PROBABILITY

10.8.2018

RECAP

* choosing the right way to plot something!

* colormaps!

  * sequential, diverging, and qualitative

BERNOULLI

* The Bernoulli distribution is like flipping a coin: the outcome is either heads (1) or tails (0)

  * (but it doesn't have to be a fair coin)

* It has one parameter: q = the probability that the outcome is 1

* X ~ Bernoulli(q) # means that X is a bernoulli random variable with parameter q

BERNOULLI

* To simulate a Bernoulli random variable (RV) we can use **np.random.rand()** and **>**

* For example, to get a random True or False with 50% probability:

* **np.random.rand() > 0.5**

BERNOULLI

* The **expected value (EV)** is (more or less) what the average value should be for an infinitely large sample

* It's defined as a sum of the possible values an RV could take, weighted by the probabilities of those values

* If $X \sim \text{Bernoulli}(q)$, then its expected value $E[X] = q$

WEIGHTED COIN?

* suppose we have a coin and we want to know if it's fair or not (i.e. does $q=0.5$?)

* we flip it 100 times, and it lands on heads 63 times

* is it fair? how do we know?

WEIGHTED COIN?

* we can test this by simulation!

* simulate 10,000 experiments where a fair coin (q=0.5) is flipped 100 times

* we know most values should be around 50

* but how often does a value as extreme as 63 come up?

P-VALUE

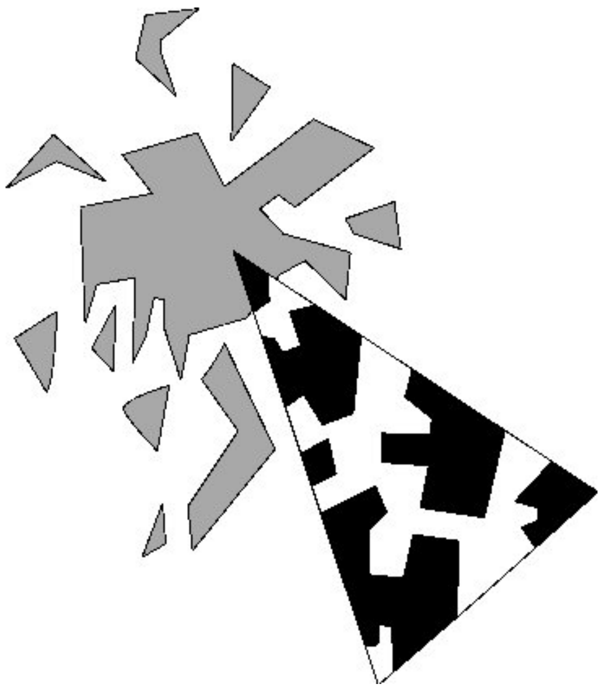
* the **p-value** is the probability of seeing an outcome *at least this extreme* if the “null hypothesis” (i.e. the coin is fair) is true

WEIGHTED NEURON?

* now we have a recording from a neuron in auditory cortex while a stimulus (sound) is being played

* at each time point the output of the neuron is either 1 (because it’s spiking) or 0 (because it’s not)

* we can model this neuron as a *Bernoulli random variable*!

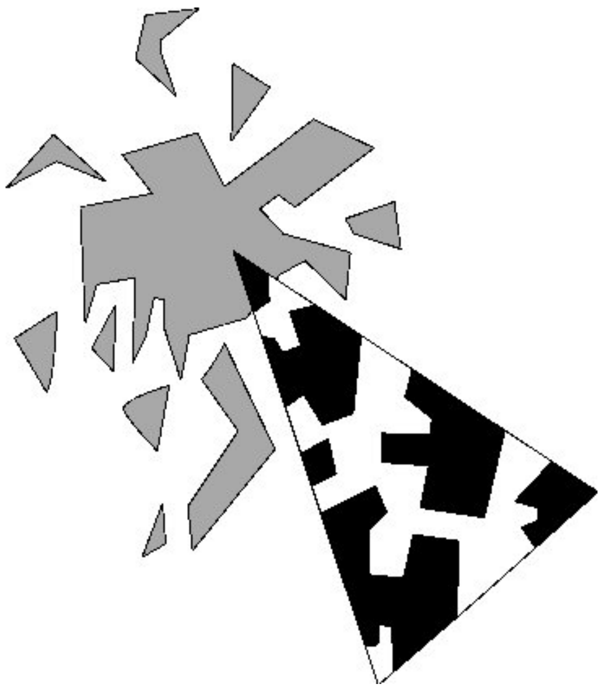


WEIGHTED NEURON?

* the baseline firing rate of this neuron (i.e. while there is no stimulus) is q=0.02114

* during the 360 time points when the stimulus was on, the neuron spiked 25 times

* is this within the range of normal fluctuation, or is the neuron actually responding to this stimulus?



BINOMIAL DISTRIBUTION

* it turns out you don’t need to simulate lots of experiments to test things about bernoulli RVs

* because you can compute it exactly using the binomial distribution!

BINOMIAL DISTRIBUTION

* Y ~ Binomial(n, q) # Y is a Binomial RV with n trials and probability q on each trial

* Y is number of 1's that came up in *n* samples from a bernoulli distribution with parameter *q*

* (this is exactly what we've been doing–generating binomial RVs!)

BINOMIAL DISTRIBUTION

* the exact shape of a binomial distribution is given by this formula:

$$\binom{n}{k} q^k (1-q)^{n-k}$$

n-choose-k: the number of ways to pick k things out of n things

BINOMIAL DISTRIBUTION

* the exact shape of a binomial distribution is given by this formula:

$$\binom{n}{k} q^k (1-q)^{n-k}$$

q^k: the probability of getting exactly k 1's

BINOMIAL DISTRIBUTION

* the exact shape of a binomial distribution is given by this formula:

$$\binom{n}{k} q^k (1-q)^{n-k}$$

(1-q)^(n-k): the probability of getting exactly (n-k) 0’s

BINOMIAL DISTRIBUTION

* the n-choose-k function is available as:

```
>>> import scipy.misc
>>> scipy.misc.comb(n, k)
```

END