

## LECTURE 6

# EDA & Visualization: Part 2

Incorporating visualizations to aid our EDA

**CSCI 3022 @ CU Boulder**

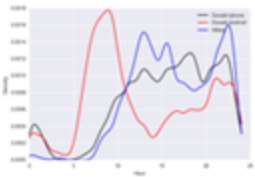
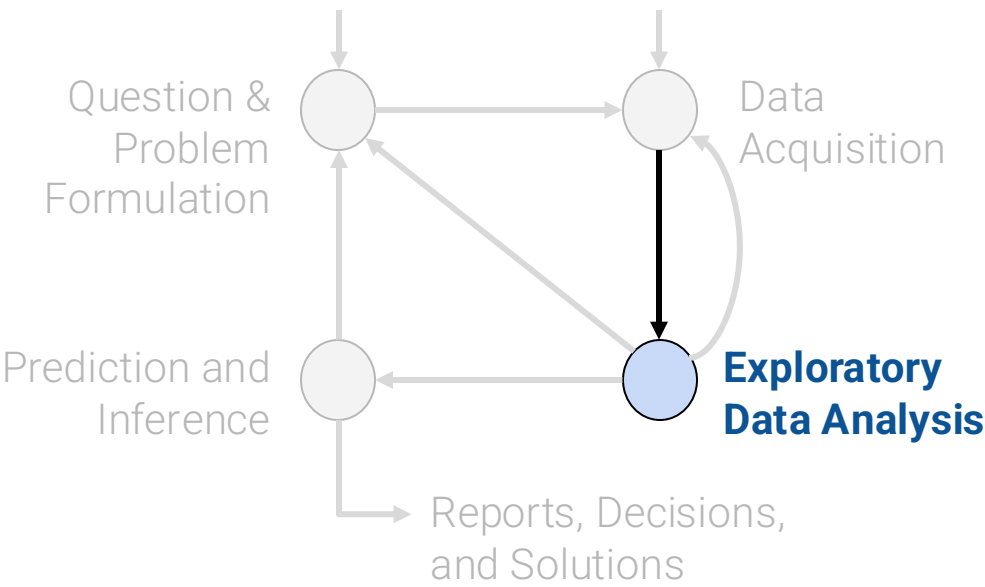
Maribeth Oscamou

Content credit: [Acknowledgments](#)

## Course Logistics: Your Third Week At A Glance

---

Mon 1/27	Tues 1/28	Wed 1/29	Thurs 1/30	Fri 1/31
Attend & Participate in Class		Attend & Participate in Class	HW 3 Due 11:59pm via Gradescope	In Class Quiz 2 (beginning of class): Scope: Lessons 1-4; HW 2 Attend & Participate in Class
			HW 2 feedback/ grades posted	HW 4 released



EDA, Wrangling, and Data Visualization

# Today's Roadmap

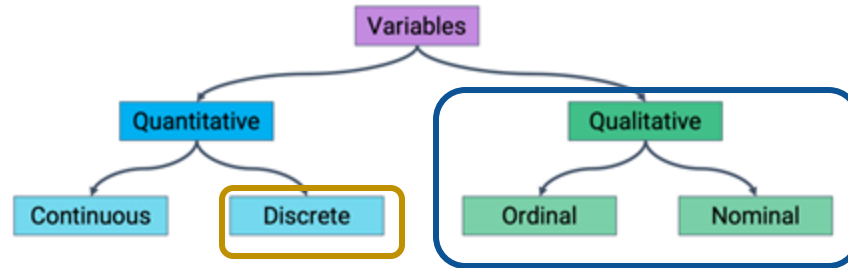
---

Finish Lesson 5: EDA & Visualization Part 1

Start Lesson 6: EDA & Visualization Part 2

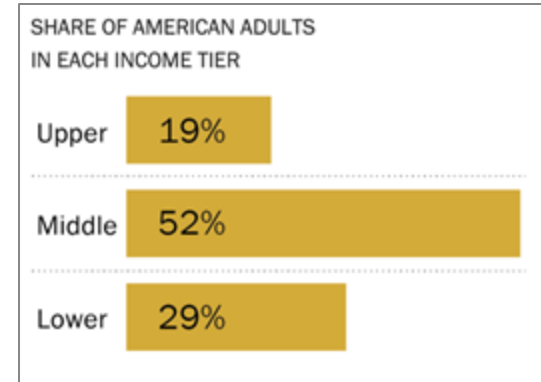
## Review from Lesson 5 - Bar Plots: Distributions of Qualitative Variables

Bar plots are the most common way of displaying the **distribution** of a **qualitative** variable.



\*Sometimes quantitative discrete data too, if there are few unique values.

- Bar Charts:
  - One bar for each category
  - Length of bar is the percent (or count) of individuals in that category
  - Widths encode nothing: but bar widths should all be the same.
  - If ordinal - order of bars should reflect category order
  - Space between bars (not connected)
  - *Color* could indicate a sub-category (but not necessarily).



## Lec05\_06\_Visualization.ipynb

We will be using data in the file `baby.csv` which contains data for 1774 mother-baby pairs in the 1960s. 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)
```

## Lesson 6 Learning Objectives:

- Use Python functions to visualize distributions of Quantitative Variables
- Describe quantitative distributions in terms of skew, modes & outliers
- Use Python functions to visualize relationships between variables

# EDA & Visualization: Part 2

---

## EDA & Visualization Part 2:

- Visualizing Distributions
  - Quantitative Variables
- Describing Quantitative Distributions
- Visualizing Relationships between variables:
  - Quantitative
  - Qualitative
  - Mixed

## Learning Objectives

- Use Python functions to visualize distributions of Quantitative Variables

# Visualizing Distributions

---

- Visualizing Distributions of Quantitative Variables:
  - Histograms & Density Curves
  - Boxplots and Violin Plots



# Visualizing Quantitative Features

	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
...	...	...	...	...	...	...

Suppose we want to plot the distribution of "Maternal Pregnancy Weight" as a bar plot.

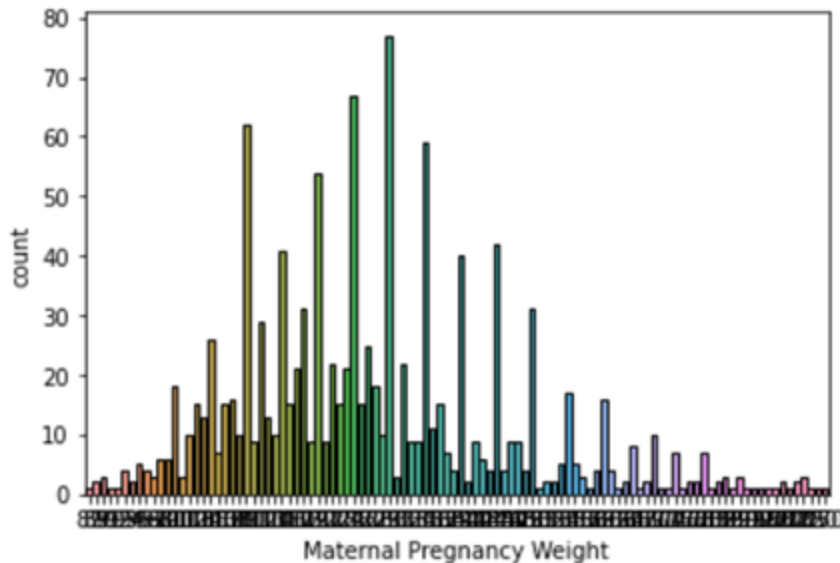
- What are our "categories"?
  - One natural choice: Each integer value, e.g. 100 is a category, 135 is a category, etc.

## Distributions of Quantitative Variables

Earlier, we said that bar plots are appropriate for distributions of qualitative variables.

Why only qualitative? Why not quantitative as well?

- For example: The distribution of maternal pregnancy weight



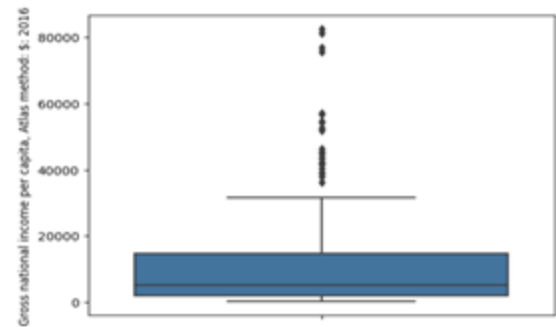
A bar plot will create a separate bar for each unique value. This leads to too many bars for continuous data!

Quick note: These weights are **self-reported pre-pregnancy weights** from the 1960s!

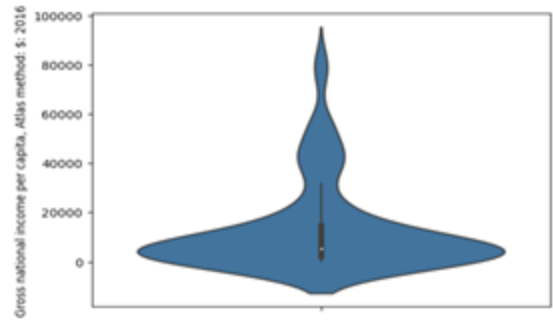
```
sns.countplot(data = births, x = 'Maternal Pregnancy Weight');
```

# Distributions of Quantitative Variables

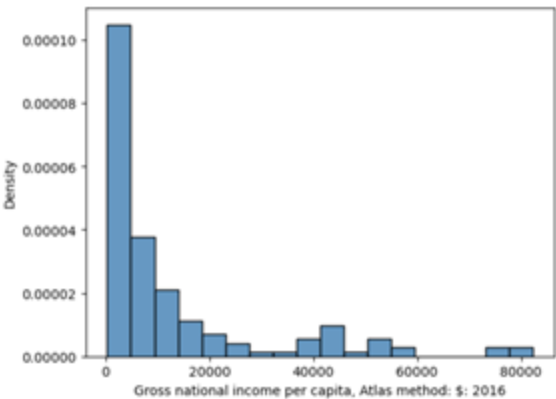
To visualize the distribution of a quantitative variable:



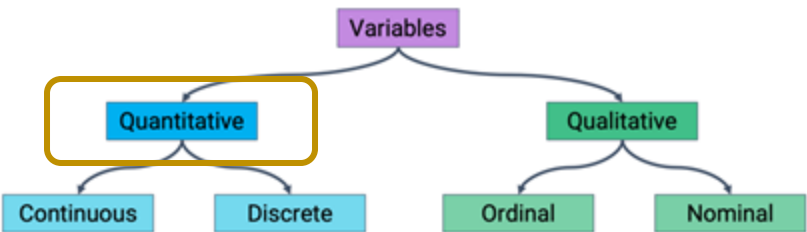
Box plot



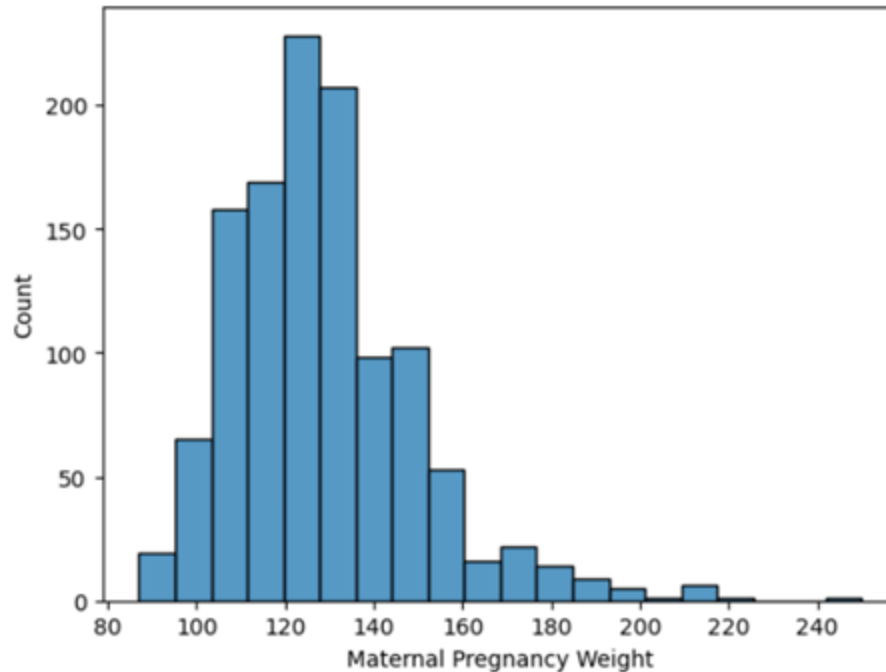
Violin plot



Histogram



# Maternal Pregnancy Weight

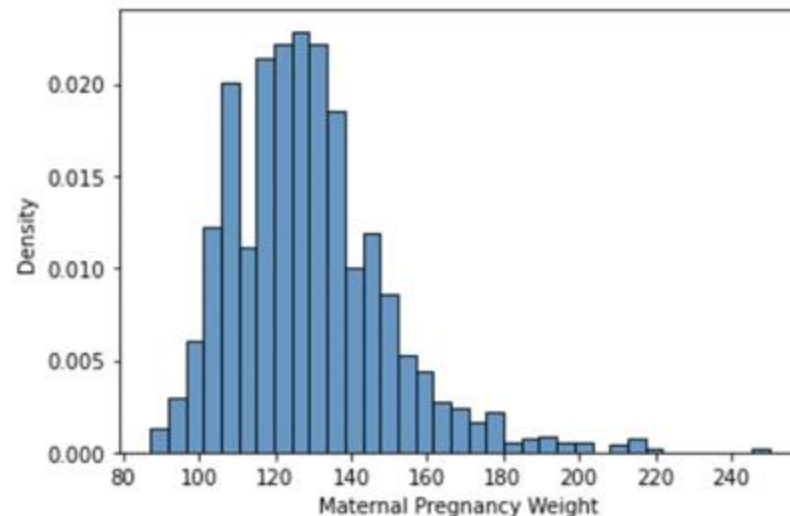
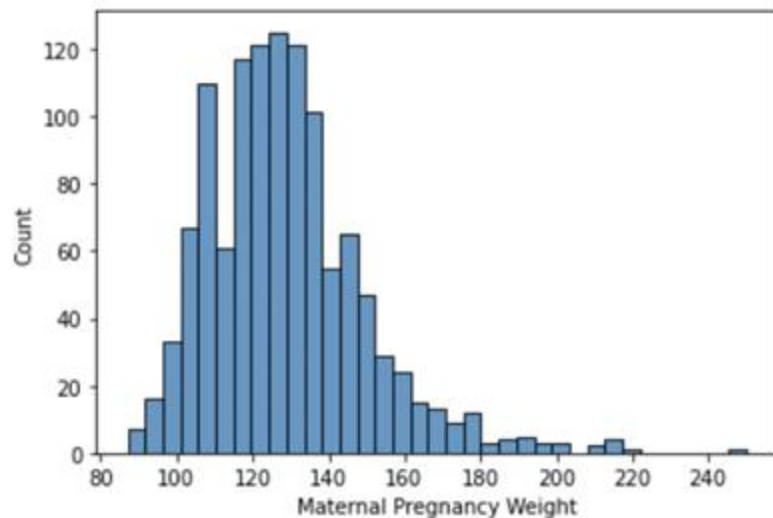


If we use bins of integer values as a category.

```
sns.histplot(data = births, x = 'Maternal Pregnancy Weight', bins = 20);
```

# Histograms and Density

Rather than labeling by counts, we can instead plot the density, as shown below:



Units of density: Proportion (fraction) of data points per unit of x-axis  
Ex above: fraction of data points per

maternal pound

```
sns.histplot(data = births, x = 'Maternal Pregnancy Weight', stat = 'density');
```

## (Intuition) Density Histogram: Proportional Areas

---

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

Suppose we wanted to create a density histogram for these 5 data points.

Using the following bins:

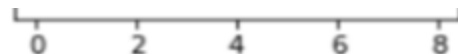
Bins	Points
------	--------

[0, 2)

[2, 4)

[4, 6)

[6, 8]



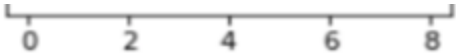
# (Intuition) Density Histogram: Proportional Areas

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

Suppose we wanted to create a density histogram for these 5 data points.

Using the following bins:

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

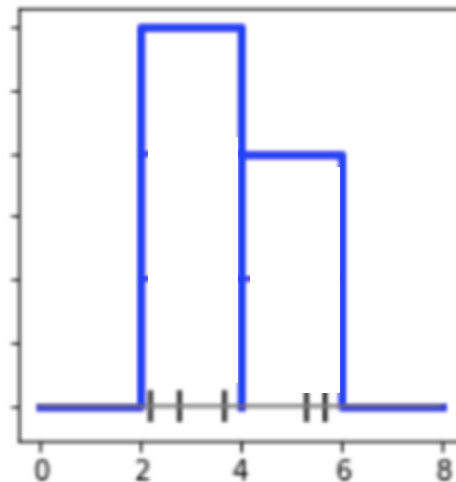


## (Intuition) Density Histogram: Proportional Areas

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

In a density histogram, **area = proportion** (i.e. fraction of data points)

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





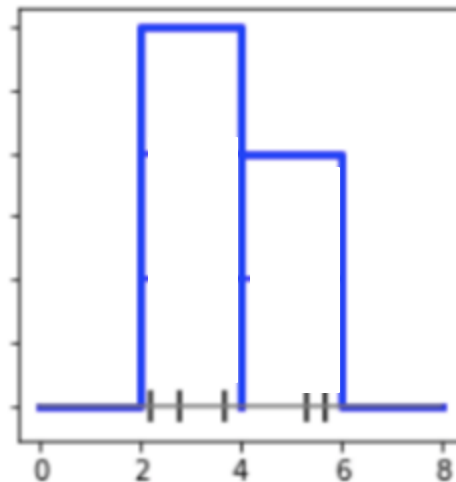
## (Intuition) Density Histogram: Proportional Areas

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

In a density histogram, **area = proportion** (i.e. fraction of data points)

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

What should the values on the y-axis be to make this a density histogram?

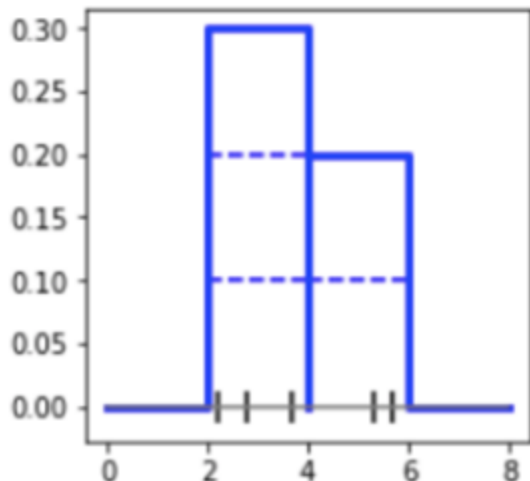


## (Intuition) Density Histogram: Proportional Areas

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

In a density histogram, **area = proportion** (i.e. fraction of data points)

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



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

Each of the  $N = 5$  points:

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

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

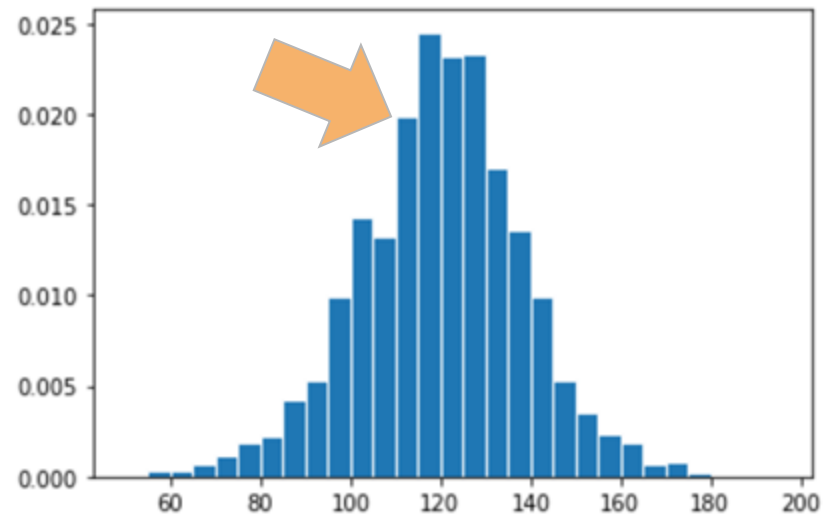
## Computing Count from Bin Size and Density

There are 1174 observations total.

Poll:

**Approximately how many observations are in the bin [110, 115) ?**

**What are the units of the y-axis?**



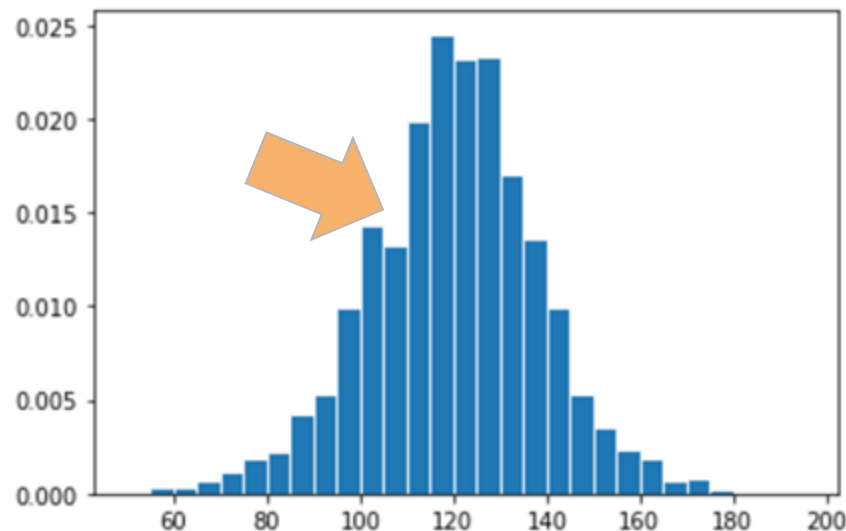
Weights of Babies (in oz)

## Computing Count from Bin Size and Density

There are 1174 observations total.

- Width of bin  $[110, 115)$ : 5
- Height of bar  $[110, 115)$ : approximately 0.02
- Proportion in bin  $= 5 * 0.02 = 0.1$
- Number in bin  $= 0.1 * 1174 = \mathbf{117.4}$

Units of y-axis: Fraction of data points per oz



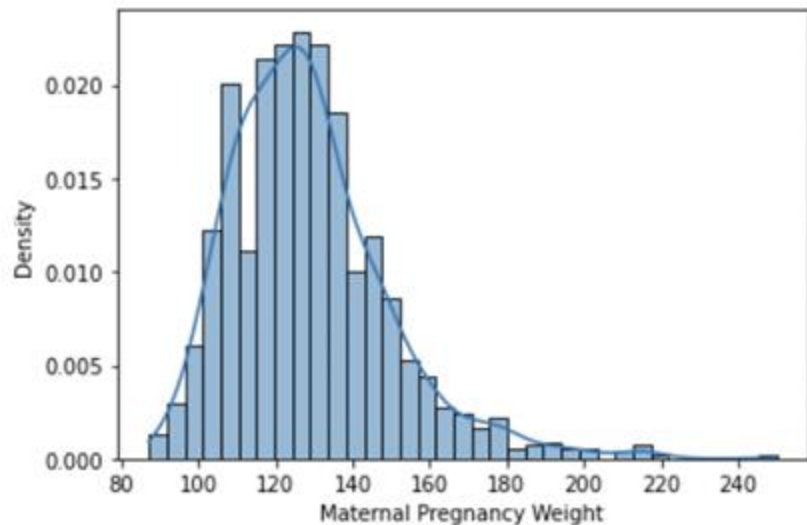
Weights of Babies (in oz)

Actual number of data points with babies weight in  $[110, 115)$ :

```
len(births.query("`Birth Weight`<115 and `Birth Weight`>=110"))
```

```
>>> 117
```

## 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 (or a Kernel Density Estimate (KDE))**.

- Density curves are a smoothed versions of histograms.
- Seaborn will calculate these automatically for you (using a technique is called Kernel Density Estimation).

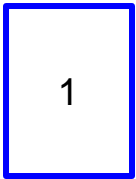
```
sns.histplot(data = births, x = 'Maternal Pregnancy Weight', stat='density', kde = True);
```

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

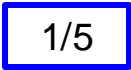
Kernel Density Estimation is used to estimate a **probability density function** (or **density curve**) from a set of data.

(We'll formally define probability density functions in a few weeks!)

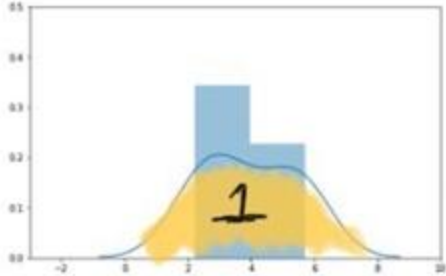
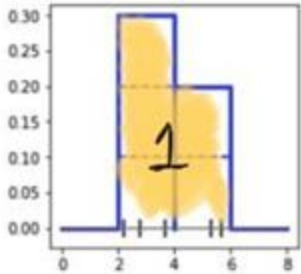
- Just like a histogram, a density curve's **total area must sum to 1**.



curve with area 1



squash per  
datapoint



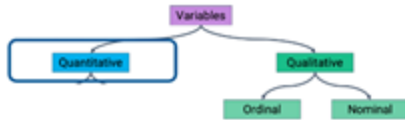
Sum together to  
make a curve

## Density curves

---

Practice: Create a KDE density plot of the babies' birth weights.

# Visualizing Distributions: Bar Chart or Histogram?

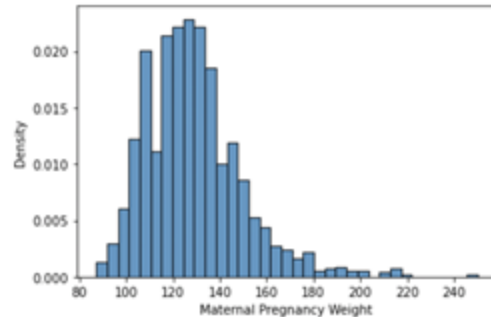


To display a **distribution** (one variable):



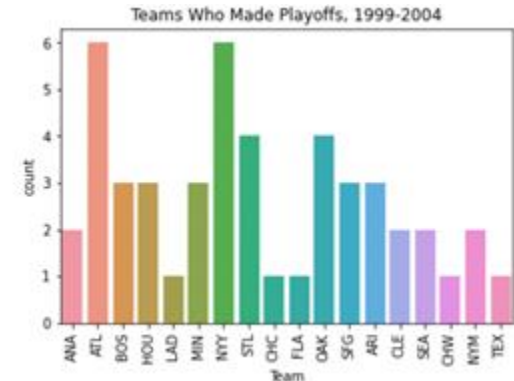
## Histogram

- Distribution of one **quantitative** variable
- Horizontal axis is numerical: drawn to scale, no gaps
- **Density Histogram**: Area of bars equals proportion of data points in a given bin; height measures density



## Bar Chart

- Distribution of one **qualitative** variable
- Bars have arbitrary (but equal) widths and spacings; in any order
- **height** of bars proportional to the percent of individuals





## Learning Objectives

- Use Python functions to visualize distributions of Quantitative Variables

# Visualizing Distributions

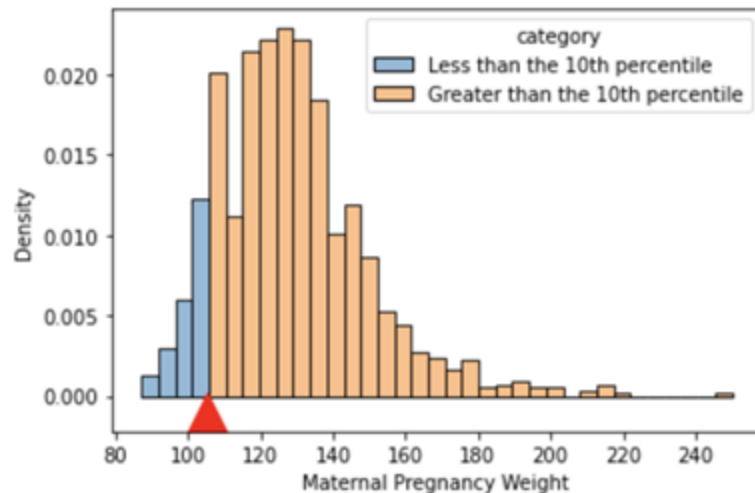
---

- Visualizing Distributions of Quantitative Variables:
  - Histograms & Density Curves
  - **Boxplots and Violin Plots**

# Percentiles

The *n*th percentile is that value  $q$  such that *n*% of the data values fall at or below it.

The value  $q$  might not be unique, and there are several approaches to select a unique value from the possibilities. With enough data, there should be little difference between these definitions.



```
p10 = np.percentile(births["Maternal Pregnancy Weight"], 10)
```

<https://numpy.org/doc/stable/reference/generated/numpy.percentile.html>

For our class we will calculate percentiles using the numpy percentile function with its default settings.

# Quartiles

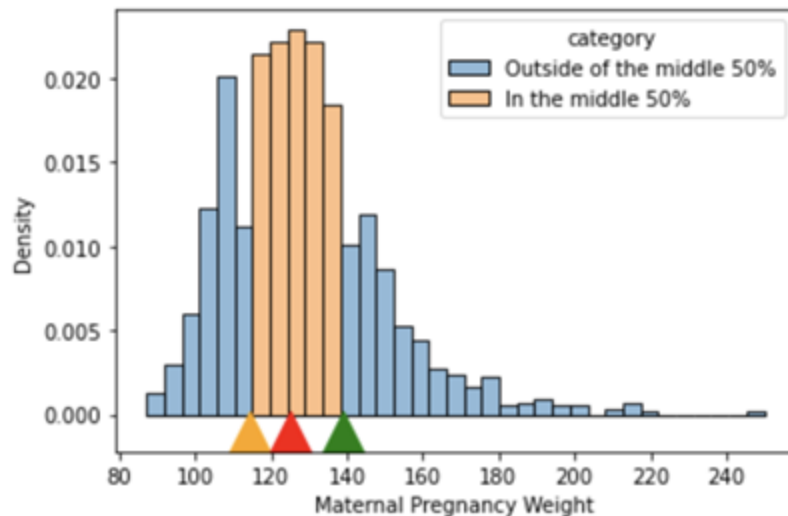
For a quantitative variable:

- 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.

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

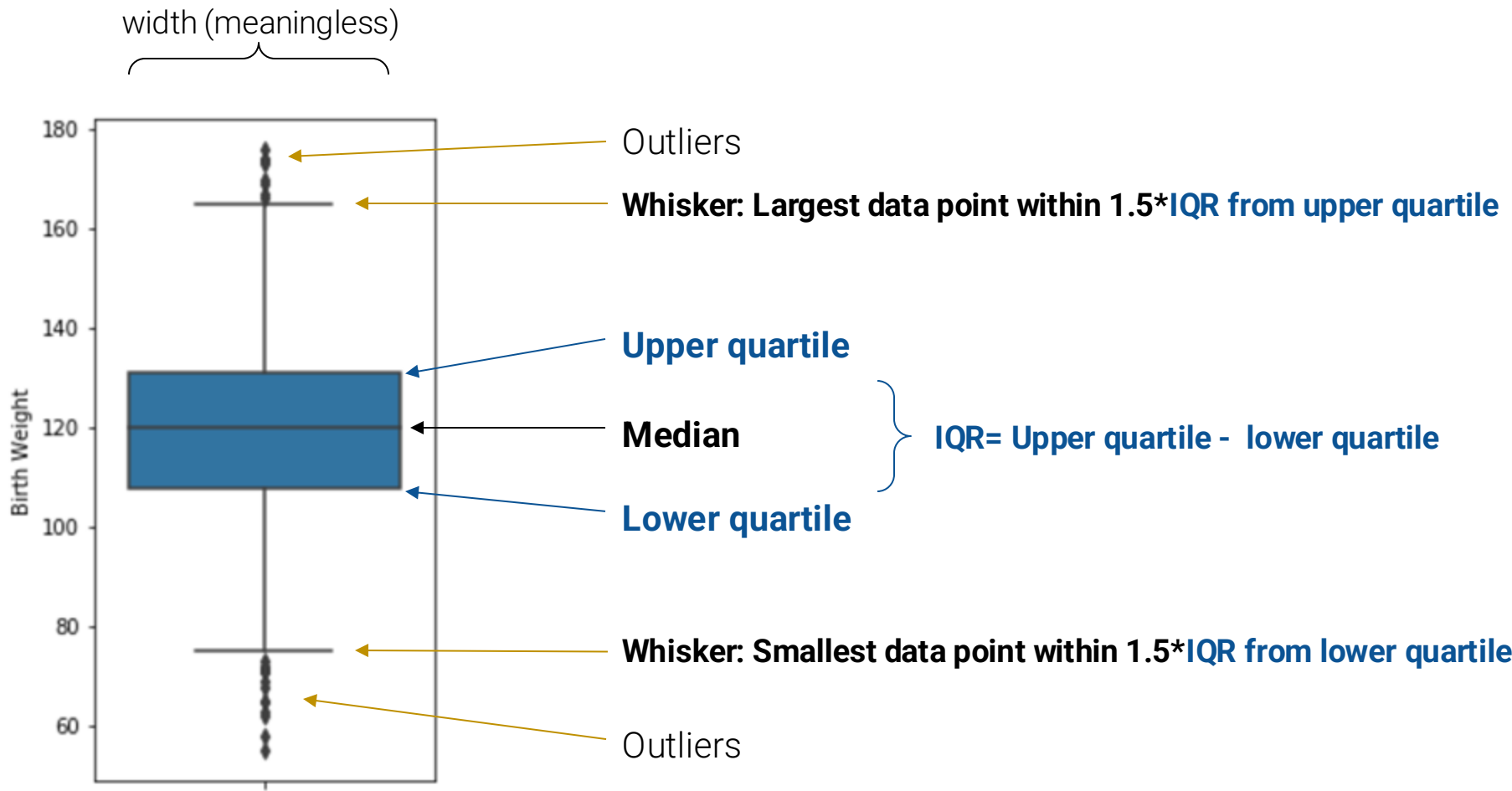


```
q1, median, q3 = np.percentile(births['Maternal Pregnancy Weight'], [25, 50, 75])
```

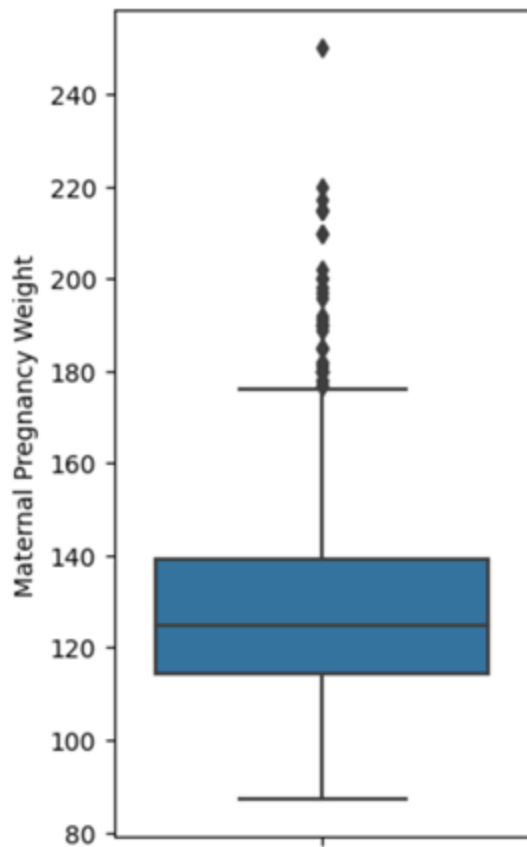
```
display([q1, median, q3])
```

```
[114.25, 125.0, 139.0]
```

# Box Plot of Baby Birth Weights:



## Box Plots

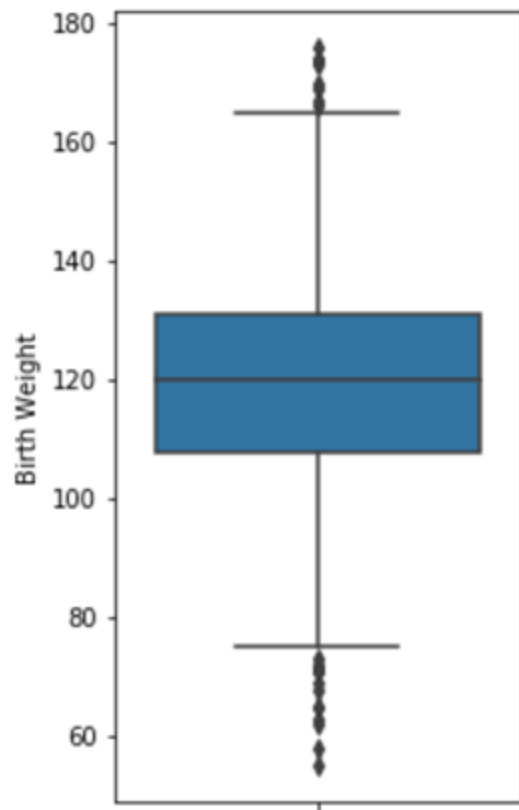


Box plots summarize several characteristics of a numerical distribution. They visualize:

- **Lower quartile.**
- **Median.**
- **Upper quartile.**
- **“Whiskers”**, The whiskers extend from the edges of box to show the range of the data. By default, they extend no more than  $1.5 * \text{IQR}$  ( $\text{IQR} = Q3 - Q1$ ) from the edges of the box, ending at the farthest data point within that interval.
- **Outliers**, which are defined as being further than  $1.5 * \text{IQR}$  from the extreme quartiles. (Arbitrary definition!)
- We lose a lot of information, too!

`sns.boxplot(data = births, y = 'Maternal Pregnancy Weight')`

## Box Plot of Baby Birth Weights:

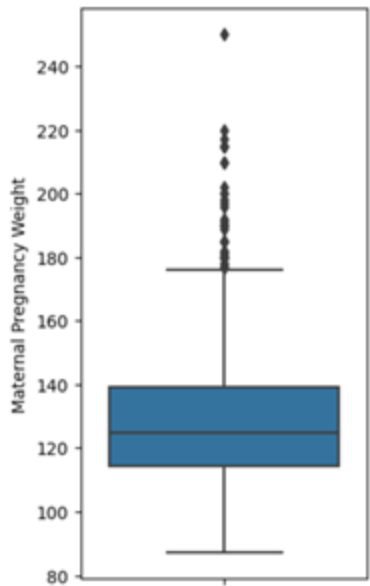


```
bweights = births['Birth Weight']  
q1 = np.percentile(bweights, 25)  
q2 = np.percentile(bweights, 50)  
q3 = np.percentile(bweights, 75)  
iqr = q3 - q1
```

q1, q2, q3

(108.0, 120.0, 131.0)

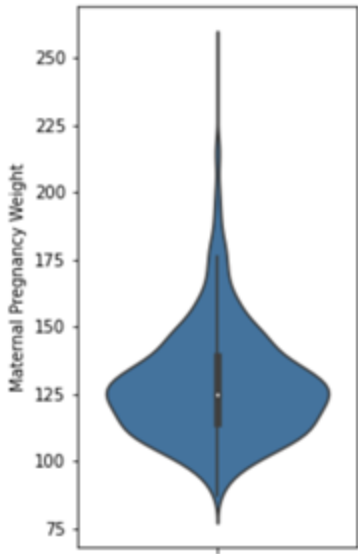
## Box Plot



```
sns.boxplot(data = births, y = 'Maternal Pregnancy Weight')
```

## Violin Plot

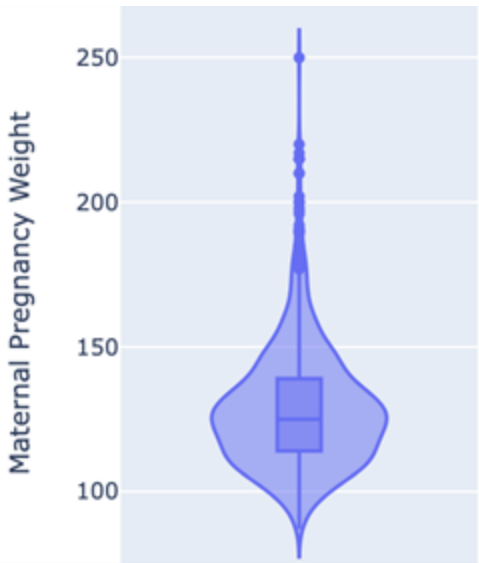
Violin plots are similar to box plots, but also show smoothed density curves.



```
sns.violinplot(data = births, y = 'Maternal Pregnancy Weight')
```

- The "width" of our "box" now has meaning!
- The three quartiles and "whiskers" are still present – look closely.

## Box plot AND Violin Plot



```
px.violin(births, y = 'Maternal Pregnancy Weight', box=True)
```

## Learning Objectives

- Describe quantitative distributions

# Describing Quantitative Distributions

---

- Describing Distributions of Quantitative Variables:
  - Measures of Center
  - Skew



## Describing distributions

---

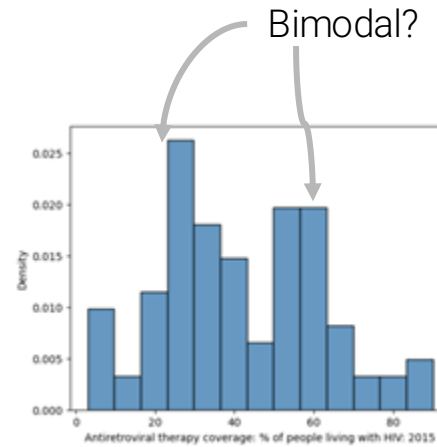
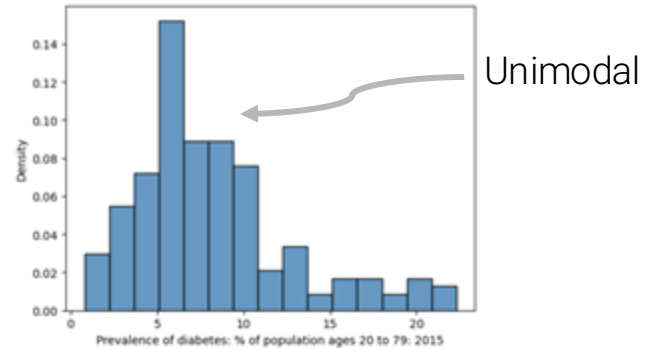
One of the benefits of a histogram or density curve is that they show us the “bigger picture” of our distribution

Some of the other terminology we use to describe distributions:

- **Modes.**
- **Skewness.**
  - Skewed left vs skewed right.
- **Outliers.**
  - Define these arbitrarily.
  - One definition is any data that is further than  $1.5 \times \text{IQR}$  from the extreme quartiles

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.

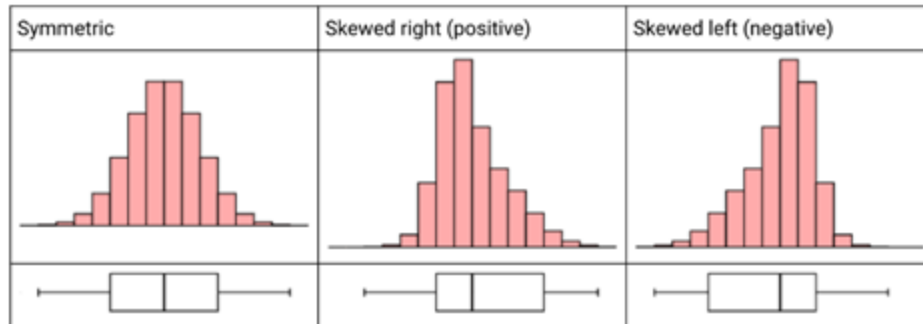
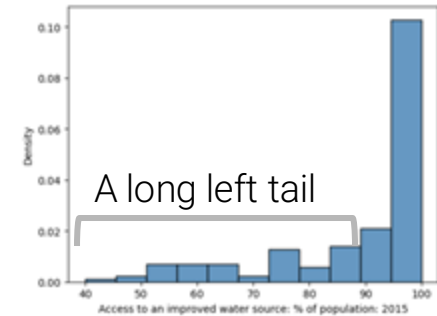
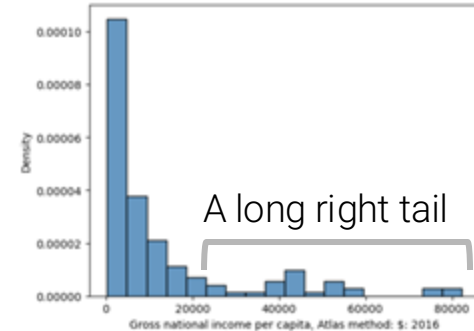


\*For discrete or categorical data, the mode is the most commonly occurring value. For continuous data, the mode is any local max.

# Skew of a Histogram

The **skew** of a histogram describes the direction in which its "tail" extends.

- A distribution with a **long right tail is skewed right (aka positively skewed)**.
  - In such cases, the mean is typically to the right of the median.
- A distribution with a **long left tail is skewed left (aka negatively skewed)**.
  - In such cases, the mean is typically to the left of the median.
- A distribution with no clear skew is called symmetric.



# Characterizing Quantitative Data

Summarizing the “center” of the sample is a popular way to characterize quantitative data.

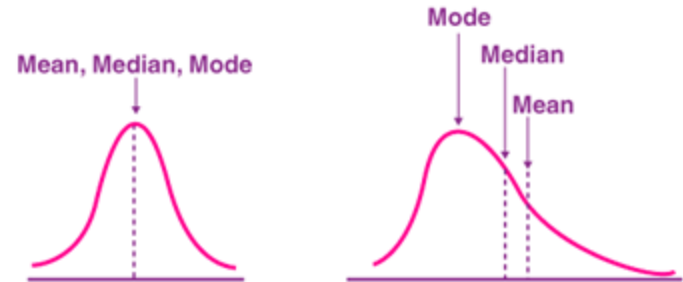
**Mean:**

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{1}{n} \sum_{k=1}^n x_k$$

**Median:** The *middle element* of the dataset when it is put in ascending order.

Note: When  $n$  is even, take the average of the middle two elements.

**Mode:** Most commonly occurring value in a data set.



\*Note about modes: For categorical or discrete variables, multiple modes are values that reach the same frequency: the highest one observed. For continuous variables, all peaks of the distribution can be considered modes even if they don't have the same frequency (i.e. any local max is a mode)

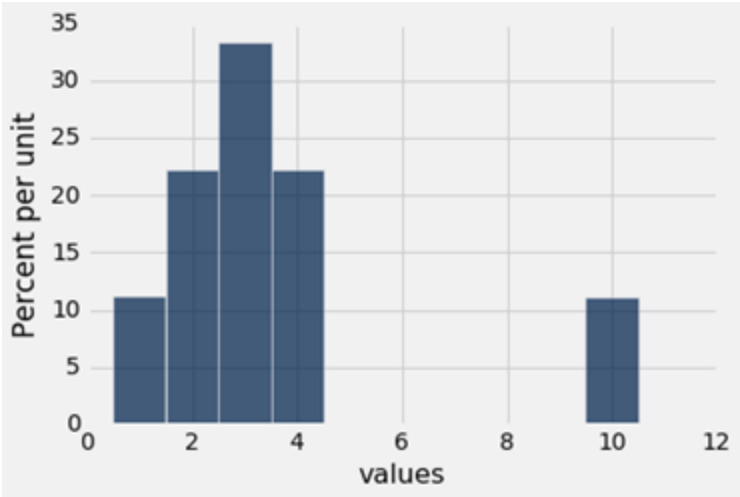
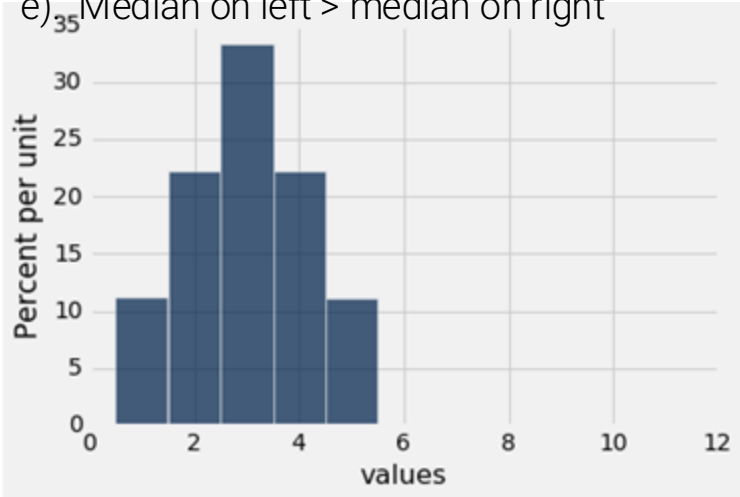
# Discussion Question: Mean vs Median

Are the medians of these two distributions the same or different? Are the means the same or different?

If you say “different,” then say which one is bigger.

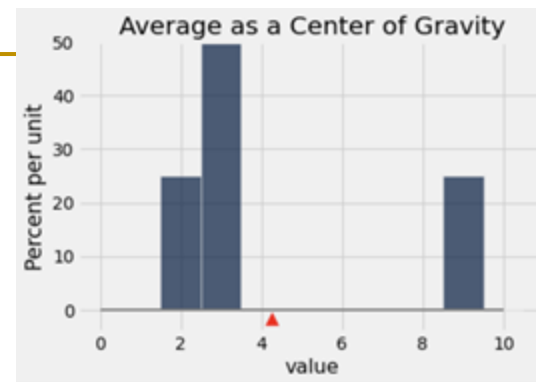
Select all that apply.

- A). Means Equal right mean
- B). Medians Equal
- C). Left mean > Right mean
- d). Left mean <
- e). Median on left > median on right

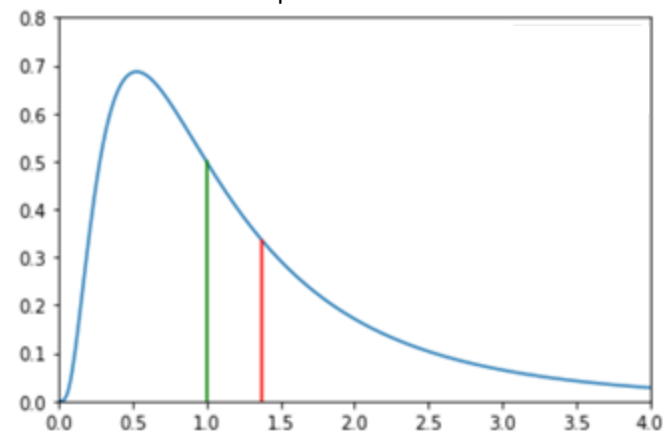


## Comparing Mean and Median

- **Mean:** Balance point of the histogram
  - Physics Analogy: Center of Gravity
- **Median:** 50th percentile of the data
- If the distribution is symmetric about a value, then that value is both the average and the median
- If the histogram is skewed, then the mean is pulled away from the median in the direction of the skew (tail)

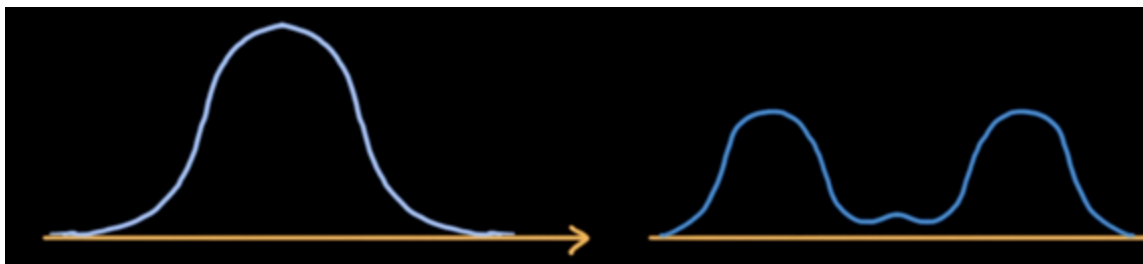
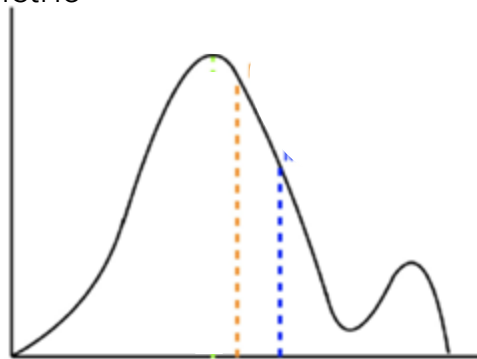
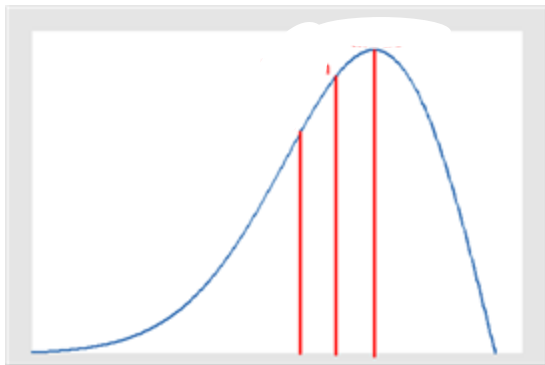


Which is the mean and which is the median in the plot below?



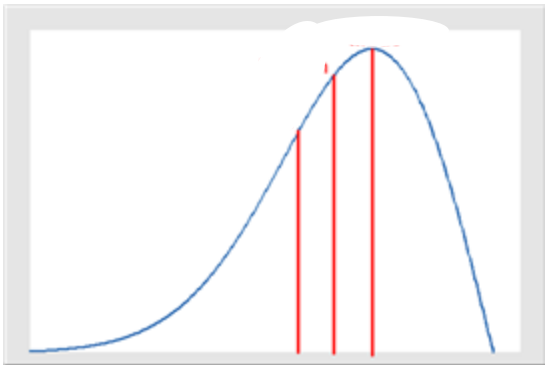
## Mean vs Median vs Mode:

For each of the density curves below, label the mean, median and any mode(s). Then classify each distribution as left-skewed, right-skewed or symmetric



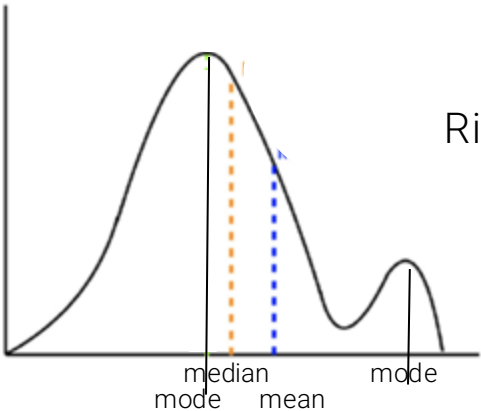
# Mean vs Median vs Mode:

For each of the density curves below, label the mean, median and any mode(s)

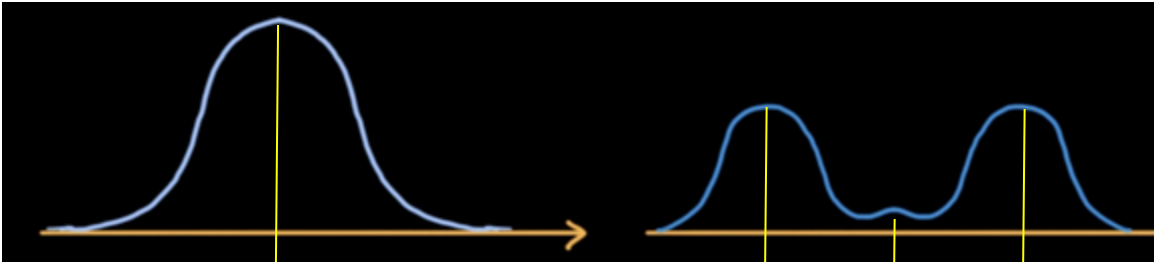


Left-skewed

Mean, median, mode



Right-skewed



Symmetric

Mode  
Median  
mean

mode      Median  
Mean  
mode



## Learning Objectives

- Use Python functions to visualize relationships between variables

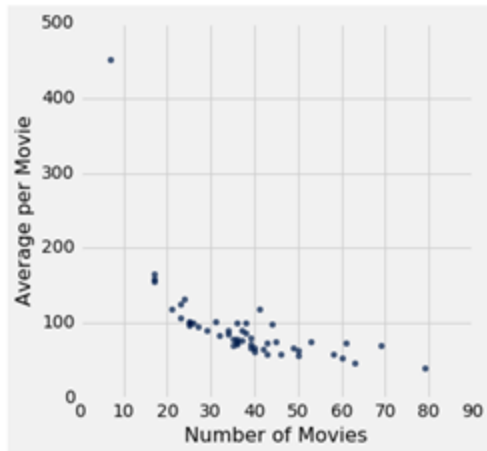
# Visualizing Relationships Between Variables

---

- Relationships between variables:
  - Quantitative
  - Qualitative
  - Mixed

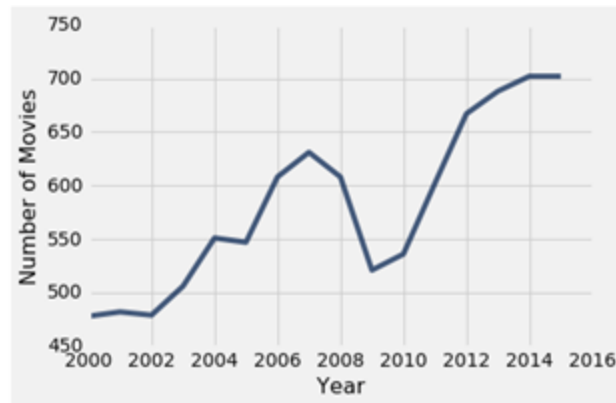
# Plotting Relationship Between Two Quantitative Features

## Scatter plot :



- Use scatter plots for non-sequential numerical data
- Plot one quantitative continuous variable on the x-axis, and second quantitative continuous variable on the y-axis.
- Each scatter point represents one datapoint in the dataset.
- Use if you are looking for associations between 2 variables
- Use there isn't a unique output for each input

## Line plot: Scatter plot with points connected by straight lines



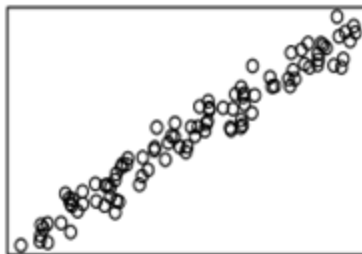
- Use line plots for sequential quantitative data: if...
  - ...your x-axis has an order
  - ...sequential differences in y values are meaningful
  - ...there's only one y-value for each x-value
    - x-axis is **time** or **distance**

# Scatter plots

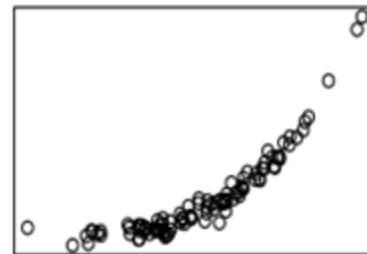
Scatter plots are used to reveal relationships between **pairs** of numerical variables.

- Visual assessment may help us decide how to model these relationships.
- Example: Linear model
  - Linear Regression (we'll introduce this later in the course)
  - Good for the left two, not so much for the right two.
- Coming Later in the Course:  
"Correlation does not imply causation."  
A linear relationship is a mathematical one.

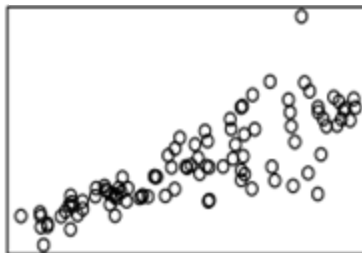
**simple linear**



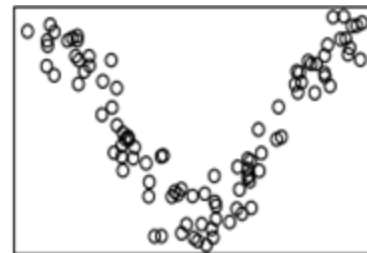
**simple nonlinear**



**linear, spreading**

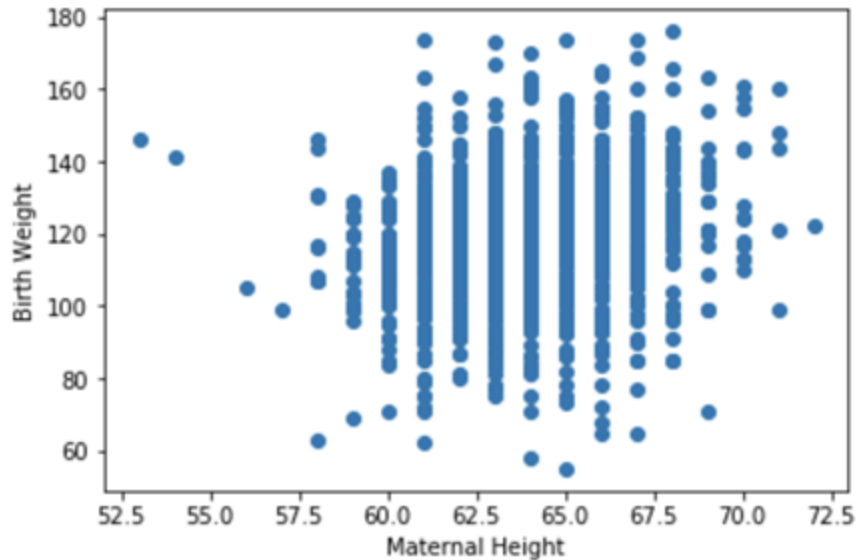


**v-shaped**



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

## Scatter Plot on our Birth Data (Matplotlib Example)



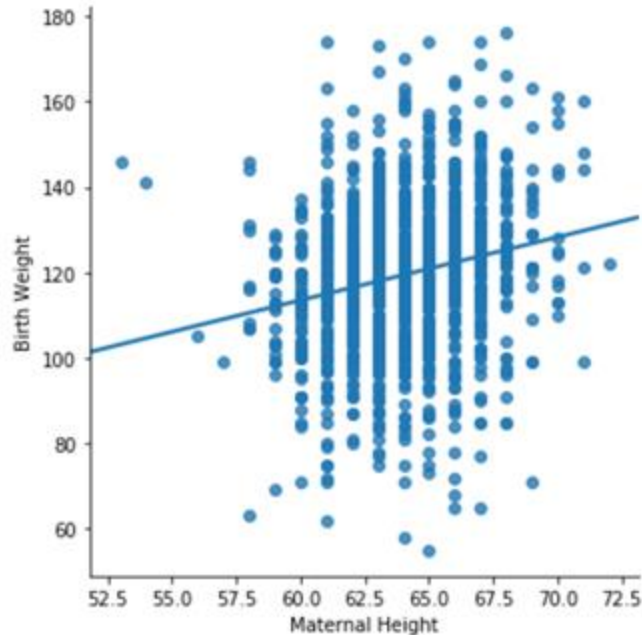
```
plt.scatter(births['Maternal Height'], births['Birth Weight']) # array/series  
# OR
```

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

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

## Scatter Plot Alternatives

Seaborn includes several built-in functions for making more complex scatter plots.



We'll learn how to create this linear model later in the course!

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

See Supporting Materials for  
scatter plot alternatives

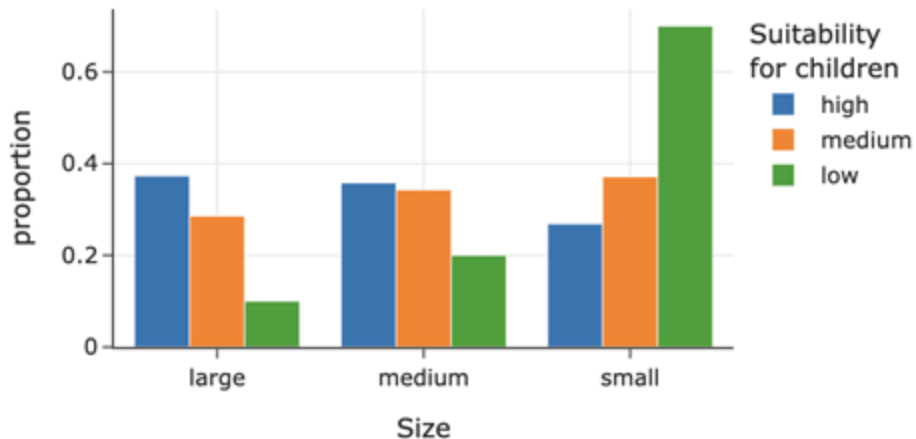
# Visualizing Relationships Between Qualitative Variables

---

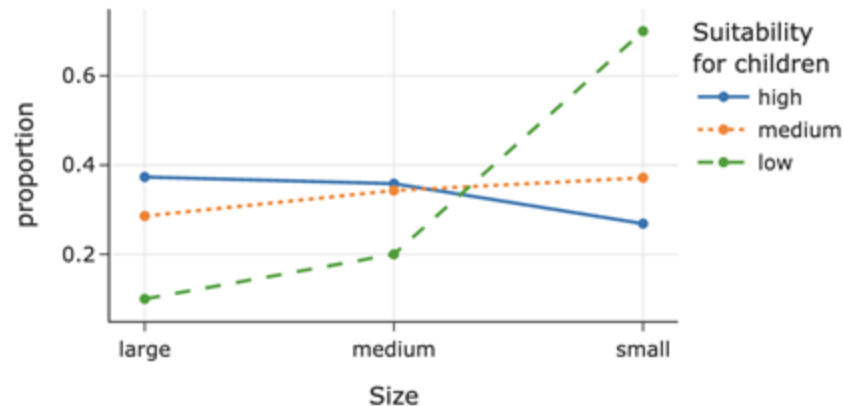
- Comparing Quantitative Distributions
- **Relationships between variables:**
  - Quantitative
  - **Qualitative**
  - Mixed

# Visualizing Relationships Between Two Qualitative Features

## Side-By-Side Bar Charts



## Overlaid Line Charts



# Visualizing Relationships Between Mixed Variables

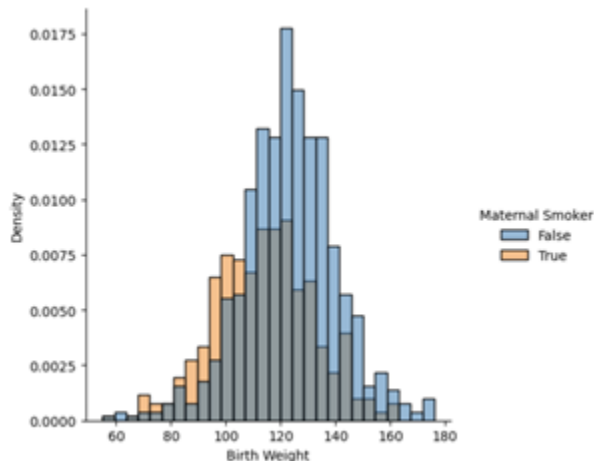
---

- Comparing Quantitative Distributions
- **Relationships between variables:**
  - Quantitative
  - Qualitative
  - **Mixed**

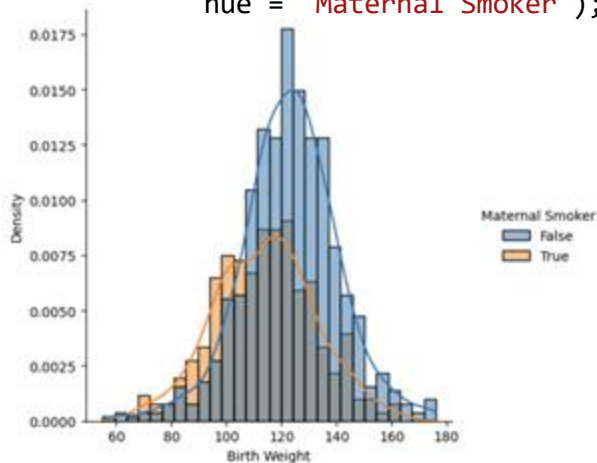


# Overlaid Histograms and Density Curves

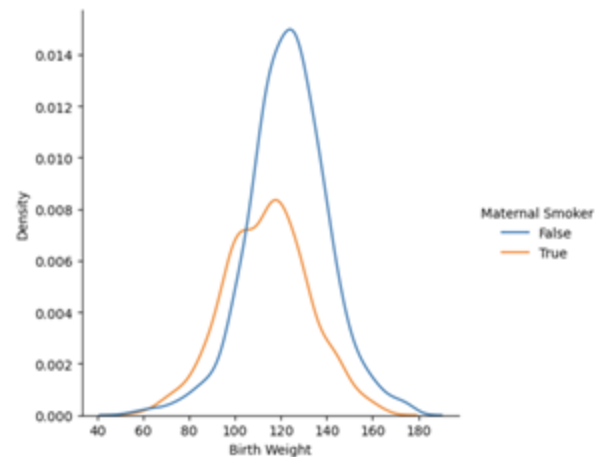
```
sns.displot(data = births,  
            x = 'Birth Weight',  
            stat = 'density',  
            hue = 'Maternal Smoker');
```



```
sns.displot(data = births,  
            x = 'Birth Weight',  
            kde = True,  
            stat = 'density',  
            hue = 'Maternal Smoker');
```



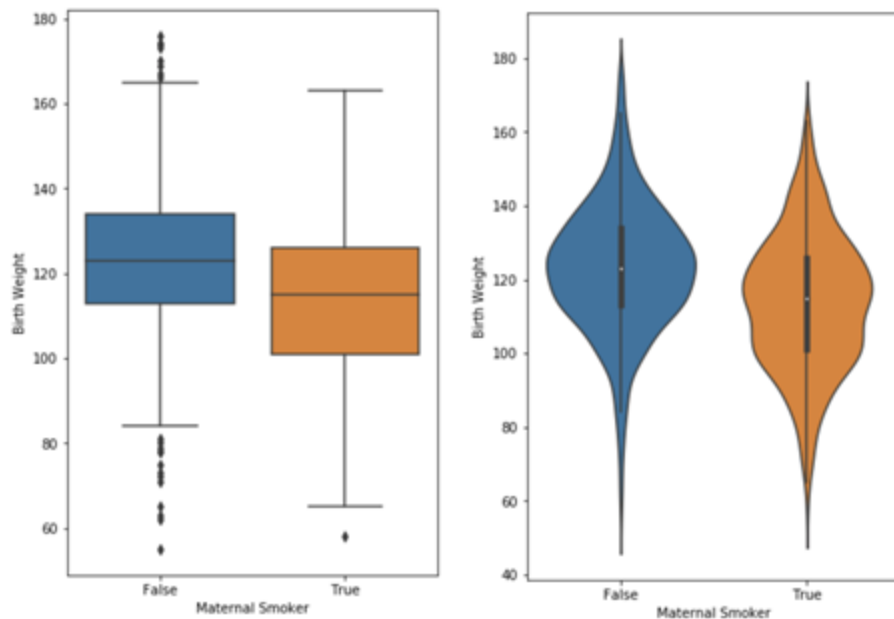
```
sns.displot(data = births,  
            x = 'Birth Weight',  
            kind = 'kde',  
            hue = 'Maternal Smoker');
```



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



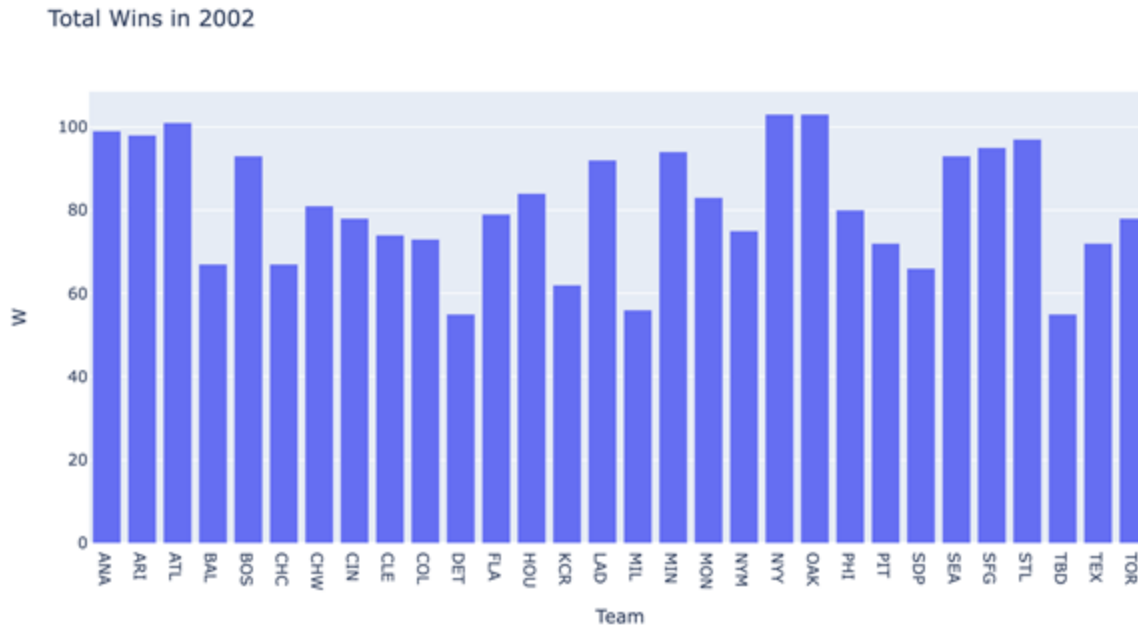
Box plots and violin plots are concise, and thus are well suited to be stacked side by side to compare multiple distributions at once.

- 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.

```
sns.boxplot(data = births, x = 'Maternal Smoker', y = 'Birth Weight')  
sns.violinplot(data = births, x = 'Maternal Smoker', y = 'Birth Weight')
```

## Plotting Relationship Between A Quantitative and Qualitative Feature:

### Bar Plots



- Vertical : Height encode values of the quantitative feature
- *Widths* encode *nothing*!
- Spaces between qualitative names to indicate not continuous and not a distribution

# Visualizing Data: Summary

---

## Review: Goals of Data Visualization

---

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

- Key part of exploratory data analysis.
- Summarize trends visually before in-depth analysis.
- Lightweight, iterative and flexible.

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

- Highly editorial and selective.
- Be thoughtful and careful!
- Fine-tuned to achieve a communications goal.
- Considerations: clarity, accessibility, and necessary context.

What do these goals imply?

Visualizations aren't a matter of making "pretty" pictures.

We need to do a lot of thinking about what stylistic choices communicate ideas most effectively.

# Summary of Types of Visualizations:

Feature Type	Dimension	Types of Visualizations: Which Visualizations on the right should be used? Select all that apply.
Quantitative	1 Feature	
Qualitative	1 Feature	
Quantitative	2 Features	
Qualitative	2 Features	
1 Quant and 1 Qual	2 Features	



A). Bar chart histograms



B). Histogram



C). Bar Chart



D). violin plot



E) Overlaid



F). Density curve



G). Scatter plot



H). Line Graph



I). Scatter plot



J)



K). Boxplot



L). side by side boxplots



M). side by side violin plots



N). overlaid density curves



O). Side by side bar charts



P). Overlaid line graphs

# Summary of Types of Visualizations:

Feature Type	Dimension	Types of Visualizations: Which Visualizations on the right (letters) should be used?
Quantitative	1 Feature	B, D, F, K
Qualitative	1 Feature	C
Quantitative	2 Features	G, H - only if scatter plot passes vertical line test J - only if one of the variables has a few discrete values (which will be used as the x-axis)
Qualitative	2 Features	O, P
1 Quant and 1 Qual	2 Features	A, E I, J, L, M, N



A). Bar chart histograms



B). Histogram



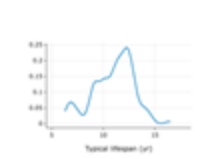
C). Bar Chart



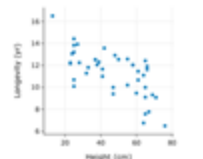
D). violin plot



E). Overlaid



F). Density curve



G). Scatter plot



H). Line Graph



I). Scatter plot



J).



K). Boxplot



L). side by side boxplots



M). side by side violin plots



N). overlaid density curves



O). Side by side bar charts



P). Overlaid line graphs

# Summary of Types of Visualizations:

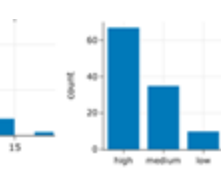
Feature Type	Dimension	Types of Visualizations: Which Visualizations on the right (letters) should be used?
Quantitative	1 Feature	B, D, F, K
Qualitative	1 Feature	C
Quantitative	2 Features	G, H - only if scatter plot passes vertical line test J - only if one of the variables has a few discrete values (which will be used as the x-axis)
Qualitative	2 Features	O, P
1 Quant and 1 Qual	2 Features	A, E I, J, L, M, N



A). Bar chart histograms



B). Histogram



C). Bar Chart



D). violin plot



E). Overlaid



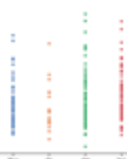
F). Density curve



G). Scatter plot



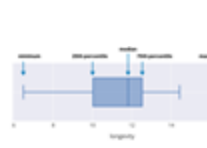
H). Line Graph



I). Scatter plot



J).



K). Boxplot



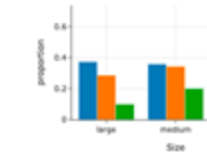
L). side by side boxplots



M). side by side violin plots



N). overlaid density curves



O). Side by side bar charts



P). Overlaid line graphs



# Supporting Materials

---

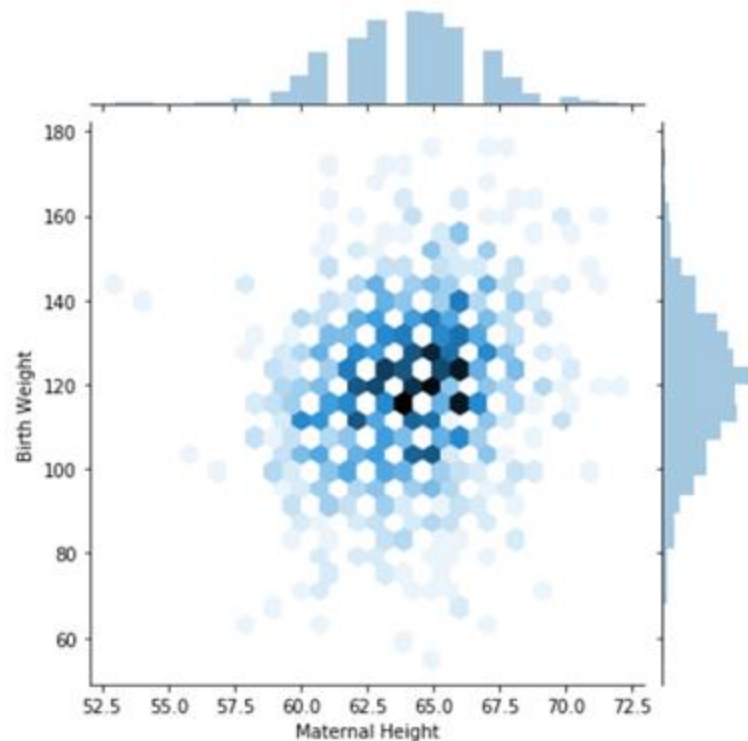
## Hex Plots

Rather than plot individual datapoints, plot the *density* of their joint distribution.

Can be thought of as a two dimensional histogram.

- The xy plane is binned into hexagons.
- More shaded hexagons typically indicate a greater density/frequency = more datapoints lie in that spot

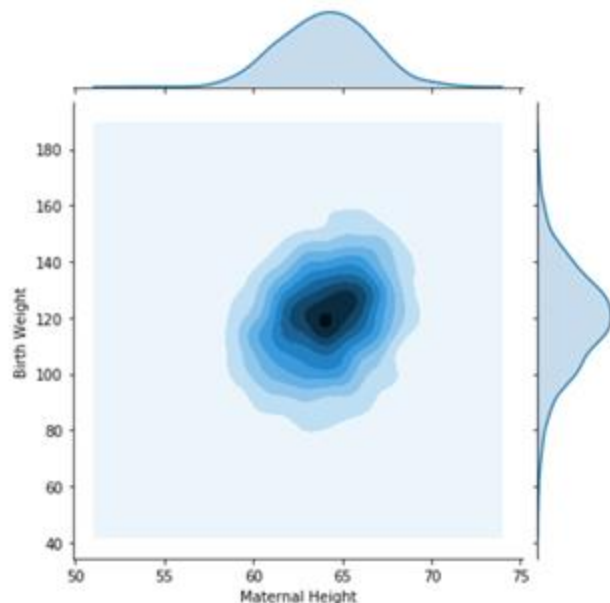
```
sns.jointplot(data=births, x='Maternal Height',  
y='Birth Weight', kind='hex')
```



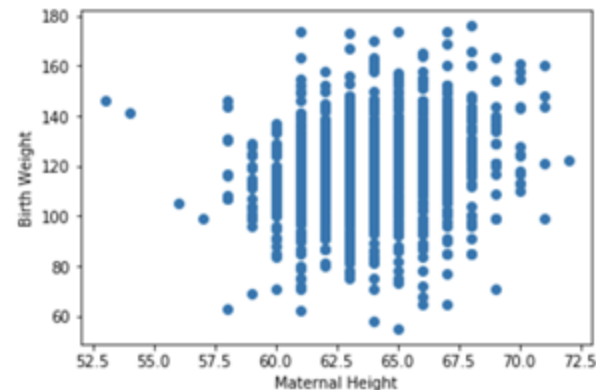
# Contour Plots

2-dimensional version of a density curves

Similar to a topographic map – contour lines represent an area that has the *same density* of datapoints throughout. Darker colors indicate more datapoints in the region.



Dark color → many datapoints



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

# Plotting Relationship Between Two Quantitative Features

