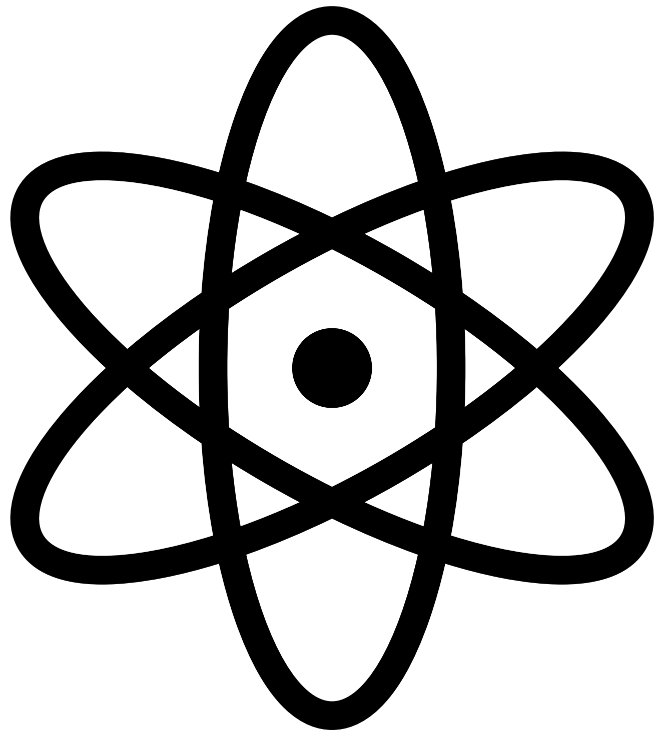
# ENTROpy: Bring the Noise

Michael Colaresi

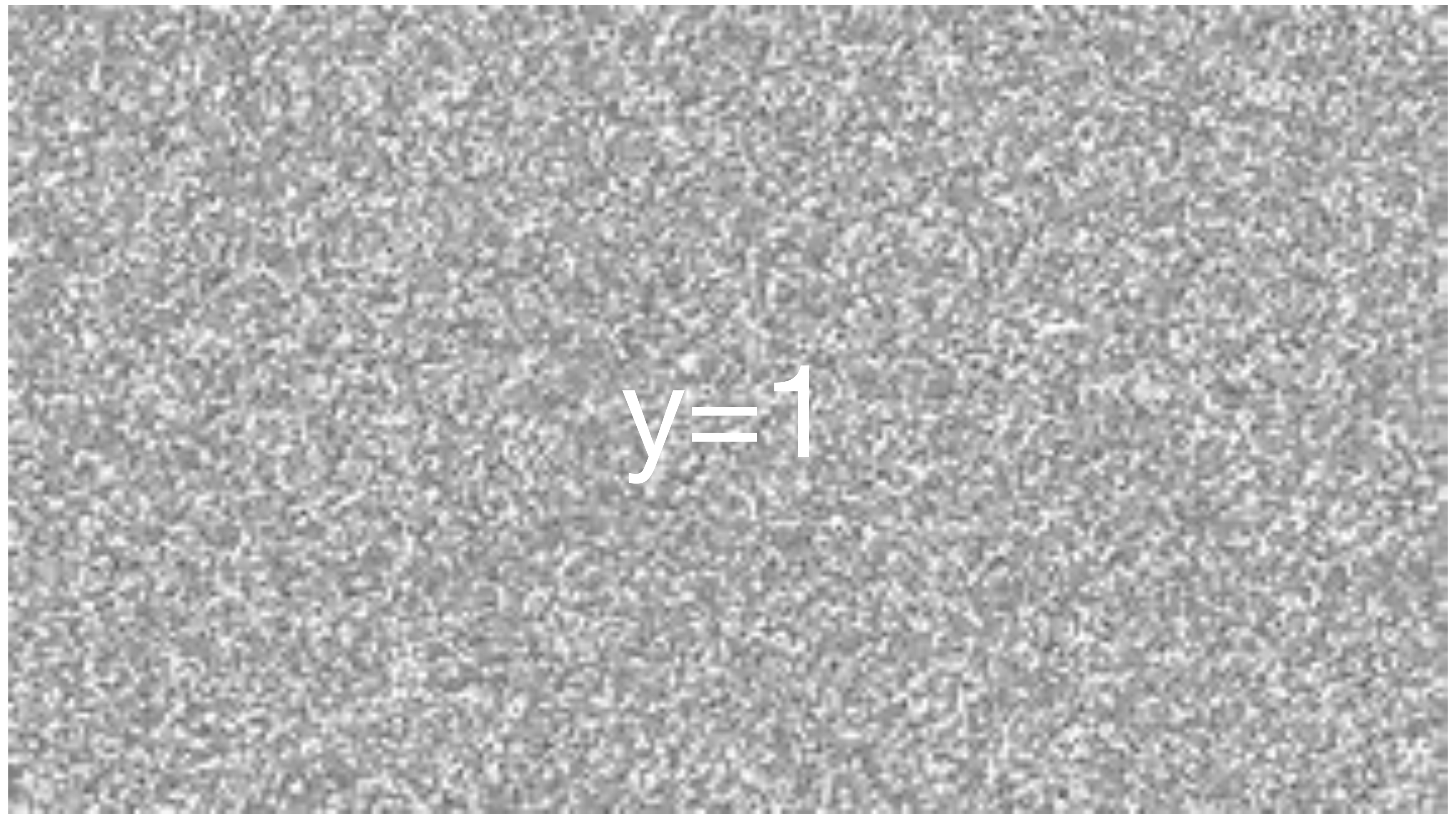
# Fundamental Elements of Research

- Prob(Y) = f(structure, noise)
  - Think about predictions of some process Y (Prob(Y))
  - Partion that process into two parts
    - Structure includes what we can hope to model
      - Eg $\hat{y} = xB$
  - Noise is the uncertainty around the structure
    - Eg $\epsilon \sim N(0,1)$



# From Entropy to Information

## Shannon and Off




- Claude Shannon birthed information theory
    - Understanding what information is (…really)
- Fundamental to computers and modern IT (it is awesome)
- What is information then you might ask?
    - The absence of entropy, of course
        - Uh ok, what is entropy, then?

# Entropy by example

- Take an event that can either happen or not happen $y \in \{0,1\}$
- Call the probability of that event $p$
- What values of $p$ implies that we have "information" on the event?
  - Think about if the event occurred (y=1)
    - If you thought p was ⬆️ like .9999, then you learned almost nothing
    - If you thought p was ⬇️ like .00001, then you are very surprised
      - How surprised: - math.log2(.00001)
  - Then do the same for the eventuality if the event did not occur (y=0)
    - If p ⬆️, then super surprised, if p ⬇️, then not as surprised

# Entropy by example

So information is already knowing something, and entropy is not knowing/ being uncertain

- Take an event that can either happen or not happen $y \in \{0,1\}$
- Call the probability of that event $p$
- What values of $p$ implies that we have "information" on the event?
  - Think about if the event occurred (y=1)
    - If you thought p was ⬆️ like .9999, then you learned almost nothing
    - If you thought p was ⬇️ like .00001, then you are very surprised
      - How surprised: - math.log2(.00001)
  - Then do the same for the eventuality if the event did not occur (y=0)
    - If p ⬆️, then super surprised, if p ⬇️, then not as surprised

# Entropy by equation

- **BUT, we do not know if the event did or will occurred or not (y=0 or y=1)**
  - So we can take the expected value of the surprise/information across outcomes
  - $-(p(log_2(p)) + (1-p)(log_2(1-p)))$
- More generally:
  - $-\sum_{i \in o} p_i(log_2 p_i)$ where o is the set of possible outcomes.
  - Also note that base of log does not have to be 2 (which are … wait for it … bits)

# Entropy by equation

So Entropy is the average surprise over all the possible outcomes from a certain perspective (you need p!)

- **BUT, we do not know if the event did or will occurred or not (y=0 or y=1)**
  - So we can take the expected value of the surprise/information across outcomes
  - $-(p(log_2(p)) + (1-p)(log_2(1-p)))$
- More generally:
  - $-\sum_{i \in o} p_i(log_2 p_i)$ where o is the set of possible outcomes.
  - Also note that base of log does not have to be 2 (which are … wait for it … bits)

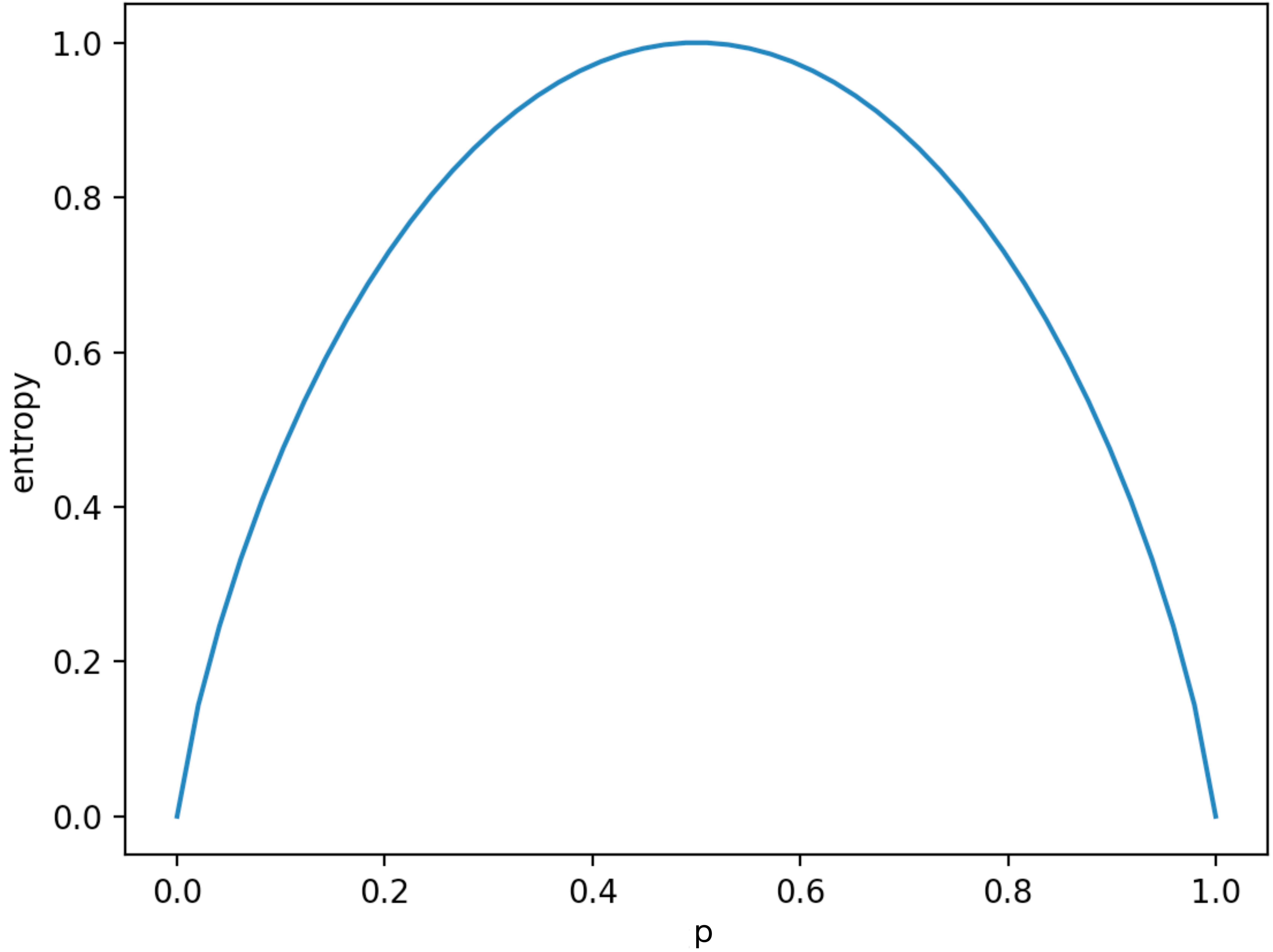
# Entropy by equation

So Entropy is the average surprise over all the possible outcomes from a certain perspective (you need p!)

- Now, we do not know if the event occurred or not (y=0 or y=1)
  - So we can take the expected value of the surprise/information across outcomes
  - $-(p(log_2(p)) + (1-p)(log_2(1-p)))$
- More generally:
  - Entropy!
  - $\boxed{-\sum_{i \in o} p_i(log_2 p_i)}$ where o is the set of possible outcomes.
  - Also note that base of log does not have to be 2 (bits)

$y \in \{0,1\}$

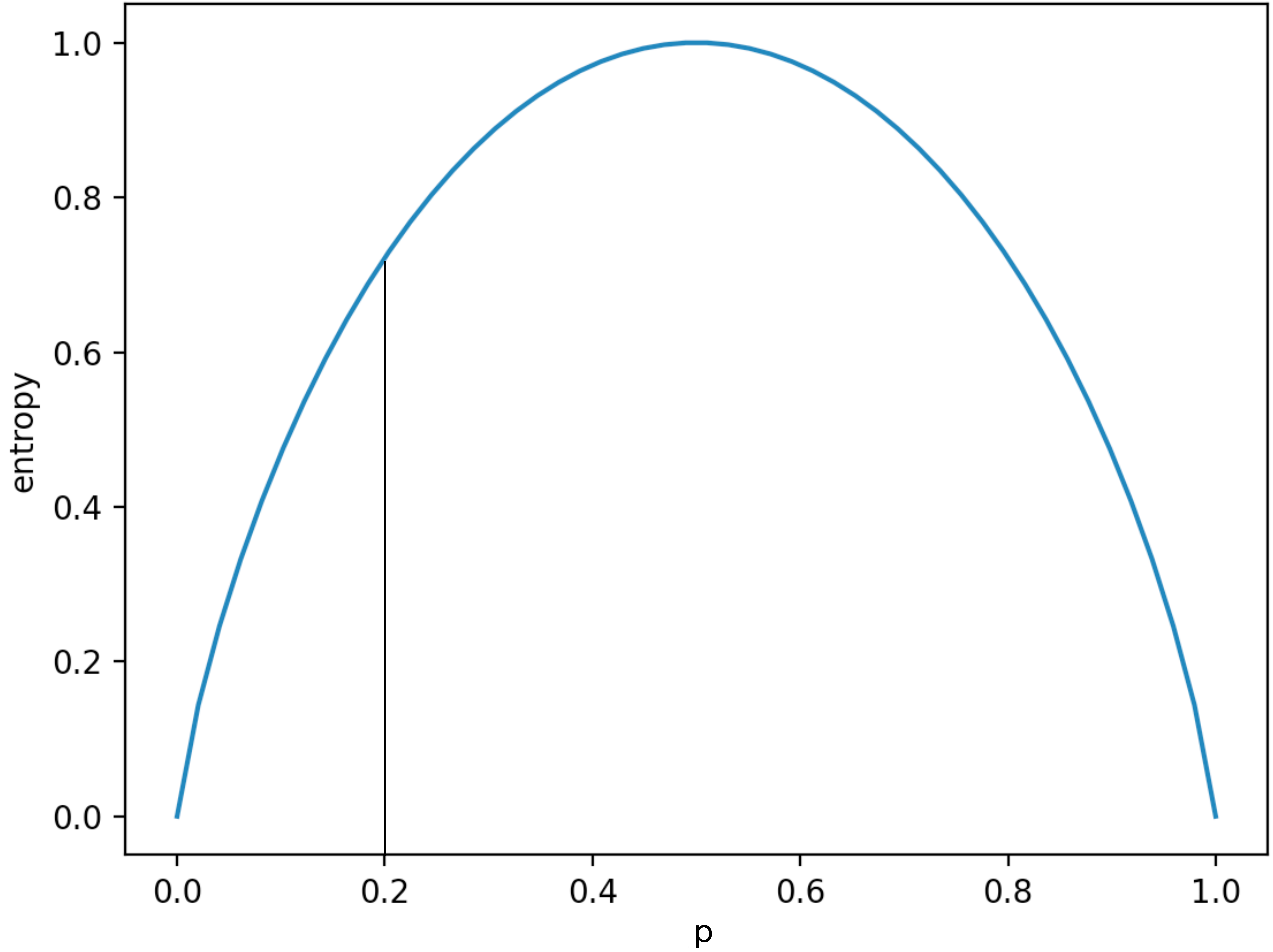
Note, each x value is a distribution over the 2 outcomes
The probabilities sum to 1
Eg .2 is the distribution where Pr(y=1) = .2 and the Pr(y=0) = .8




How “surprised” would you be on average

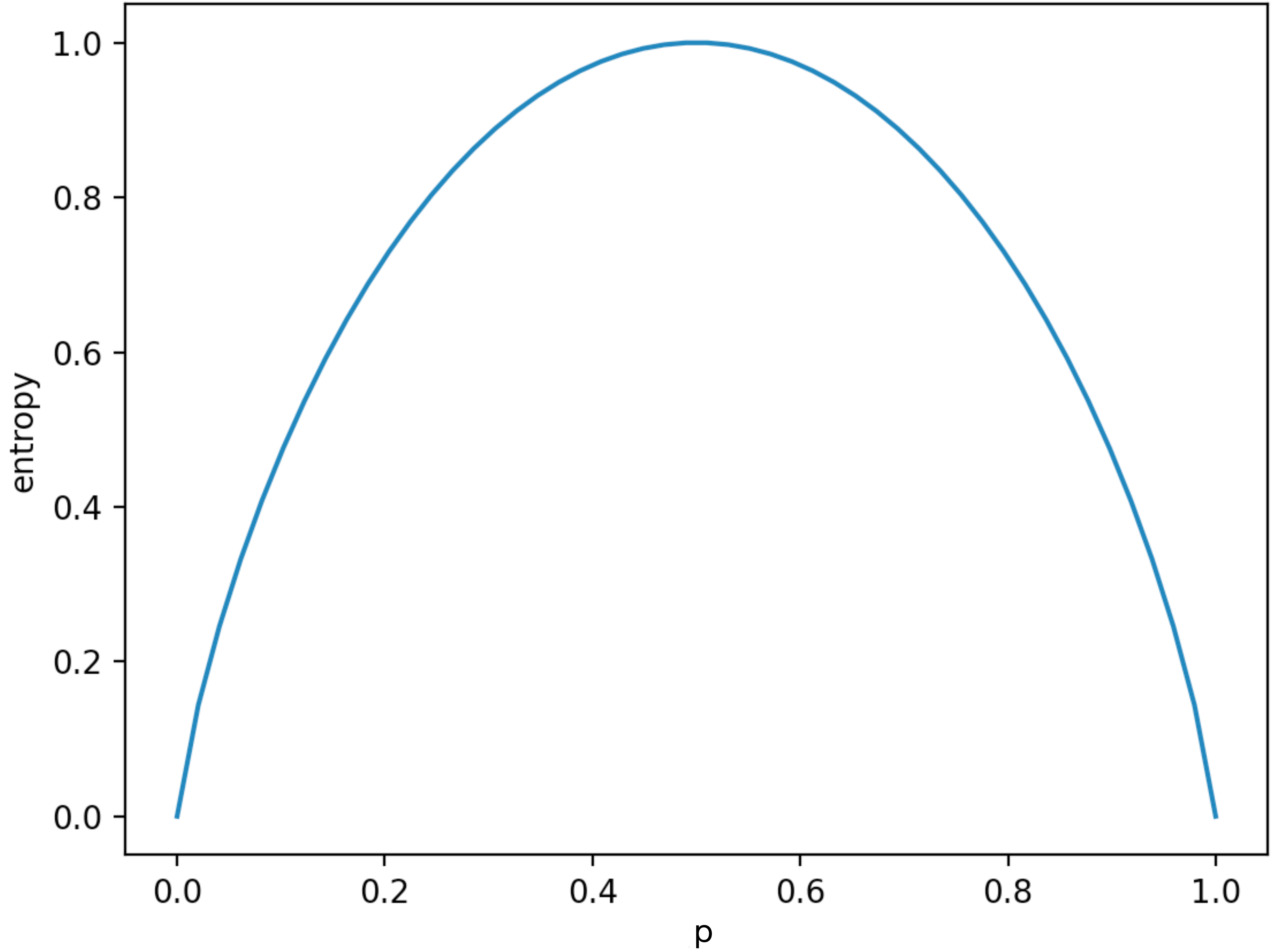
$y \in \{0,1\}$

Each distribution/point on the x-axis) has an entropy value

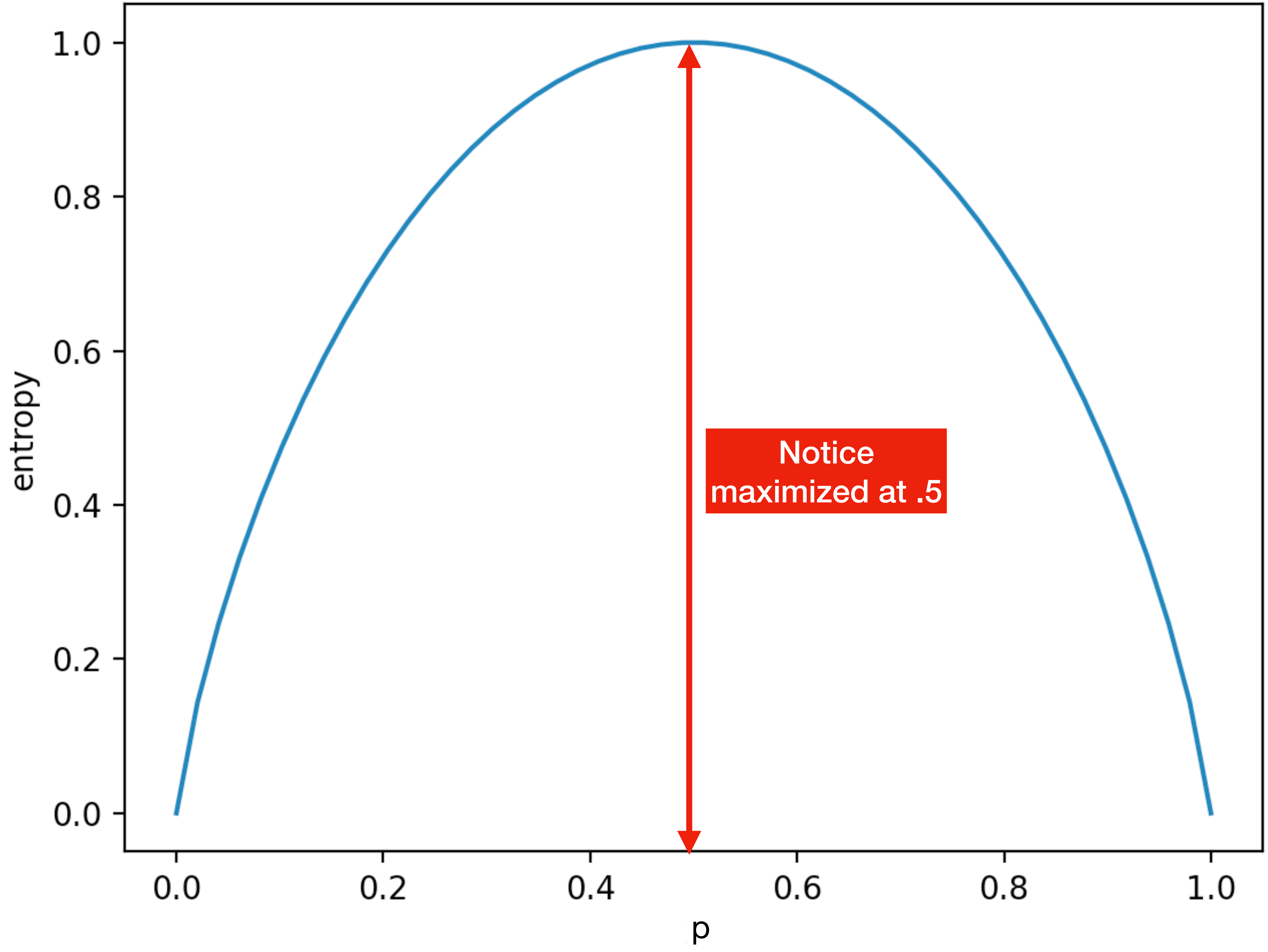



How "surprised" would you be on average

$y \in \{0,1\}$




The entropy measures how “surprised” you would be on average if you held this state of information/shape of information about this system




The most surprised you could be on average

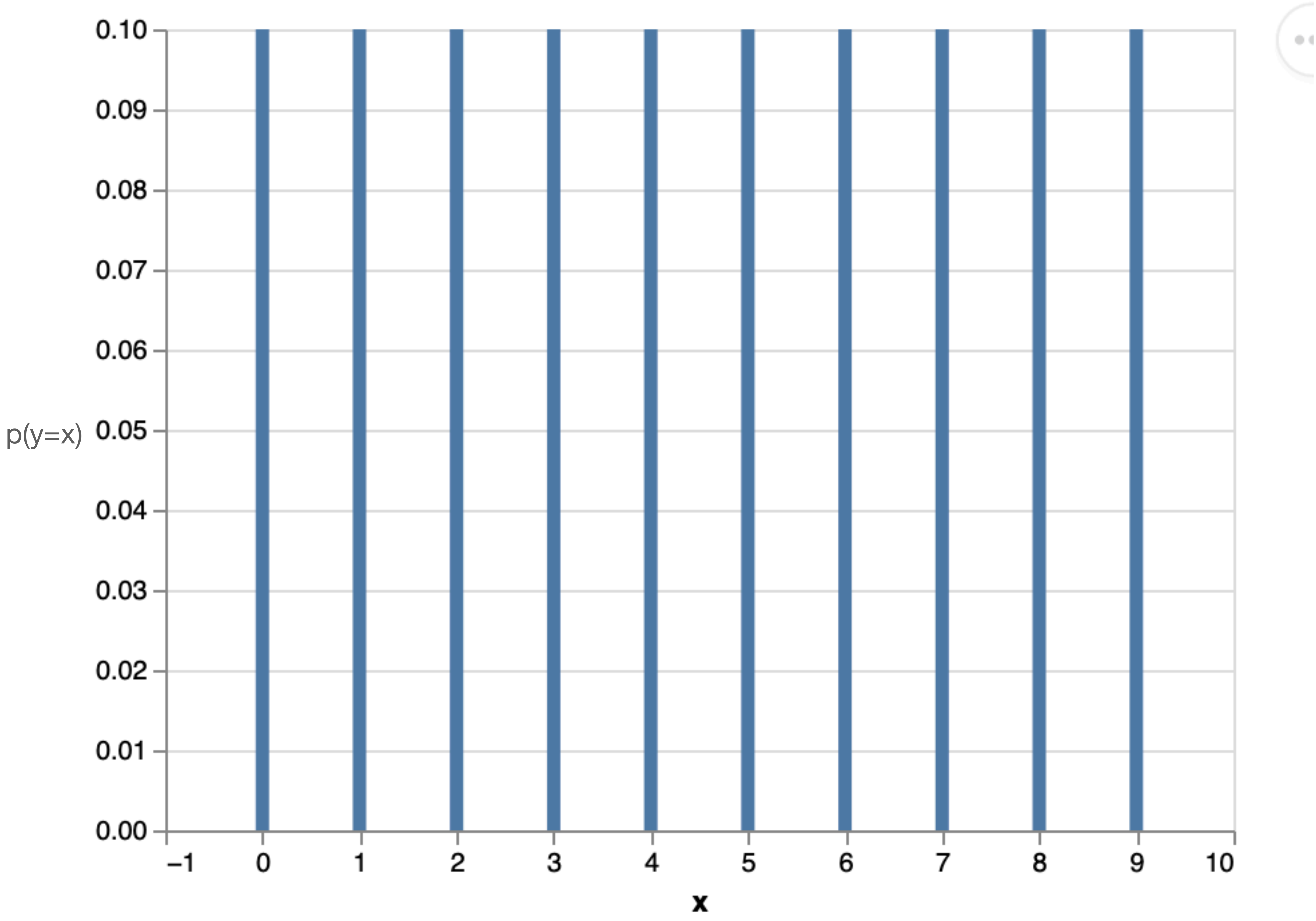
“Maximum entropy”

# Maximum Entropy is Maximumly Important

**Given constraints, what is the most surprise a state of information can hold?**

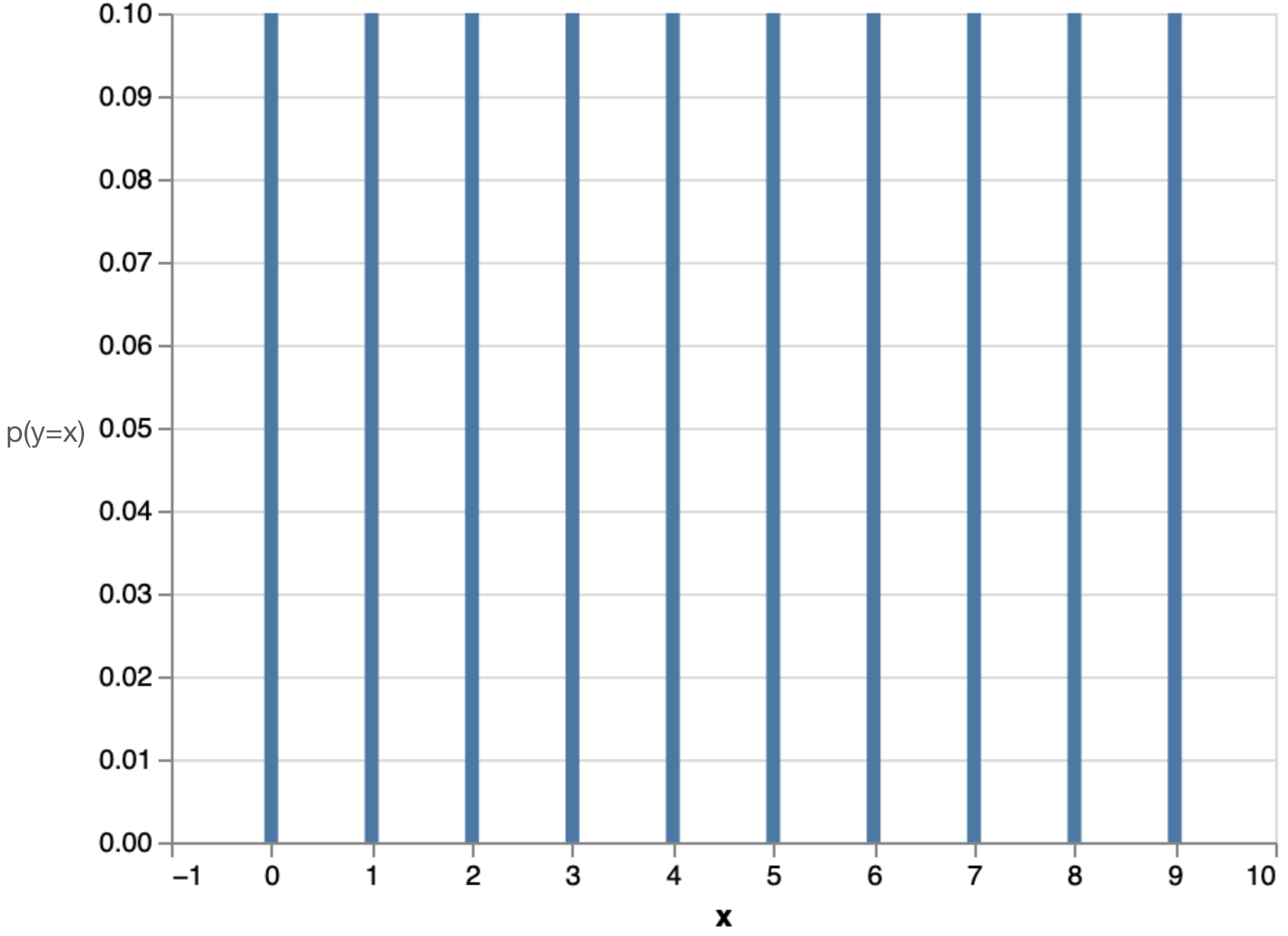
- Let's look at systems that have more than 2 states.
  - Pick an integer between 0 and 9, y = {0, 1, 2, …, 9}
    - What is the probability of each possible value where you would be maximally surprised?
      - Define $p_i$ as the probability that y=i.

# Maximize entropy for discrete values 0 to 9




This is a pmf, a probability mass function
It gives you the mass of probability placed across values… and its sums to 1
There are infinitely many others, just shrink one bar and raise others in turn so sum remains 1

# Maximize entropy for discrete values 0 to 9




Uniform distribution over possible values

Why?

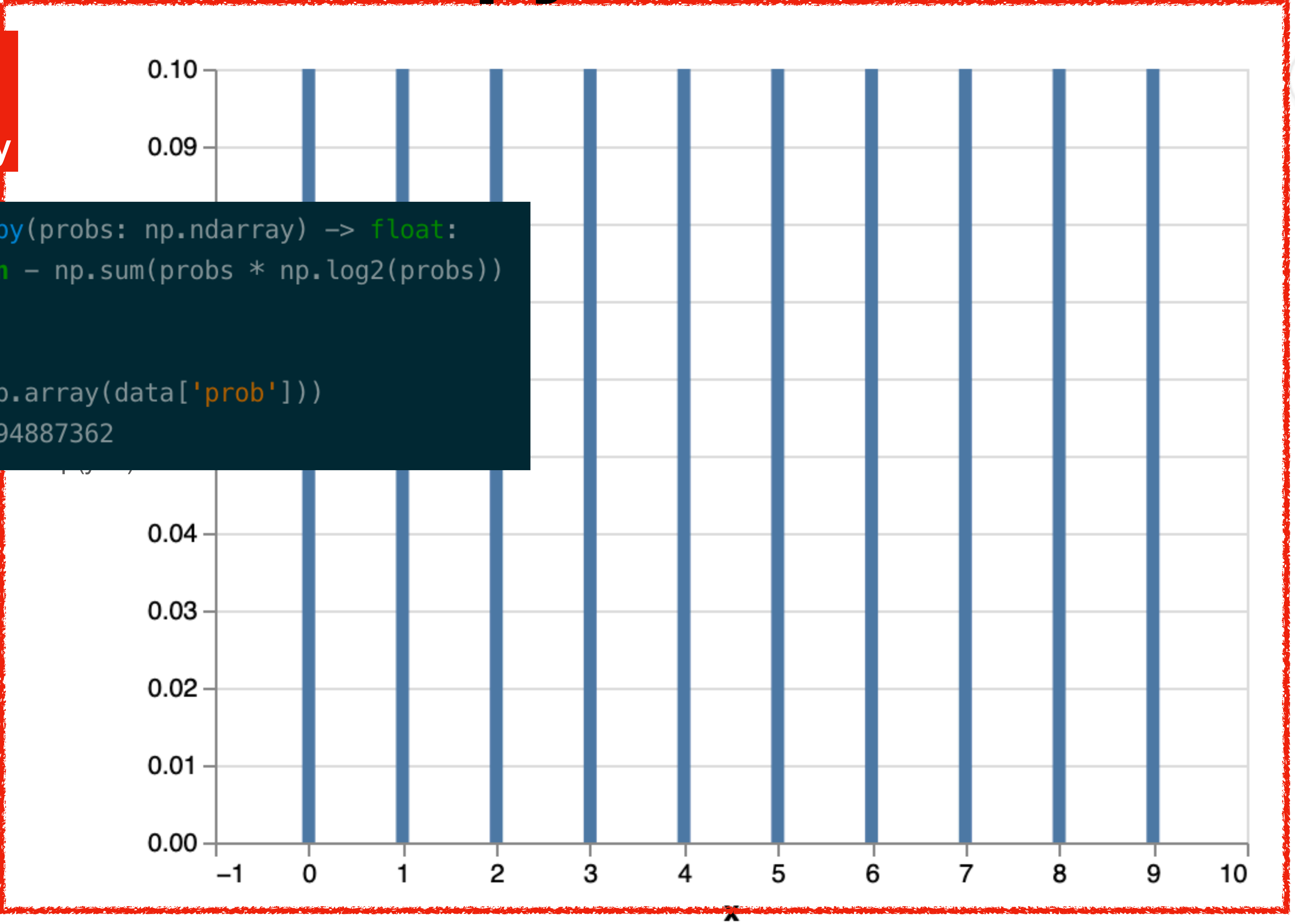
Maximizes

$$-\sum_{i \in o} p_i(log_2 p_i)$$

This is a pmf, a probability mass function
It gives you the mass of probability placed across values… and its sums to 1
There are infinitely many others, just shrink one bar and raise others in turn so sum remains 1

# Maximize entropy for discrete values 0 to 9

This whole distributions has 1 average entropy




```
In [28]: def entropy(probs: np.ndarray) -> float:
    ...:     return - np.sum(probs * np.log2(probs))
    ...:

In [29]: entropy(np.array(data['prob']))
Out[29]: 3.321928094887362
```

Uniform distribution over possible values

Why?

Maximizes

$$-\sum_{i \in o} p_i(log_2 p_i)$$

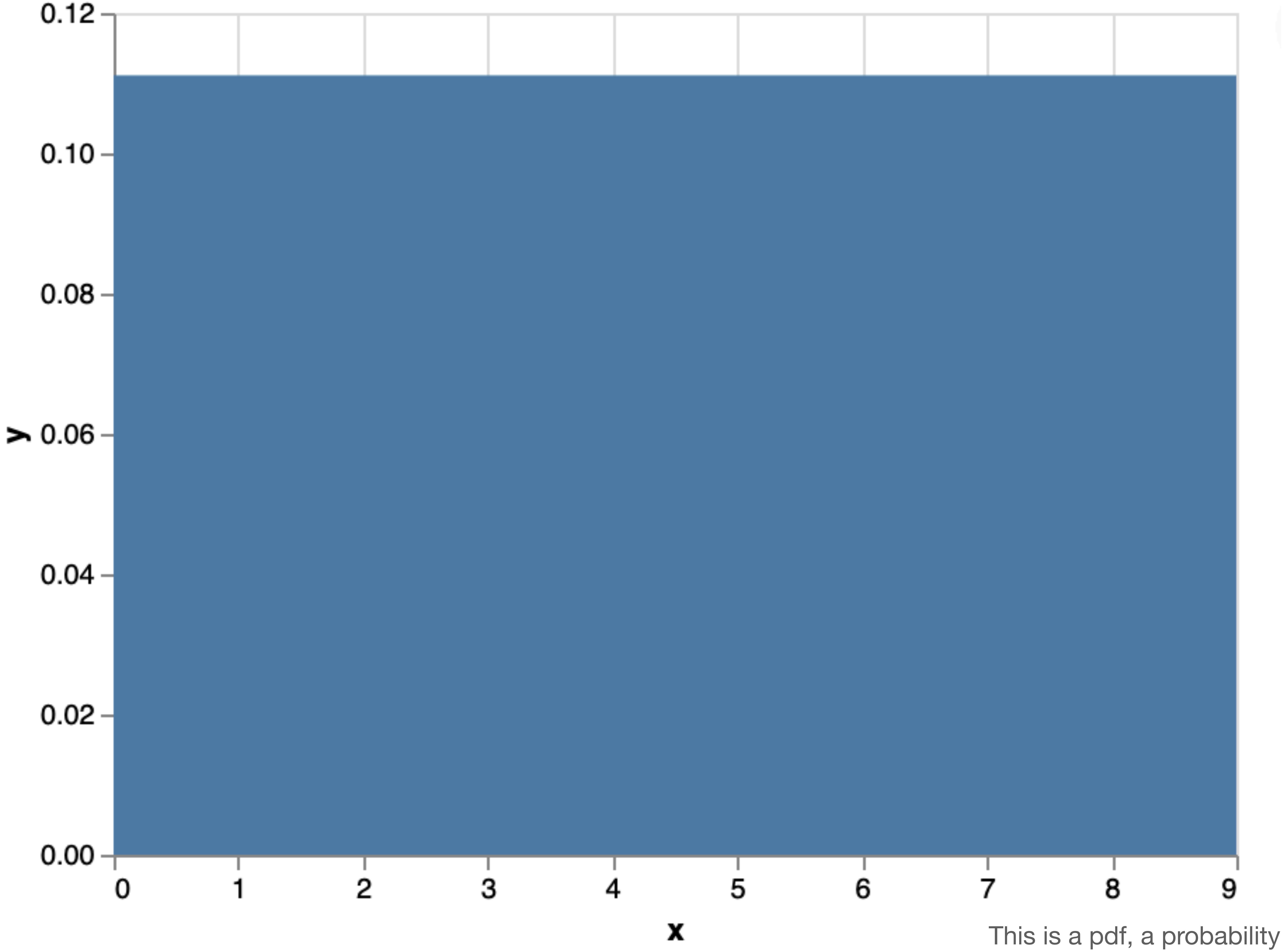
This is a pmf, a probability mass function
It gives you the mass of probability placed across values… and its sums to 1
There are infinitely many others, just shrink one bar and raise others in turn so sum remains 1

# Maximum Entropy for Continuous Outcomes

**Given constraints, what is the most surprise a state of information can hold?**

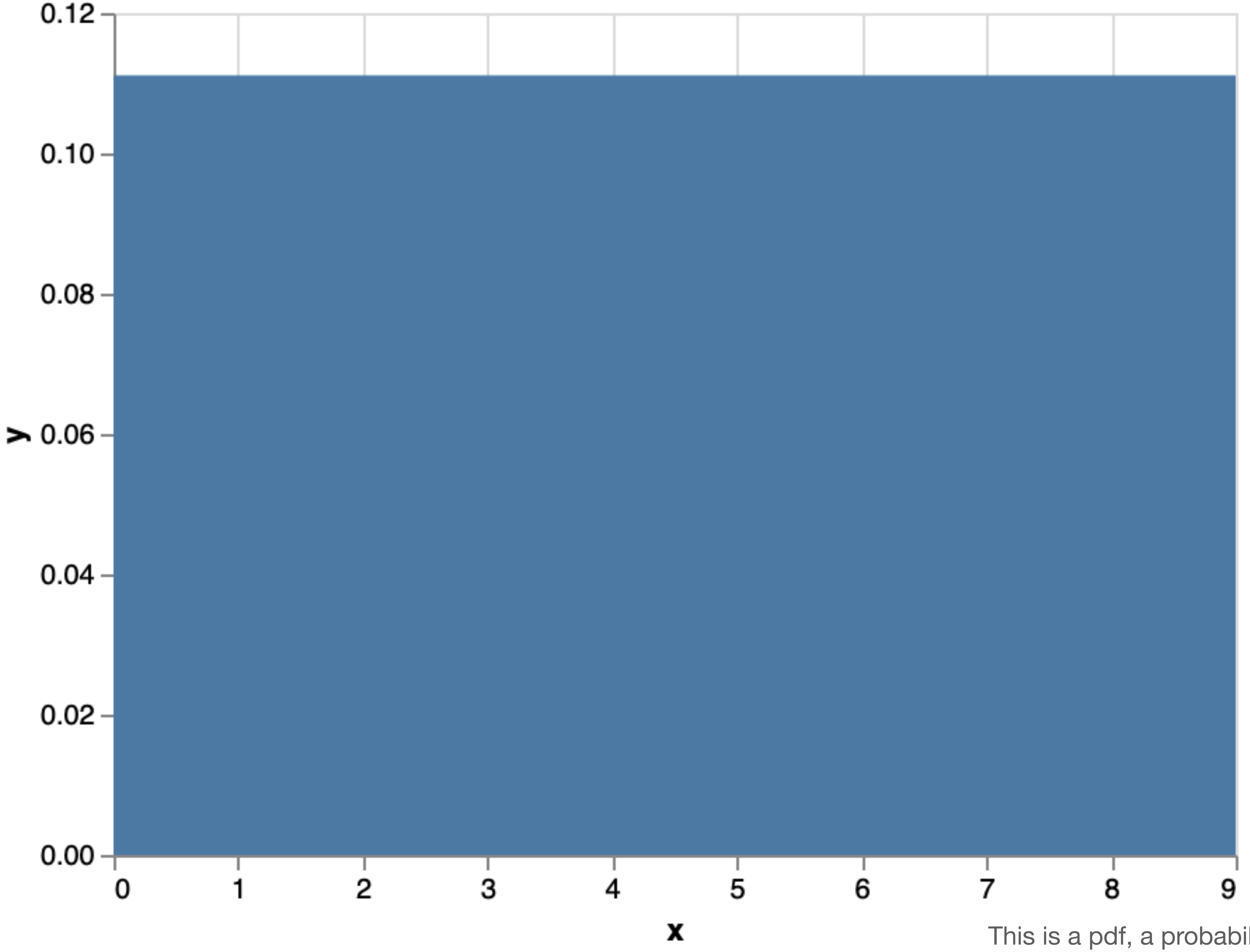
- Continuous outcomes have infinite possible states
- Pick a float between 0 and 9.
  - What is the probability of each possible value where you would be maximally surprised?
  - Continuous entropy is $-\int p(x)(log_2 p(x))dx$
    - Where p(x) is probability of outcome y=x.

# Maximize entropy for continuous values from 0 to 9




This is a pdf, a probability density function
It gives you the density of probability placed across values… and it sums to 1
There are infinitely many others, just change the shape but keep the area the same (so it equals 1)

# Maximize entropy for continuous values from 0 to 9




Uniform distribution over possible values

Why?

Maximizes

$$-\int p(x)(log_2 p(x))dx$$

Not scary… just turn integral into
A summation over really small discrete values

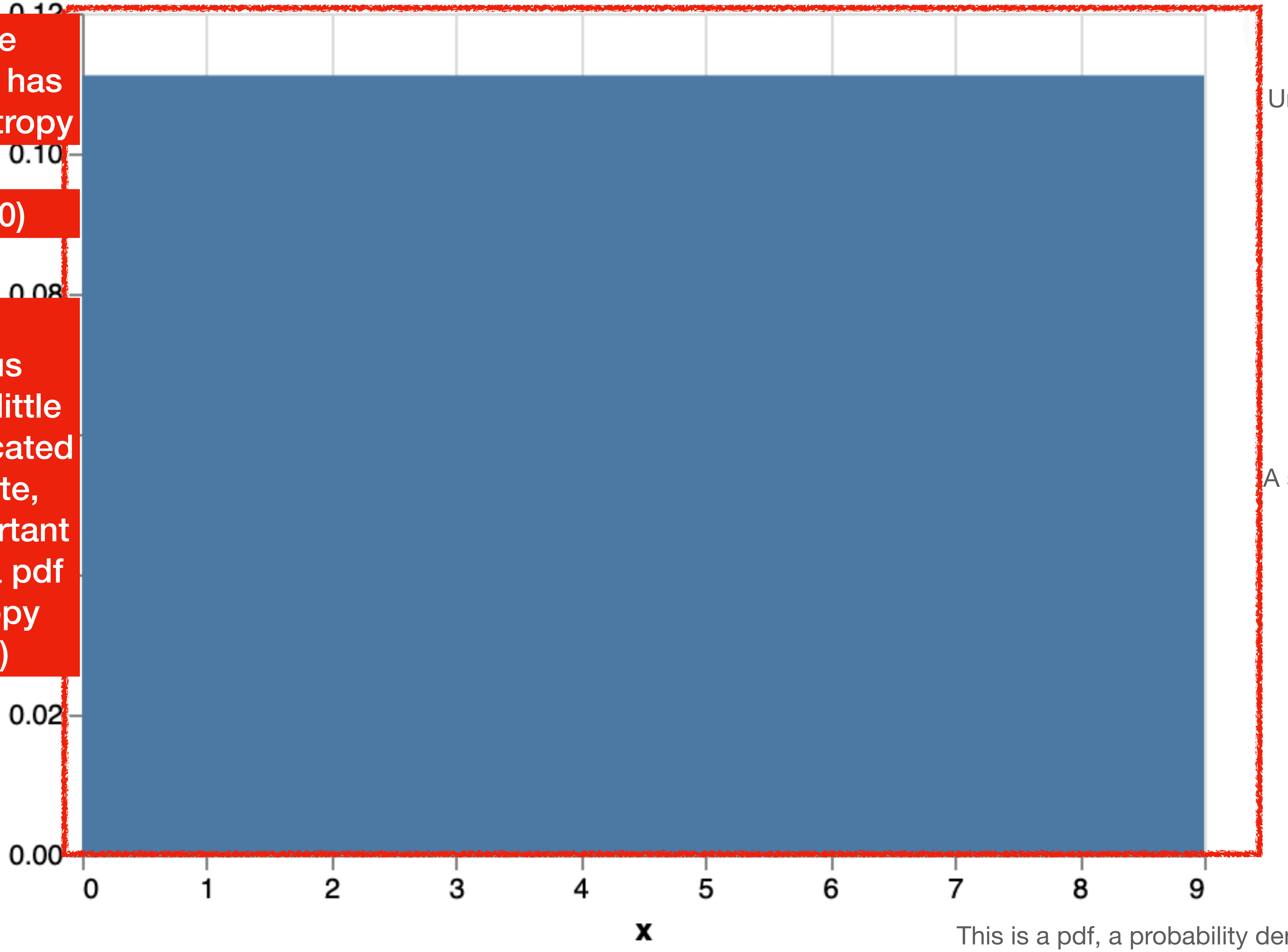
This is a pdf, a probability density function
It gives you the density of probability placed across values… and it sums to 1
There are infinitely many others, just change the shape but keep the area the same (so it equals 1)

# Maximize entropy for continuous values from 0 to 9

This whole distributions has 1 average entropy

np.log2(9-0)

Why?
Continuous entropy is a little more complicated than discrete,
But the important idea is that a pdf holds entropy (surprisal)




Uniform distribution over possible values

Why?

Maximizes

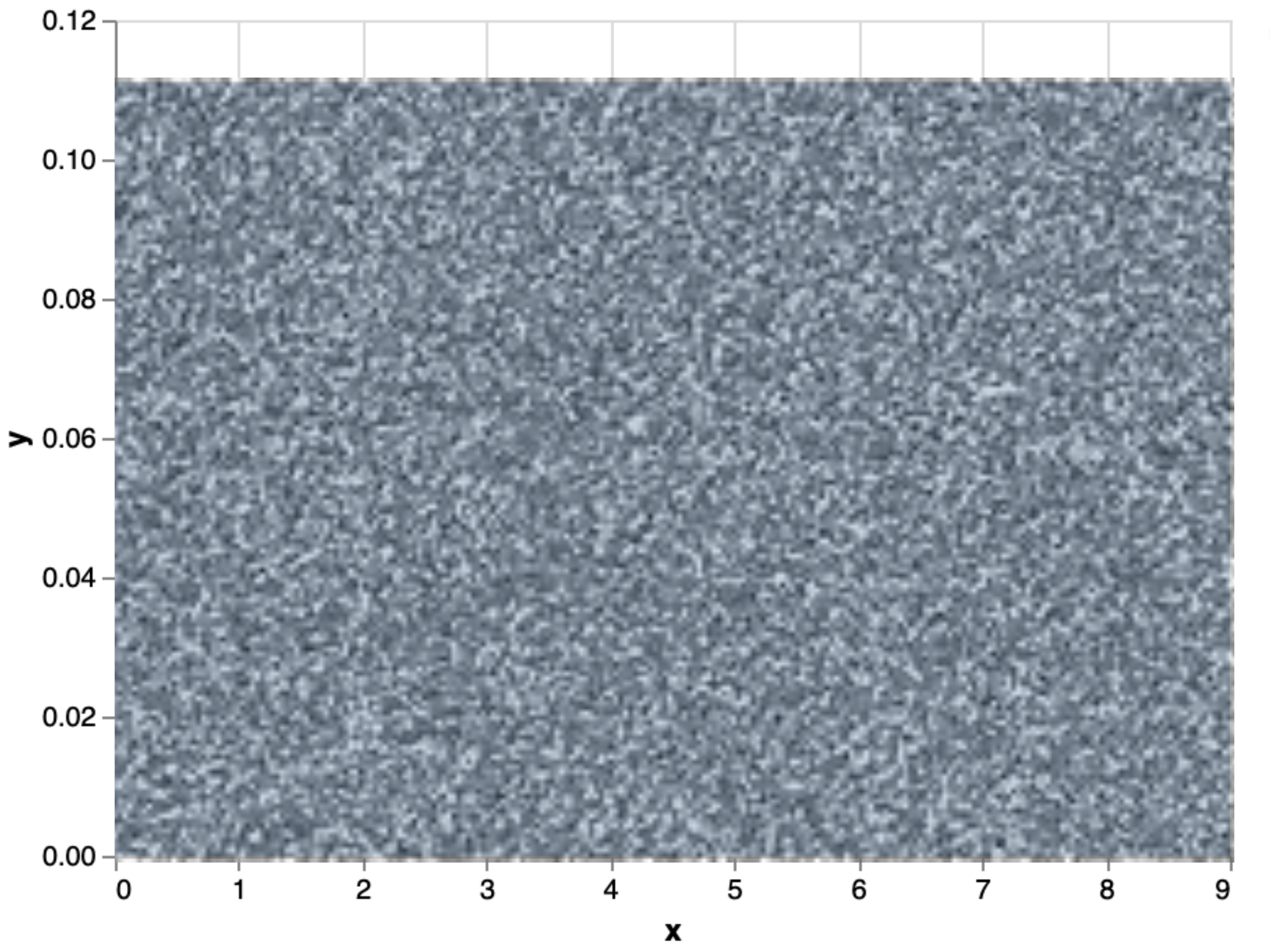
$$-\int p(x)(log_2 p(x))dx$$

Not scary… just turn integral into
A summation over really small discrete values

This is a pdf, a probability density function
It gives you the density of probability placed across values… and it sums to 1
There are infinitely many others, just change the shape but keep the area the same (so it equals 1)

# Bring the noise

## Entropy related to noise



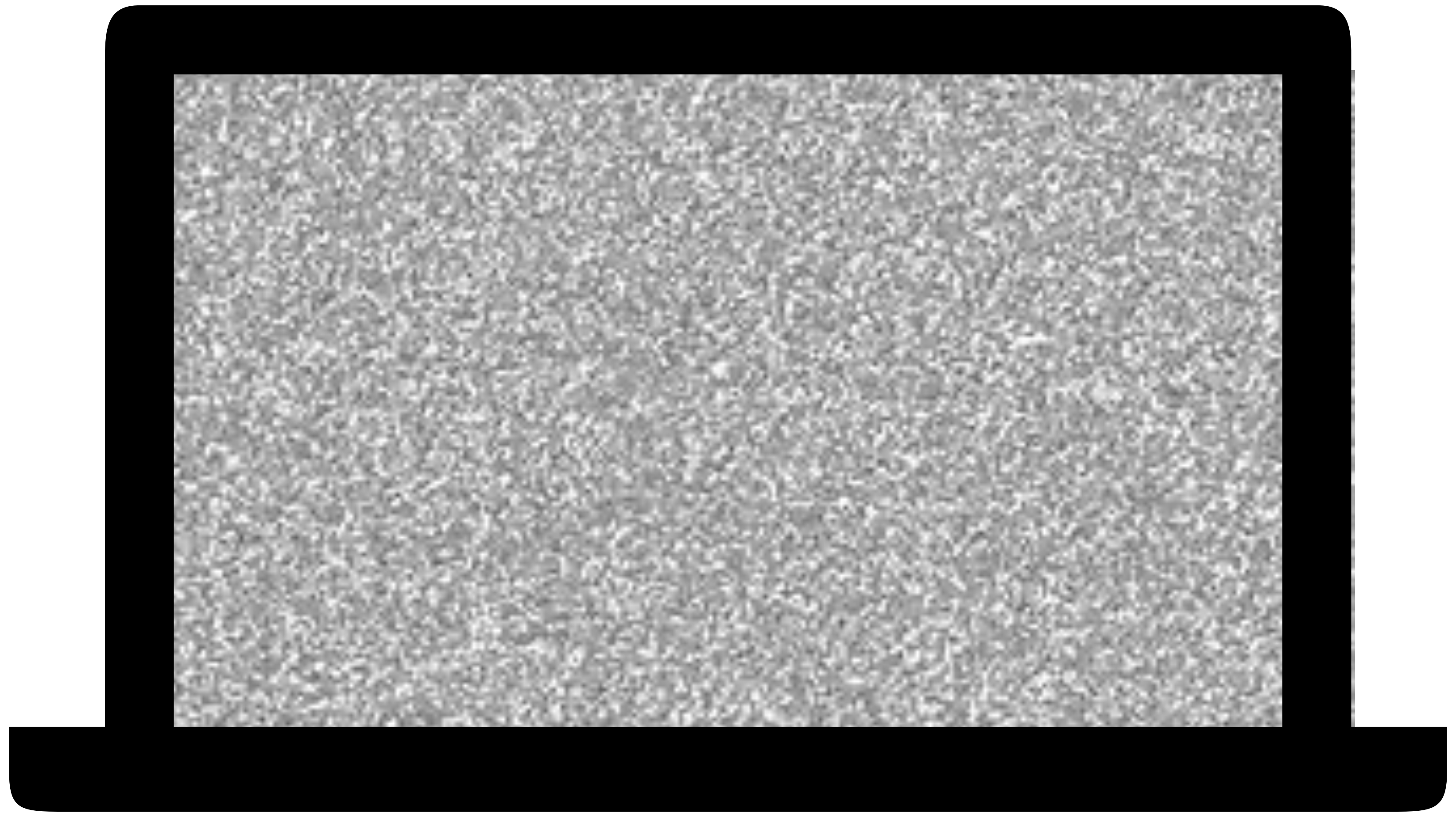

- Uncertainty comes from…
- Randomness
  - From a certain point of view
- We can have different types of noise
  - Patterns with different entropies
- Uniform distribution just has a min and max as a constraint, and then we fill up the entropy to the max between them.

# Other assumptions/constraints —> distributions

- Some probability of success or failure, but you do not know what it is?
  - Bernoulli distribution
- Outcomes are counts of trials, that each have a probability of success?
  - Binomial distribution
- Assume a finite mean and variance, and nothing else
  - Normal distribution
- Assume outcomes are only positive real numbers, and there is a finite mean
  - Exponential distribution

# Computing noise

## We often want noise/randomness



- Generate random numbers
  - To plot a distribution with discrete points
  - Run a simulation that generates data
  - To explore different solutions for optimizers that might get stuck
  - Simulate from a distribution (like from a posterior distribution in Bayes)

# Random and Not random

- Not random — 0011001100110011 0011..
  - Why? Because you have knowledge of what the next and previous number is, there is a pattern (absence of entropy)
  - $Pr(y_t = 1 \mid y_{t-1} = 0, y_{t-2} = 0) = 1$ and $Pr(y_t = 0 \mid y_{t-1} = 1, y_{t-2} = 1) = 1$
- 10100011010010010100011101
  - More random, ~.5 chance 1 follows a 0 or a 1, and vice versa

# But computers are not random

- So we use psuedo-random numbers
  - We want close to a uniform distribution across all real numbers over some interval, like [0,1].
  - But we also want to be able to REPLICATE these random numbers
    - For ourselves and others
    - For that we use a **seed**
      - This is a “starting place” for the generator
        - If you start at the same place, you get the same pseudo-random numbers

# Go to notebook

- Generate replicable pseudo-random numbers with numpy
- Use them to generate different **shapes** of noise/distributions
- Put them together with structure/patterns