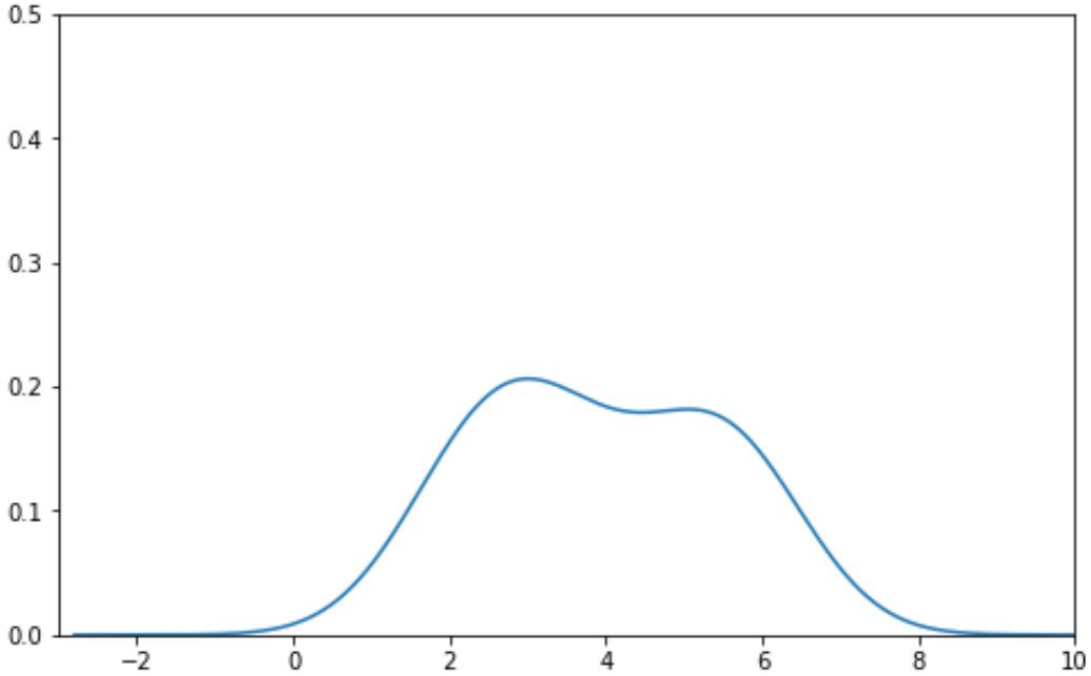




Step 3 – sum kernels

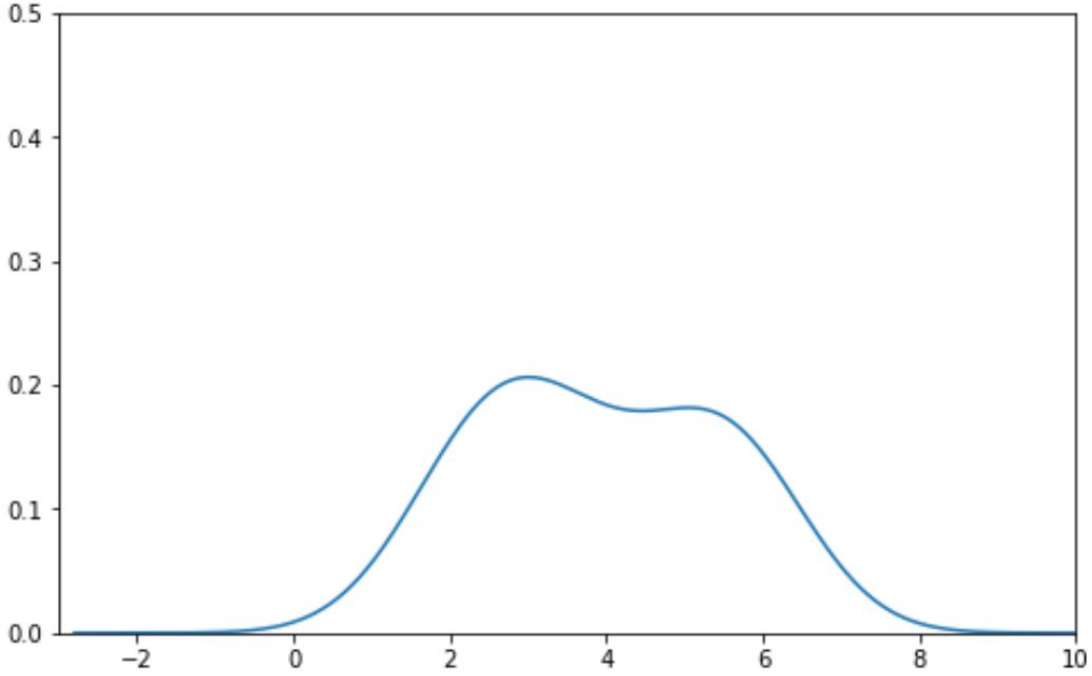
Our **kernel density estimate** is the **sum of the normalized kernels at each point.** It is depicted below on the right.

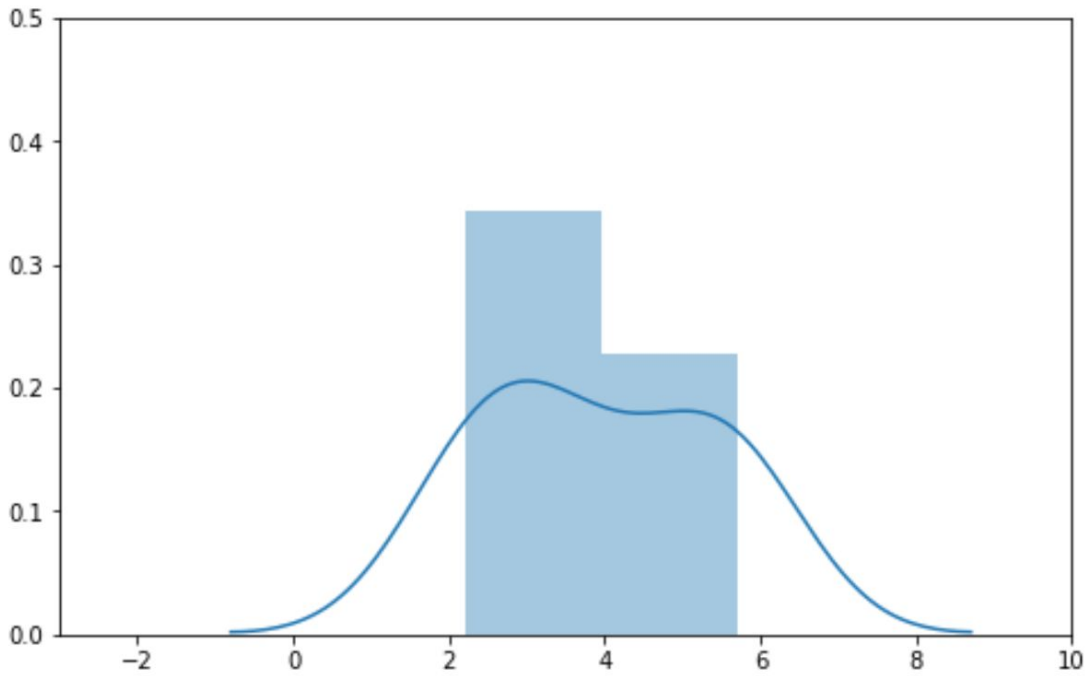




Kernel density estimates

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





Kernels

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

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

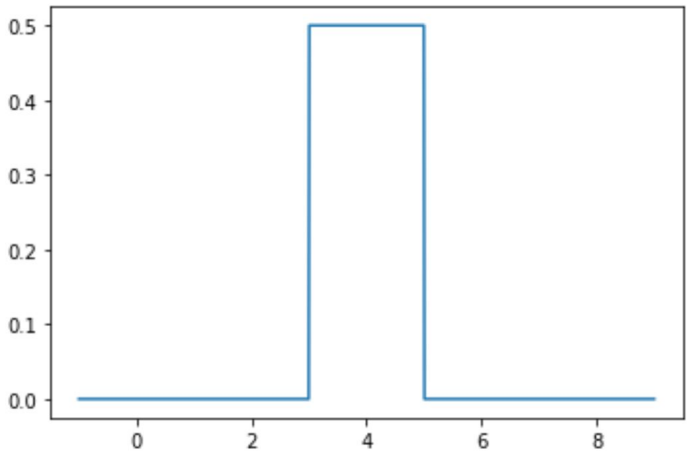
가우시안 sum 형태

Kernels

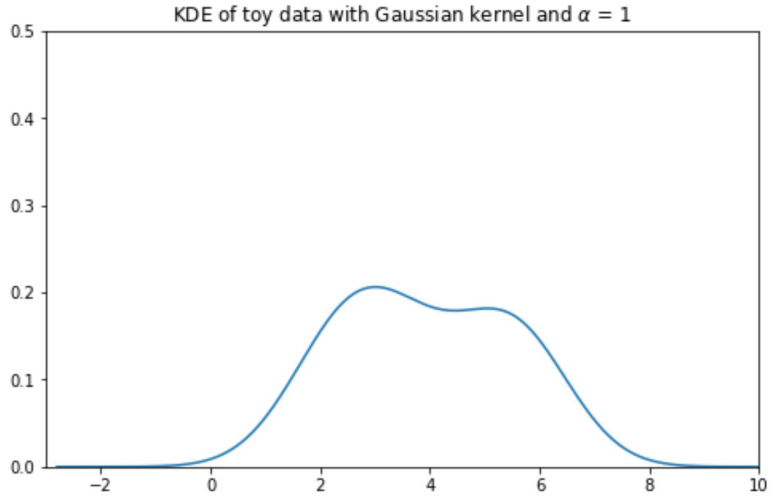
- Another common kernel is the **boxcar** kernel.
  - It assigns uniform density to points within a "window" of the observation, and 0 elsewhere.
  - Resembles a histogram… sort of.

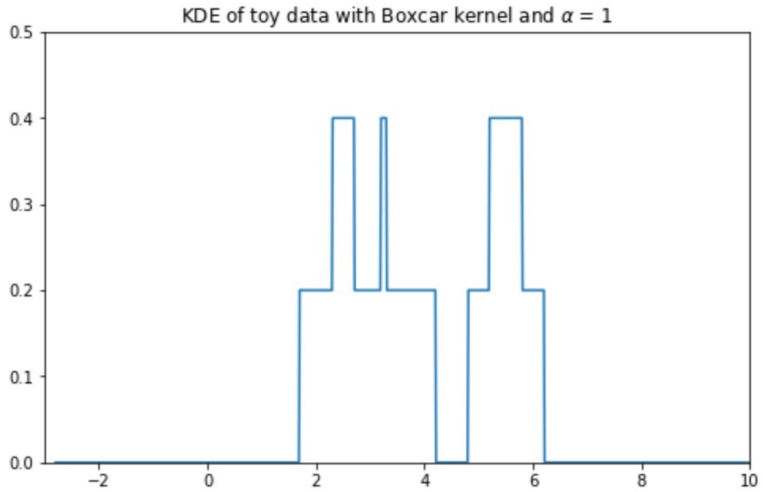
$$K_\alpha(x, x_i) = \begin{cases} \frac{1}{\alpha}, & |x - x_i| \leq \frac{\alpha}{2} \\ 0, & \text{else} \end{cases}$$

가우시안 대신 박스 plot으로 Gaussian -> 정규 분포

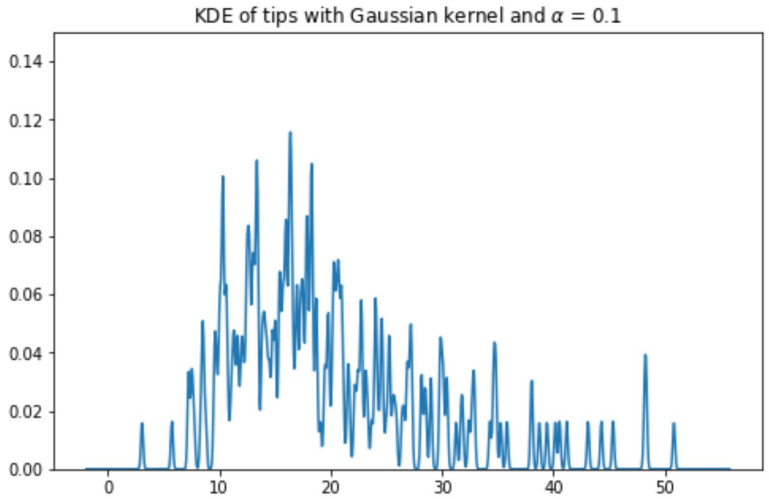


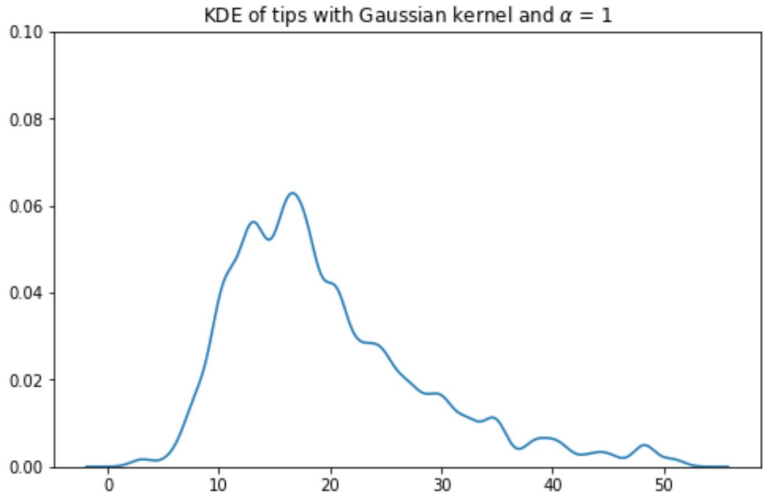
A boxcar kernel centered on $x_i = 4$ with $\alpha = 2$.

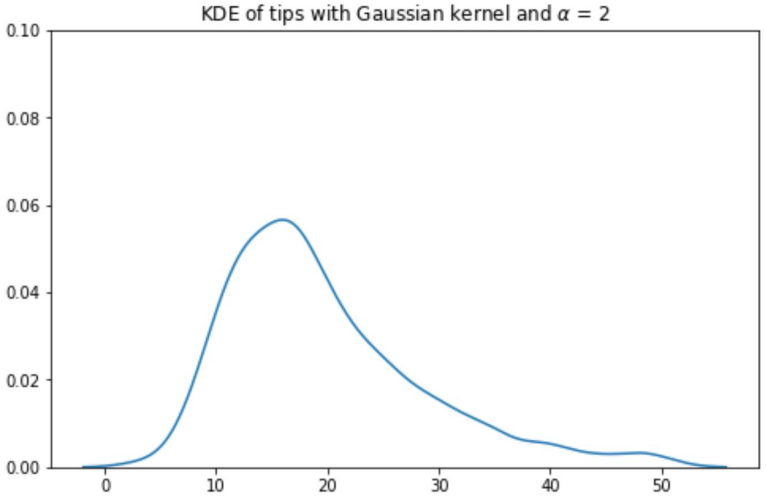


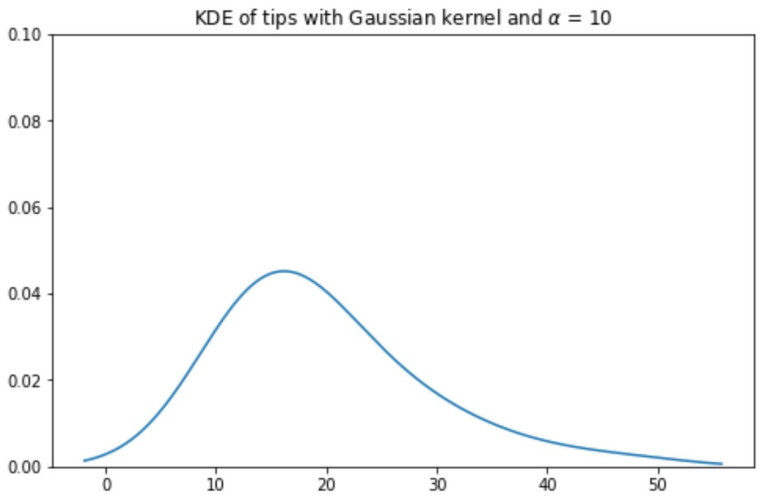


Effect of bandwidth on KDEs



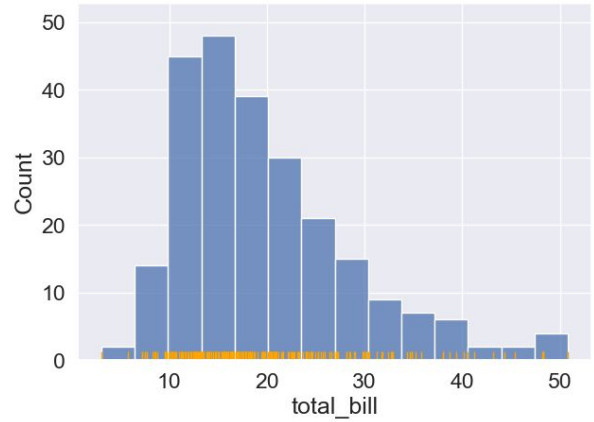






Bandwidth is analogous to the width of each bin in a histogram.

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



Summary of KDE

현재 이해가 잘 안되는 부분

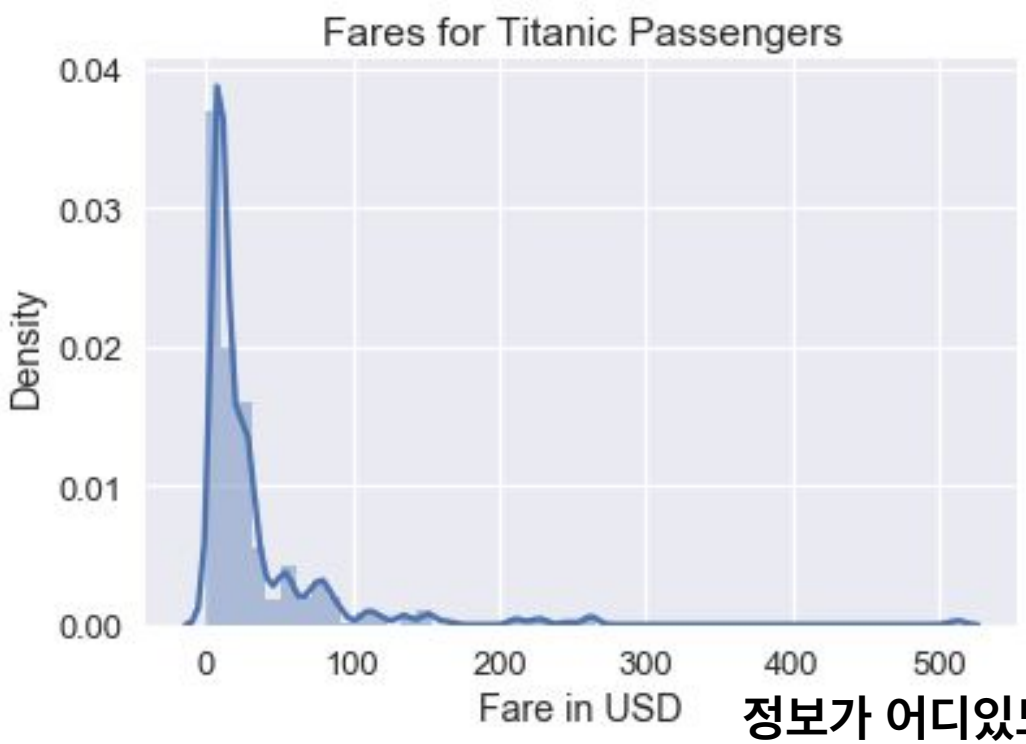
$$f_\alpha(x) = \frac{1}{n}\sum_{i=1}^{n} K_\alpha(x, x_i)$$

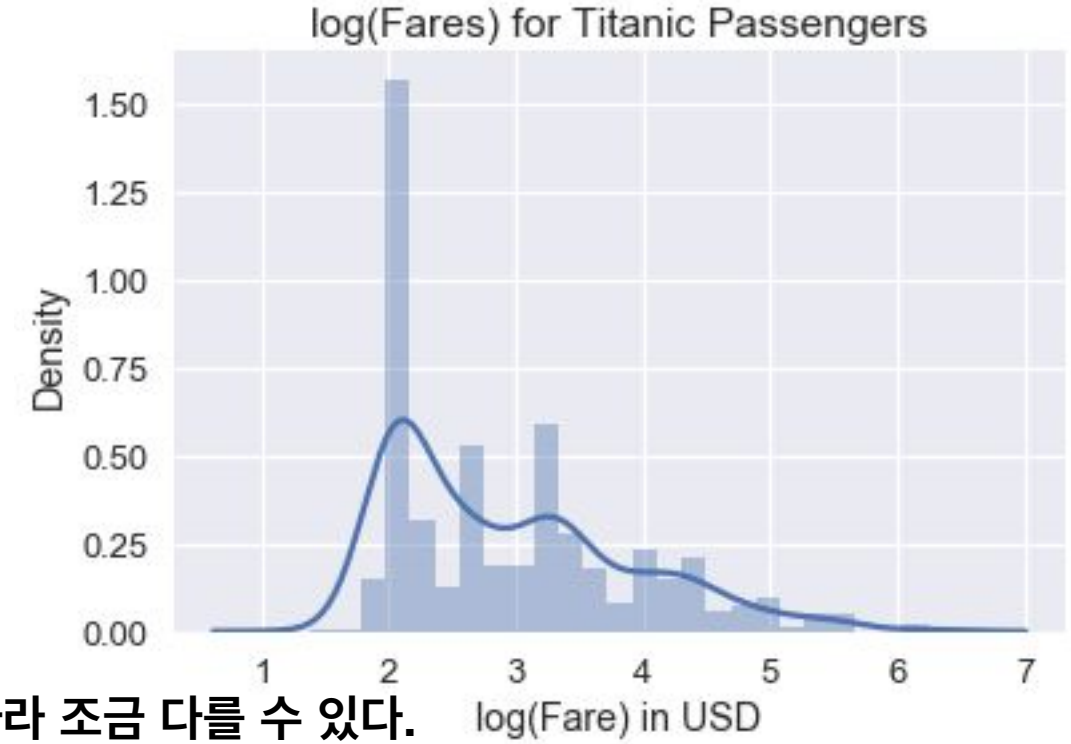
The “KDE formula” is above.

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

Transformations

Transforming data can reveal patterns





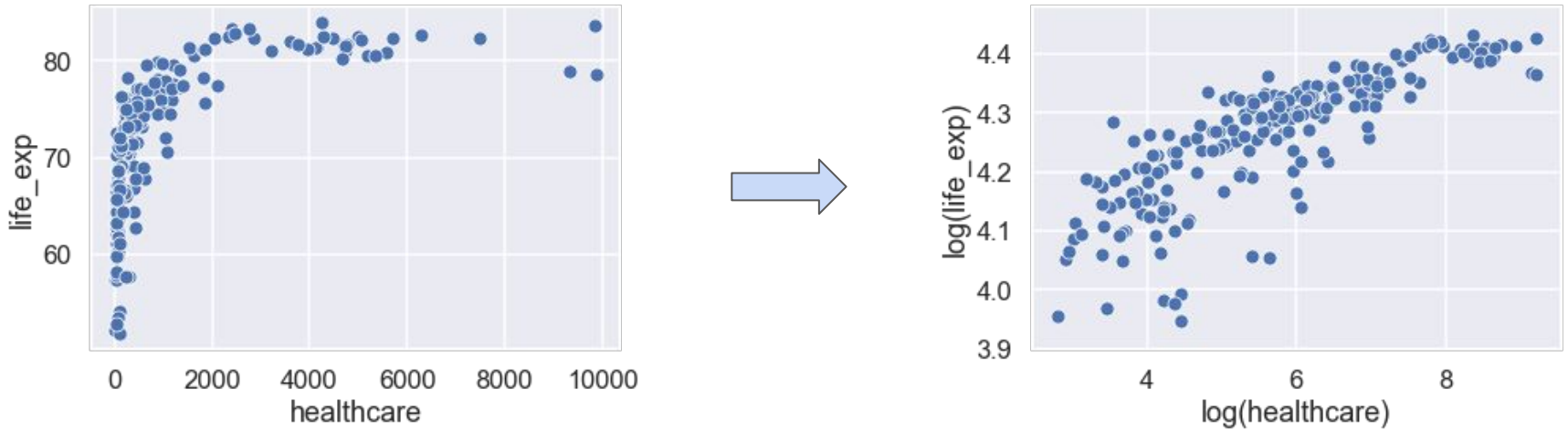
정보가 어디있느냐에 따라 조금 다를 수 있다.
0 ~ 100 밀도가 100 ~ 500보다 중요하다는 전제
0 ~ 100 사이의 값이 더 잘 보여준다는 정도?

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

Why straighten relationships?

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

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



Log of y-values

로그를 붙여서 liear로 만들 수 있다면
역추적해서 실제 관계를 복원하는 가능하는 것이 가능하다

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

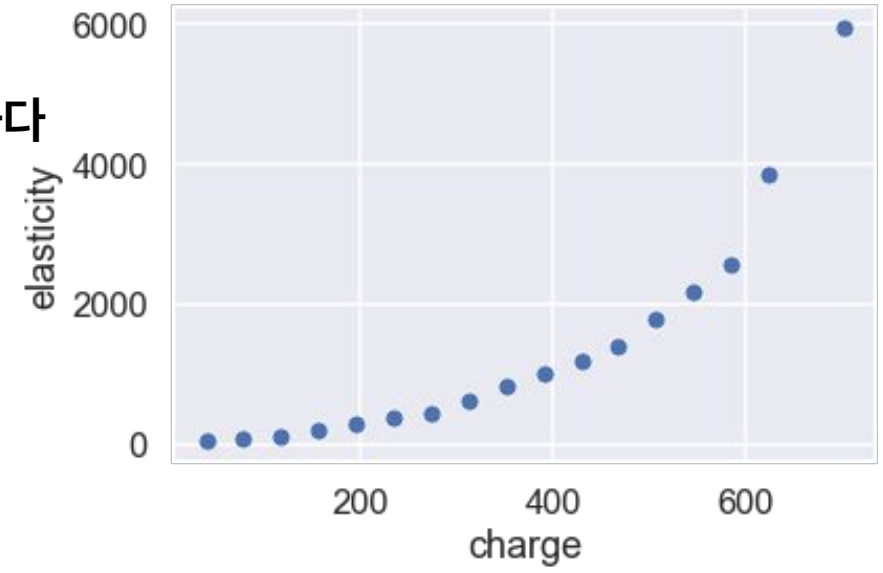
$$\log y = ax + b$$

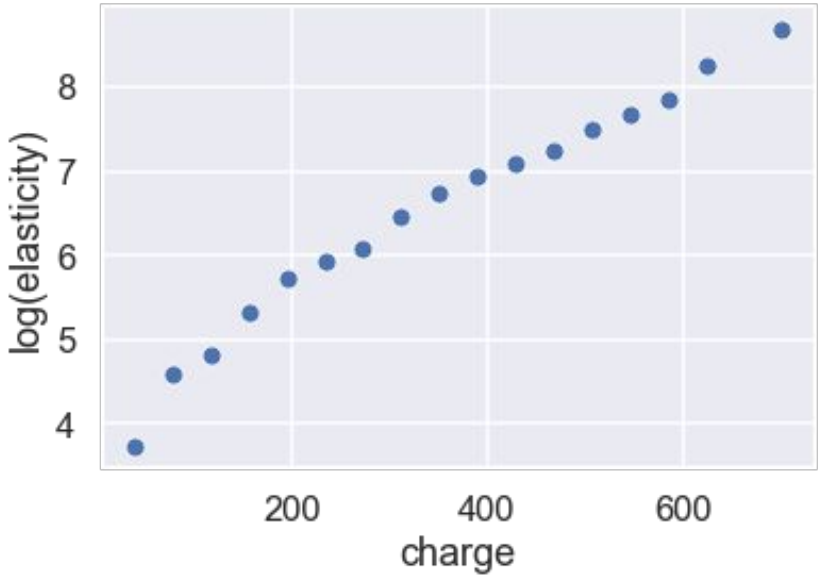
Working backwards:

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

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

지수 사이의 관계





Log of both x and y-values

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

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

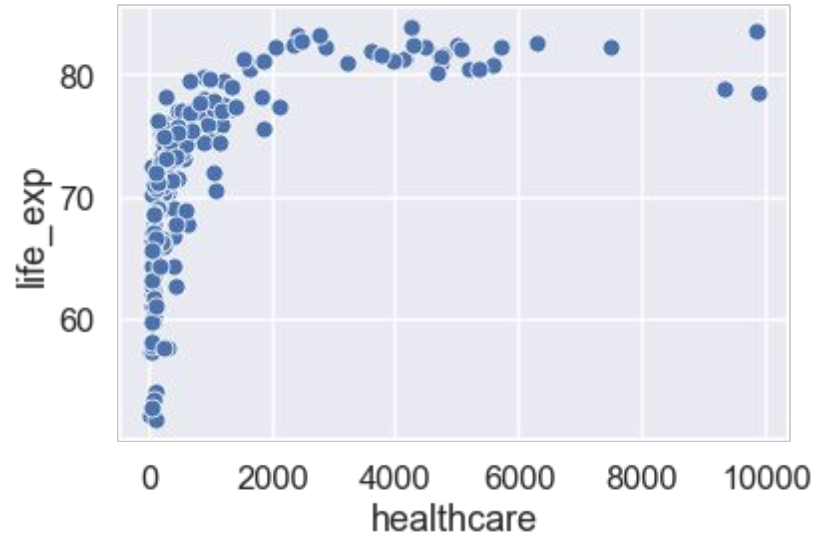
Working backwards:

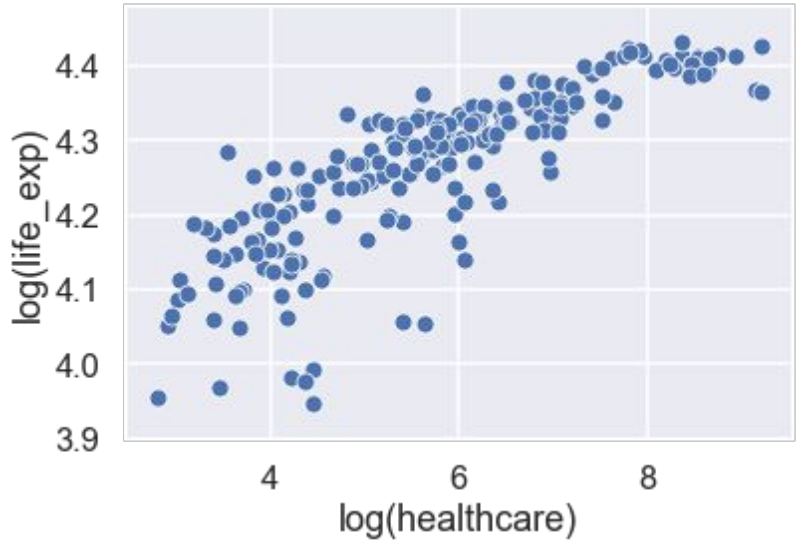
$$y = e^{a \cdot \log x + b}$$

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

$$y = Cx^a$$

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





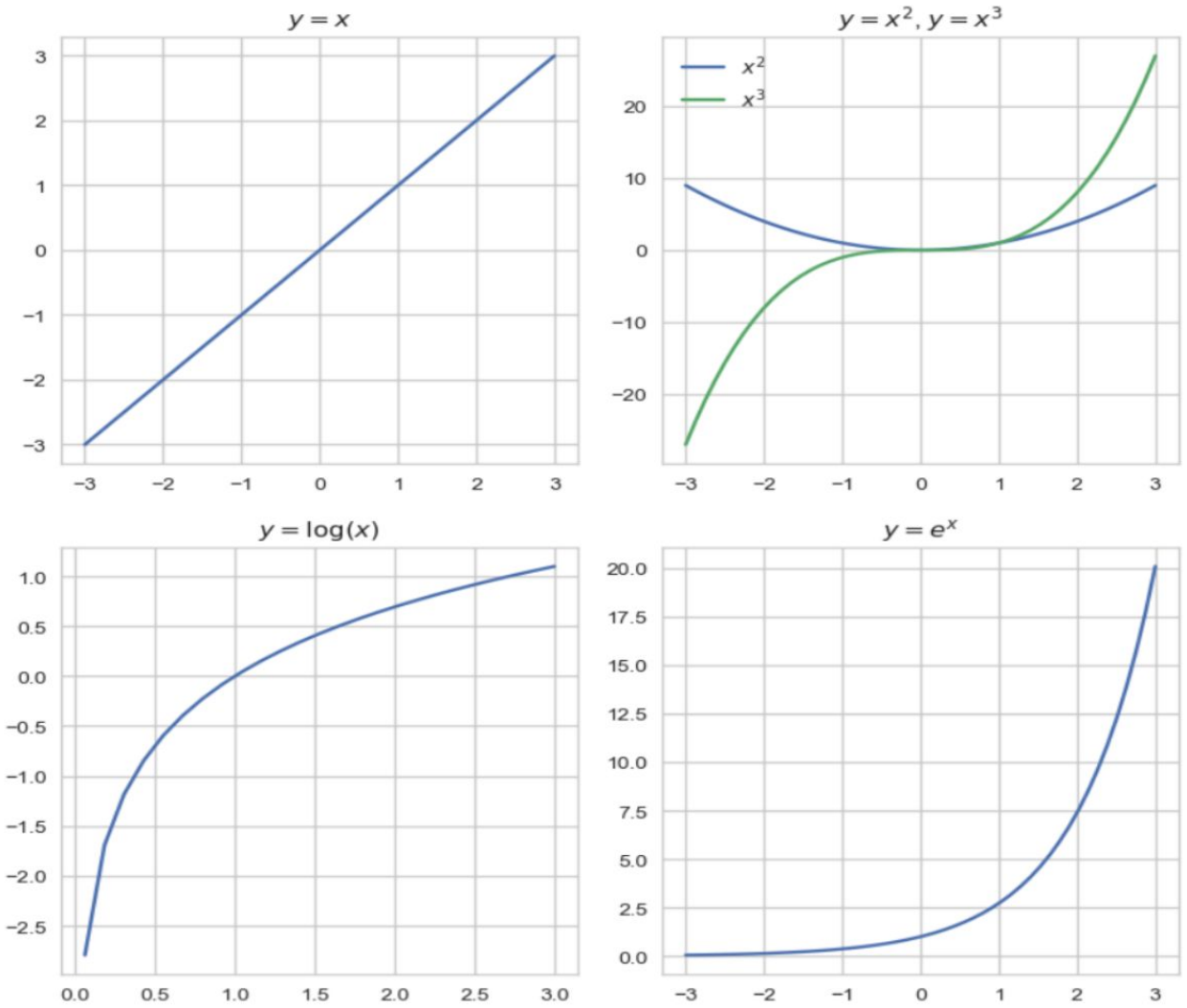
Log transform as a “Swiss army knife”

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

Properties of logarithms make them very powerful!

Basic functional relations

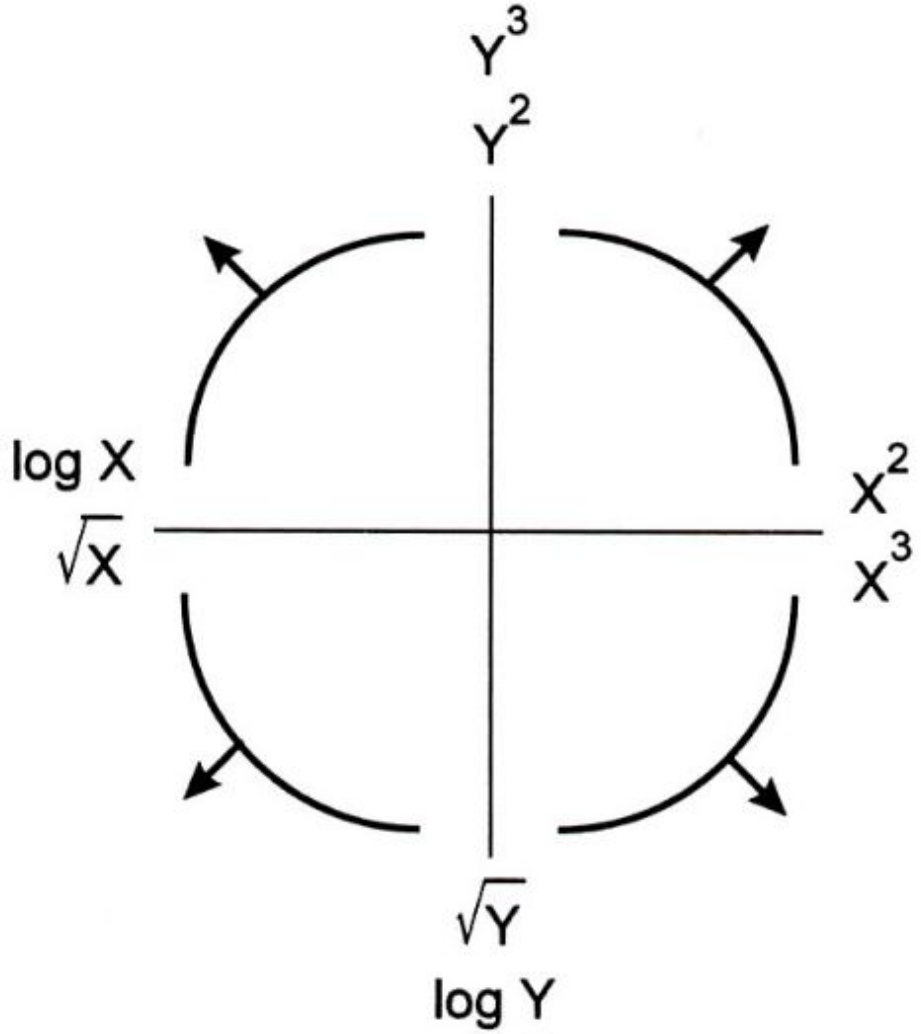
Knowing the general shapes of polynomial, exponential, and logarithmic curves (regardless of base) will go a long way.



Tukey-Mosteller Bulge Diagram

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

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



Summary

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