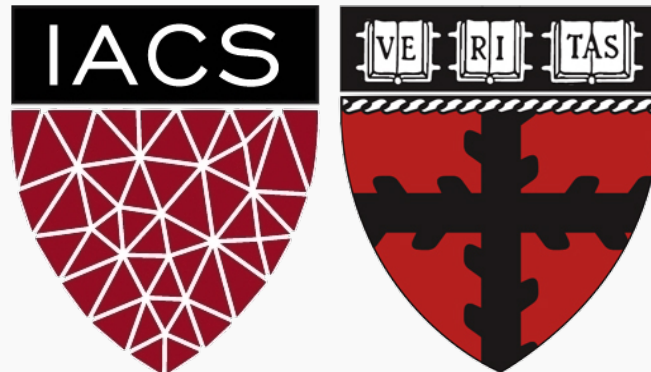


# From Probability to Maximum Likelihood Estimation (MLE)

CS109A Introduction to Data Science  
Pavlos Protopapas, Natesh Pillai



# Outline

---

- What is a random variable?
- Point estimates of random variables. Confidence Intervals, Histogram and PDF/PMF
- Known random variables: Uniform, Binomial. Normal
- Modeling Data with Probability Distributions
- Likelihood Theory
- Modeling Linear Regression Probabilistically

# Outline

---

- **What is a random variable?**
- Point estimates of random variables. Confidence Intervals, Histogram and PDF/PMF
- Known random variables: Uniform, Binomial. Normal
- Modeling Data with Probability Distributions
- Likelihood Theory
- Modeling Linear Regression Probabilistically



# CS 109A Olympics





**WHO WILL WIN THE 100M DASH?**

# CS109A 100m dash

**OPTION A**



**THE PROFESSOR**

**AVERAGE PACE: 13 SECONDS**

**CONSISTENCY: HIGH**

# CS109A 100m dash

**OPTION B**



**THE HOT SHOT**

**AVERAGE PACE: 13.0 SECONDS**

**CONSISTENCY: VERY LOW**

**OPTION A**





**THE BANGALORE CHAMPION**

**AVERAGE PACE: 13.1 SECONDS**

**CONSISTENCY: MEDIUM**

**OPTION A**



**OPTION B**



**OPTION C**



**THE RESEARCHER**

**AVERAGE PACE: 14 SECONDS**

**CONSISTENCY: LOW**

**OPTION A**



**OPTION B**



**OPTION C**



**OPTION D**



OPTION A



OPTION B



OPTION C



OPTION D



RACE #1 13.1



RACE #2 13.2

# RACE TIME

13.51

RACE #3 13.64

14.04

14.01

13.63

RACE #4 12.87

13.52

13.12

13.91

RACE #5 13.22

13.24

12.78


12.32





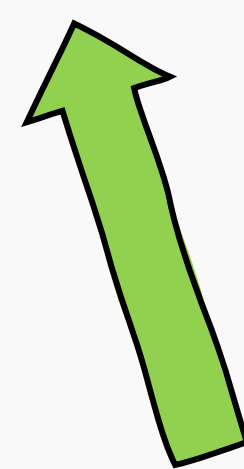
# CS109A 100m dash

Let  $X$  be the race pace for a given 100m dash, then  $X$  is called a **random** variable

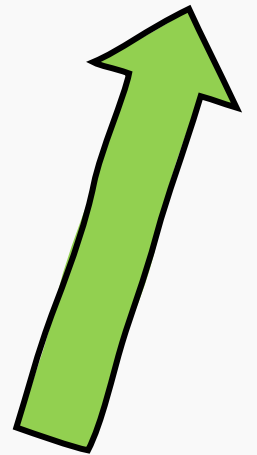


My race pace will always vary a little, despite my efforts!

$$\textit{Race Pace} = \textit{Average Pace} + \epsilon$$



Constant



Varying

# RECAP: Python variables

We have seen variables as something we assign a value to.

- Integer <int>
- Float <float>
- List <list>
- Dictionary <dict>

```
In [2]: a = 2
In [3]: b = 2.5
In [4]: pavloslist = [1,2,3,4,5]
In [5]: pavlosdict = {'John':2, 'Pavlos':2, 'Eric':7}

In [6]: type(a)
Out[6]: int
In [7]: type(b)
Out[7]: float
In [8]: type(pavloslist)
Out[8]: list
In [9]: type(pavlosdict)
Out[9]: dict
```

# Random Variable

- A random variable can be thought of as an **outcome** of a random experiment.
- Unlike the python variables defined before, the **value** of a random variable is not fixed.
- The output of a random variable could be either **discrete** (integers) or **continuous** (floats).

```
In [26]: x = RandomVariable()
```

```
In [27]: x.random
```

```
Out[27]: 0.5632899481539281
```

```
In [28]: x.random
```

```
Out[28]: 0.630954141651853
```







$$X = \textit{Average Pace} + \epsilon$$

- What are the possible values of  $X$ ?
- What is the maximum value of  $X$ ?
- What is the minimum value of  $X$ ?
- What is the expected value of  $X$ ?
- Are the values of  $X$  spread out, or consistent?

AND MANY MORE QUESTIONS ...

```
In [5]: pavlos = Sprinter()
```

```
In [6]: pavlos.time
```

```
Out[6]: 13.431656720548697
```

```
In [7]: pavlos.time
```

```
Out[7]: 13.42798180661262
```

```
In [8]: pavlos.time
```

```
Out[8]: 11.78189462795882
```

```
In [9]: pavlos.time
```

```
Out[9]: 14.77745984741147
```

# Simulations



RUNNER: PAULOS PROTOPAPAS

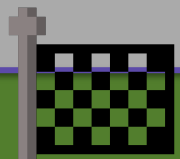
COUNTRY: CYPRUS

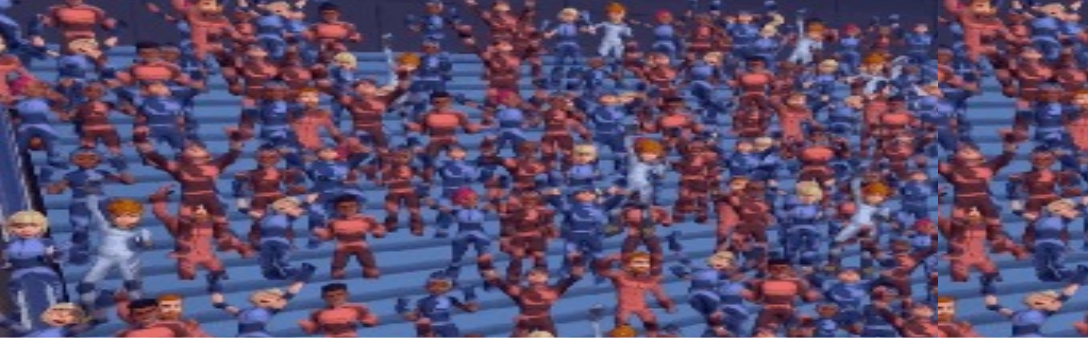
CURRENT TIME: 12.35 S



START

FINISH





RUNNER: PAULOS PROTOPAPAS

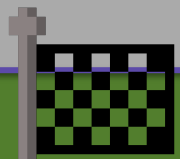
COUNTRY: CYPRUS

CURRENT TIME: 12.47 S



START

FINISH





RUNNER: PAULOS PROTOPAPAS

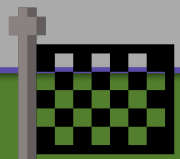
COUNTRY: CYPRUS

CURRENT TIME: 12.75 S



START

FINISH





# Random Variable

$$X = \textit{Average Pace} + \epsilon$$

[13.75, 15.21, 13.65, 13.58, 12.93, 14.23, 12.81, 11.50, 13.09, 12.26, ...]

- We could run the experiment **multiple** times and record the results.
- This will give us a list of **possible values** of the random variable  $X$ .
- An **exhaustive** list of all possible values is often called the **population** space of the random experiment.

Let's do it. I will run many runs for you





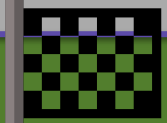
RUNNER: PAUL  
COUNTRY: CYPRUS  
CURRENT TIME: 00:00

Come on guys! How many more races do you need me to run? I can't do this all day!



START

FINISH



# Random Variable

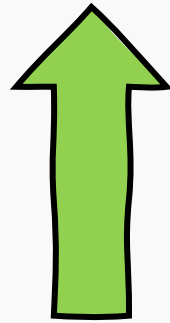
---

[13.75, 15.21, 13.65, 13.58, 12.93, 14.23, 12.81, 11.50, 13.09, 12.26, ...]

# Random Variable

[13.75, 15.21, 13.65, 13.58, 12.93, 14.23, 12.81, 11.50, 13.09, 12.26, ...]

[13.75, 13.65, 12.93, 12.81, 12.26]



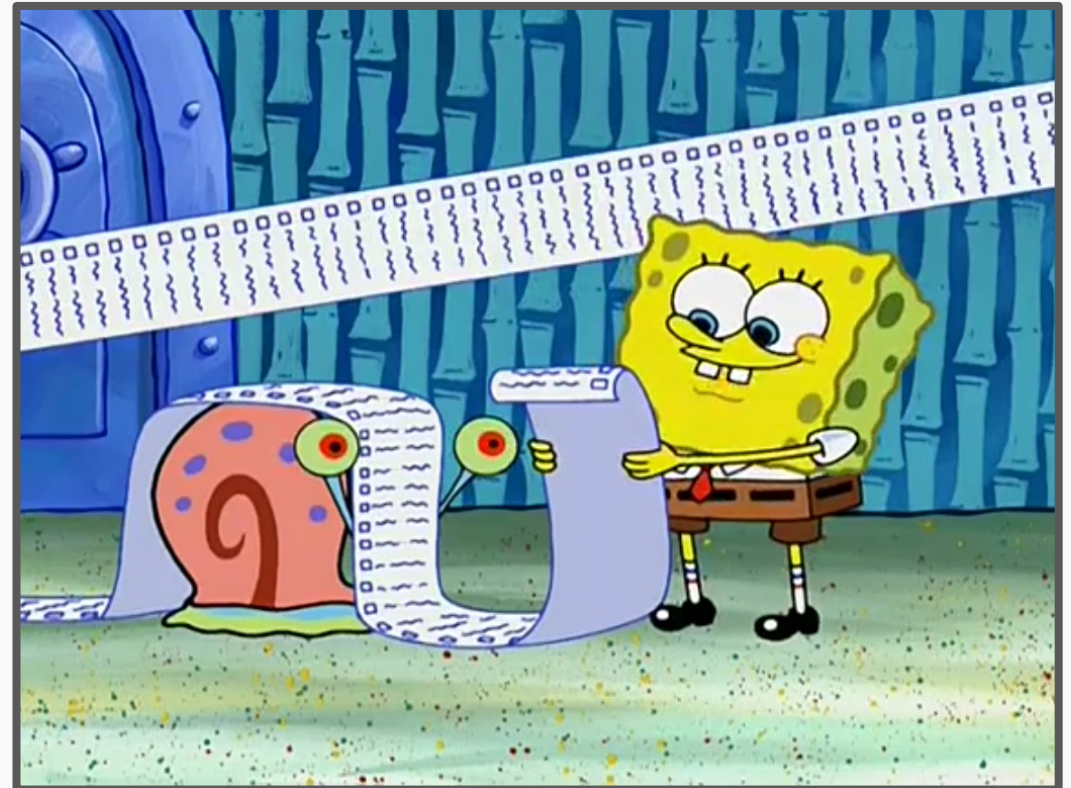
Sample

# Properties of a Random Variable

## ISSUES?

The outcomes of a random variable captured over multiple simulations is difficult to interpret and consequently difficult to compare to other random variables.

- **ISSUE #1:** We do not have estimates to compare with other random variables.
- **ISSUE #2:** It is difficult to visualize the spread of the outcome.





# Properties of a Random Variable

## ISSUES?

The outcomes of a random variable captured over multiple simulations is difficult to interpret and consequently difficult to compare to other random variables.

- **ISSUE #1:** We do not have estimates to compare with other random variables.
- **ISSUE #2:** It is difficult to visualize the spread of the outcome.

## Description of Random Variables

Low

Point Estimates

Medium

Confidence Intervals

Histogram

High

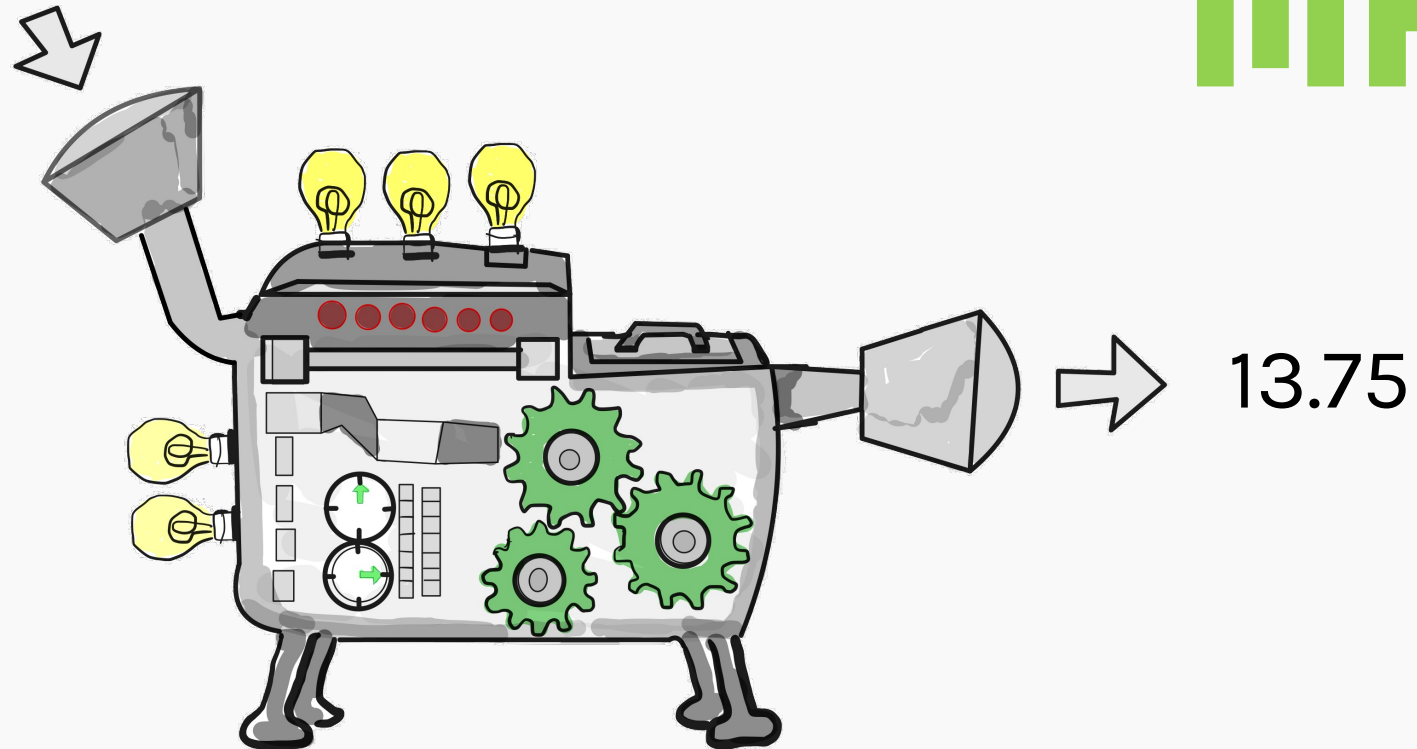
Probability Density Function  
or Probability Mass Functions

# Point estimates

Sample

[13.75, 13.65, 12.93, 12.81, 12.26]

MAX

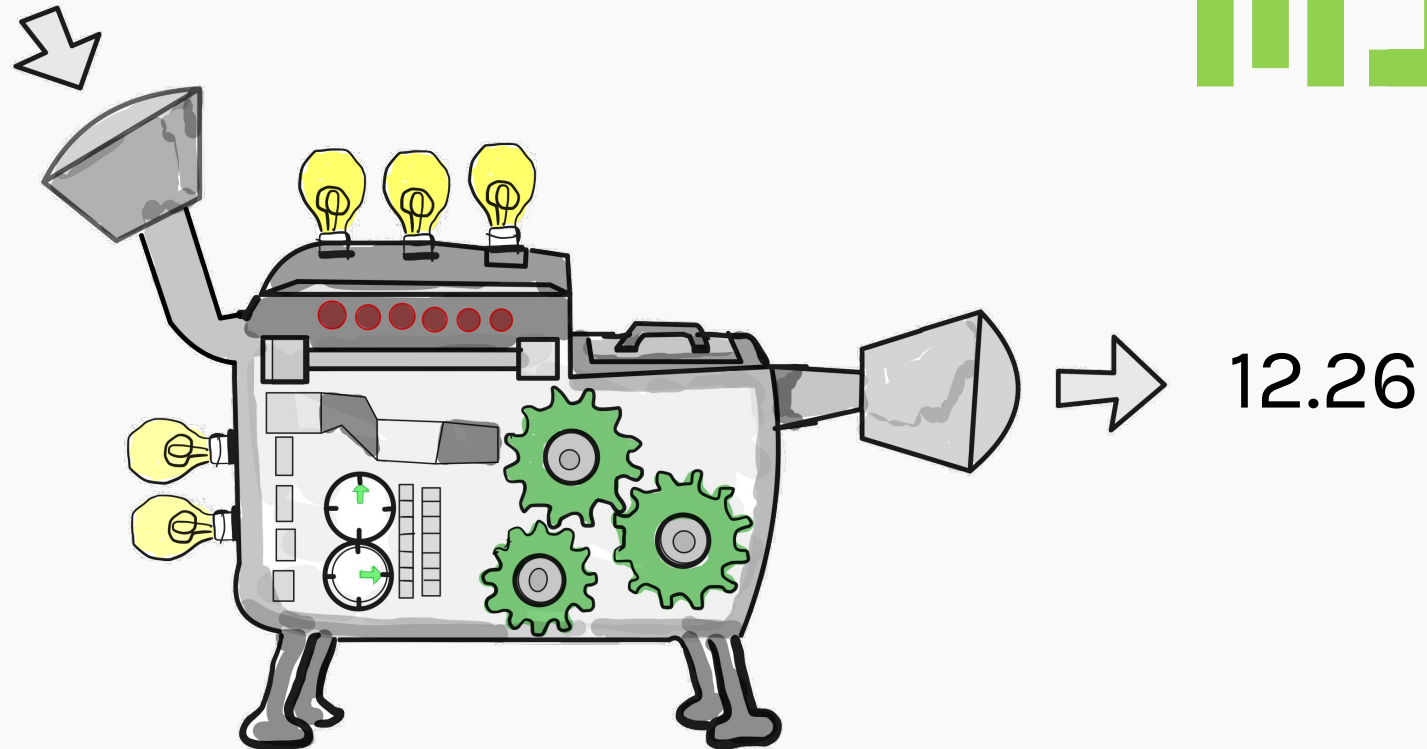


# Point estimates

Sample

[13.75, 13.65, 12.93, 12.81, 12.26]

MIN

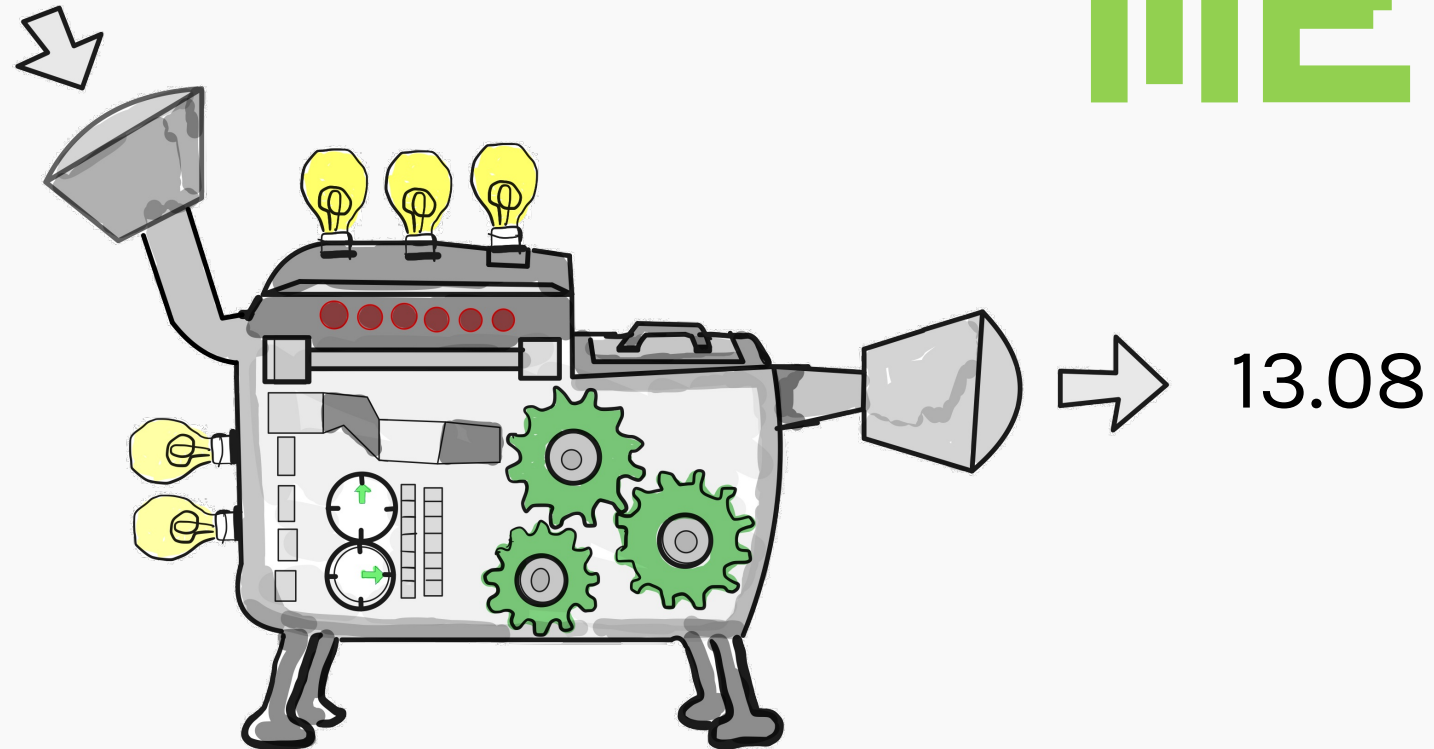


# Point estimates

Sample

[13.75, 13.65, 12.93, 12.81, 12.26]

MEAN



# What we want

## POINT ESTIMATES

- Point estimates can be defined as numbers that give some **information** of the random variable.
- Commonly used point estimates include **max**, **min**, **mean**, **mode**, **variance**, interquartile range, etc.
- Two major categories describe the **central tendency** & the **spread**

Population Parameters		Sample Statistics
Mean	$\mu$	$\bar{X}$
Standard Deviation	$\sigma$	$s^2$

Remember! Population estimates are in Greek and sample estimates are written in roman style!

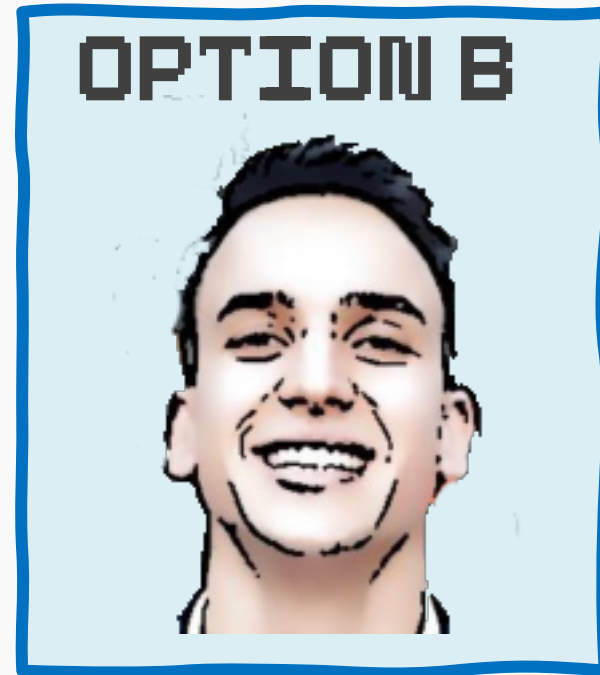




# Random Variables - Point estimates



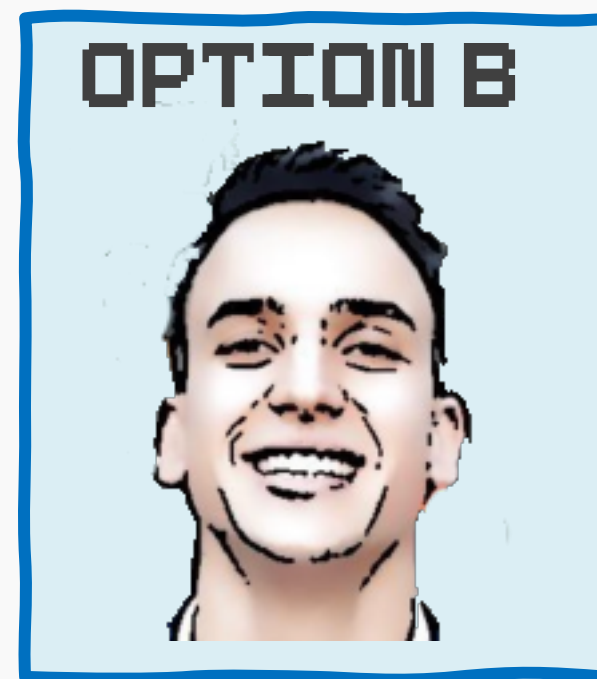
V/S



# Random Variables Comparison - Point estimates



<b>13.31</b>	<b>MAX</b>	<b>14.78</b>
<b>12.67</b>	<b>MIN</b>	<b>11.31</b>
<b>13.00</b>	<b>MEAN</b>	<b>13.00</b>
<b>13.01</b>	<b>MEDIAN</b>	<b>13.00</b>
<b>12.67</b>	<b>MODE</b>	<b>12.25</b>



# Measure of Central Tendency

# Central Tendency

- Mean is the same as the ‘average’ that we are used to. If we know all the outcomes of the population:

$$\text{Population mean, } \mu = \frac{\sum_i x_i}{n}$$

- Sample statistics are calculated in a manner which best approximates the population parameters.
- Sample mean is calculated like population mean:

$$\text{Sample mean, } \bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

# Measure of Spread



# Spread

- Standard deviation is a measure of how spread out the data is from the mean. Assuming all of population is known:

$$\text{Population std}(\sigma) : \sqrt{\frac{\sum (x_i - \mu)^2}{n}}$$

- Sample std has a slight correction to population std:

**Sample std is used as an estimate for the population std, using n-1 gives more accurate results.**

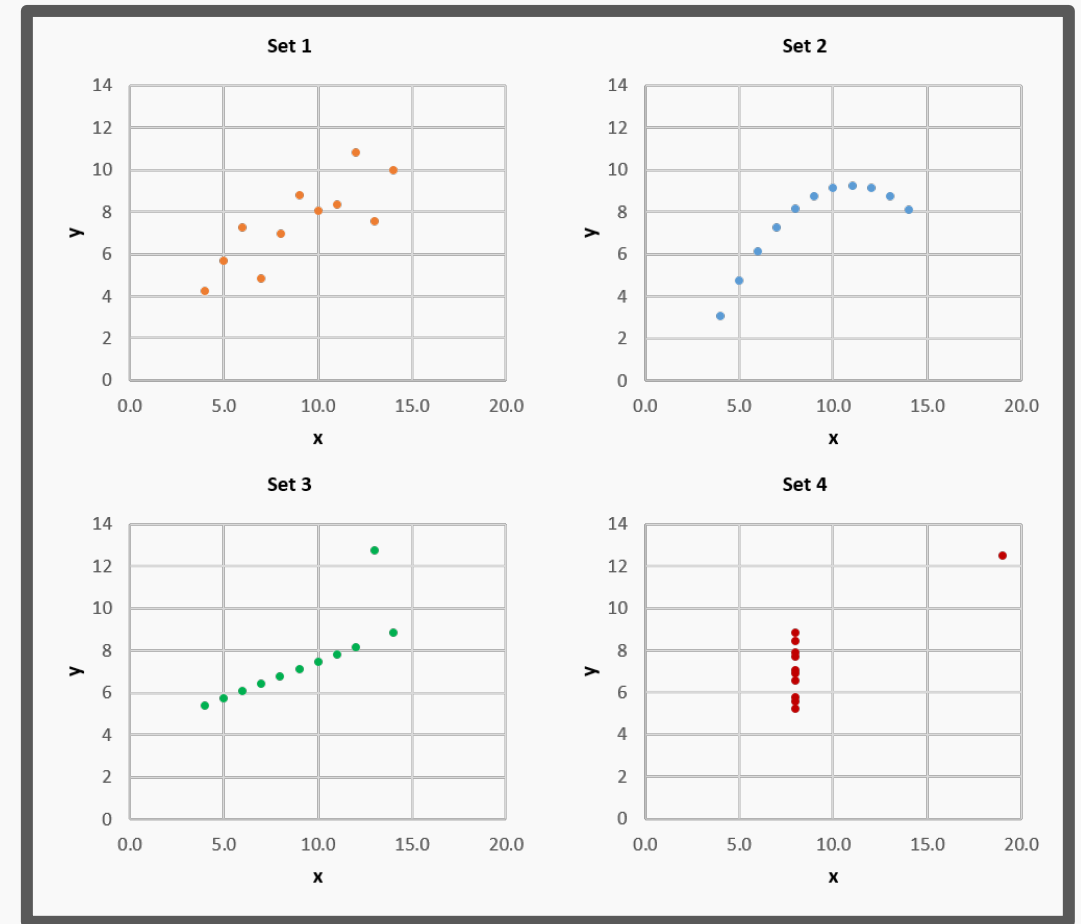
$$\text{Sample std}(s) : \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

# Confidence Interval

# Confidence Intervals

- Point estimates can often be misleading and lead to **imprecise** understanding of the random variable.

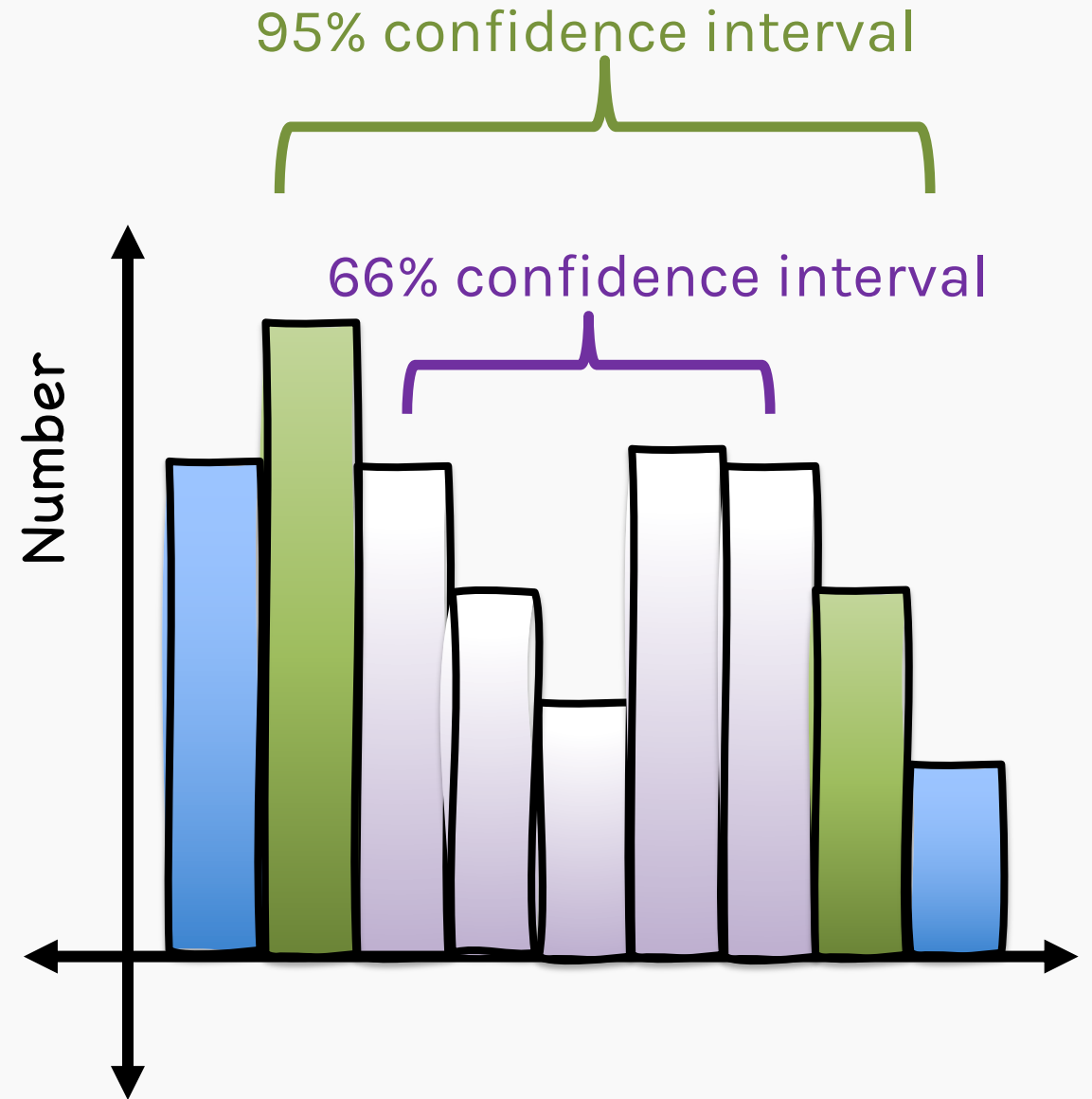
## Anscombe's Quartet



# Confidence Intervals

- Point estimates can often be misleading and lead to **imprecise** understanding of the random variable.
- Unlike point estimates, a confidence interval is a **range** that represents the likely output of a random experiment.
- We often set the **confidence level** before examining the data and it is expressed as %, e.g.


 95% confidence



# Confidence Intervals


---

[13.75, 15.21, 13.65, 13.58, 12.93, 14.23, 12.81, 11.50, 13.09, 12.26, ...]

 Step #1: Sort the original data from lowest to highest

# Confidence Intervals

[13.75, 15.21, 13.65, 13.58, 12.93, 14.23, 12.81, 11.50, 13.09, 12.26, ...]


 Step #1: Sort the original data from lowest to highest

[11.50, 12.26, 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21, ...]

 Step #2: Find the lower confidence range using np.percentile

# Confidence Intervals

[13.75, 15.21, 13.65, 13.58, 12.93, 14.23, 12.81, 11.50, 13.09, 12.26, ...]

 Step #1: Sort the original data from lowest to highest

[11.50, 12.26, 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21, ...]

 Step #2: Find the lower confidence range using np.percentile

`np.percentile`( [11.50, 12.26, 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21, ... ], 2.5 ) = 12.80

[11.50, 12.26, 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21, ...]



2.5% of data are on the left of this value



# Confidence Intervals

[ 11.50, 12.26 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21,, ... ]



Step #3: Find the upper confidence range again using np.percentile

`np.percentile`( [11.50, 12.26 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21,, ... ], 97.5) = 13.71

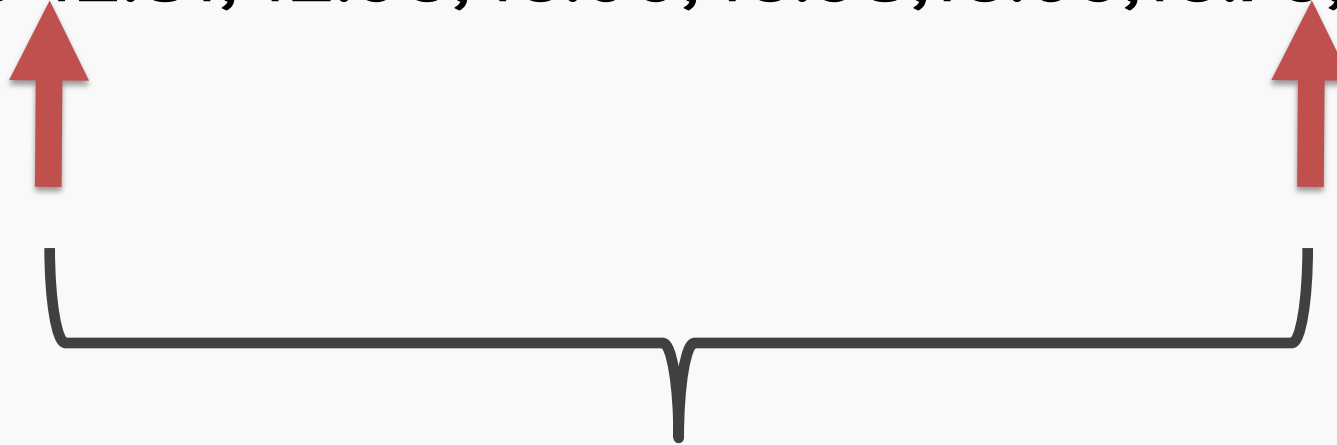
[ 11.50, 12.26 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21,, ... ]



2.5% of data are on the right of this value

# Confidence Intervals

[ 11.50, 12.26 12.81, 12.93, 13.09, 13.58, 13.65, 13.75, 14.23, 15.21,, ... ]



95% confidence intervals

# Random Variables - Point estimates



**13.31**

**MAX**

**14.78**

**12.67**

**MIN**

**11.31**

**13.00**

**MEAN**

**13.00**

**13.01**

**MEDIAN**

**13.00**

**12.67**

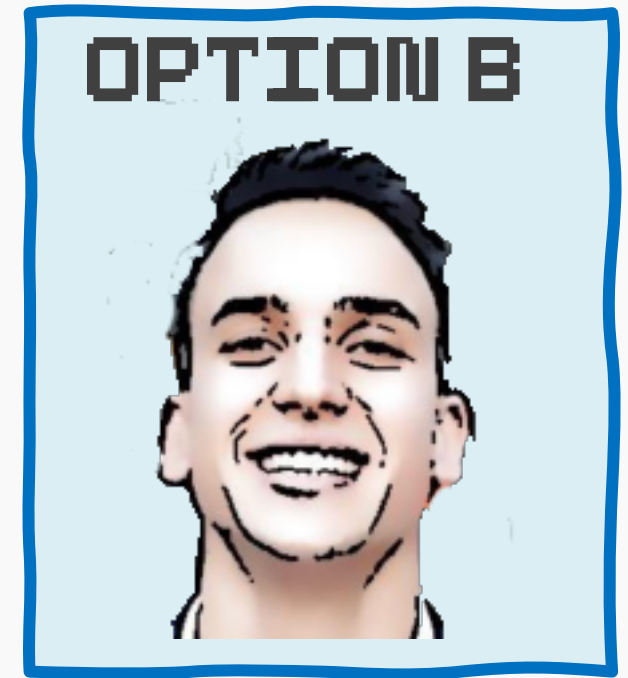
**MODE**

**12.25**

**12.80, 13.20**

**CI**

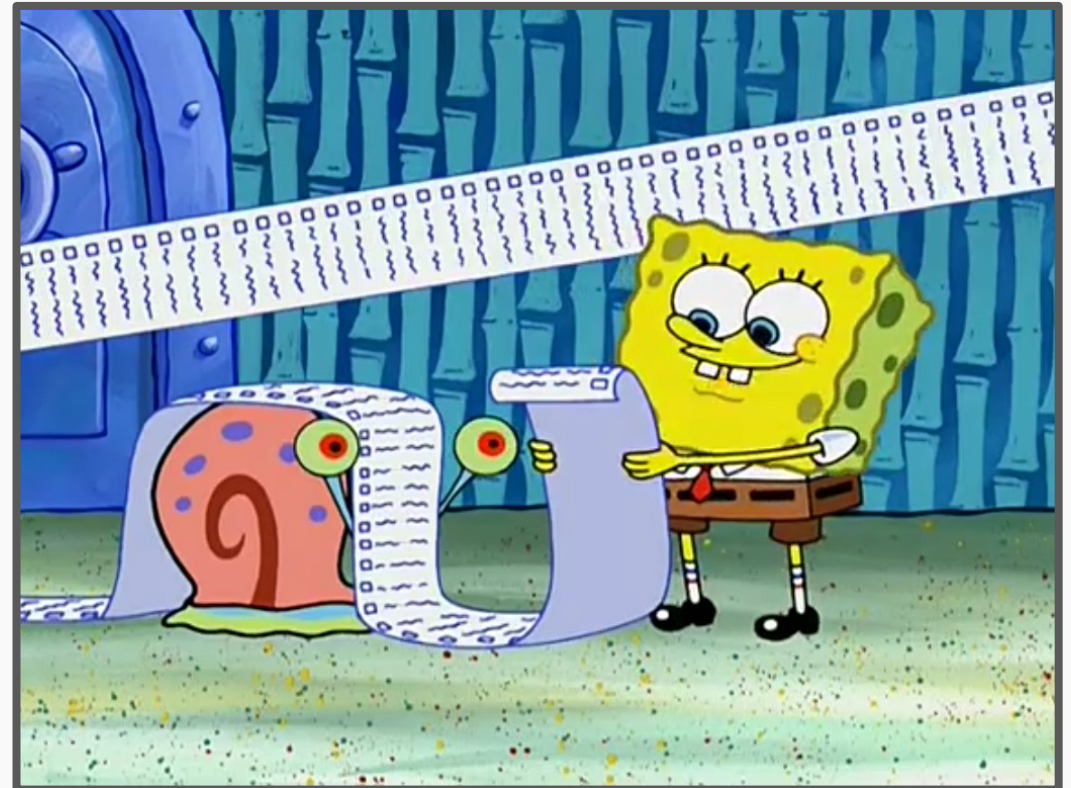
**12.80, 13.20**



# Properties of a Random Variable

## ESTIMATE ISSUES

- Point or interval estimates of random variables do not guarantee a **unique** description of the output.
- Due to its **approximate** nature, it may lead to confounding of different processes.
- A popular example of this is the **Anscombe's Quartet**; a set of four datasets with same estimates but different distributions.



# Histogram

# Histogram

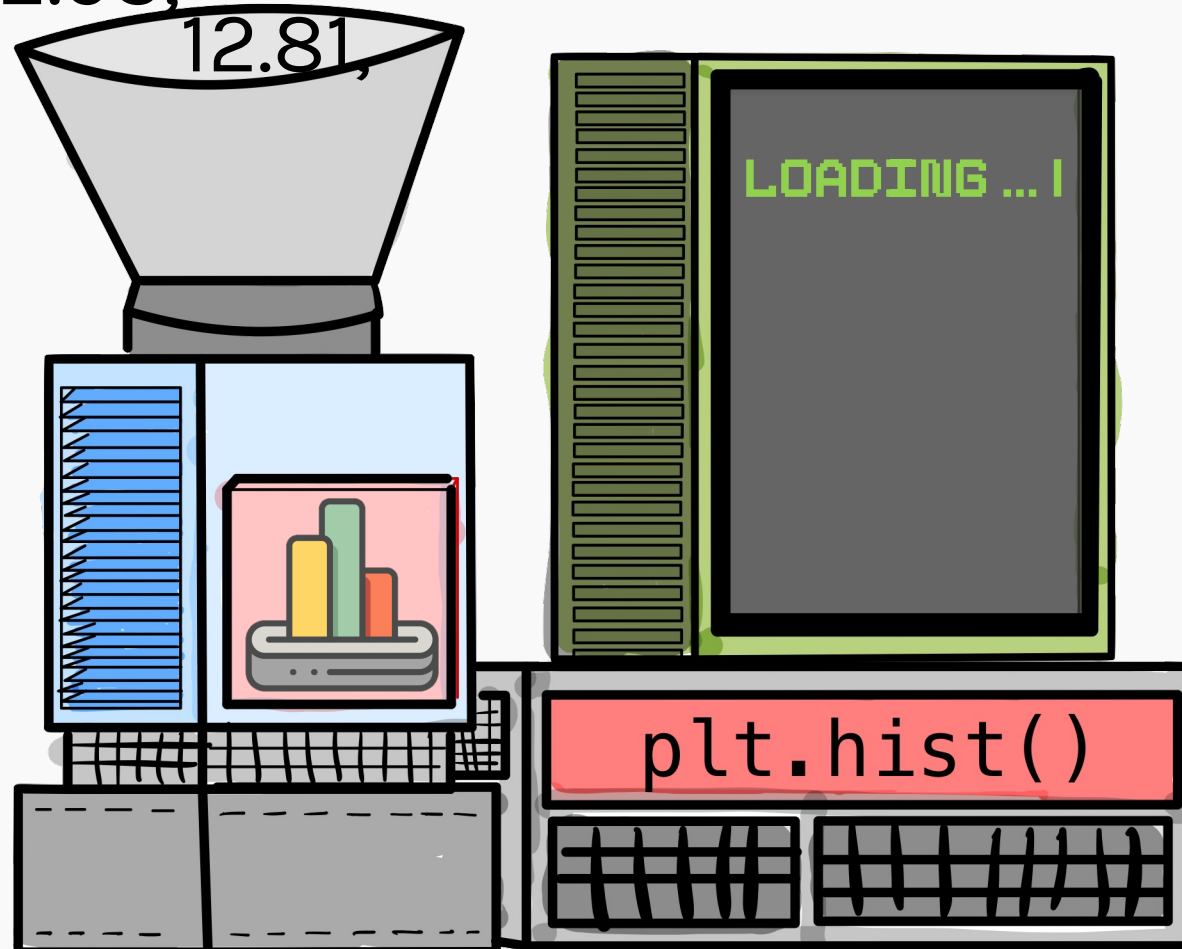
---

Sample

[ 13.75, 13.65, 12.93, 12.81, 12.26 ]

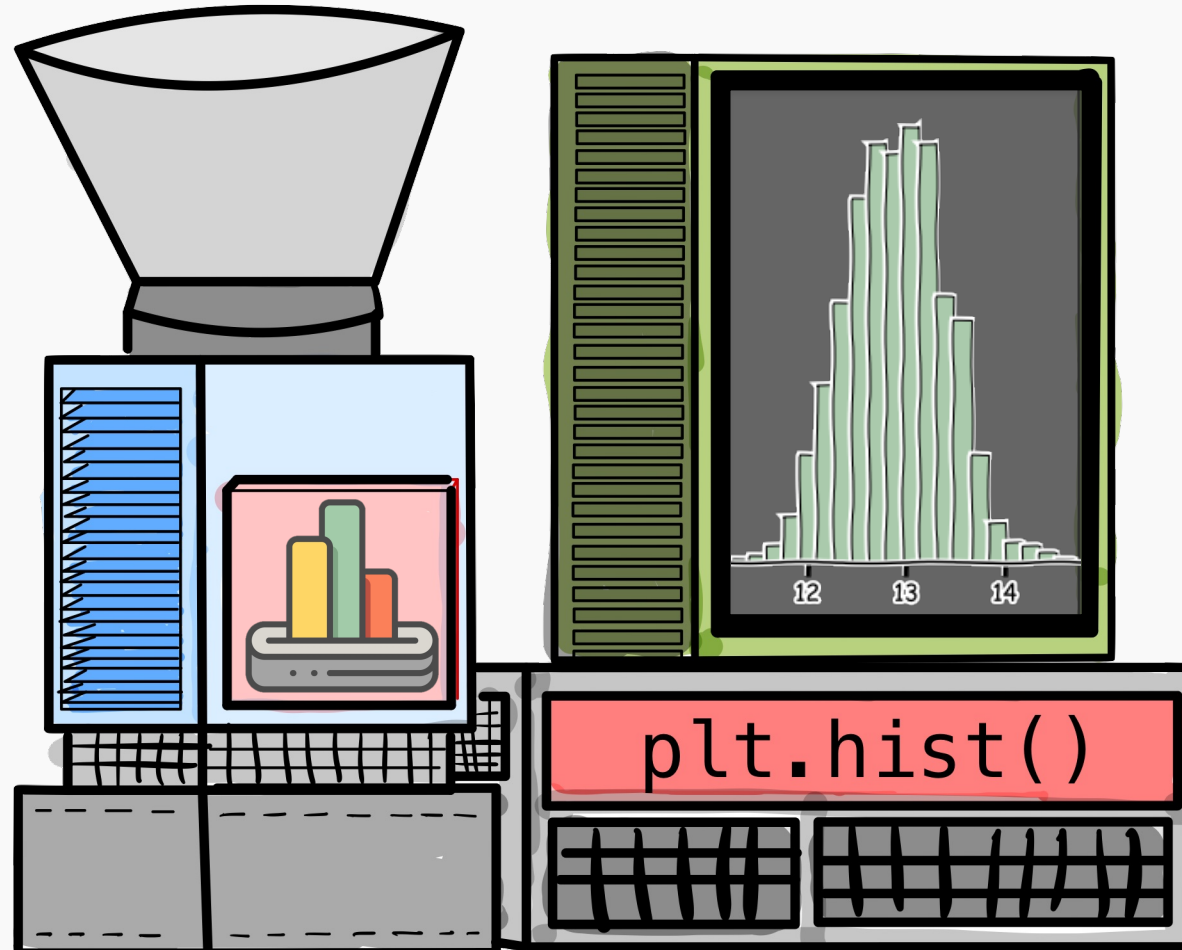
# Histogram

13.75, 13.75,  
13.75, 13.65, 12.81,  
12.93, 12.26





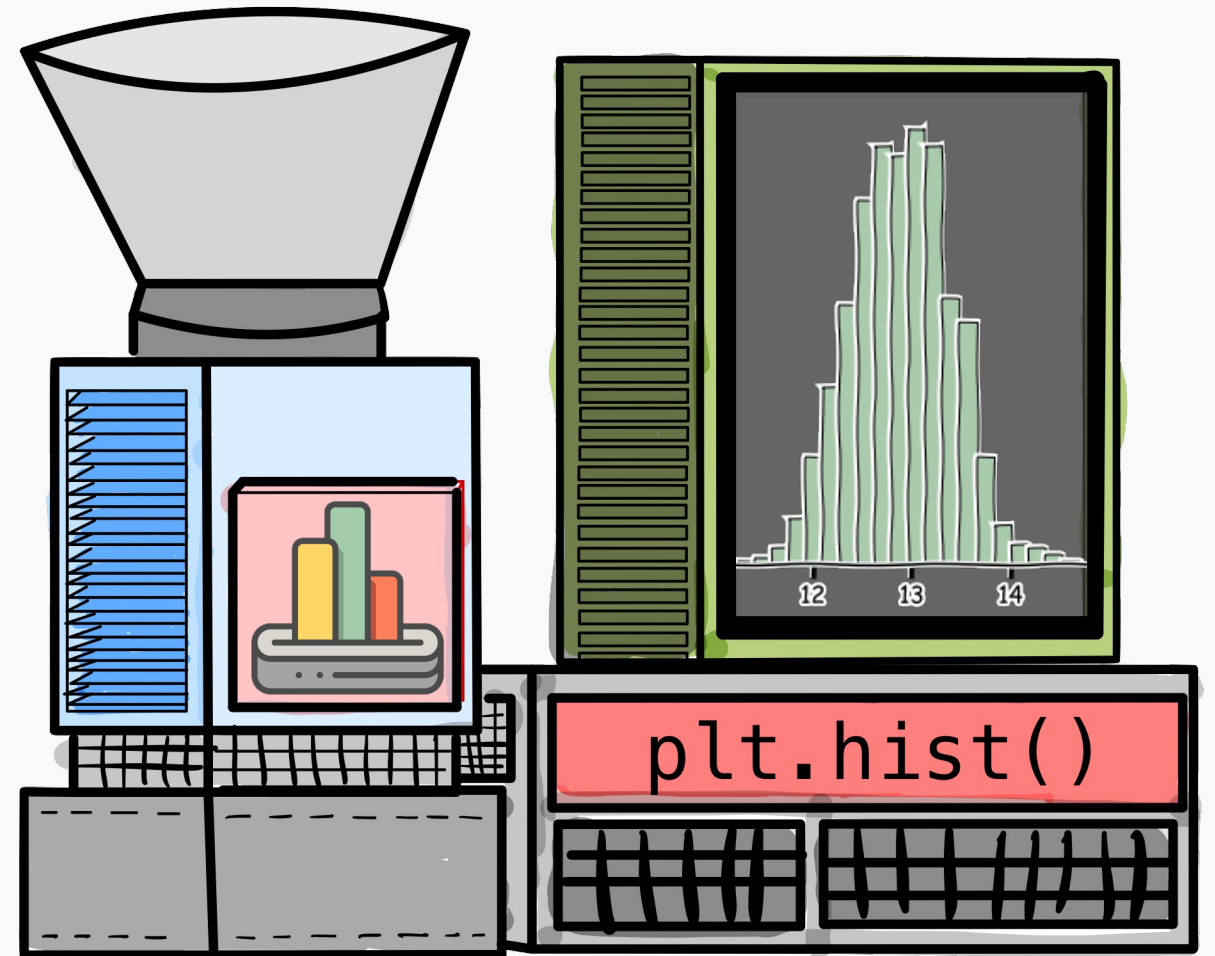
# Histogram



# Histogram

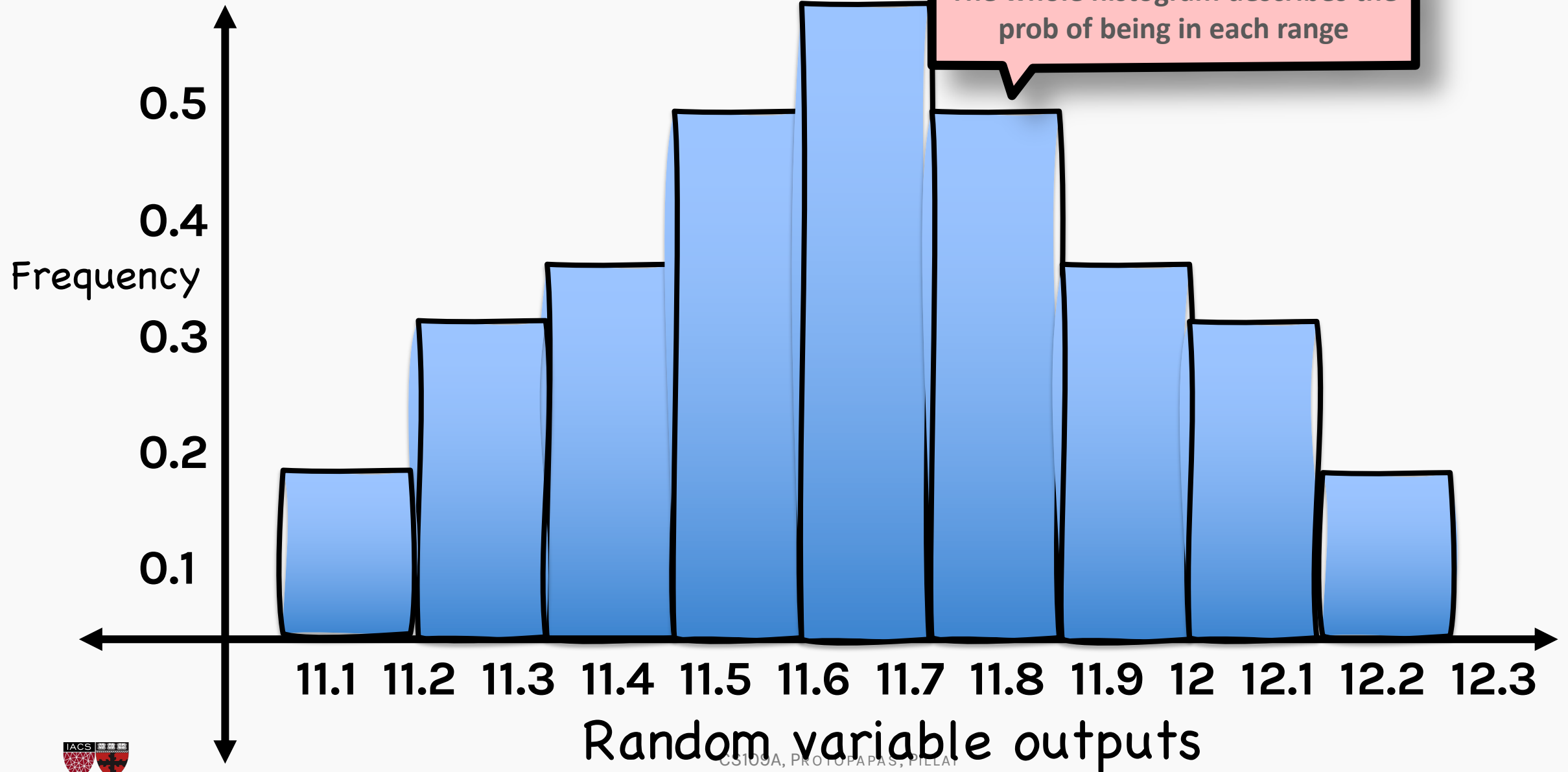
## DISTRIBUTIONS

- Histogram (from the Greek word *histos* meaning pole & *gram* meaning chart) is a **visual representation** of the sample.
- It is defined by the **relative frequency** on the y-axis and the exhaustive **outcomes** of the random variable on the x-axis.
- It can be decorated with point estimates for better description.

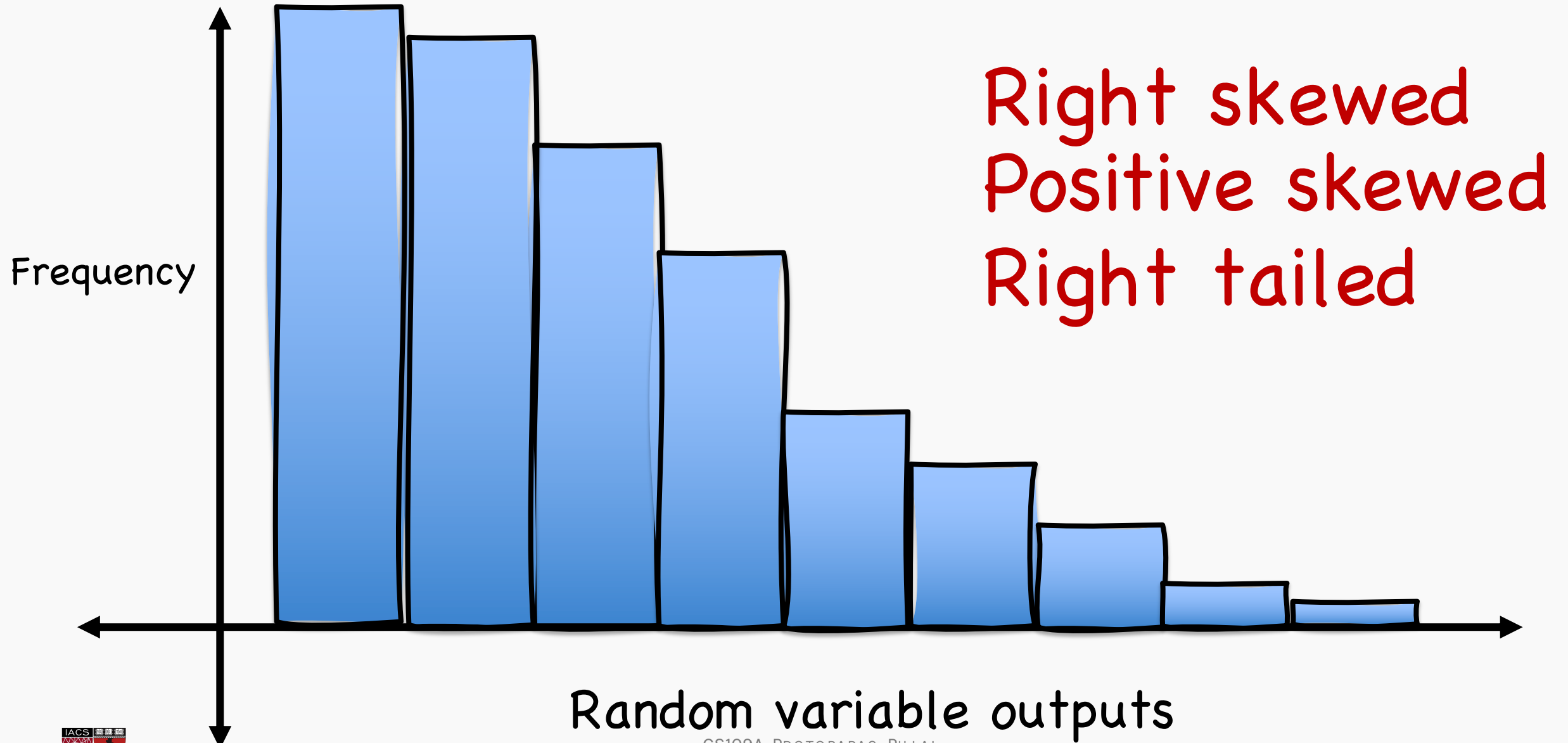


# Anatomy of a histogram

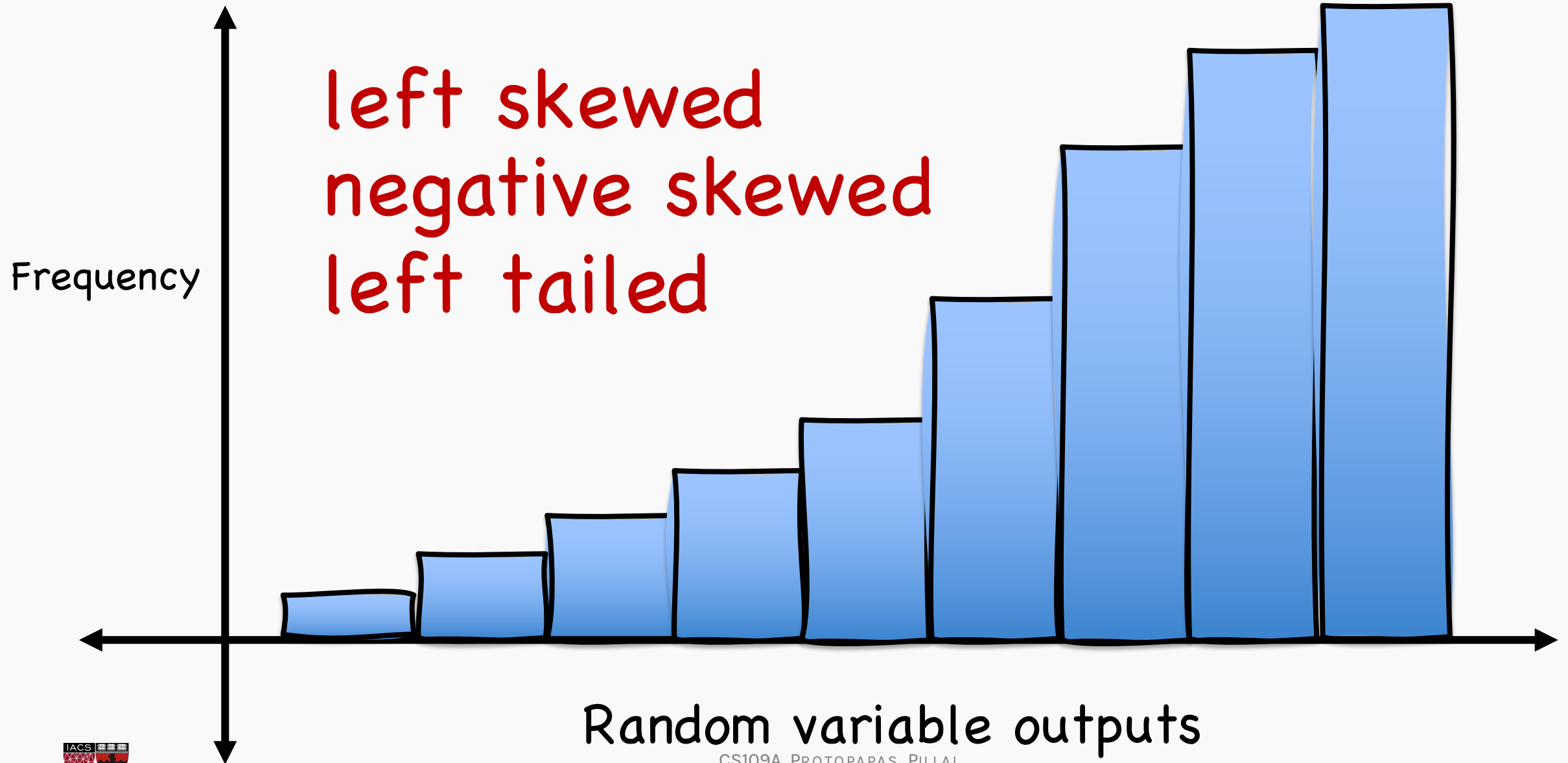
This bin describes the probability of the variable to be in this range. The whole histogram describes the prob of being in each range



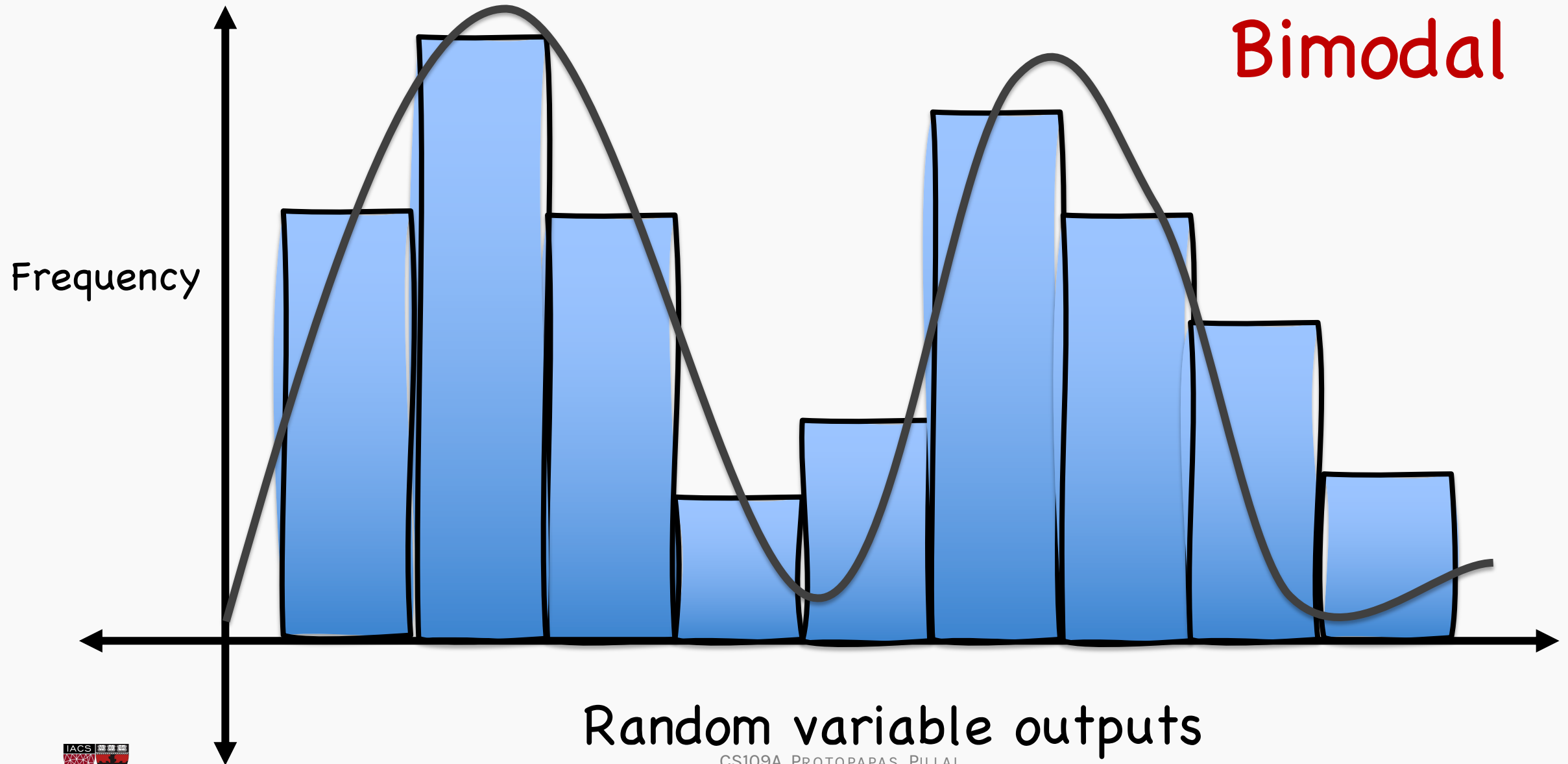
# Anatomy of a histogram



# Anatomy of a histogram



# Anatomy of a histogram

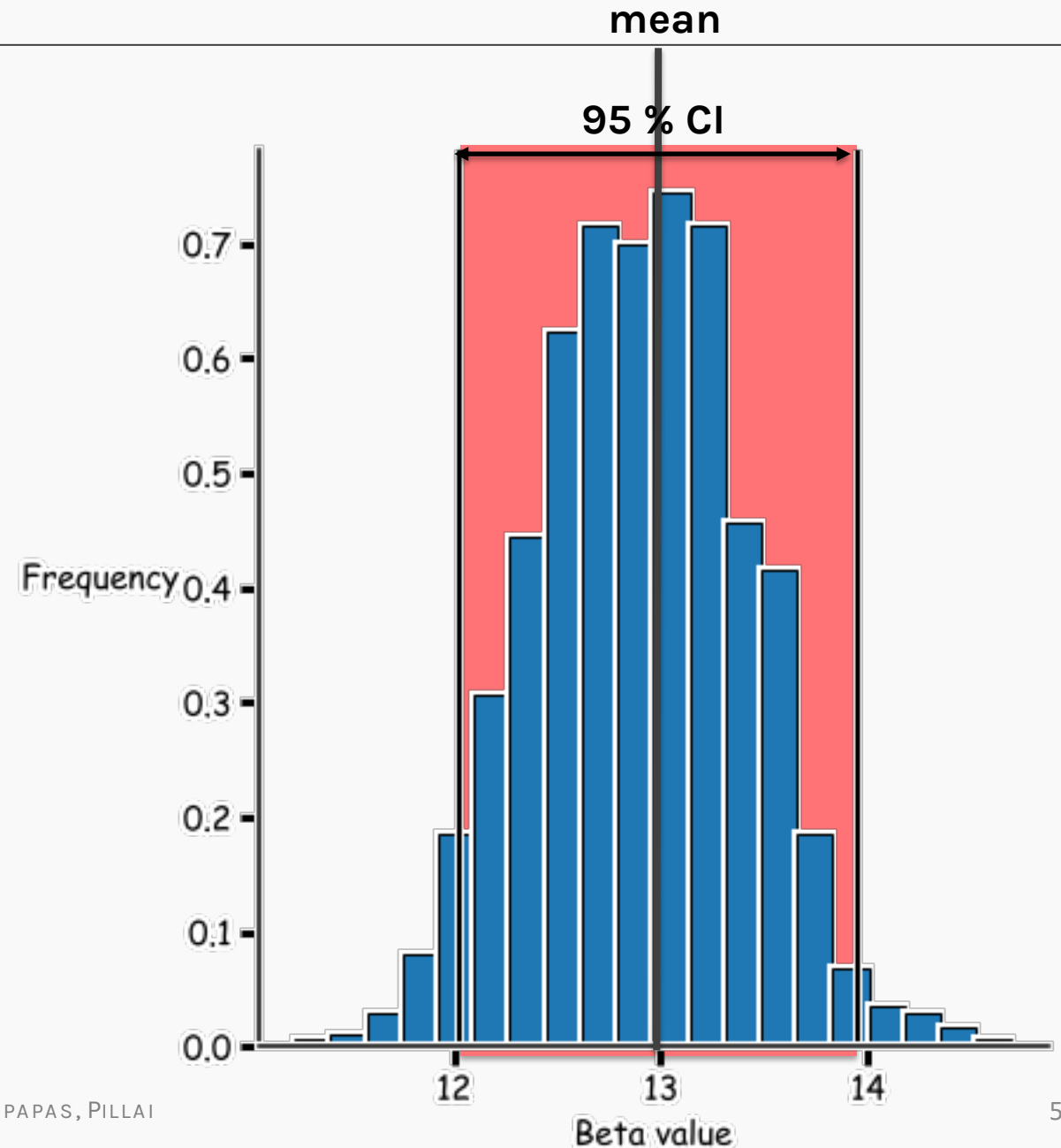


# Histogram

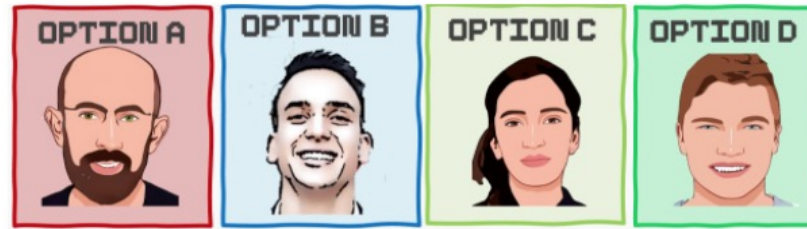
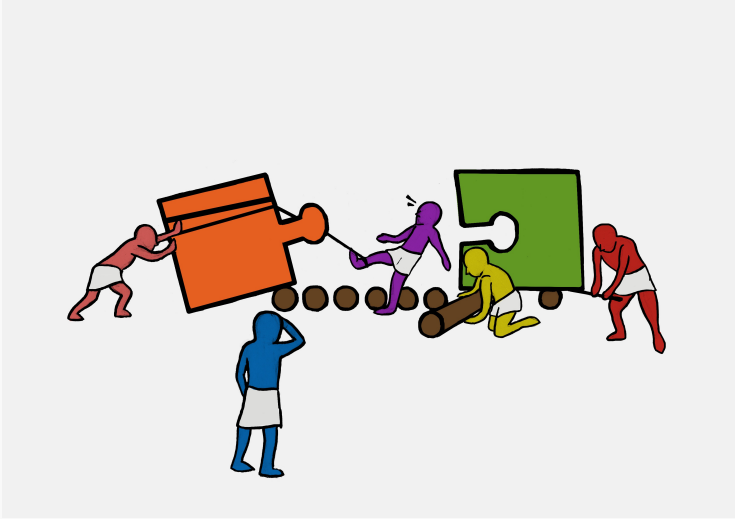
## ISSUES?

The outcomes of a random variable captured over multiple simulations is difficult to interpret and consequently difficult to compare to other random variables.

- ~~ISSUE #1: We do not have estimates to compare with other random variables.~~
- ~~ISSUE #2: It is difficult to visualize the spread of the outcome.~~

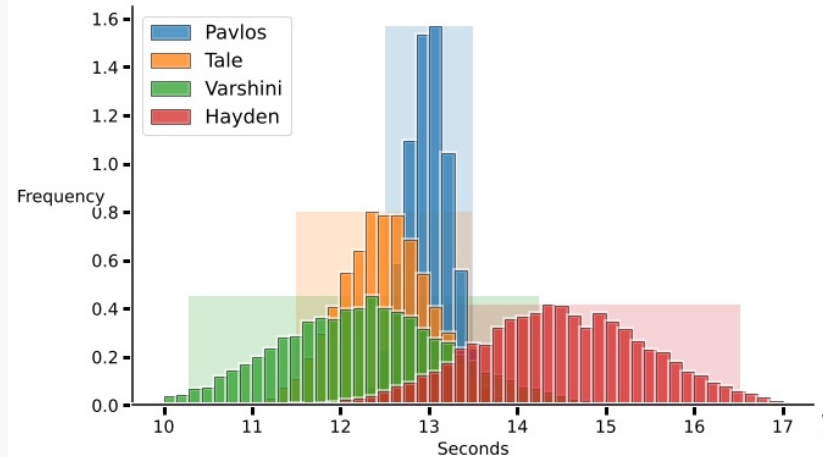






Instructions:

- In this exercise, you will simulate the 100m sprint race discussed during the lecture.
- We have already defined for you a `Sprinter()` class which has two characteristics for each sprinter:
  - Base time
  - Performance variance
- Run the code cell that makes four instances of the `Sprinter()` class. You will work with those for the entire exercise.
- Call `time` attribute of the helper class to get the time taken by a competitor in the actual race.
- First run the race simulation for five times; you will do this by making a dictionary of participant name as keys, and time taken in a simulated race as the values. You will sort this dictionary by values and determine the winner of the simulated race.
- You will repeat the simulation of the race for 10,000 times and count who won the race for how many times. Based on this observation, you will then investigate *why* a particular participant won as many times.
- You will again repeat the simulation for 10,000 times but this time get the distribution of times for each participant over these runs.
- You will calculate the mean race time, standard deviation of the race time and the confidence interval for each participant.
- You will then run the helper code to observe a plot similar to the one given below:



# Exercise Review

```
In [3]: pavlos = Sprinter(base_speed=13, variance=0.1)
```

```
In [4]: pavlos.time
```

```
Out[4]: 13.029063370231649
```

```
In [5]: pavlos.time
```

```
Out[5]: 13.034317928671507
```

# Exercise Review

```
In [3]: pavlos = Sprinter(base_speed=13,variance=0.1)
```

```
In [4]: pavlos.time
```

```
Out[4]: 13.029063370231649
```

```
In [5]: pavlos.time
```

```
Out[5]: 13.034317928671507
```

```
In [2]: class Sprinter:
```

```
...:
```

```
...:     def __init__(self,base_speed,variance):
```

```
...:         self.base_speed = base_speed
```

```
...:         self.variance = variance
```

```
...:
```

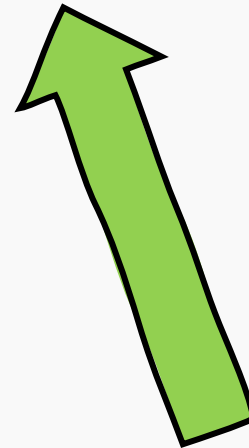
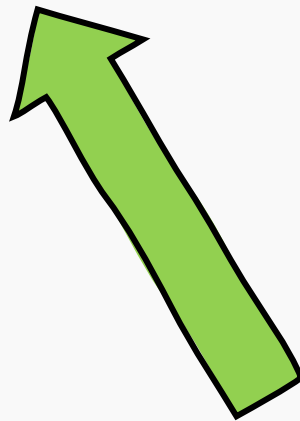
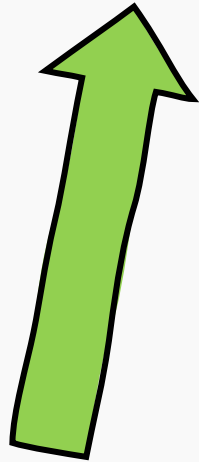
```
...:     @property
```

```
...:     def time(self):
```

```
...:         return np.random.normal(loc= self.base_speed,scale=self.variance)
```

# Exercise Review

```
np.random.normal(loc= self.base_speed, scale=self.variance)
```



what is "normal"?

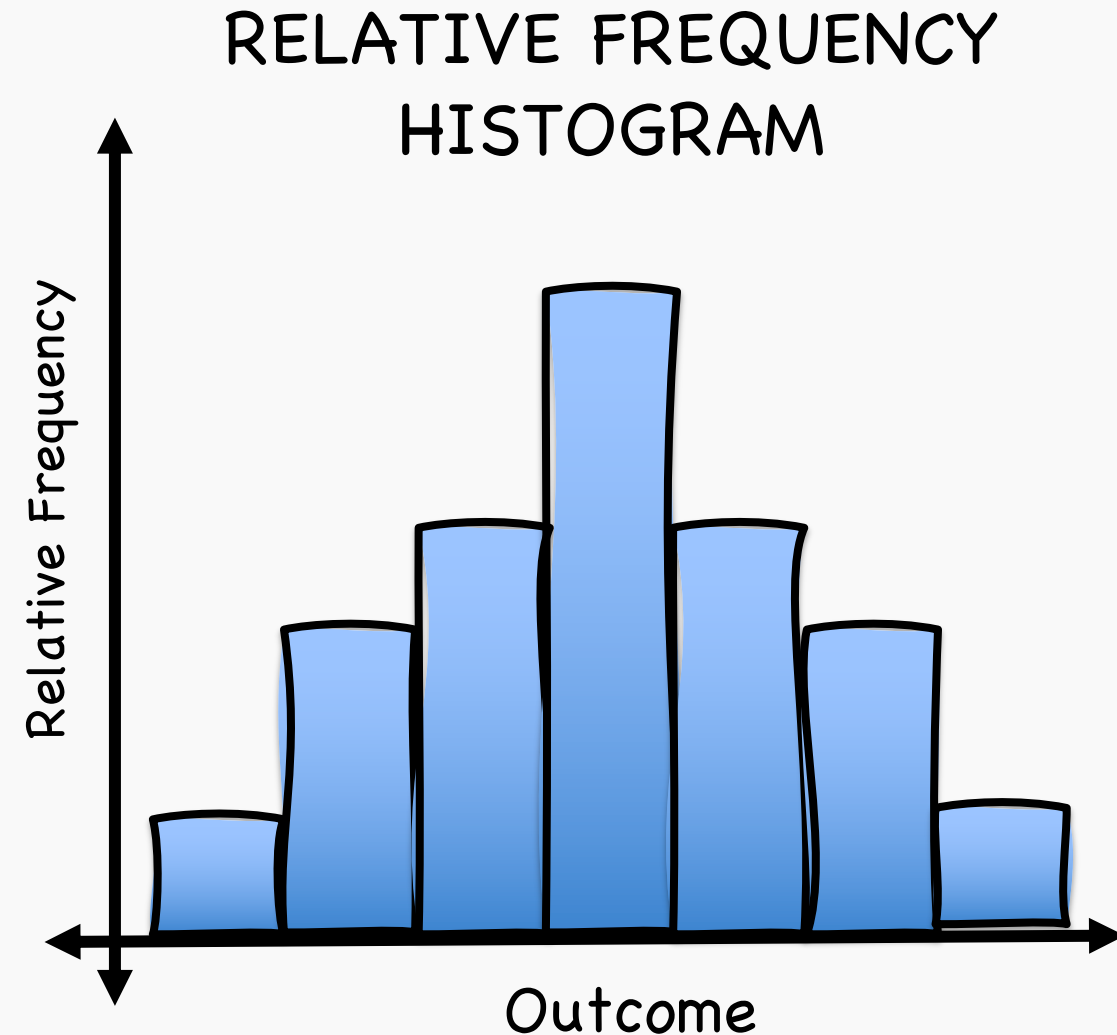
what is location?

what is scale

# Known Distributions

# Histogram as a Probability Density Function (PDF)

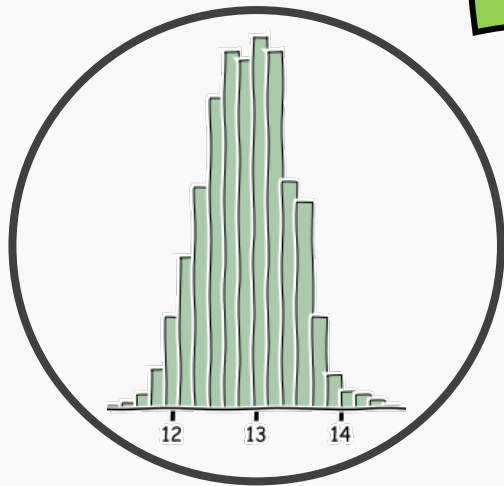
- Recall that a histogram describes the probability of being in a given range (**relative frequency** of the "bins").
- We can describe the probability of being a range with a function.
- This function could be defined for either discrete or continuous variables:
  - In case of continuous, this is the **probability density function (PDF)** for values in a range.
  - In case of discrete, the range is the discrete value itself. The function is the **probability mass function (PMF)**.



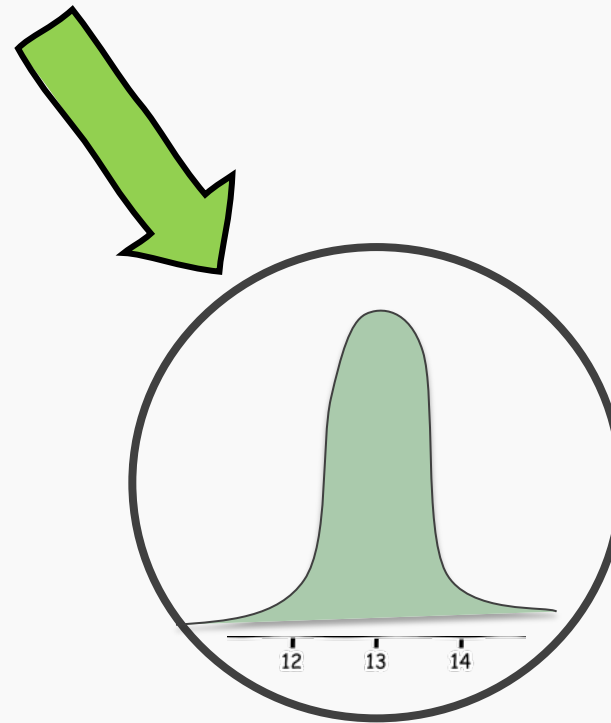
# PDF vs PMF

Random Variable

X



Discrete Random Variable



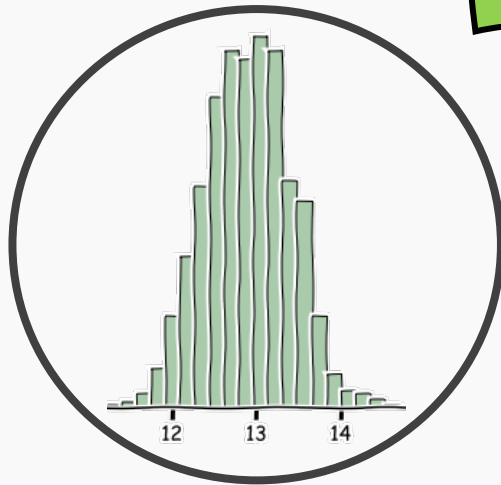
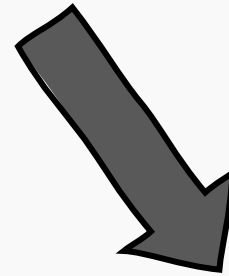
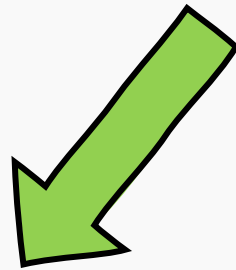
Continuous Random Variable



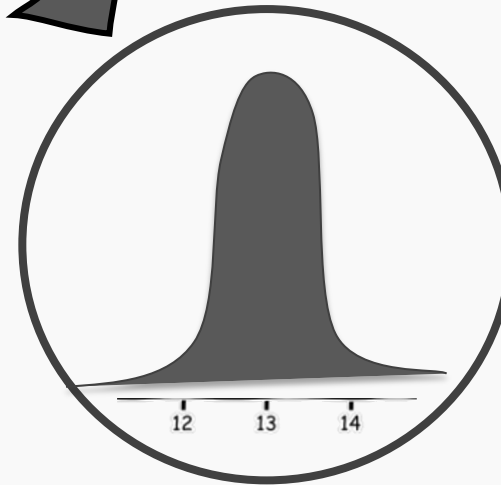
# PDF vs PMF

Random Variable

X



Discrete Random Variable



Continuous Random Variable

# Discrete Uniform Distribution

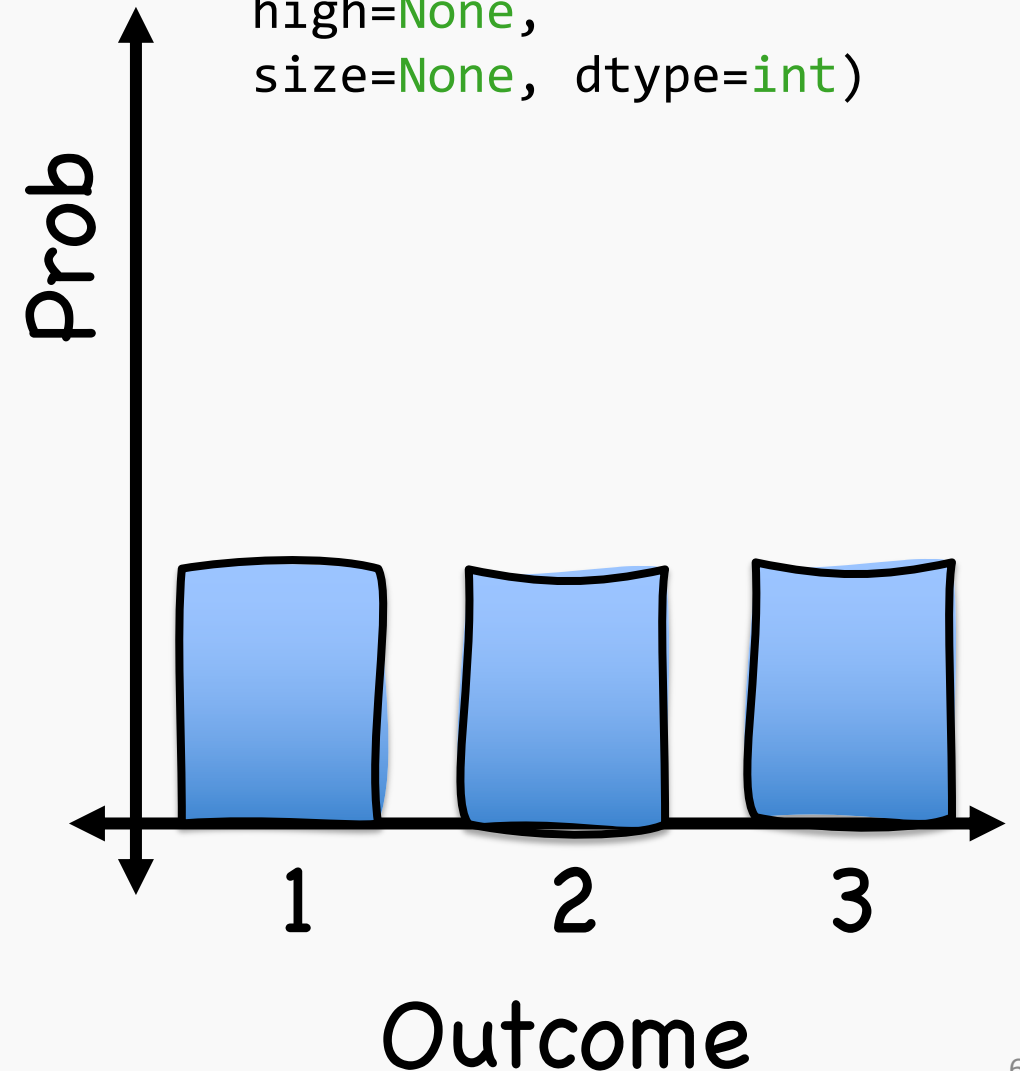
- This distribution occurs when there are a **finite** number of equally likely outcomes possible.

$$\text{PMF: } P(X = \text{"red"}) = \frac{1}{N}$$

$$\text{mean } \mu = \frac{a+b}{2}$$

$$\text{Variance } \sigma^2 = \frac{N^2-1}{12}$$

```
np.random.randint(low,  
high=None,  
size=None, dtype=int)
```



# Bernoulli Distribution

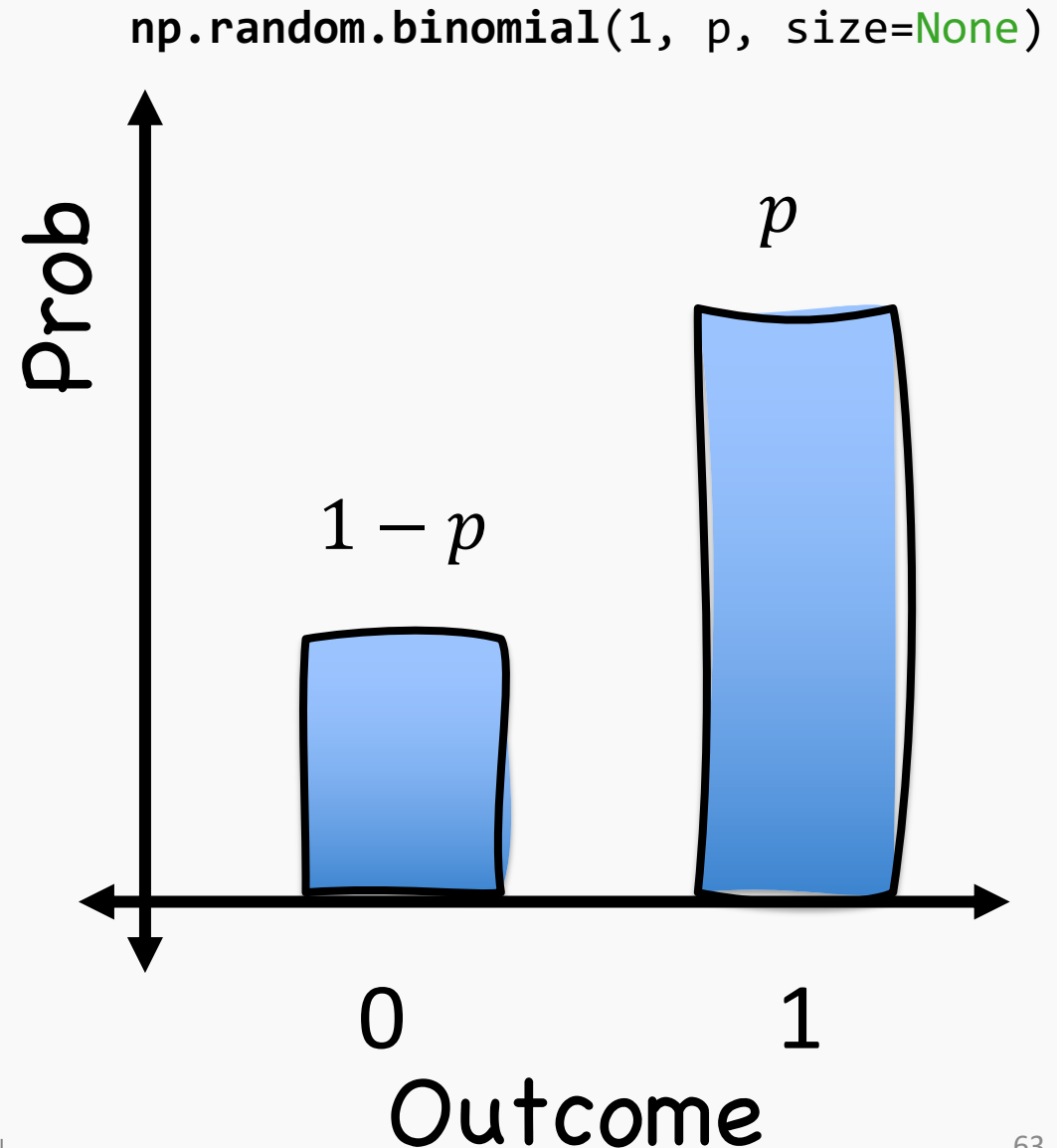
- This distribution can be thought of as a model of possible outcomes of an experiment that asks a **yes-no** question.
- E.g., If you toss a coin, will you get a **head** or a **tail** ?

$$\text{PMF: } P(X = x) = p^x (1 - p)^{1-x}$$

where  $p$  is the probability of success and  $q = 1 - p$  is the probability of failure.

$$\text{mean } \mu = p$$

$$\text{Variance } \sigma^2 = pq$$



# Binomial Distribution

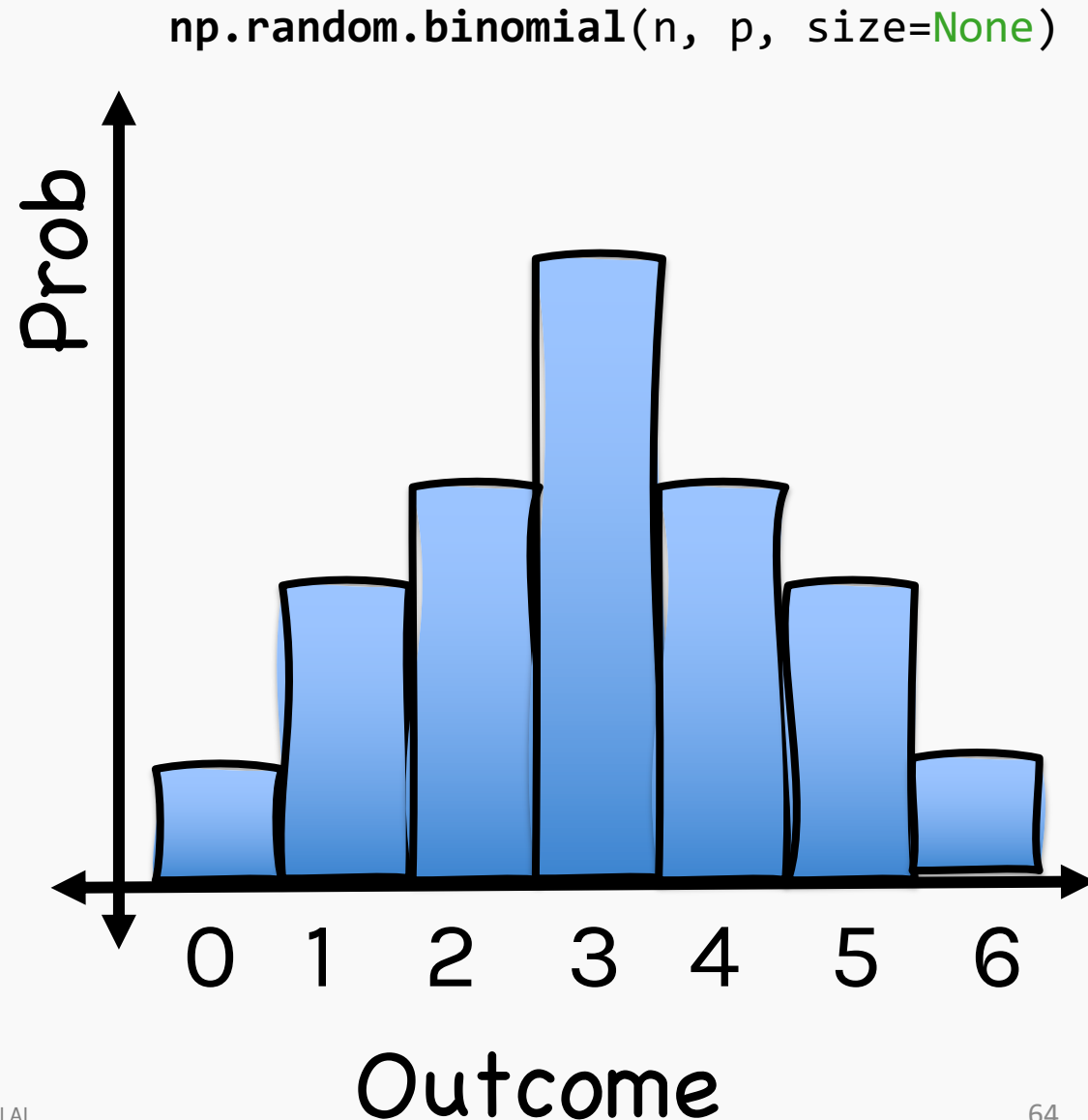
- A binomial distribution with parameters  $n$  and  $p$  is the distribution of the number of  $k$  successes in a sequence of  $n$  independent experiments, each asking a yes-no questions.

$$\text{PMF: } P(X = x) = \binom{n}{k} p^k q^{n-k}$$

where  $p$  is the probability of success and  $q = 1 - p$  is the probability of failure.

$$\text{mean } \mu = np$$

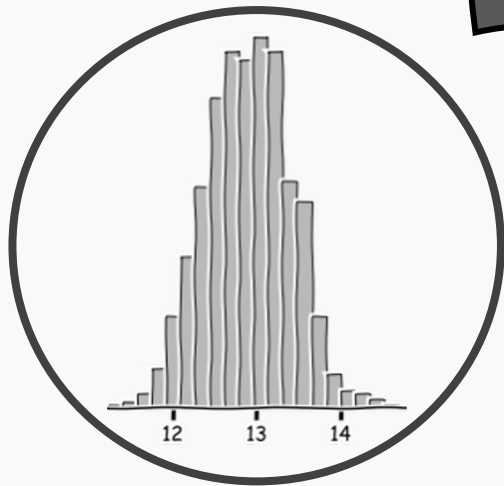
$$\text{Variance } \sigma^2 = npq$$



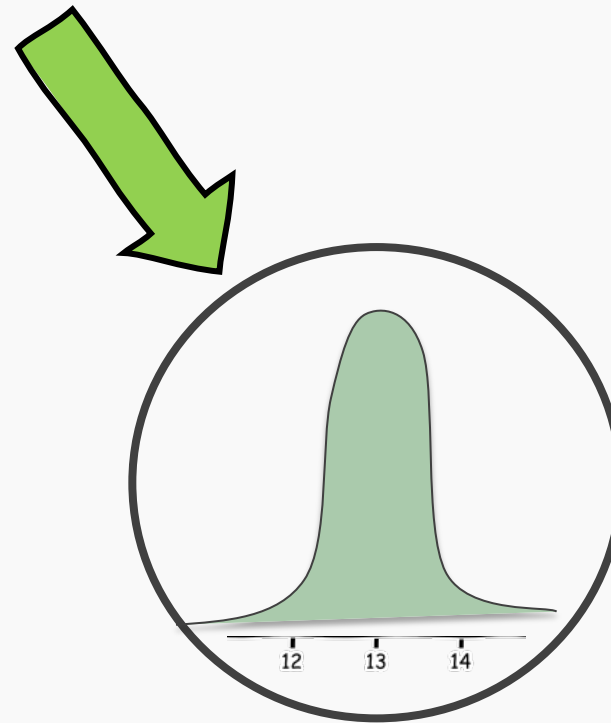
# PDF vs PMF

Random Variable

X



Discrete Random Variable



Continuous Random Variable

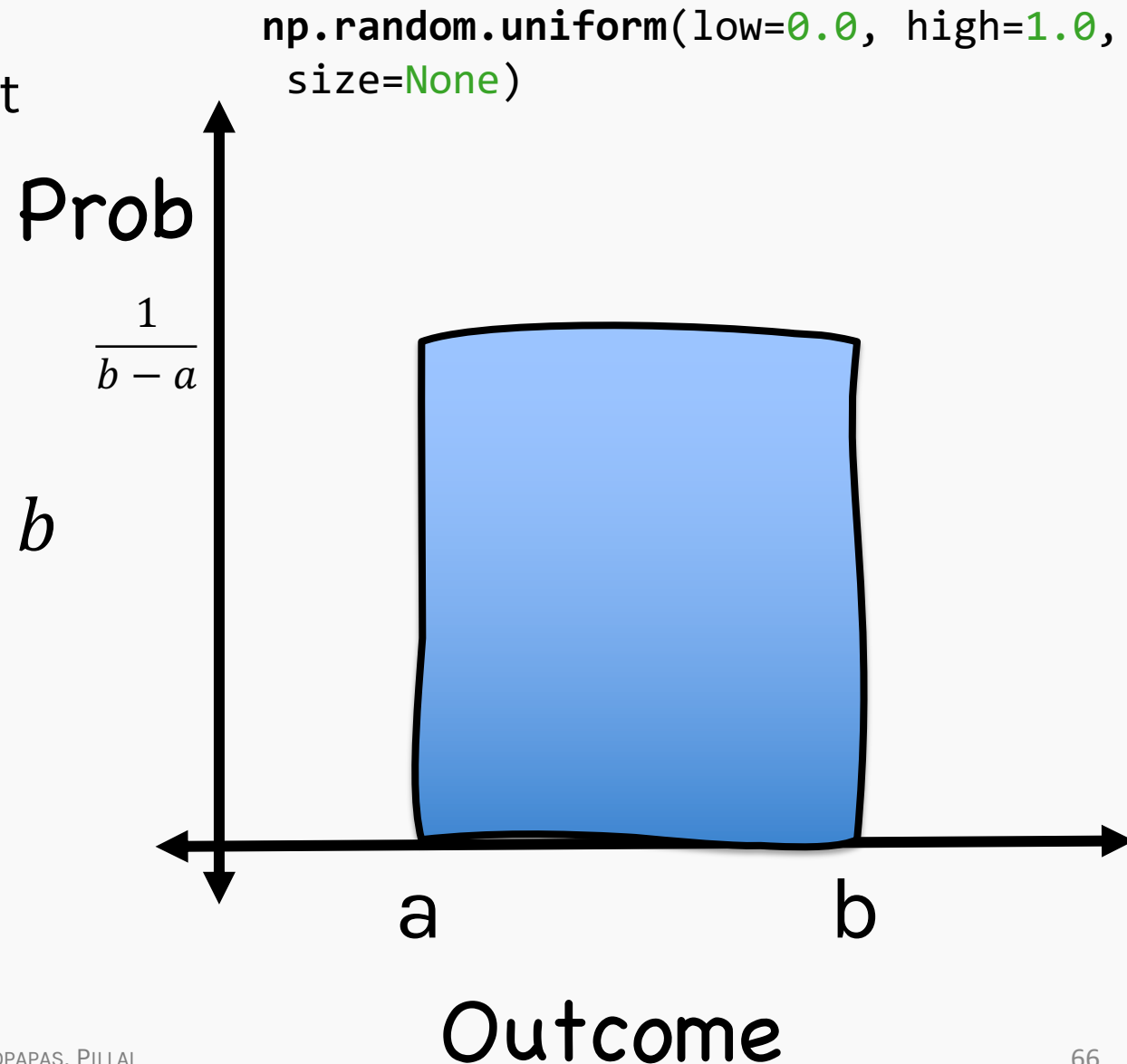
# Uniform continuous distribution

- This distribution describes an experiment where there is an arbitrary outcome that lies between certain **bounds**, defined by parameters  $a$  and  $b$ .

$$\text{PDF: } P(X = x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \end{cases}$$

$$\text{mean } \mu = \frac{a+b}{2}$$

$$\text{Variance } \sigma^2 = \frac{(b-a)^2}{12}$$



# Normal distribution

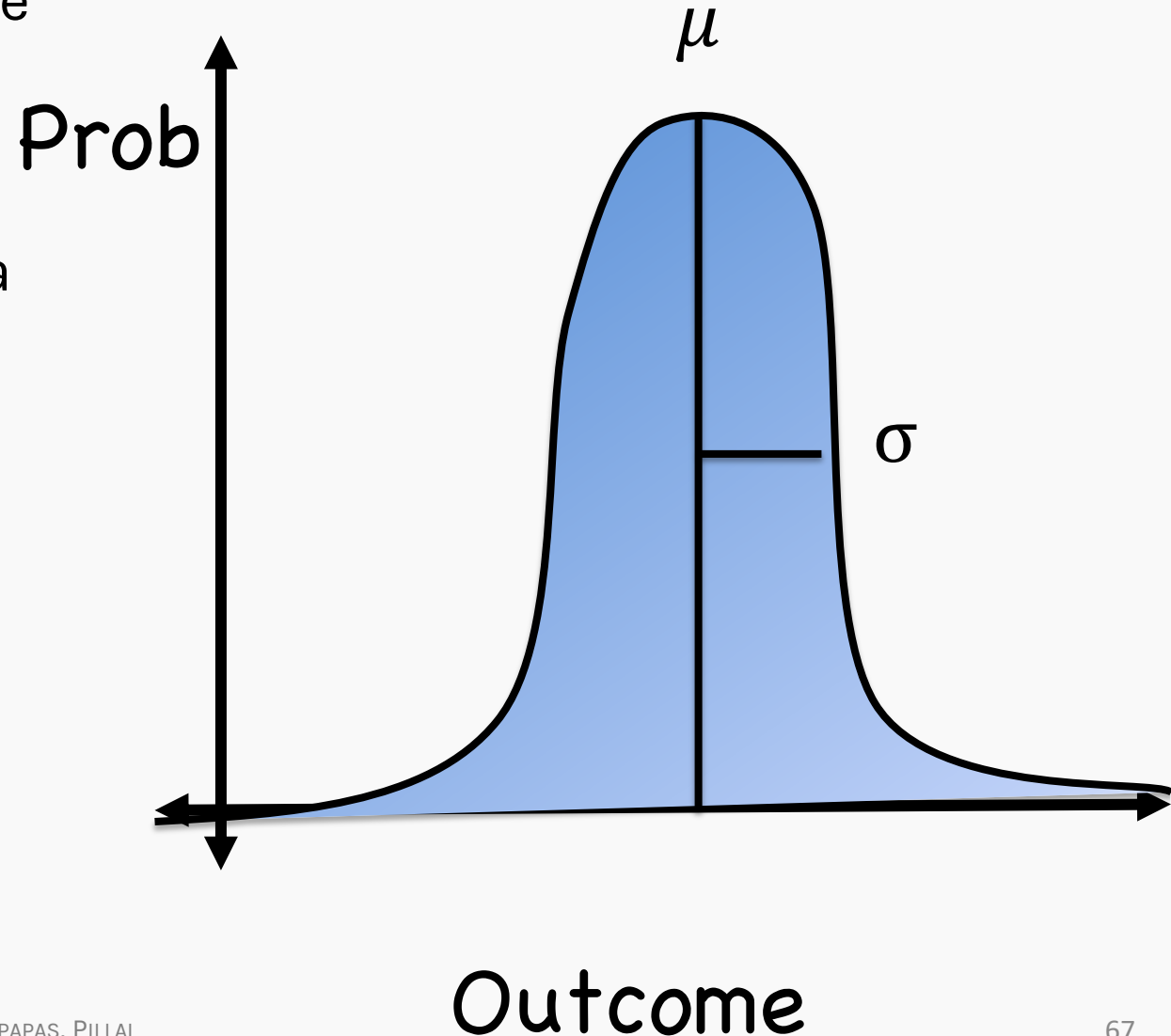
```
np.random.normal(loc=0.0, scale=1.0,  
size=None)  
np.random.randn()
```

- A normal (or Gaussian) distribution is one of the most used continuous random variables.
- As a result of the [central limit theorem](#), a random variable with an unknown distribution could be approximated with the normal distribution.

$$\text{PDF: } P(X = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$\text{mean } E[X] = \mu \quad (\text{location})$$

$$\text{Variance } E[(X - \mu)^2] = \sigma^2 \quad (\text{scale})$$





Why is the normal distribution used so often?

The **Central Limit Theorem**: random variables that are averages or sums of many other random variables will be approximately normally distributed.

More specifically: if  $X_1, X_2, \dots, X_n$  are independent random variables (representing individual observations of data) with mean  $\mu$  and standard deviation  $\sigma$  (not necessarily normal themselves), then the sample mean

$$\bar{X} = \frac{X_1 + X_2 + \dots + X_n}{n}$$

will have approximate distribution:

$$\bar{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$



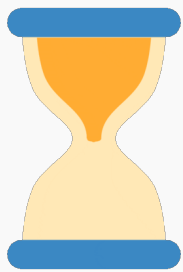


What happens to these probability distributions (PMFs and PDFs) when there are multiple random variables involved (aka, multiple observations in a data set)?

Let  $f(x_1, x_2, \dots, x_n)$  be the **joint distribution** of  $n$  separate random variables. If they all come from the same generative marginal distribution,  $f(x_i)$ , and are independent, what is the resulting distribution?

$$f(x_1, x_2, \dots, x_n) = f(x_1) \cdot f(x_2) \cdots f(x_n) = \prod_{i=1}^n f(x_i)$$

GIVEN THE PACE OF  
TECHNOLOGY, I PROPOSE  
WE LEAVE MATH TO THE  
MACHINES AND GO PLAY  
OUTSIDE.



# Digestion Time

# Modeling Data with Probability Distributions

# The Probability of Data

In a typical probability problem (like in Stat 104 or 110), you would be told something like “20% of Harvard College students are collegiate athletes. What is the probability that there are 50 athletes in a random sample of 200 students from Harvard College?”

$$P(X = 50) = \binom{200}{50} (0.20)^{50} (0.80)^{150} = 0.0149$$

$$P(X \geq 50) = \sum_{x=50}^{200} \binom{200}{x} (0.20)^x (0.80)^{200-x} = 0.0494$$

An alternative question: what is more likely to occur: 50 athletes or 40 athletes in a sample of 200 students? How can we make the determination?

# Inference: the inverse of probability

---

In the last problem, how did we know that the statement “20% of Harvard College students are collegiate athletes” is accurate? Where did this come from?

In most applications, the true population parameter (here, the proportion in all of Harvard College) is unknown. What we get to observe is the data, and we want to make a statement about the unknown parameter. So a more poignant question would be:

“There are 50 athletes in a random sample of 200 students from Harvard College. Is a binomial distribution with  $p = 0.2$  or  $p = 0.25$  more reasonable?”

This approach of using the data to make a statement about a parameter (in a statistical model) is called **inference**.

# Likelihood Theory

# The idea of likelihood

The **likelihood** approach to inference is based on exactly what was presented in the last slide: given observed values of data (summarized by specific sample statistics), what values of the model's parameters are likely?

It simply just **flips** a PDF or PMF on its head: instead of writing this function with the data ( $X$ ) as the unknown, it uses the same function but uses the parameter(s) as the unknown(s). The **likelihood function**,  $\mathcal{L}$ , measures how well a model (and its set of parameters) describes the observed data.

For a set of independent and normally distributed random variables,  $X_i \sim N(\mu, \sigma^2)$ :

$$\mathcal{L}(\mu, \sigma^2 | x_1, \dots, x_n) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{x_i - \mu}{\sigma}\right)^2}$$

# The log-likelihood

If the goal is optimization, why is transforming via the log function a good choice?



The likelihood function measures how well a model describes observed data. So, it makes sense that we want a model (or set of parameters) that maximizes this function.

Likelihood functions are typically products of many similar pieces, and products are difficult to maximize (both mathematically and numerically). Why?

So instead, the log of the likelihood function, called the **log-likelihood function**,  $\ell$ , is used. For the Normal distribution model:

$$\ell(\mu, \sigma^2 | x_1, \dots, x_n) = \ln \left( \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{x_i - \mu}{\sigma}\right)^2} \right) = - \sum_{i=1}^n \sqrt{2\pi\sigma^2} - \sum_{i=1}^n \left(\frac{x_i - \mu}{\sigma}\right)^2$$



# Maximizing the likelihood



In order to choose the best Normal distribution to describe a set of data, we should maximize the likelihood that chooses the best set of parameters given the data.

The **maximum likelihood estimates** for a statistical model are those that maximize the likelihood function given the observed data.

How do we do this mathematically? How could we do this computationally?

With Math: \_\_\_\_\_

With Computers: \_\_\_\_\_

# Modeling Linear Regression Probabilistically

# The Simple Linear Regression Model

---

We've defined the linear regression model to predict the  $i$ -th observation's response,  $Y_i$ , from a predictor,  $X_i$ , to be:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

For any random variable,  $\epsilon$ , that has zero mean

$$E(Y_i) = \beta_0 + \beta_1 X_i$$

The error term,  $\epsilon_i$ , represents the distance the observation lies from the line in the vertical distance (direction of  $Y$ ).

# The Probabilistic Regression Model

If we assume that  $\epsilon_i \sim N(0, \sigma^2)$

This regression model can be rewritten as:

$$Y_i | X_i \sim N(\beta_0 + \beta_1 X_i, \sigma^2)$$

The likelihood of a measurement having value  $Y_i$  given  $X_i$  for a model  $\beta_0, \beta_1$

$$L(\beta_0, \beta_1, \sigma^2 | Y_i, X_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{Y_i - (\beta_0 + \beta_1 X_i)}{\sigma}\right)^2}$$

# The Probabilistic Regression Model

The likelihood of a measurement having value  $Y_i$  given  $X_i$  for a model  $\beta_0, \beta_1$

$$L(\beta_0, \beta_1, \sigma^2 | Y_i, X_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{Y_i - (\beta_0 + \beta_1 X_i)}{\sigma}\right)^2}$$

This formulation allows us to write out the joint likelihood function for this probability model.

The joint likelihood function for this probability model becomes:

$$L(\beta_0, \beta_1, \sigma^2 | \mathbf{Y}, \mathbf{X}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{Y_i - (\beta_0 + \beta_1 X_i)}{\sigma}\right)^2}$$

# The Likelihood of Linear Regression

What does this function look eerily similar to? What does maximize this function lead to with regards to the best estimates of  $\beta_0, \beta_1$ ?



The joint likelihood function for this probability model becomes:

$$L(\beta_0, \beta_1, \sigma^2 | \mathbf{Y}, \mathbf{X}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{Y_i - (\beta_0 + \beta_1 X_i)}{\sigma}\right)^2}$$

Which leads to the log-likelihood:

$$l(\beta_0, \beta_1, \sigma^2 | \mathbf{Y}, \mathbf{X}) = \ln L(\beta_0, \beta_1, \sigma^2 | \mathbf{Y}, \mathbf{X}) = - \sum_{i=1}^n \ln(\sqrt{2\pi\sigma^2}) - \sum_{i=1}^n \left(\frac{y_i - (\beta_0 + \beta_1 x_i)}{\sigma}\right)^2$$

What should we do with this log-likelihood?

# The Likelihood of Linear Regression

Instead of **maximize** the log-likelihood we can **minimize** the *negative-log-likelihood*:

$$-l(\beta_0, \beta_1, \sigma^2 | \mathbf{Y}, \mathbf{X}) = \sum_{i=1}^n \ln(\sqrt{2\pi\sigma^2}) + \sum_{i=1}^n \left( \frac{y_i - (\beta_0 + \beta_1 x_i)}{\sigma} \right)^2$$

Which is equivalent to **minimizing**

$$MSE = \frac{1}{n} \sum_{i=1}^n \left( \frac{y_i - (\beta_0 + \beta_1 x_i)}{\sigma} \right)^2$$

# Take home message

---

By taking a probabilistic approach to linear regression and assuming the residuals are normally distributed, we see that **maximizing the likelihood** for this model is equivalent to **minimizing mean squared error** around the line!

So, if we believe our residuals are normally distributed, then minimizing mean square error is a natural choice.

But by choosing this specific probability model, we get much more than simply motivation for our loss function. We get *instructions* on how to perform inferences as well 😊

We will see this in more details in next lecture!



# Checking the assumptions of this model:

---

The probabilistic model of linear regression leads to 4 main assumptions that can be checked with the data (the first 3 at least):

1. Linearity: relationships are linear and there is no clear non-linear pattern around the line (as evidenced by the residuals).
2. Normality: the residuals are normally distributed.
3. Constant Variance: the vertical spread of the residuals is constant everywhere along the line.
4. Independence: the observations are independent of each other.

**Note:** collinearity is not a violation of an assumption, but can certainly muck up the model.

# Thank you