BINOMIALS ETC.

10.10.2018

PROBLEM SET 2

* Due Friday 10/12 before class!
* Office hours:
  * (me) today 1:30-3pm
  * (Manu) thursday 4-5:30pm

RECAP

* bernoulli distribution
  * X ~ Bernoulli(q)
* binomial distribution
  * Y ~ Binomial(n, q)
* p-values
  * what is the probability (under the null hypothesis) that you would see something *at least* as extreme as what you did see?

BINOMIAL DISTRIBUTION

* If you flip a weighted coin (where Pr(heads) = q) n times, what's the probability that you get k heads?

BINOMIAL DISTRIBUTION

* Simpler question: if you flip a coin twice, what's the probability of heads both times?
  * What's the probability of one heads and one tails?
  * What's the probability of tails both times?

BINOMIAL DISTRIBUTION

* The probability that two things both happen is the product of the probabilities of each thing happening
  * (assuming that the two things are independent)
* This can be extended to an arbitrary number of things

BINOMIAL DISTRIBUTION

* Back to the original question: if you flip a weighted coin (where Pr(heads) = q) n times, what's the probability that you get k heads?

BINOMIAL COEFFICIENT

* n-choose-k (aka the binomial coefficient) is the number of ways to select k things out of n things

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

BINOMIAL COEFFICIENT

```
* the n-choose-k function is available as:
  >>> import scipy.misc
  >>> scipy.misc.comb(n, k)
```

BINOMIAL DISTRIBUTION

* Let p1 = the probability of flipping a coin k times and getting heads every time

* Let p2 = the probability of flipping a coin $(n-k)$ times and getting tails every time

* Let c = the number of ways to choose k things out of n things

BINOMIAL DISTRIBUTION

* Pr(k heads in n flips) = p1 * p2 * c

BINOMIAL TEST

* (As on monday) If we flipped a coin 100 times and got 63 heads, is it a fair coin?

* Formally: if the coin was fair (q=0.5), what is the probability that we would see a result at least as extreme as 63 in 100 trials?

BINOMIAL TEST

* How do we compute this probability?
  * We can simulate, as we did on monday
  * But, since we know the Binomial distribution we can just compute the probability for each k and sum!

BINOMIAL TEST

* (but in actuality we would always use scipy.stats.binom_test)

MEAN

* as we’ve already seen when talking about numpy, the **mean** of a collection of numbers is the same as the average

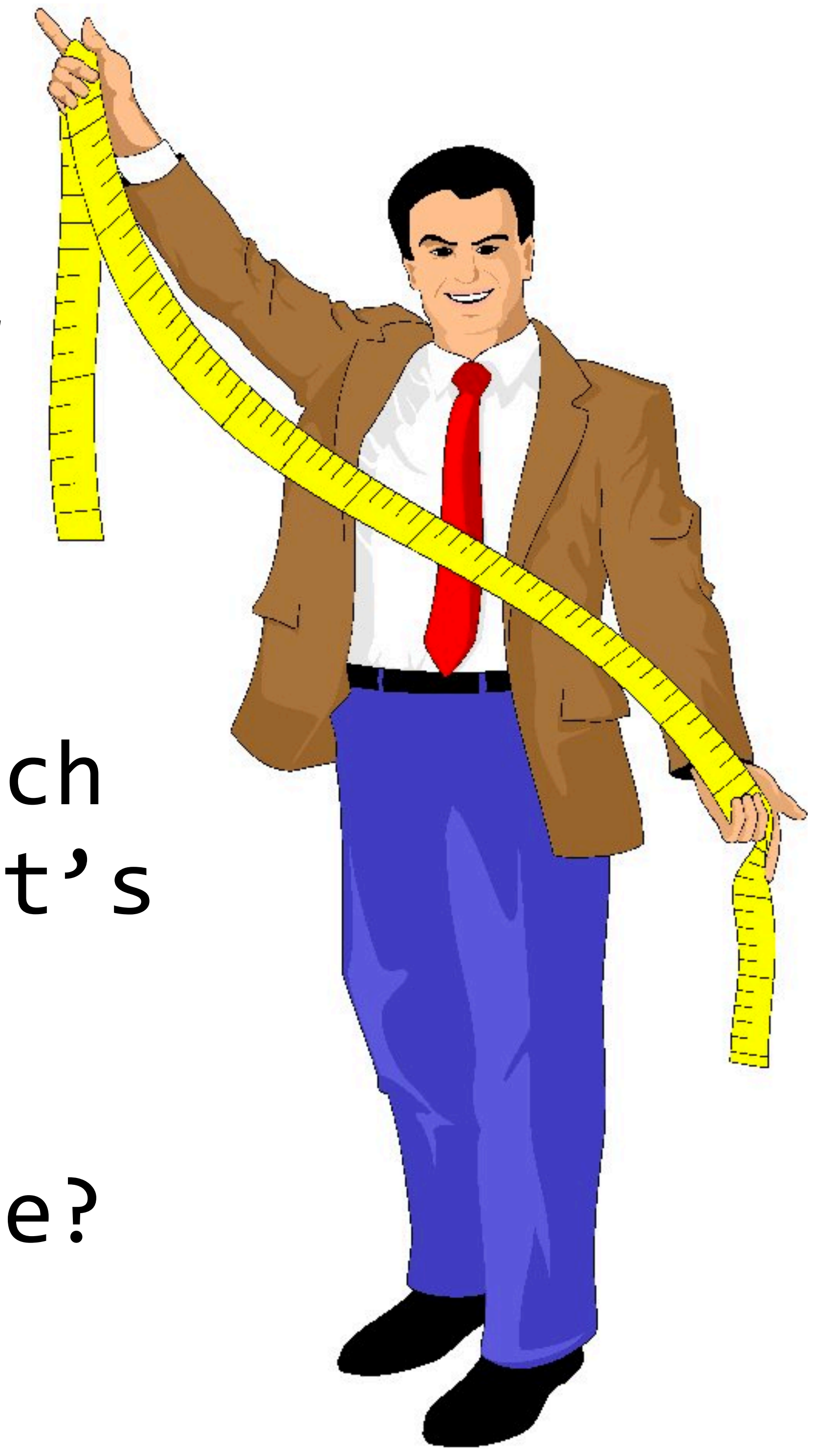
* i.e. mean(arr) = sum(arr) / len(arr)

VARIABILITY

* What if we want to measure how variable the data is around the mean?

* We could do compute how far each data point is from the mean—let’s call this the deviation

* What will the mean deviation be?



VARIABILITY

* The mean deviation is always zero!

* So obviously we can't just average deviations to get a sense of how variable the data is

* One thing we could do is take the mean *squared* deviation

* This is the ***variance*** of the data

VARIABILITY

* Variance can also be obtained using **arr.var()** in numpy

VARIABILITY

* Another useful number is the mean absolute deviation (or square root of the mean squared deviation)

* This is the ***standard deviation***

* Standard deviation can be obtained using **arr.std()** in numpy

END