LECTURE 8

Visualization

Visualizing distributions and KDEs

Data Science, Spring 2024 @ Knowledge Stream

Sana Jabbar

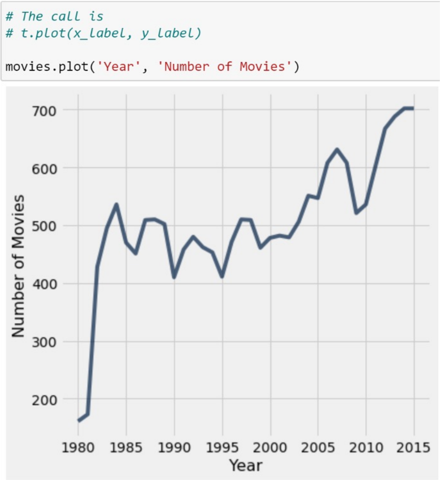
Visualization of Distribution

Lecture 8, Spring 2024

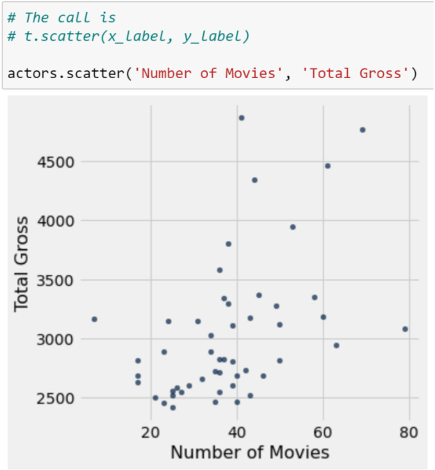
- Regex
  - Regex review and regex functions
- Visualization
  - Goals of visualization
  - **Visualizing distributions**
  - Kernel density estimation

Visualizations in BS (and in Data Science, so far)

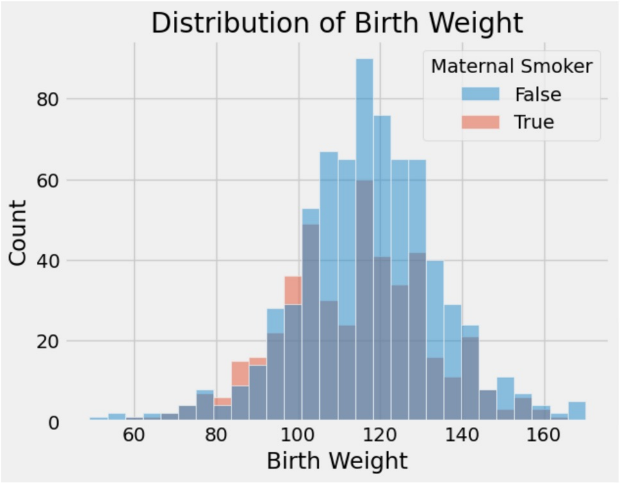
You worked with many types of visualizations throughout.



Line plot



Scatter plot



Histogram

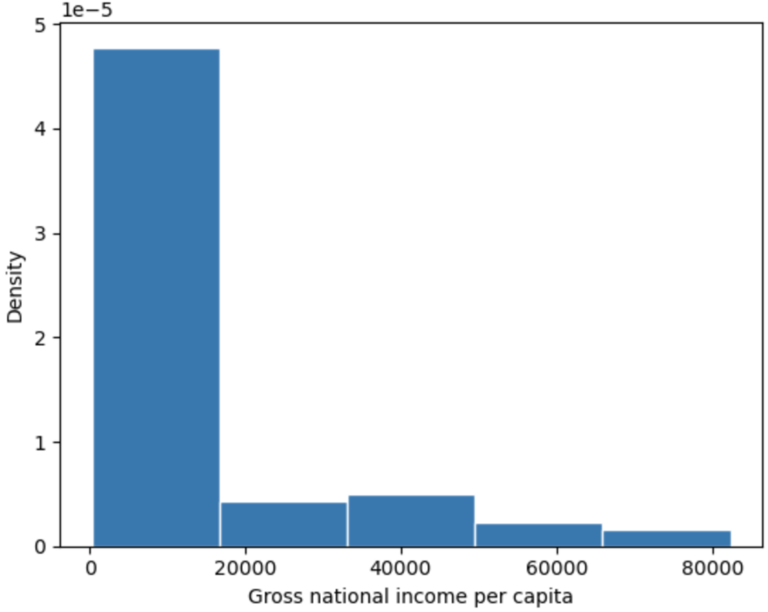
What did these achieve?

- Provide a high-level overview of a complex dataset.
- Communicated trends to viewers.

Histograms

A histogram:

- Collects datapoints with similar values into a shared "bin".
- Scales the bins such that the **area** of each bin is equal to the **percentage** of datapoints it contains.



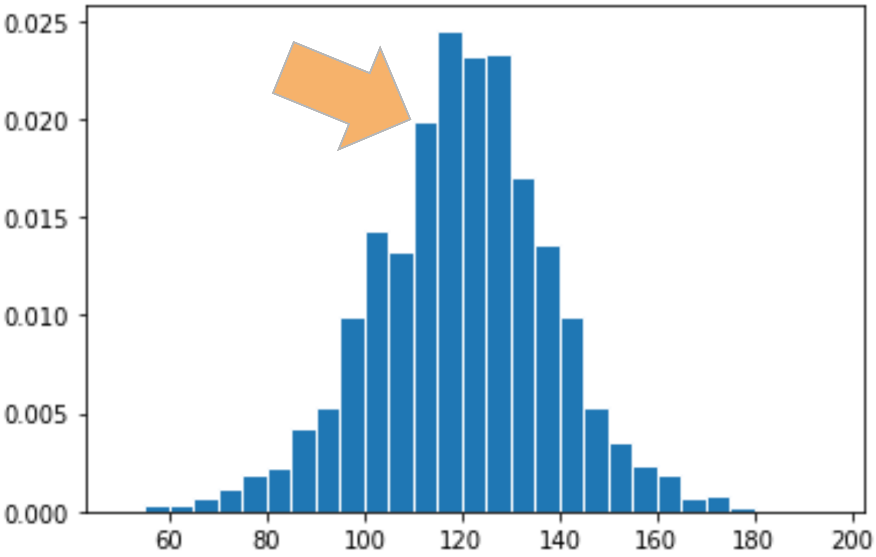
The first bin has a width of \$16410
height of 4.77×10^{-5}

This means that it contains $16410 \times (4.77 \times 10^{-5}) = 78.3\%$ of all datapoints in the dataset.

Answer

There are 1174 observations in total.

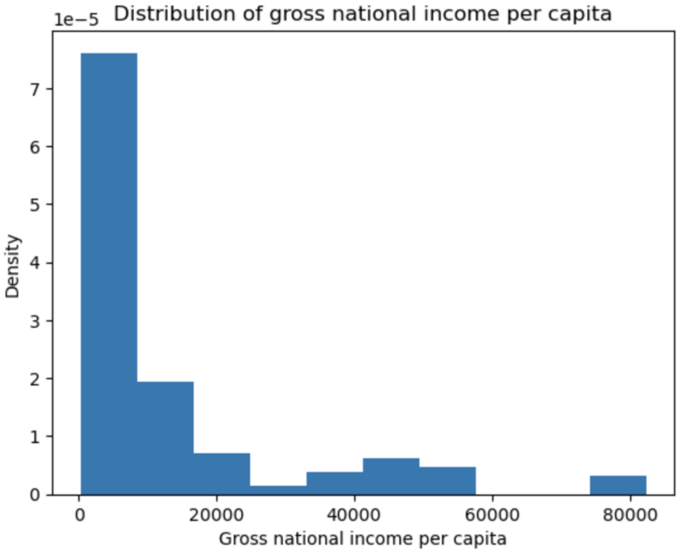
- Width of bin [110, 115): 5
- Height of bar [110, 115): 0.02
- Proportion in bin = 5 * 0.02 = 0.1
- Number in bin = 0.1 * 1174 = **117.4**



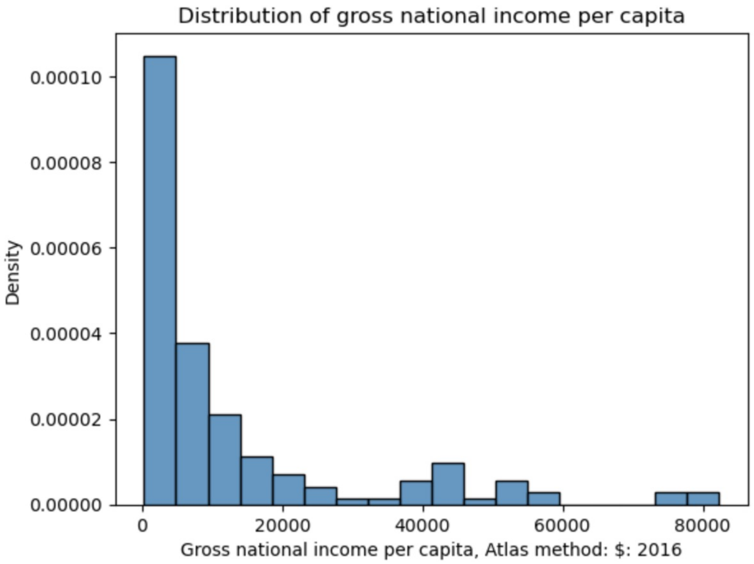
Histograms in Code

In Matplotlib: `plt.hist(x_values, density=True)`

In Seaborn: `sns.histplot(data=df, x="x_column", stat="density")`



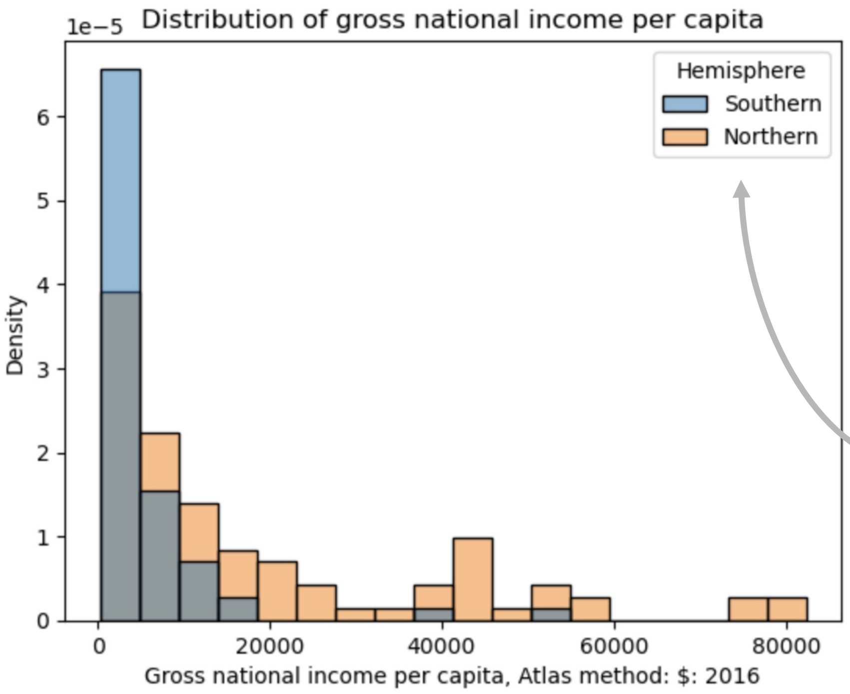
Matplotlib



Seaborn

Overlaid Histograms

To compare a quantitative variable's distribution across qualitative categories, overlay histograms on top of one another.



The **hue** parameter of Seaborn plotting functions sets the column that should be used to determine color.

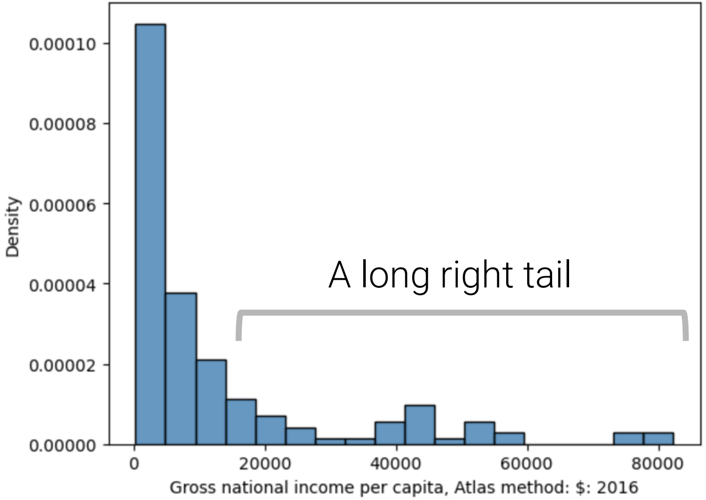
```
sns.histplot(data=wb, hue="Hemisphere",
x="Gross national income…")
```

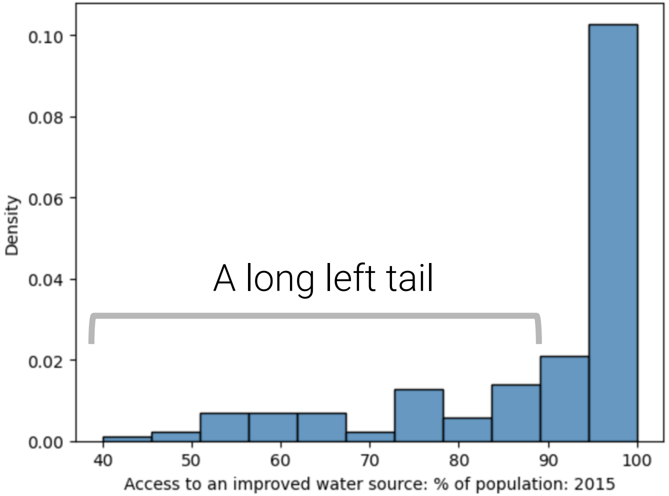
Always include a legend when color is used to encode information!

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

- A distribution with a long right tail is skewed right.
- A distribution with a long left tail is skewed left.

A histogram with no clear skew is called symmetric.

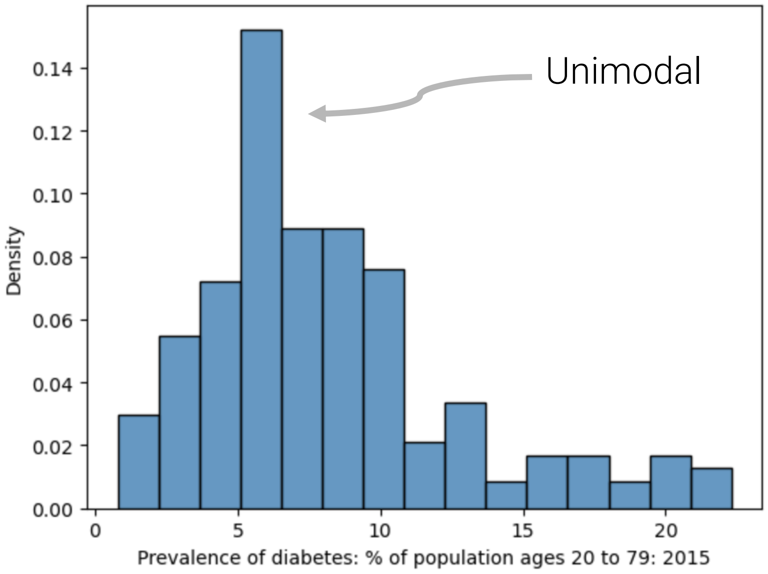


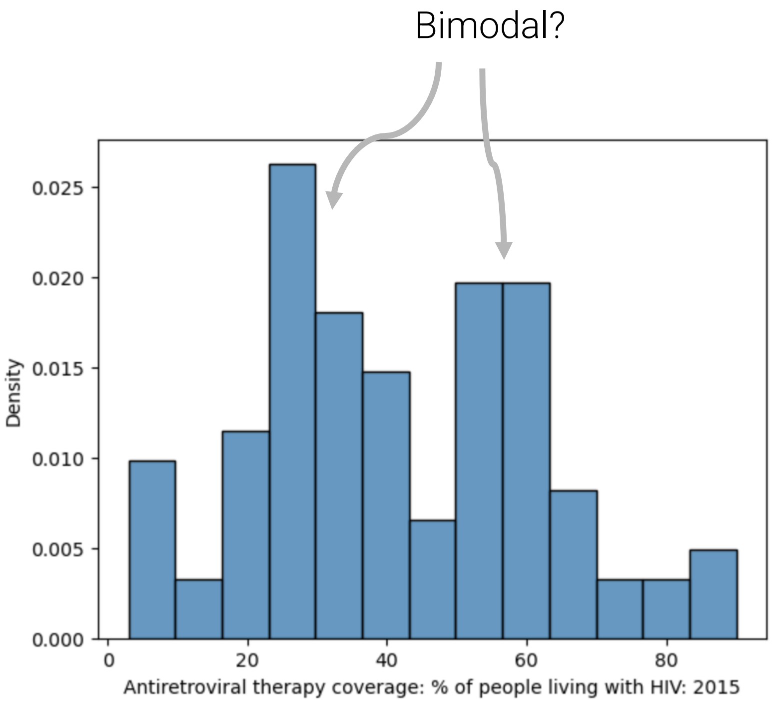


Interpreting Histograms

The **mode(s)** of a histogram are the peak values in the distribution.

- A distribution with one clear peak is called unimodal.
- Two peaks: bimodal.
- More peaks: multimodal.





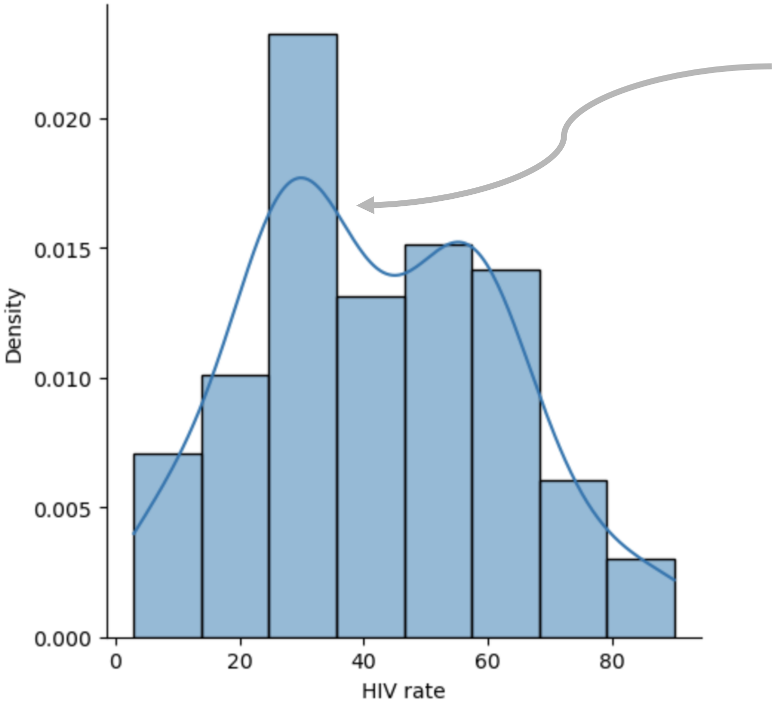
Kernel Density Estimation

Lecture 08, Spring 2024

- Regex
    - Regex review and regex functions
- **Visualization**
    - Goals of visualization
    - Visualizing distributions
    - **Kernel density estimation**

Kernel Density Estimation: Intuition

Often, we want to identify *general* trends across a distribution, rather than focus on detail.
Smoothing a distribution helps generalize the structure of the data and eliminate noise.



A KDE curve

Idea: approximate the probability distribution that generated the data.

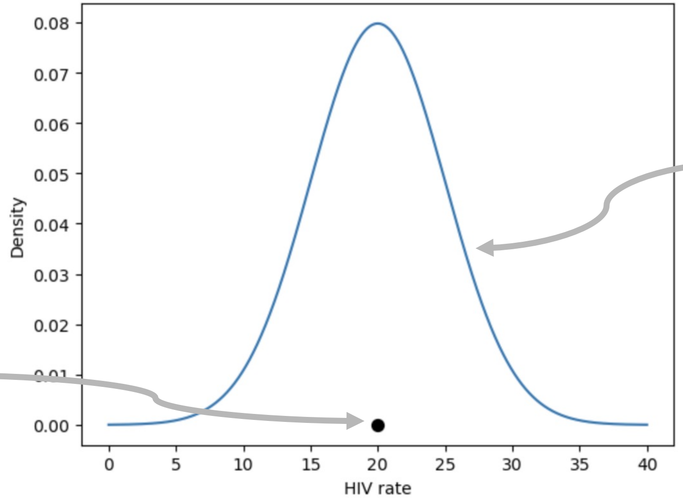
- Assign an “error range” to each data point in the dataset – if we were to sample the data again, we might get a different value.
- Sum up the error ranges of all data points.
- Scale the resulting distribution to integrate to 1.

Kernel Density Estimation: Process

Idea: Approximate the probability distribution that generated the data.

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

A **kernel** is a function that tries to capture the randomness of our sampled data.



The **kernel** models the probability of sampling that datapoint.

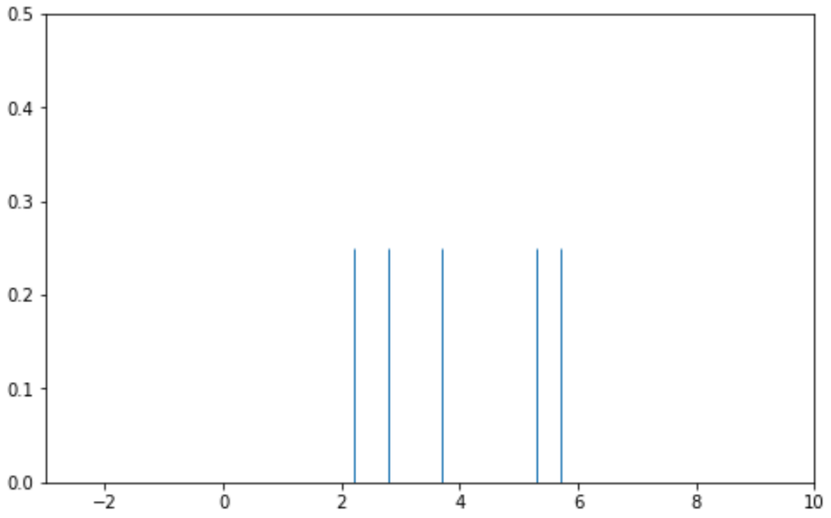
Area below integrates to 1

A **datapoint** in our dataset

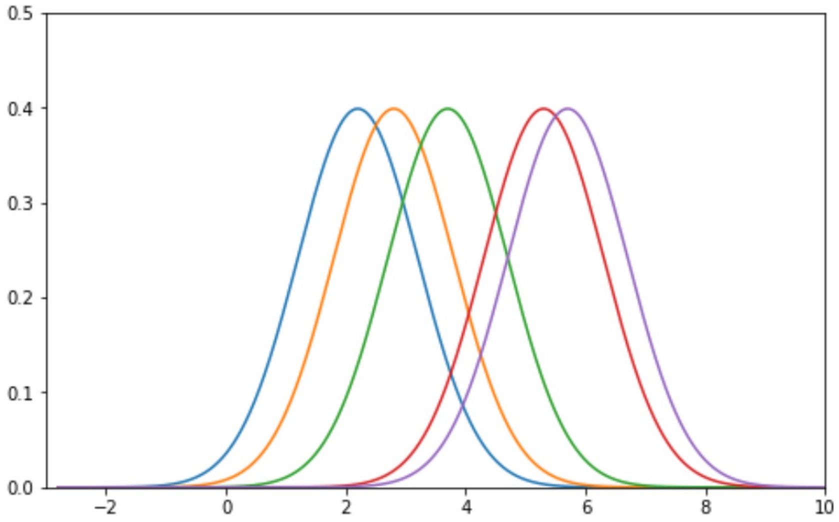
Step ① – Place a Kernel at Each Data Point

Consider a fake dataset with just five collected datapoints.

- Place a **Gaussian kernel** with **bandwidth** of **alpha = 1**.
- We will precisely define both the **Gaussian kernel** and **bandwidth** in a few slides.



Each line represents a datapoint in the dataset (e.g. one country's HIV rate).

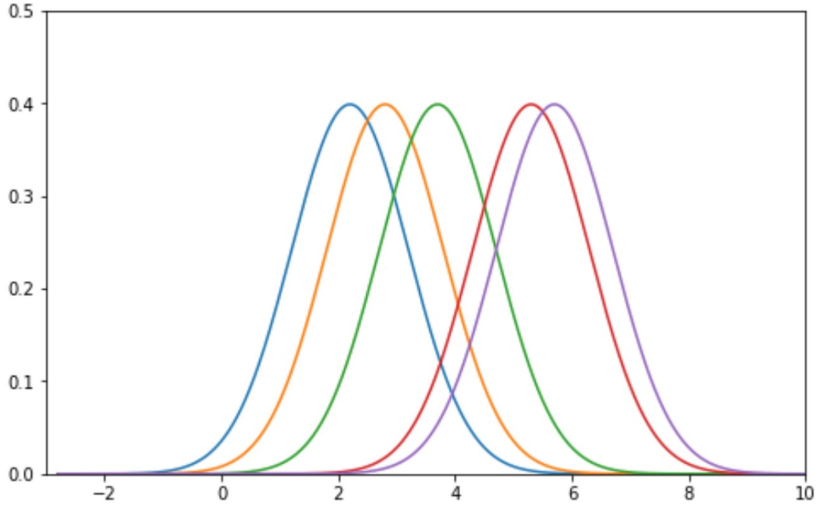


Place a kernel on top of each datapoint.

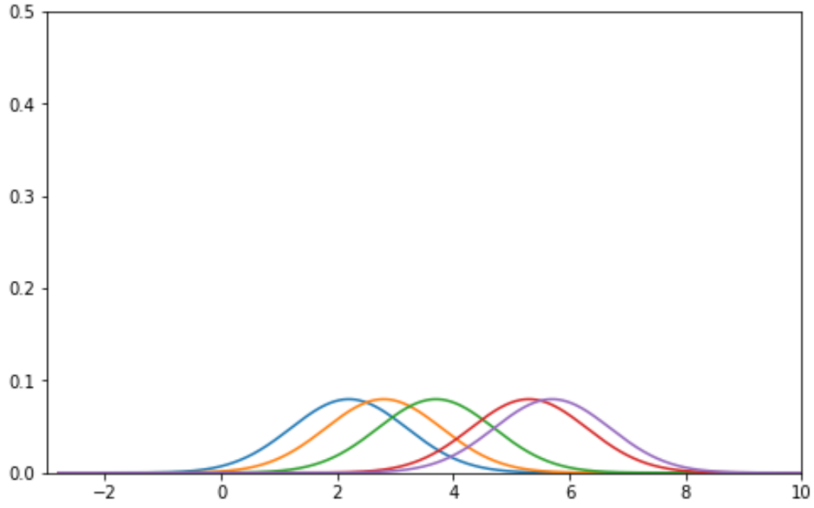
Step 2 – Normalize Kernels

In Step 3, We will be summing each of these kernels to produce a probability distribution.

- We want the result to be a valid probability distribution that has area 1.
- We have 5 different kernels, each with an area 1.
- So, we normalize by multiplying each kernel by ⅕.



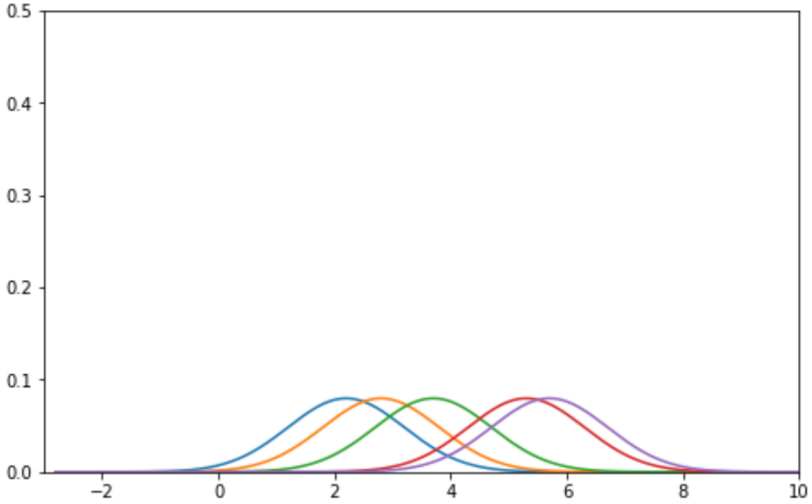
Each kernel has area 1.



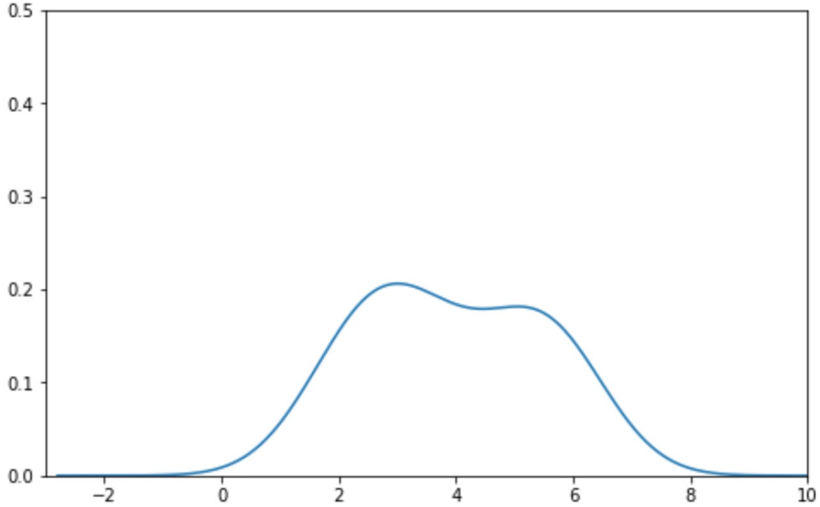
Each normalized kernel has density ⅕.

Step③ – Sum the Normalized Kernels

At each point in the distribution, add up the values of all kernels. This gives us a smooth curve with area 1 – an approximation of a probability distribution!



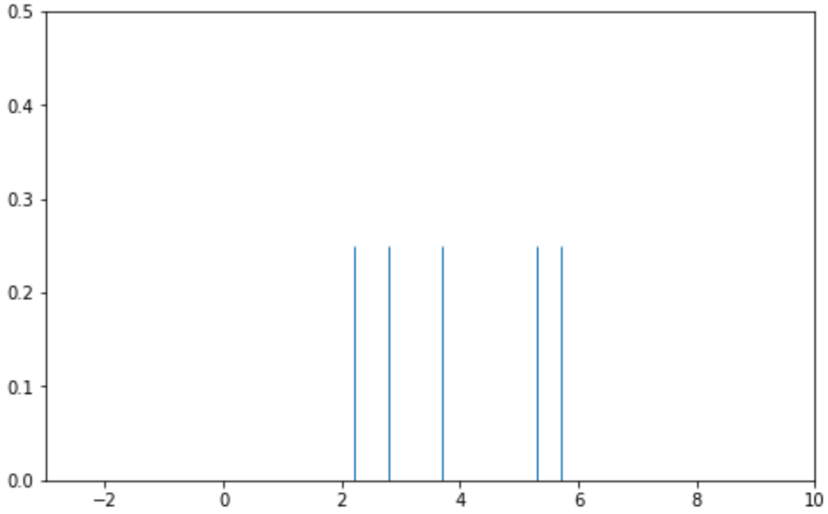
Sum these five normalized curves together.



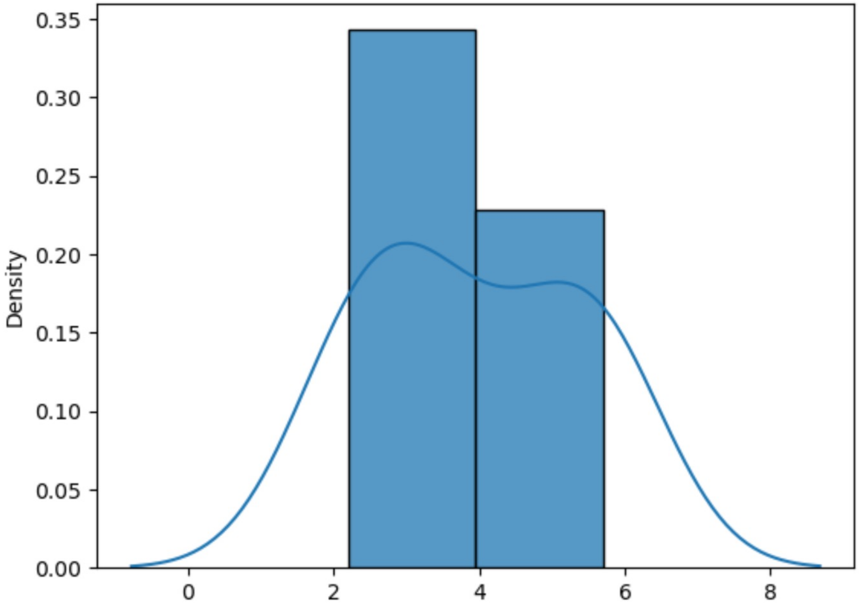
The final KDE curve.

Result

- A summary of the distribution using KDE.



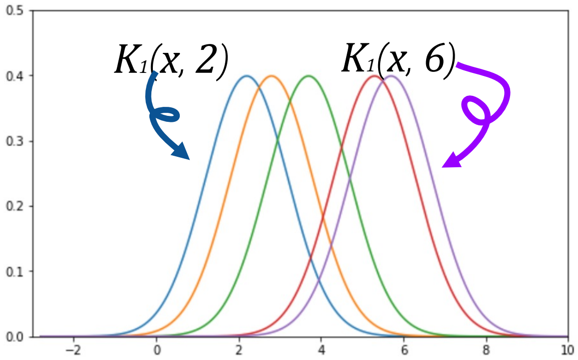
Each line represents a datapoint in the dataset (e.g. one country's HIV rate).



The density at each point corresponds to the KDE calculated based on kernels placed on all data points

Summary of KDE

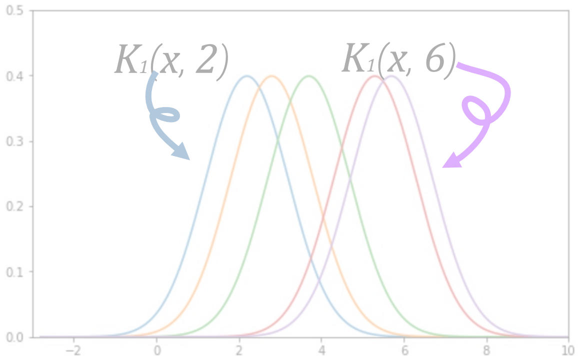
$$f_\alpha(x) = \overset{\boxed{2}}{\frac{1}{n}} \overset{\boxed{3}}{\sum_{i=1}^{n}} K_\alpha \overset{\boxed{1}}{(x, x_i)}$$



A general "KDE formula" function is given above.

- $\boxed{1}$ $K_\alpha(x, x_i)$ is the **kernel** function centered on the observation i.
    - Each kernel individually has area 1.
    - K represents our kernel function of choice. We'll talk about the math of these functions soon.

$$f_\alpha(x) = \overset{②}{\frac{1}{n}} \overset{③}{\sum_{i=1}^{n}} K_\alpha\overset{①}{(x, x_i)}$$



A general “KDE formula” function is given above.

- ① $K_\alpha(x, x_i)$ is the **kernel** centered on the observation i.
  - Each kernel individually has area 1.
  - x represents any number on the number line. It is the input to our function.
- ② n is the number of observed data points that we have.
  - We multiply by $1/n$ to normalize the kernels so that the total area of the KDE is still 1.
- ③ Each x_i $(x_1, x_2, \ldots, x_n)$ represents an observed data point. We sum the kernels for each datapoint to create the final KDE curve.

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

Kernels

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

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

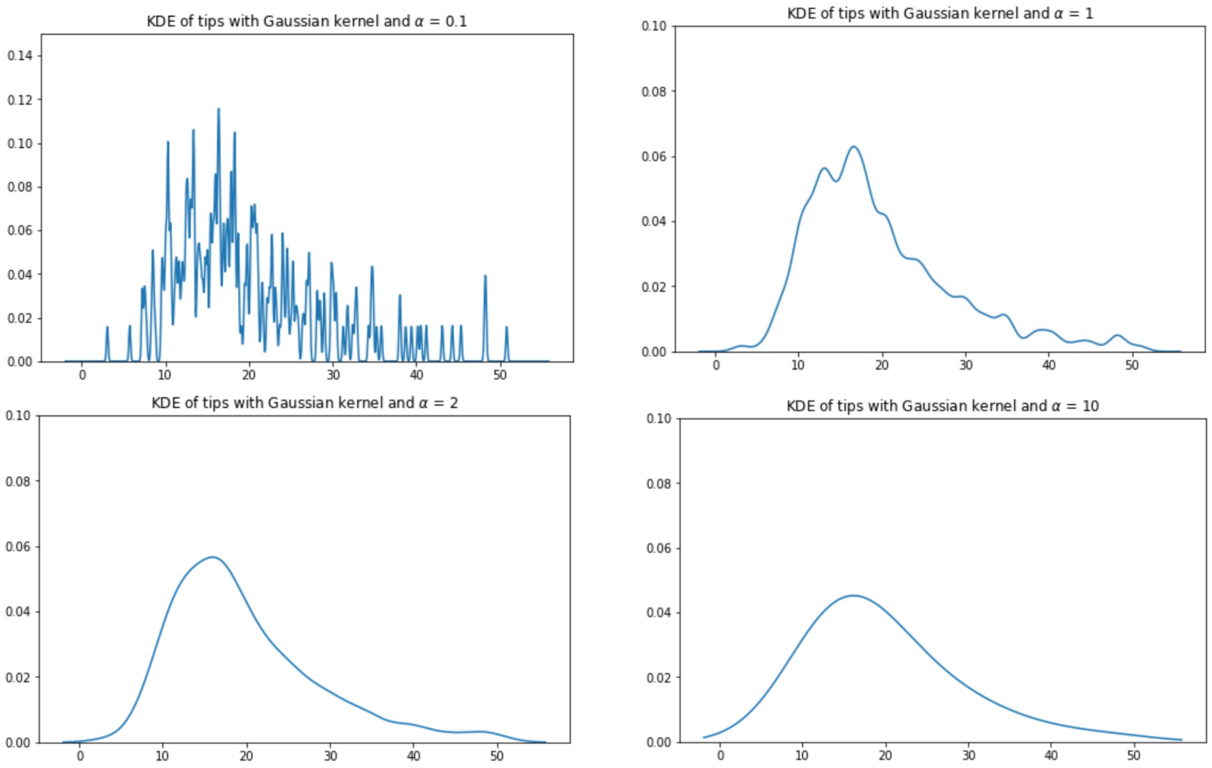
The most common kernel is the **Gaussian kernel**.

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

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

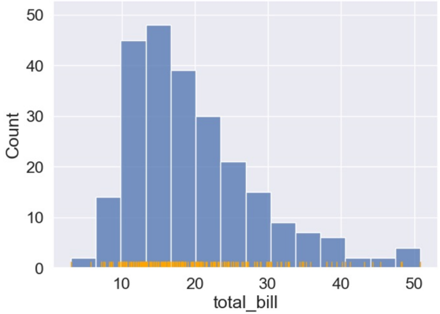
Memorizing this formula is less important than knowing the shape and how the bandwidth parameter **α** smoothes the KDE.

Effect of Bandwidth on KDEs



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

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



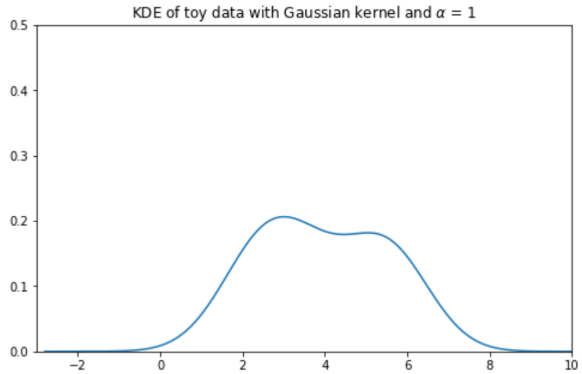
Other Kernels: Boxcar

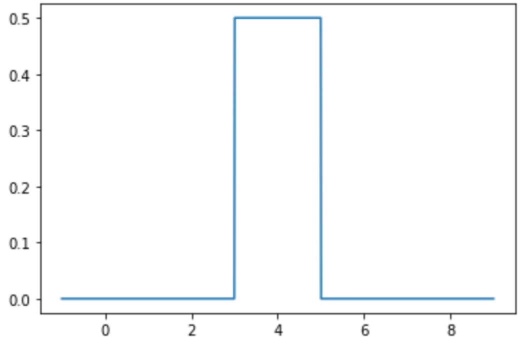
As an example of another kernel, consider the **boxcar kernel**.

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

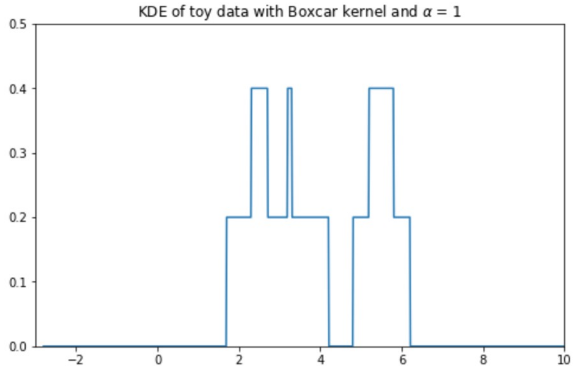
$$K_\alpha(x, x_i) = \begin{cases} \frac{1}{\alpha}, & |x - x_i| \le \frac{\alpha}{2} \\ 0, & \text{else} \end{cases}$$

- Not of any practical use in Data 100! Presented as a simple theoretical alternative.





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



LECTURE 8

Visualization

Content credit: [Acknowledgments]