

LECTURE 10

Visualization, Part 1

Why we visualize, and how to visualize common combinations of variables.

Data 100/Data 200, Fall 2021 @ UC Berkeley

Fernando Pérez and Alvin Wan

Content credit: Suraj Rampure, Ani Adhikari, Sam Lau, Yifan Wu

What is visualization?



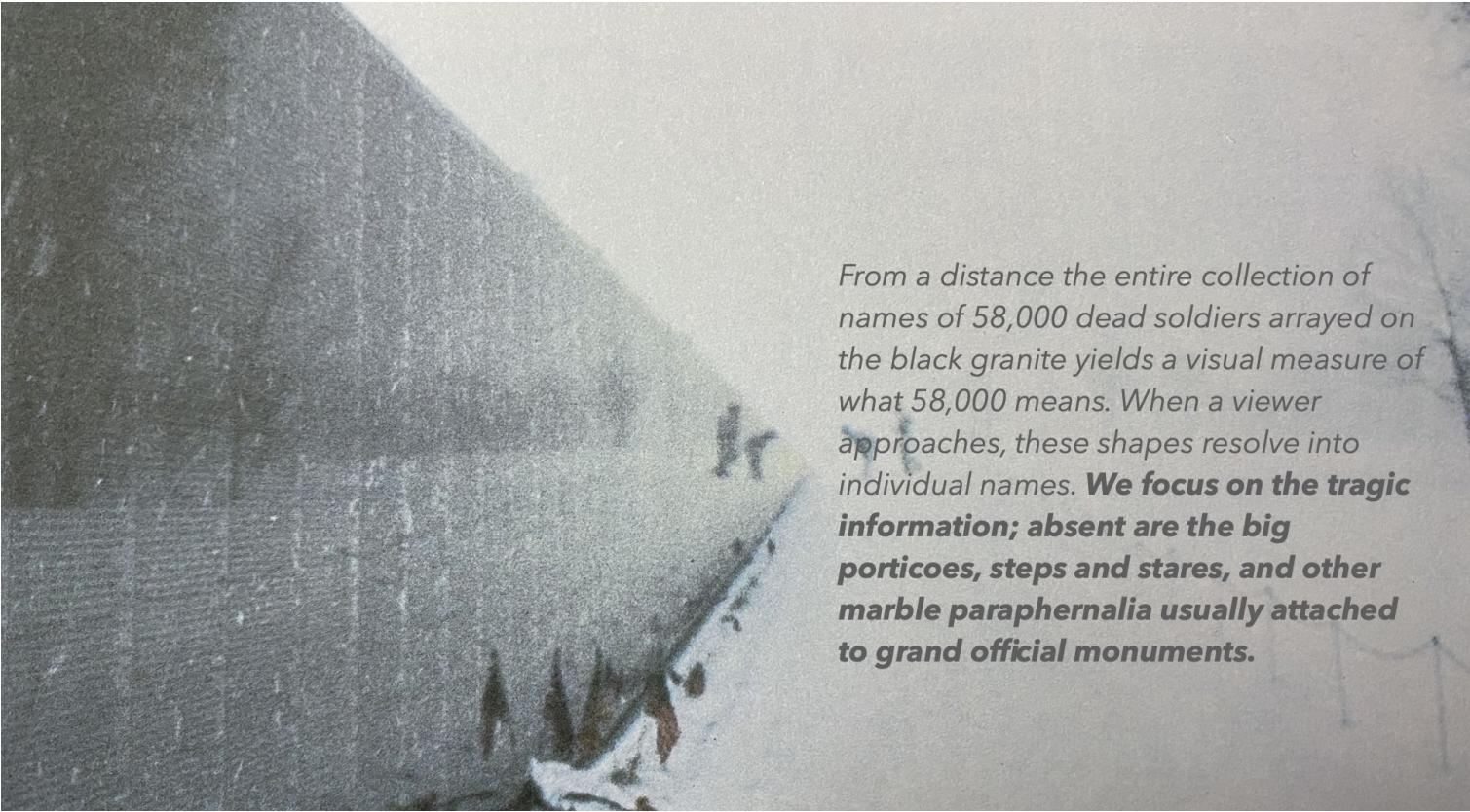
What is this a visualization of?



Squash ball bounce positions in a gym.



What is this a visualization of?



*From a distance the entire collection of names of 58,000 dead soldiers arrayed on the black granite yields a visual measure of what 58,000 means. When a viewer approaches, these shapes resolve into individual names. **We focus on the tragic information; absent are the big porticoes, steps and staves, and other marble paraphernalia usually attached to grand official monuments.***

Soldiers who died in the Vietnam War (names are ordered by their time of death).

What is visualization?

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



Card, Mackinlay, & Shneiderman 1999

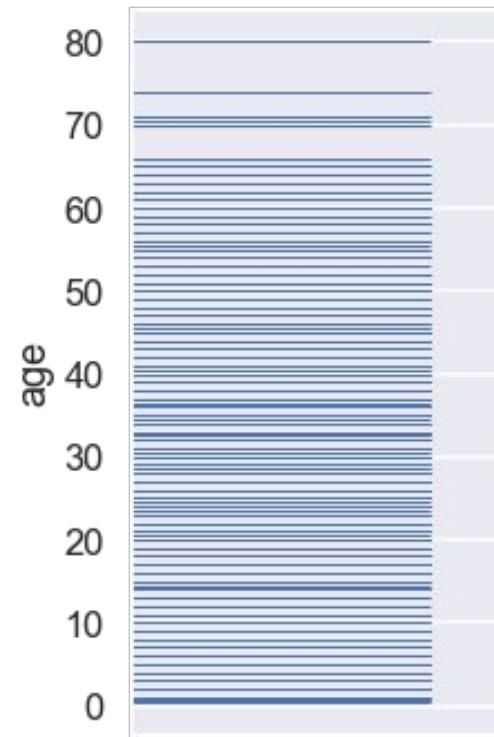
*...finding the **artificial memory** that best **supports**
our natural means of **perception***



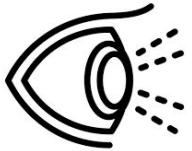
[Bertin 1967]

Take advantage of the human visual perception system

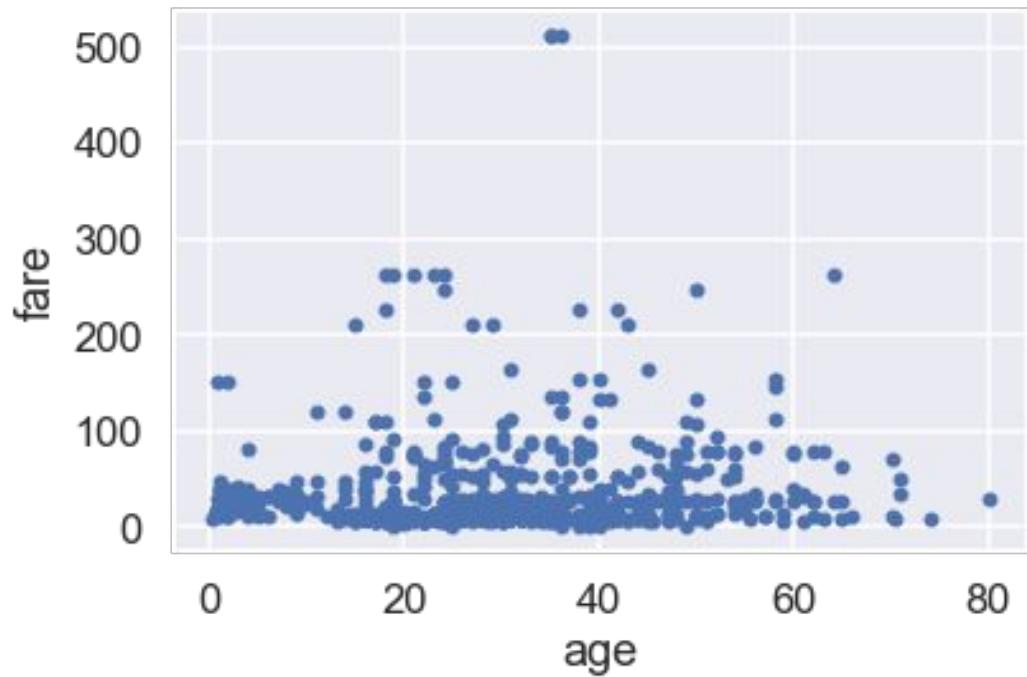
| age | |
|-----|------|
| 0 | 22.0 |
| 1 | 38.0 |
| 2 | 26.0 |
| ... | ... |
| 888 | NaN |
| 889 | 26.0 |
| 890 | 32.0 |



Visualizations are for humans



"Looks like older people didn't spend more than younger people."

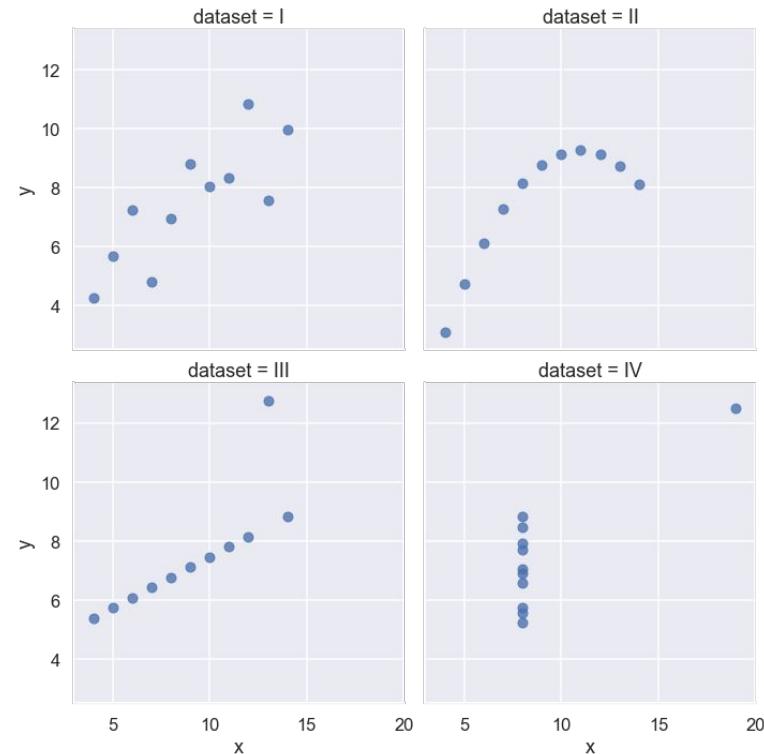


Visualize, then quantify!

| 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.

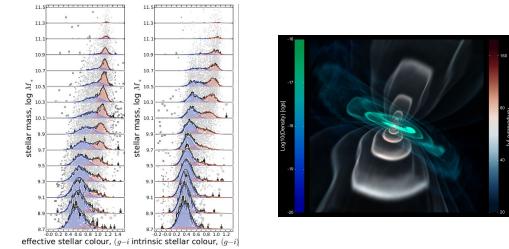
Visualization complements statistics.



Anscombe's Quartet

Goals of data visualization

1. To help **your own understanding** of your data/results
 - o Key part of exploratory data analysis.
 - o Useful throughout modeling as well.
 - o Lightweight, iterative and flexible.
2. To **communicate results/conclusions to others.**
 - o Highly editorial and selective.
 - o Be **thoughtful and careful!**
 - o Fine tuned to achieve a communications goal.
 - o 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:

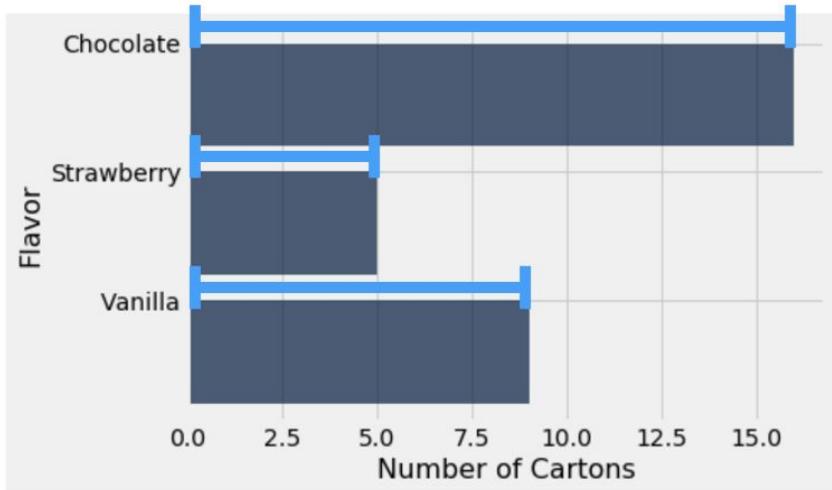
- Today: Establish when to use certain types of visualizations.
- Next lecture: Discuss various principles of visualization, along with kernel density estimation and transformation.

Encoding

Encoding

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

For instance, in Data 8, we saw that length can visually encode a numerical quantity (here, the number of cartons).

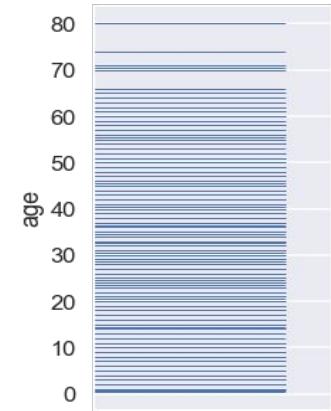


Encoding

| 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
...
0px
11px
15px



Encoding
(Maps datum to visual position)

Encoding

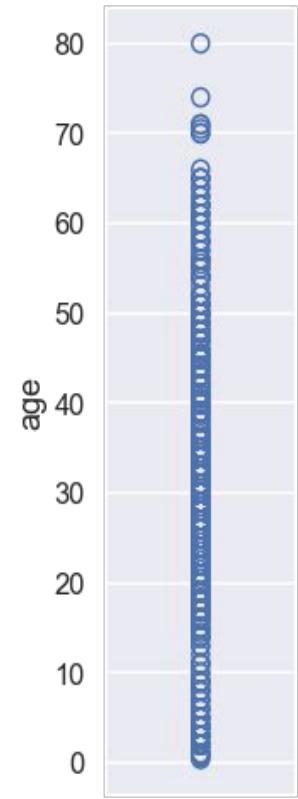
| age | |
|-----|------|
| 0 | 22.0 |
| 1 | 38.0 |
| 2 | 26.0 |
| ... | ... |
| 888 | NaN |
| 889 | 26.0 |
| 890 | 32.0 |



Mark
(Represents a datum)

10px
16px
11px
...
0px
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)

(10px, 7px)

(70px, 60px)

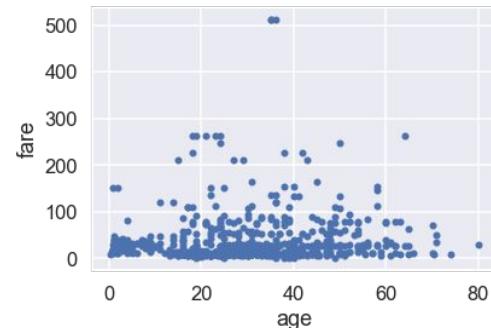
(45px, 9px)

...

(5px, 24px)

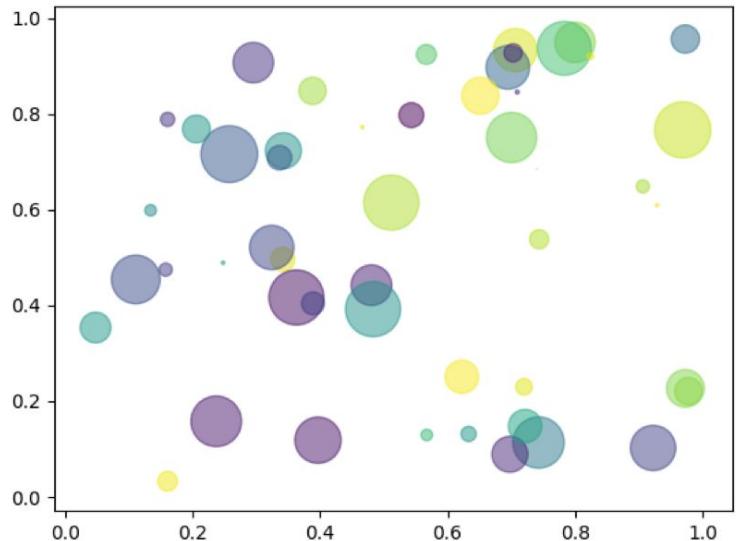
(45px, 37px)

(66px, 8px)

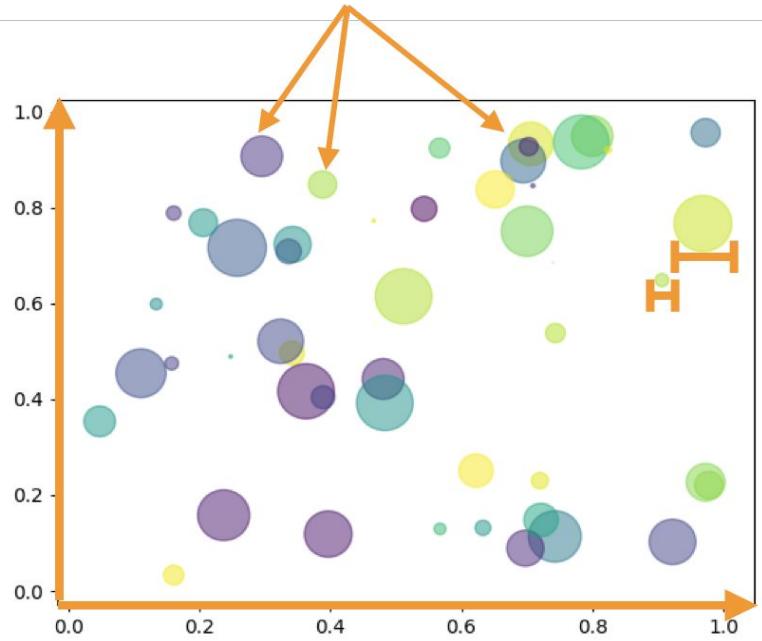


Encoding

(Maps datum to visual
position)



How many variables are we encoding here?



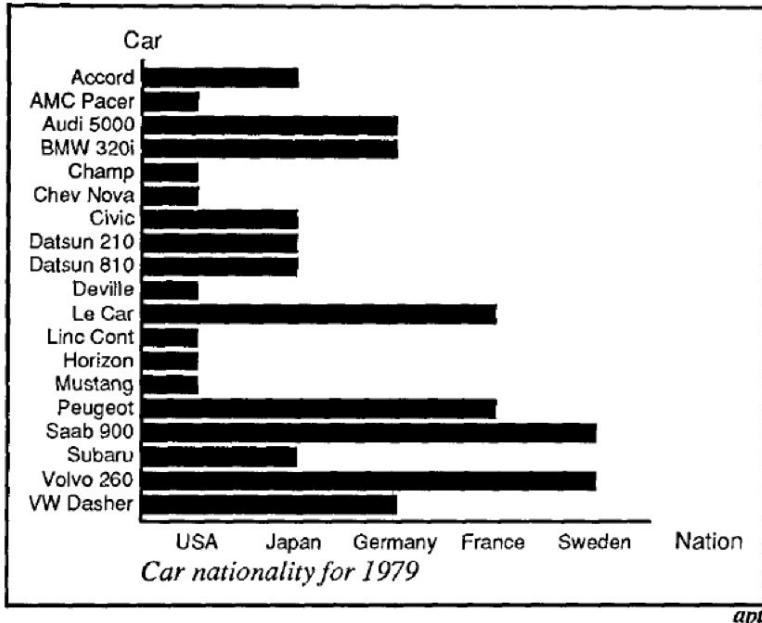
How many variables are we encoding here?

Answer: 4.

- X
- y
- area
- color

What's wrong?

Not all encoding channels are exchangeable.



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

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.

Distributions

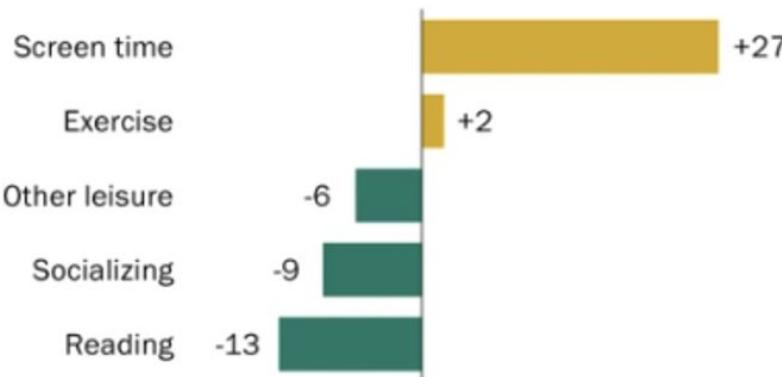
What is a distribution?

- 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.

For older Americans, leisure time looks different today than it did a decade ago

Change in daily time use 2005-2015 (minutes),
for people 60 and older



Note: Based on non-institutionalized people.

Source: Pew Research Center analysis of 2003-2006 and 2014-2017 American Time Use Survey (IPUMS).

PEW RESEARCH CENTER

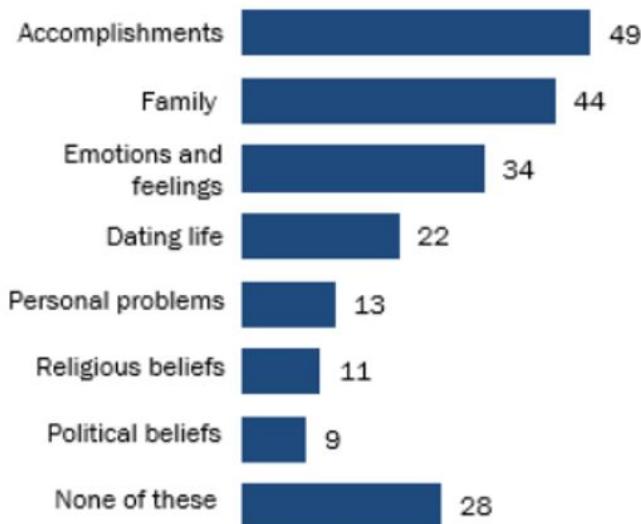
Does this chart show a distribution?

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.

While about half of teens post their accomplishments on social media, few discuss their religious or political beliefs

% of U.S. teens who say they ever post about their ___ on social media



Note: Respondents were allowed to select multiple options.

Respondents who did not give an answer are not shown.

Source: Survey conducted March 7–April 10, 2018.

“Teens’ Social Media Habits and Experiences”

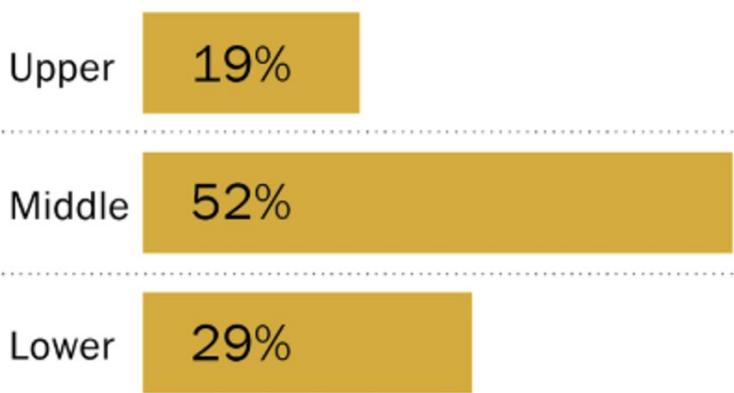
Does this chart show a distribution?

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?

SHARE OF AMERICAN ADULTS
IN EACH INCOME TIER



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 at Berkeley 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 from Data 8 for most of our plots today. Here is what that looks like.

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

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

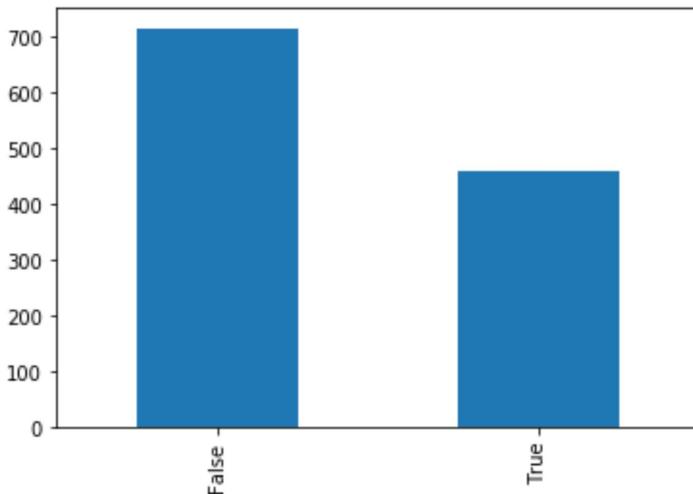
| | Birth Weight | Gestational Days | Maternal Age | Maternal Height | Maternal Pregnancy Weight | Maternal Smoker |
|---|--------------|------------------|--------------|-----------------|---------------------------|-----------------|
| 0 | 120 | 284 | 27 | 62 | 100 | False |
| 1 | 113 | 282 | 33 | 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
```

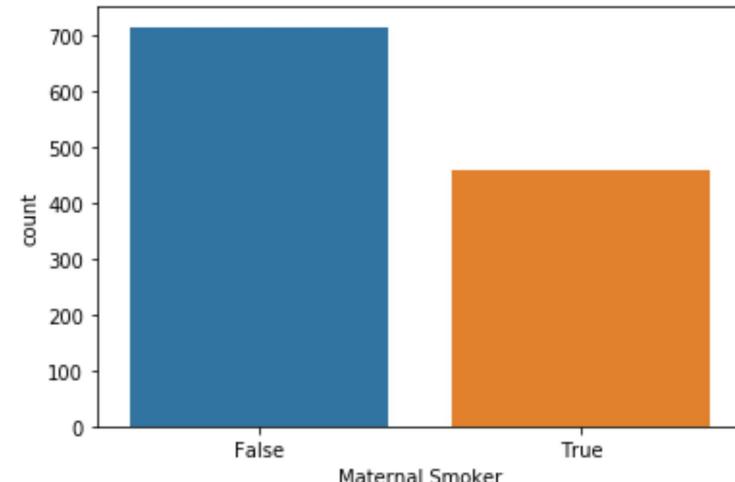
(1174, 6)

Bar plots

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



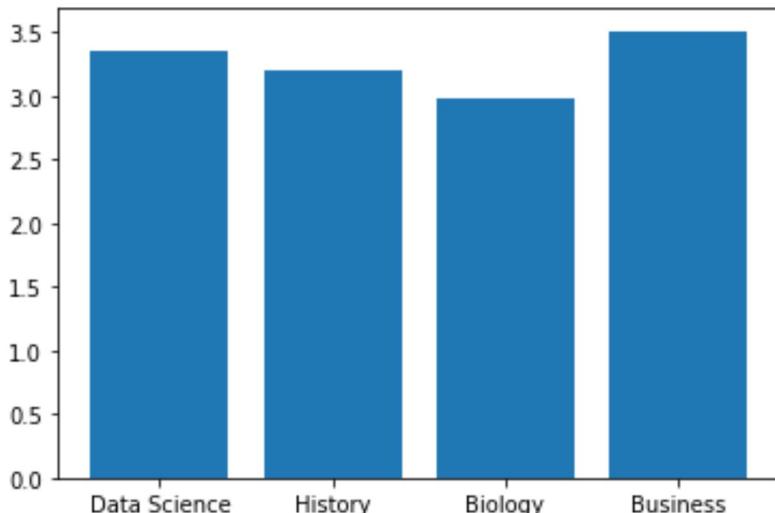
```
births['Maternal Smoker']
.value_counts().plot(kind = 'bar');
```



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

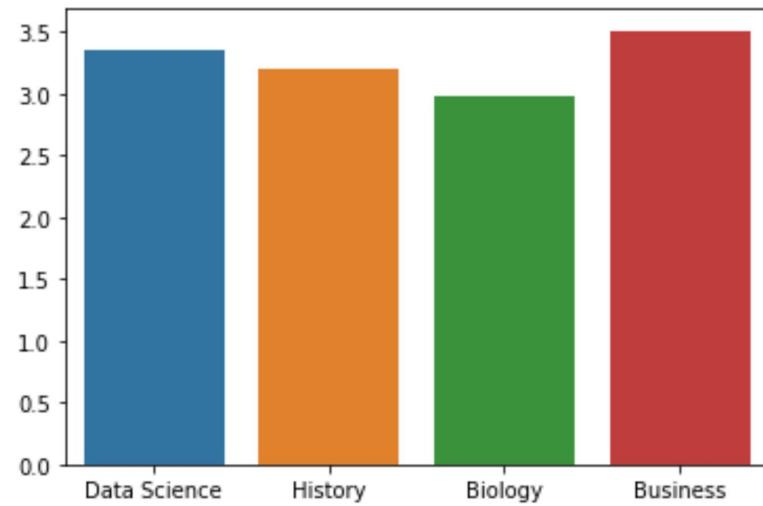
Bar plots

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.bardata(majors, gpas)



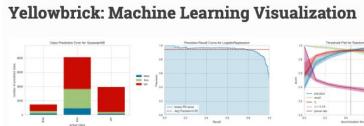
```
sns.barplot(majors, gpas)
```

Note: Here, color is meaningless.

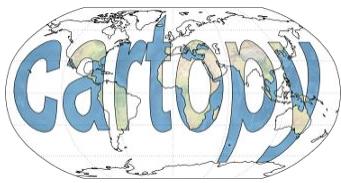
Three ways to plot

- 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.
- Moving forward, we won't necessarily show you all of the ways to plot something.
 - But we will give you the code for at least one way!
 - Play around with arguments in the supplemental notebook.

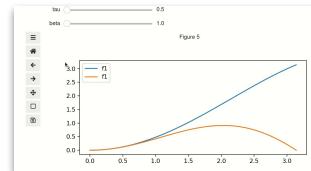
The rich Python plotting ecosystem - this is not all!



bokeh

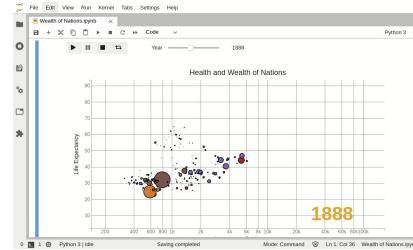


mpl_interactions



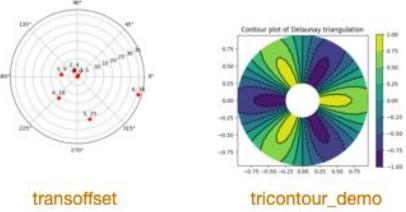
bqplot - Jupyter widgets

matplotlib

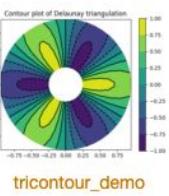


matplotlib

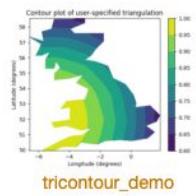
<https://matplotlib.org/gallery.html>



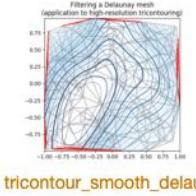
transoffset



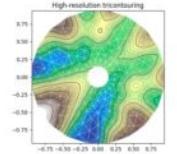
tricontour_demo



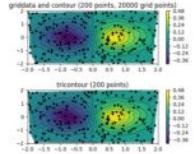
tricontour_demo



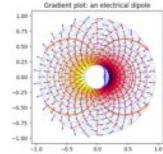
tricontour_smooth_delaunay



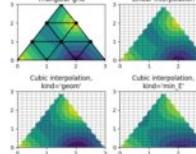
tricontour_smooth_user



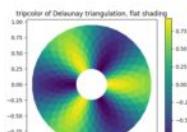
tricontour_vs_griddata



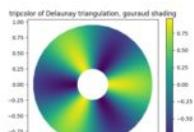
trigradient_demo



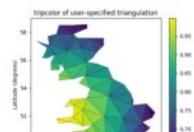
triinterp_demo



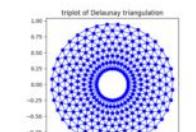
tripcolor_demo



tripcolor_demo



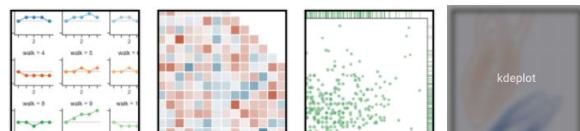
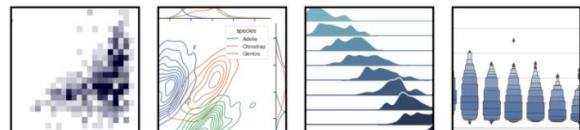
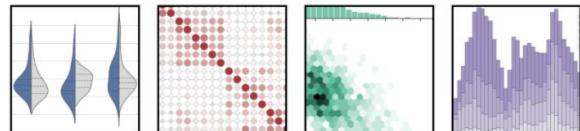
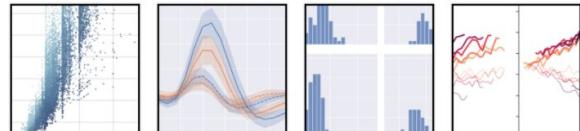
tripcolor_demo



triplot_demo



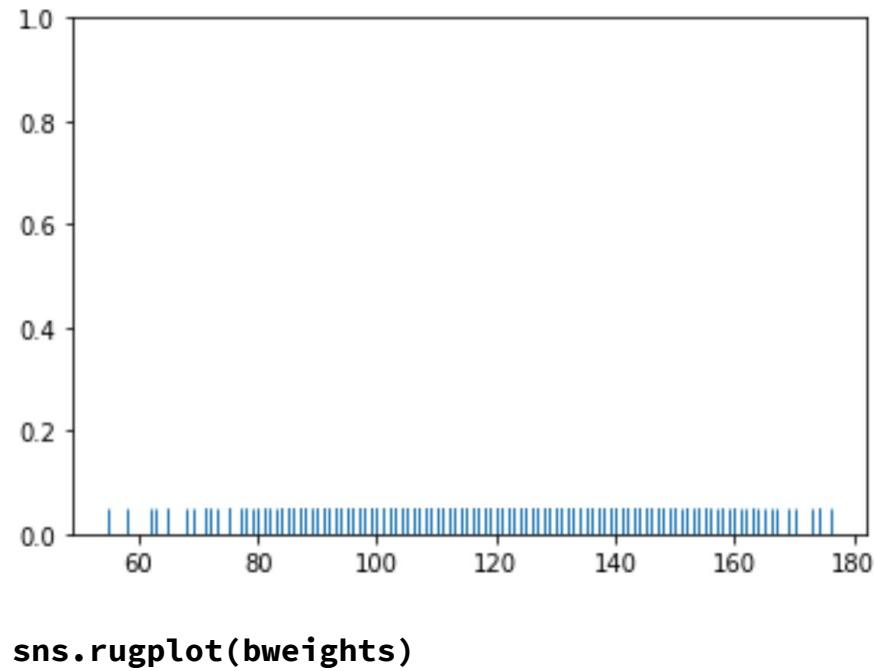
<https://seaborn.pydata.org/examples/index.html>



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.



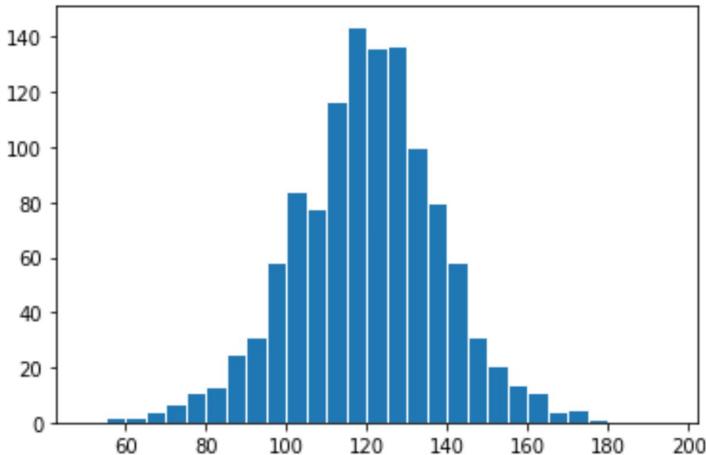
Histograms

- 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”.

proportion in bin = width of bin · height of bar

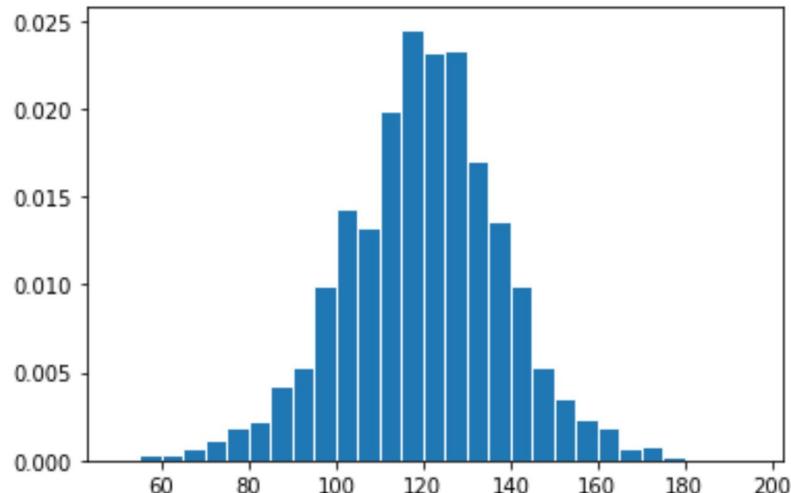
Histograms

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.

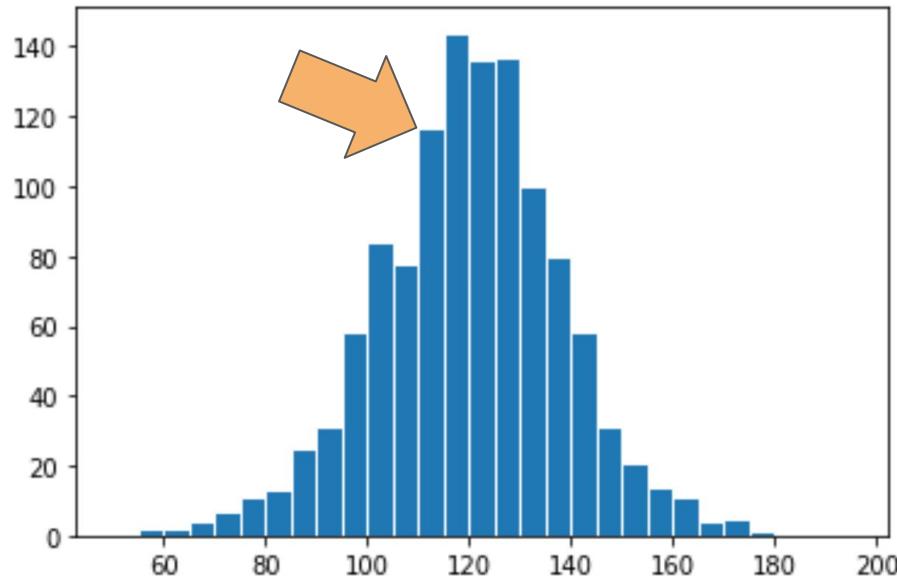


```
plt.hist(bweights, density=True,  
        bins=bw_bins, ec='w')
```

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.

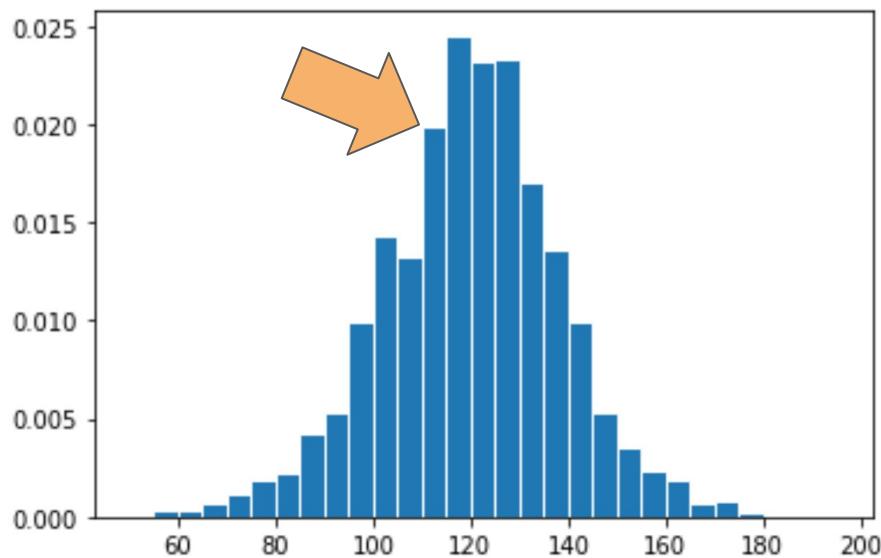
Width of bin [110, 115]: 5

Height of bar [110, 115]: 0.02

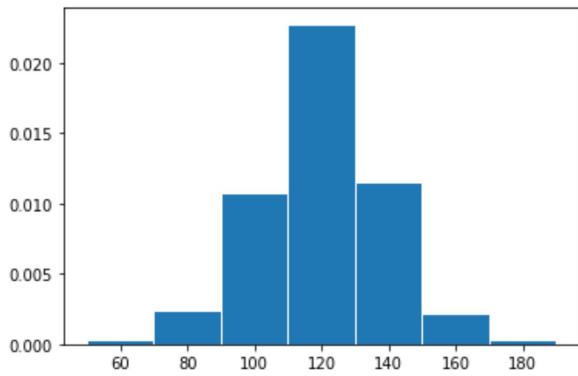
Proportion in bin = $5 * 0.02 = 0.1$

Number in bin = $0.1 * 1174 = \mathbf{117.4}$

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



Histograms



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

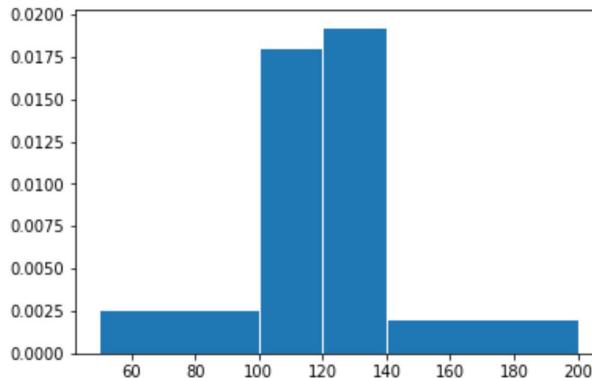
The Freedman-Diaconis rule:

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

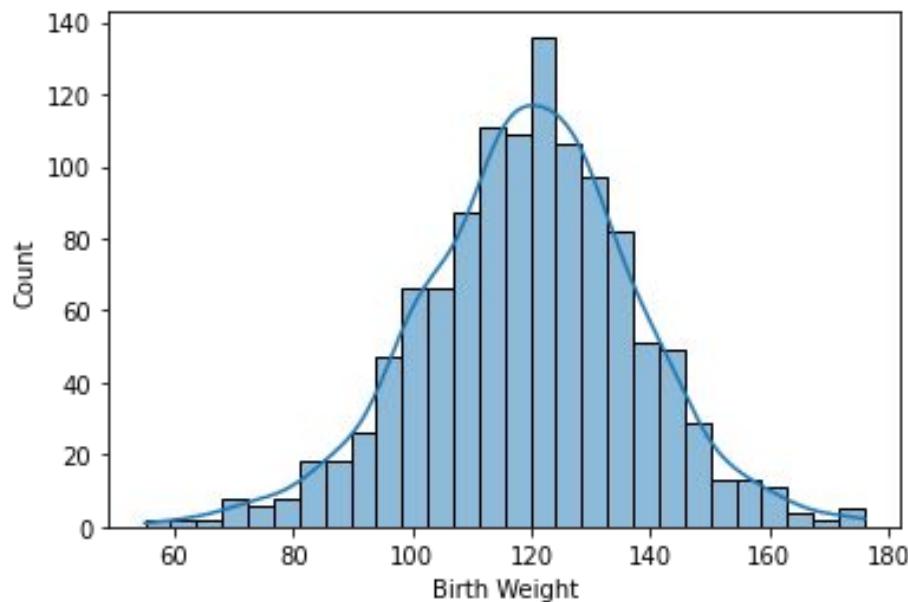
```
plt.hist(bweights, bins = np.arange(50, 200, 20),  
density=True, ec='w')
```

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

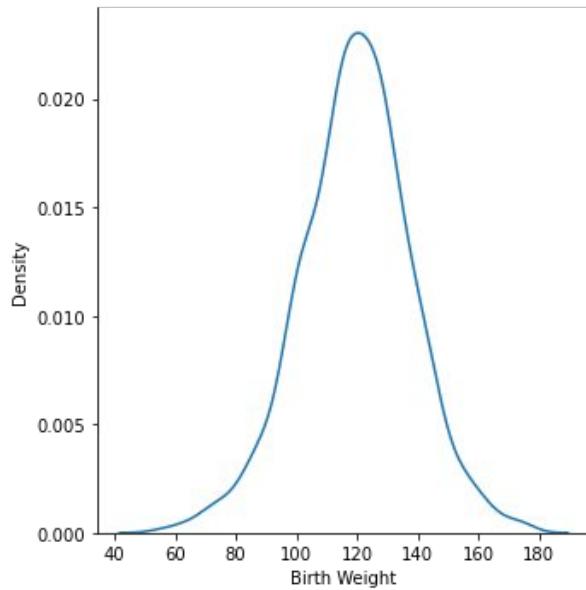


```
sns.histplot(bweights, kde=True)
```

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**.

Density curves



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

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

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

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

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).

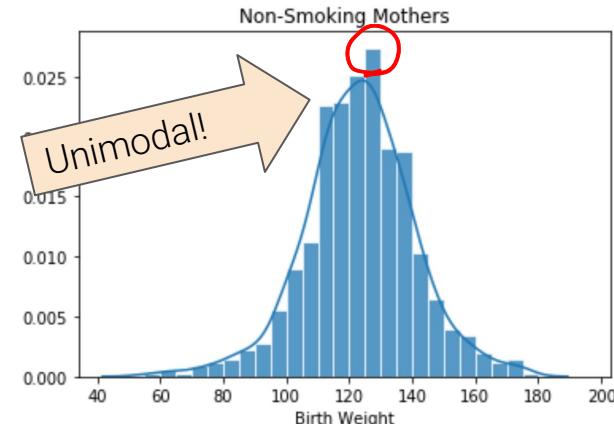
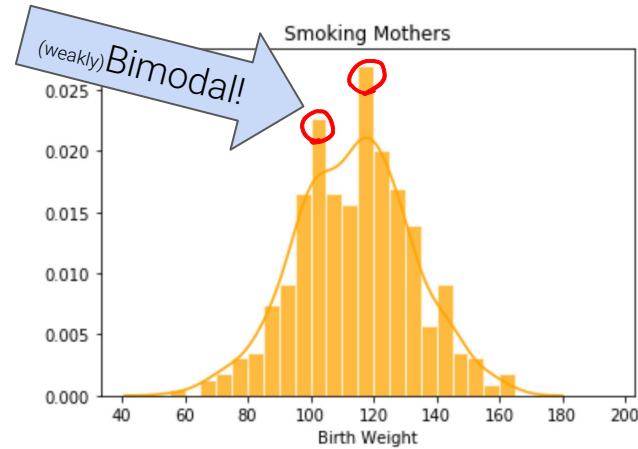
Some of the terminology we use to describe distributions:

- **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.

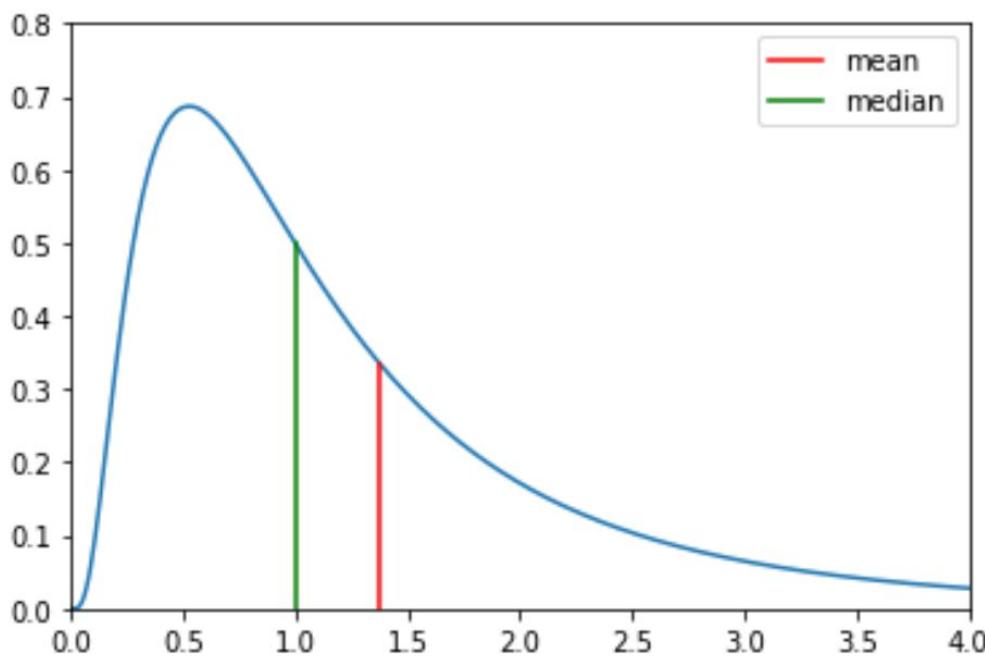
Modes

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.



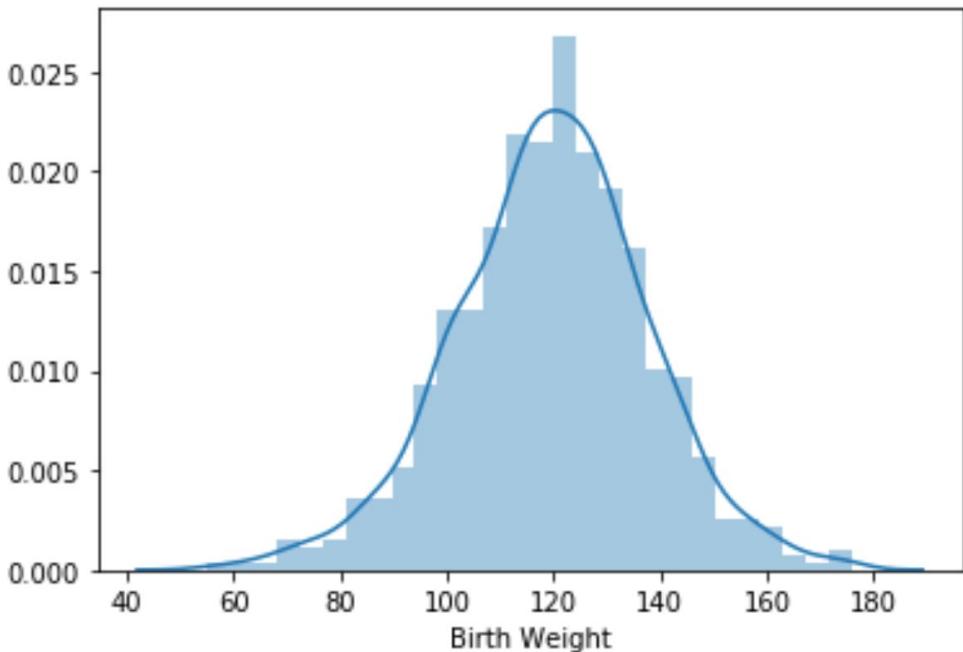
Skew and tails



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

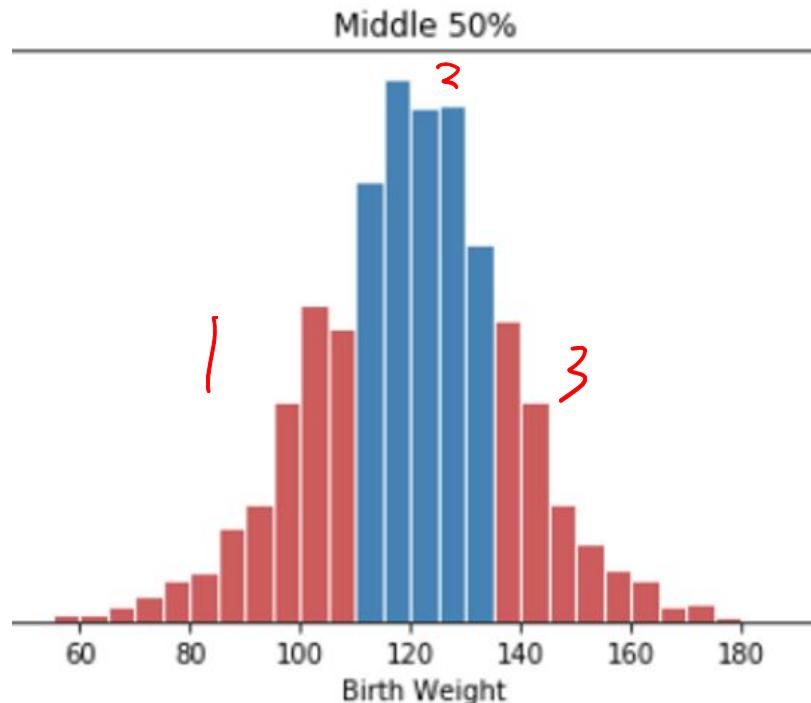
For a quantitative variable:

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

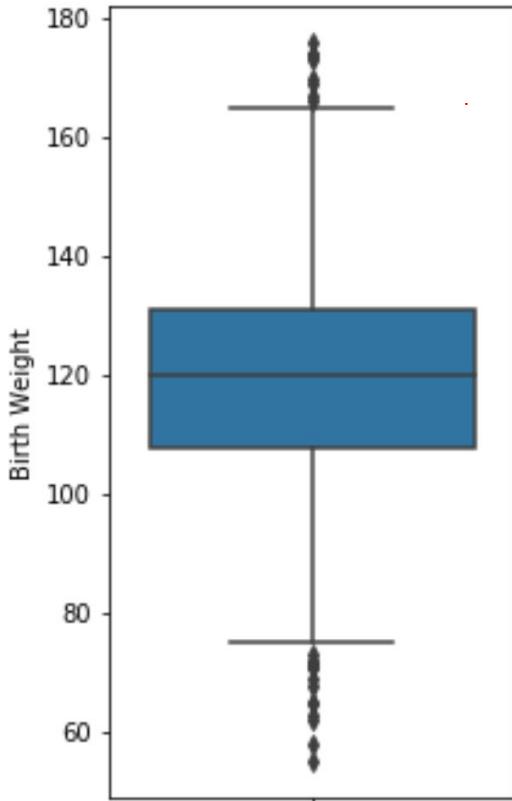
The interval [first quartile, third quartile] contains the “middle 50%” of the data.

Interquartile range (IQR) measures spread.

- $IQR = \text{third quartile} - \text{first quartile}$.



Box plots

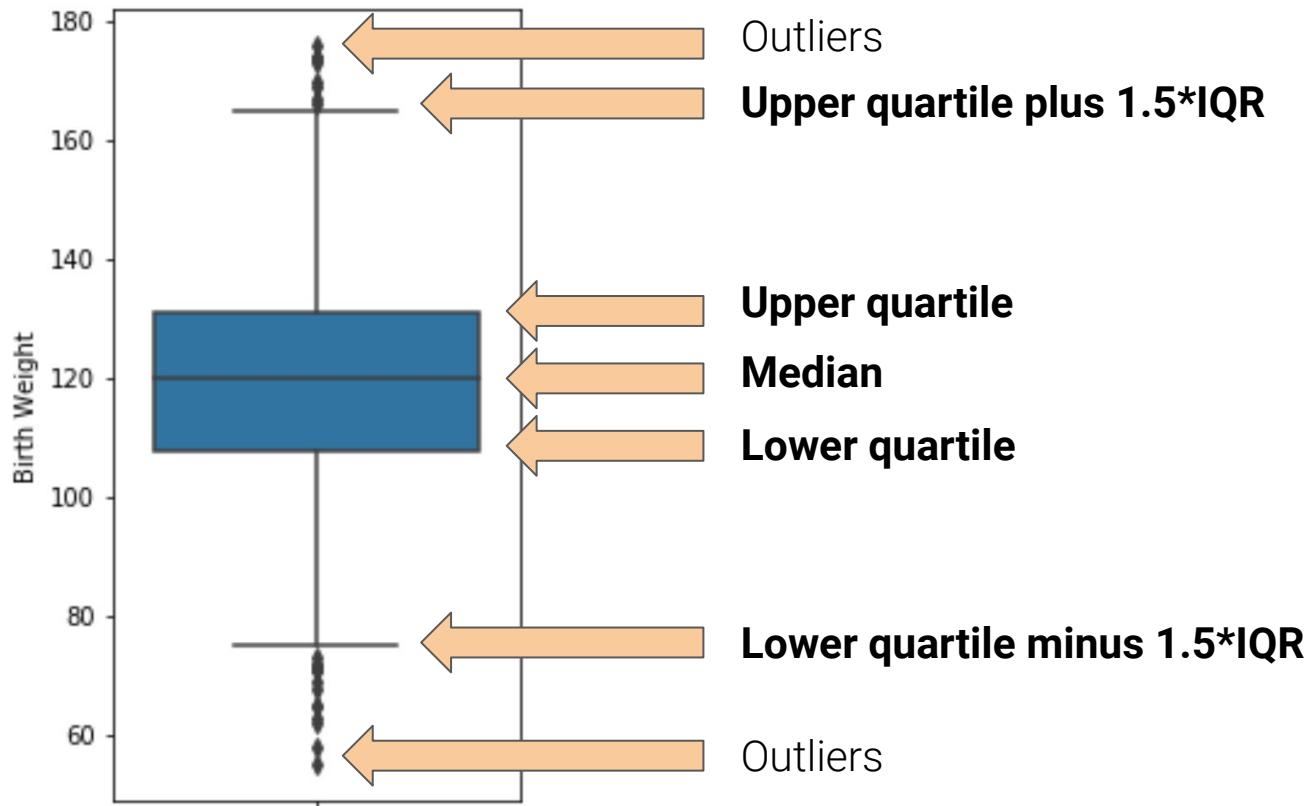


Box plots summarize several characteristics of a numerical distribution. They visualize:

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

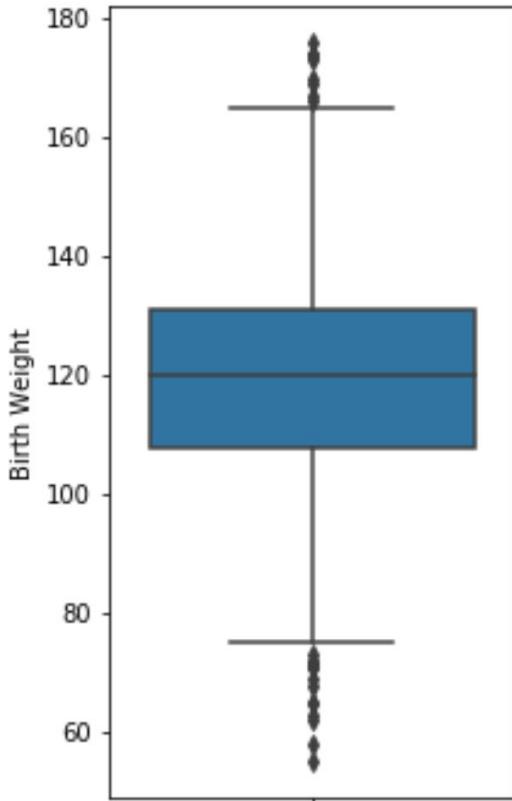
`sns.boxplot(bweights)`

Box plots



Note: The box width is meaningless.

Box plots

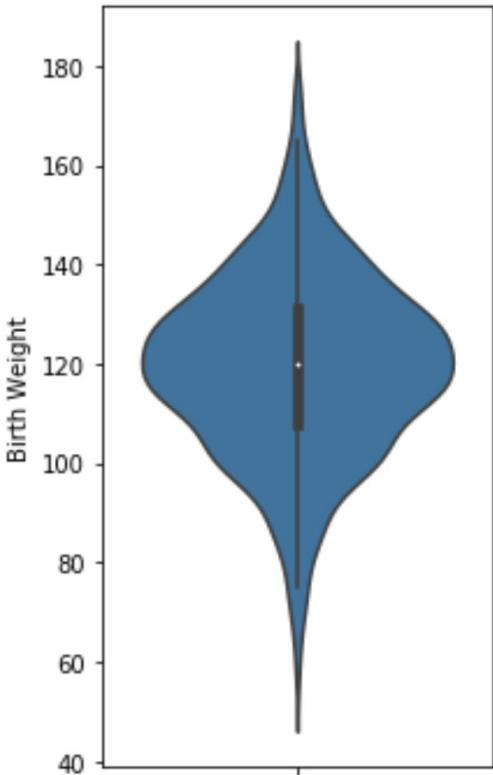


```
1 q1 = np.percentile(bweights, 25)
2 q2 = np.percentile(bweights, 50)
3 q3 = np.percentile(bweights, 75)
4 iqr = q3 - q1
5 whisk1 = q1 - 1.5*iqr
6 whisk2 = q3 + 1.5*iqr
7
8 whisk1, q1, q2, q3, whisk2
```

(73.5, 108.0, 120.0, 131.0, 165.5)

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

Violin plots

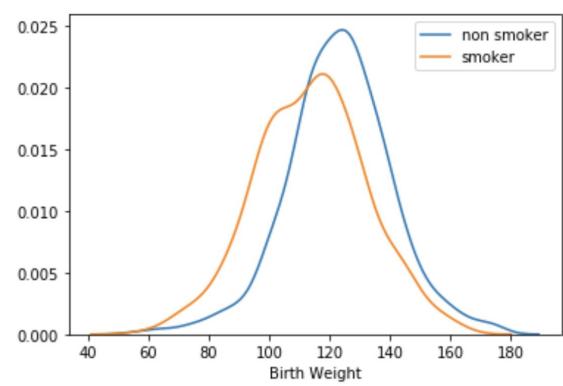
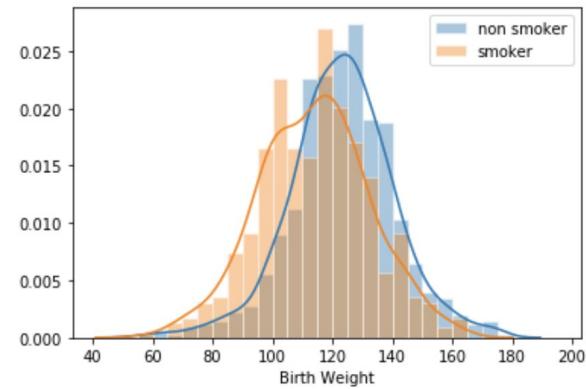
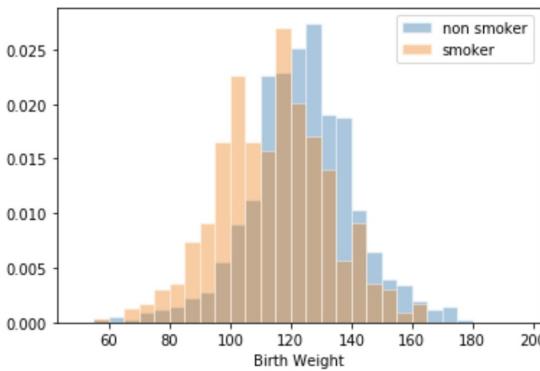


Violin plots are similar to box plots, but also show smoothed density curves.

- 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

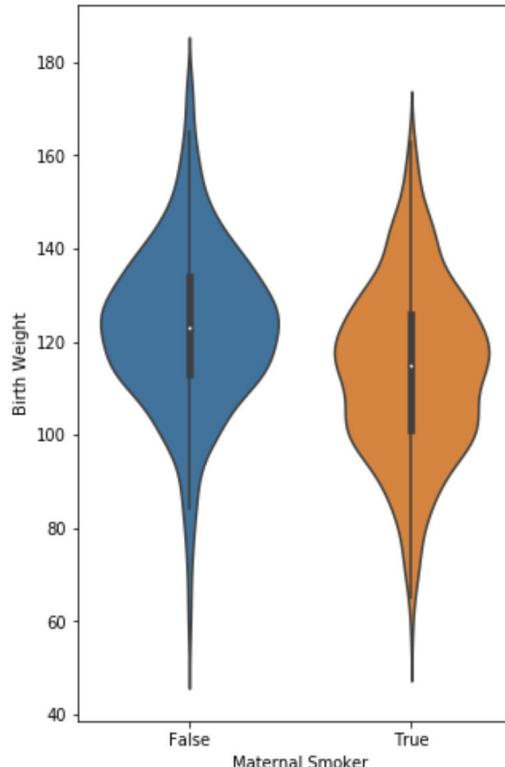
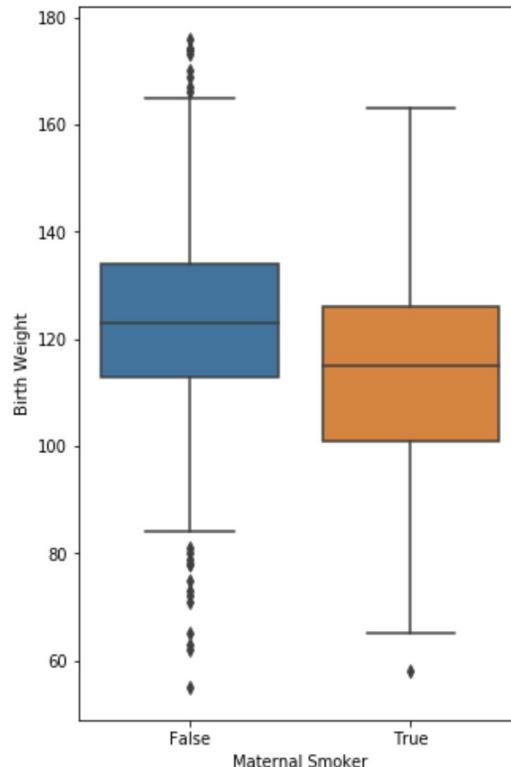


We can overlay multiple histograms and density curves on top of one another.

- 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.

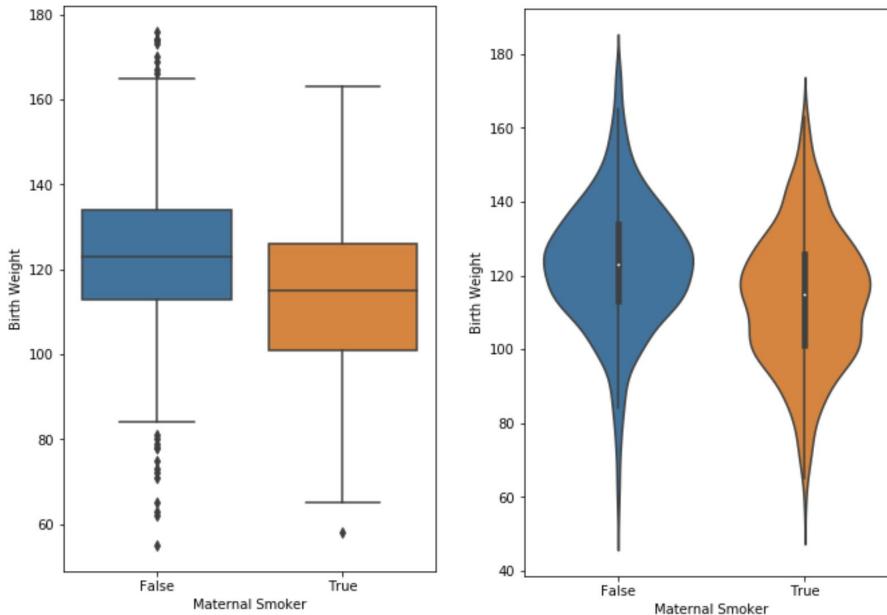
Code is in the speaker's notes.

Side by side box plots and violin plots



Code is in the speaker's notes.

Side by side box plots and violin plots



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.

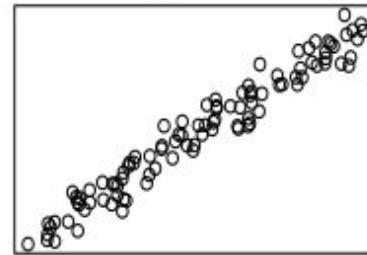
Relationships between two quantitative variables

Scatter plots

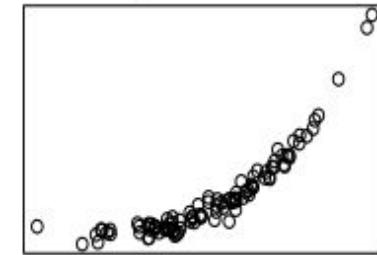
Scatter plots are used to reveal relationships between pairs of numerical variables.

- 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.

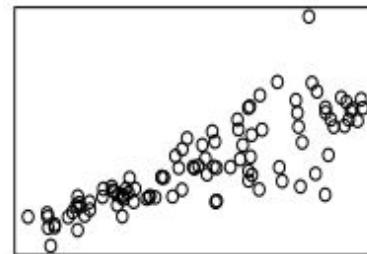
simple linear



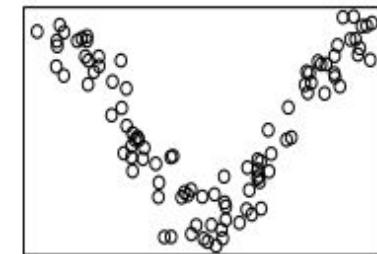
simple nonlinear



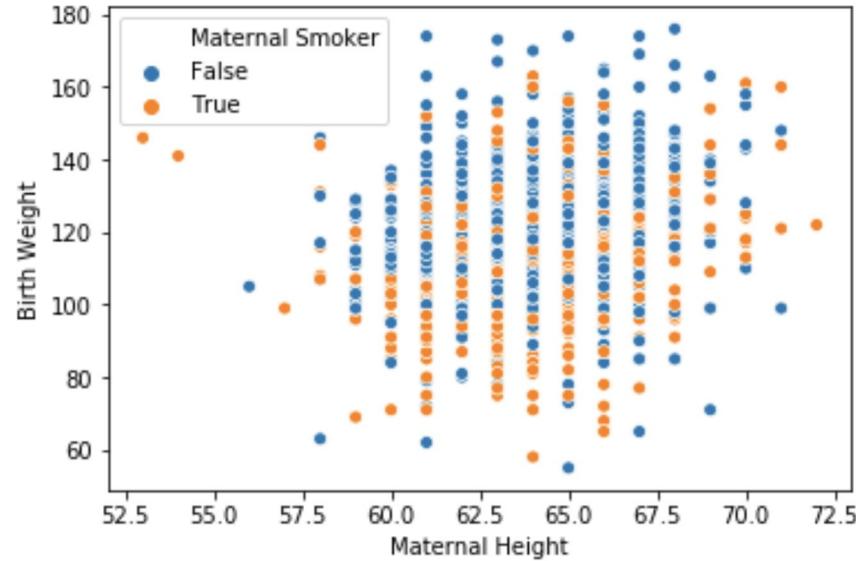
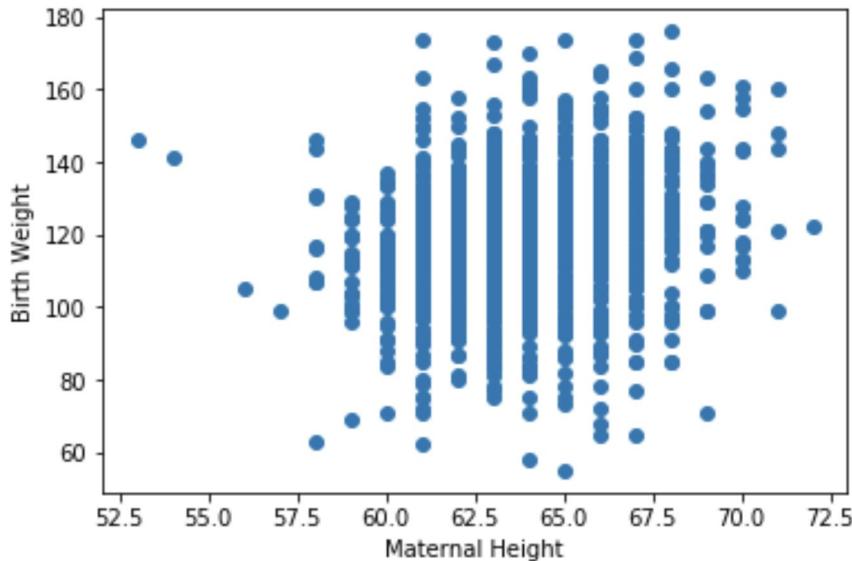
unequal spread



complex nonlinear



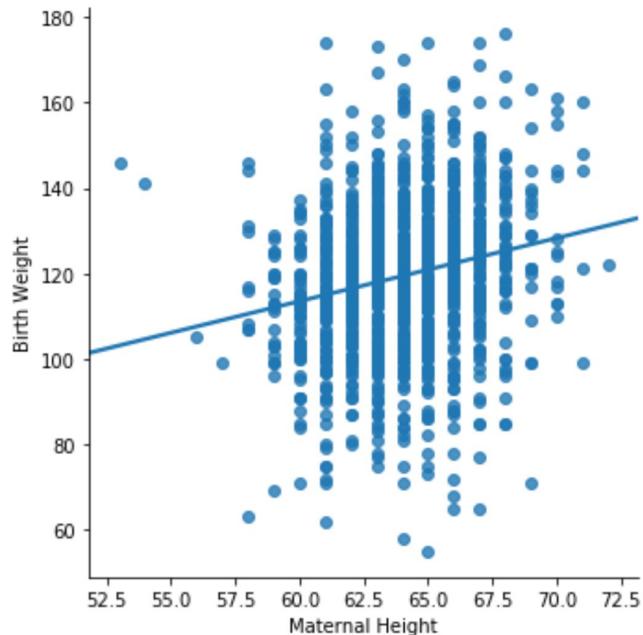
Scatter plots



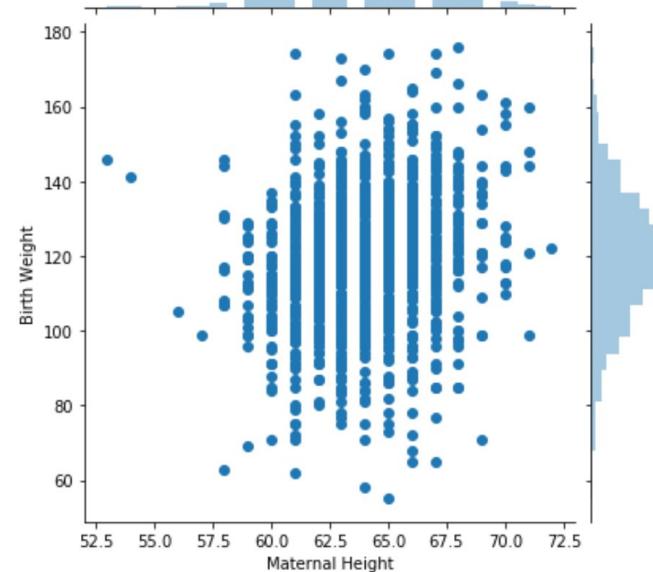
- We can also use color to encode categorical variables.
- These plots suffer from overplotting – many of the points are on top of one another!
 - One solution: add a small amount random noise in both the x and y directions.

Code is in the speaker's notes.

Scatter plots



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



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

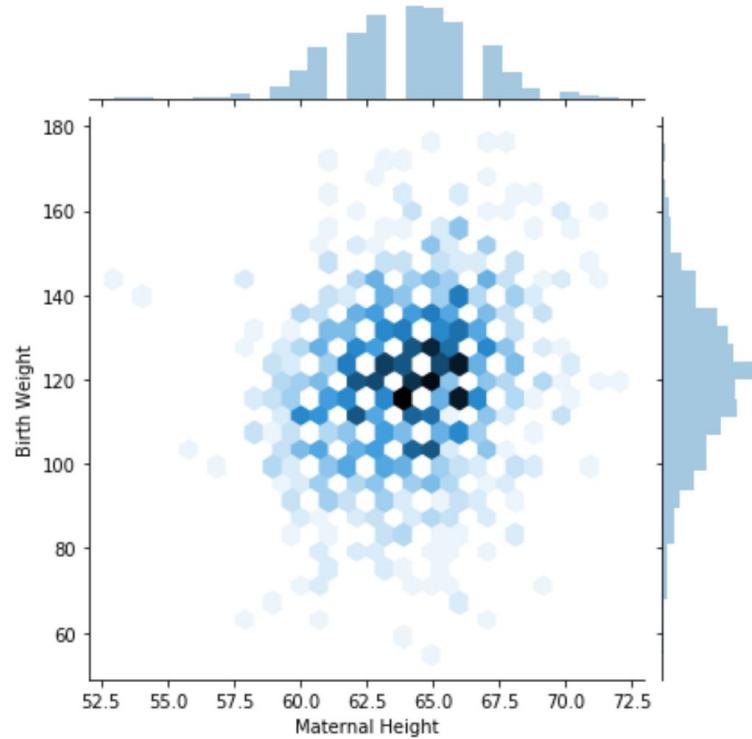
Hex plots

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')
```

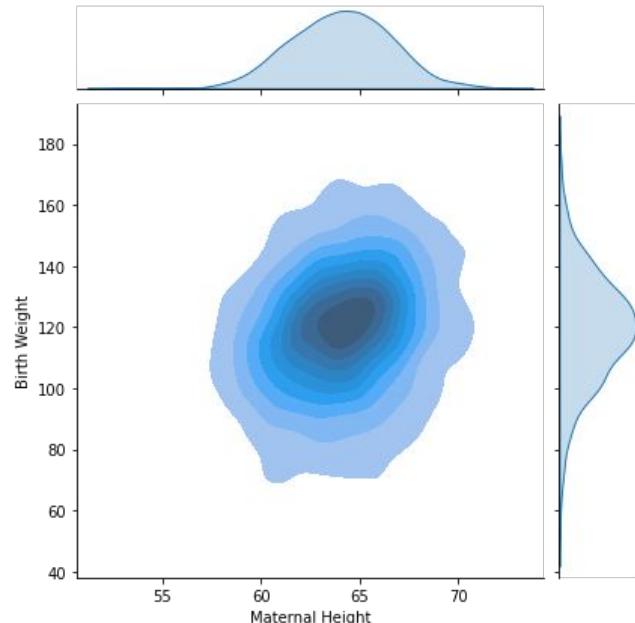
Contour plots

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**.

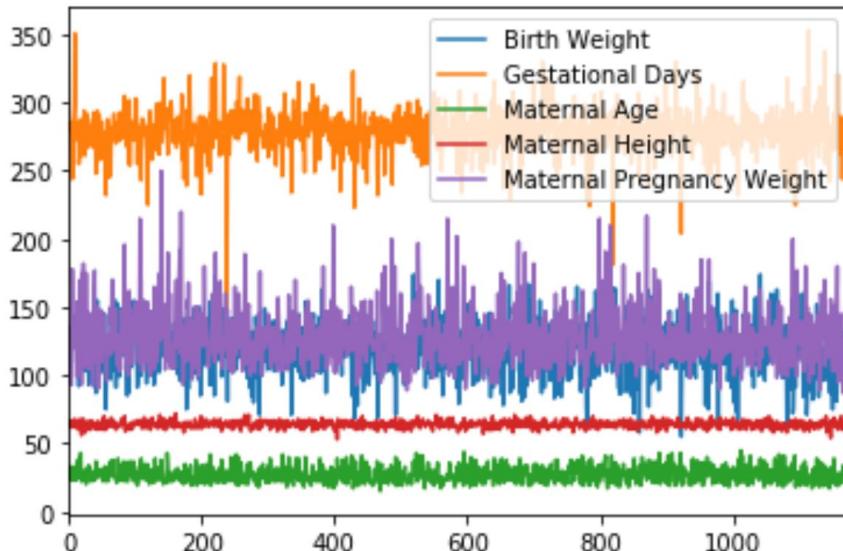
- By default, shows **marginal** distributions on the horizontal and vertical axes.
- These are the histograms/density curves of each variable independently.



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

Summary

- **Visualization requires a lot of thought!**
- Types of variables constrain the charts that you can make.
 - Single quantitative: rug plot, histogram, density plot.
 - Two quantitative: scatter plot, hex plot, contour plot.
 - Combination: bar plot, overlaid histograms/density plots, SBS box/violin plots.
- This class primarily uses seaborn and matplotlib.
 - Pandas also has basic built-in plotting methods.
 - Many other visualization libraries exist. **plotly** is one of them.
 - It very easily creates **interactive** plots.
 - It will appear in lecture code and assignments!



```
births.plot()
```

This is the result of calling **births.plot()**. If you don't provide any specifications, pandas just guesses what you want visualized. It often makes no sense!

Next time

In this lecture, we looked at when to plot what. In the next lecture, we will:

- Discuss four principles of visualization.
 - Scale.
 - Conditioning.
 - Perception.
 - Context.
- Explore how to create Kernel Density Estimates.
- Learn how to transform variables in order to linearize visualized relationships.