TESTS &
DESCRIPTIVE STATISTICS

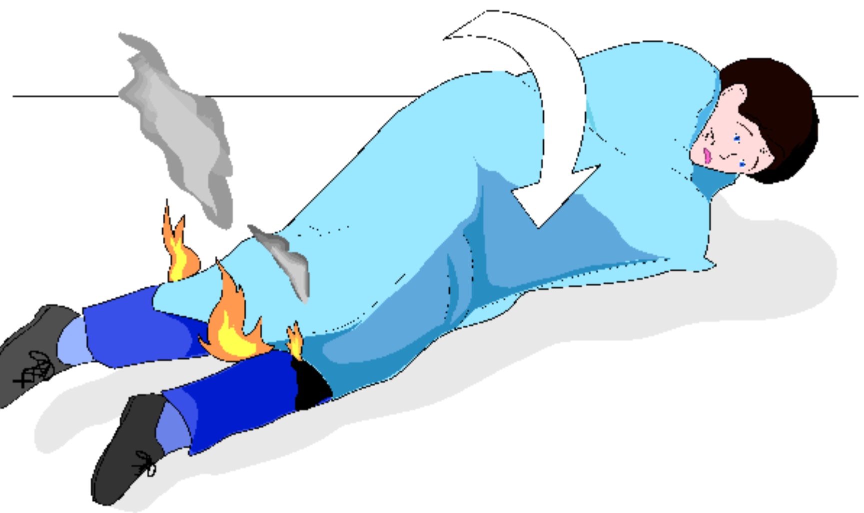
10.2.2020

PROBLEM SET 2

* was due TODAY!

PROBLEM SET 3

* is assigned TODAY! (due in 2 weeks)



RECAP

* bernoulli distribution
  * X ~ Bernoulli(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

* 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 come up in *n* samples from a bernoulli distribution with parameter *q*

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

BINOMIAL WHAT?

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

BINOMIAL WHAT?

* 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 WHAT?

* 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 WHAT?

* 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 WHAT?

* 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 WHAT?

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

BINOMIAL TEST

* (As last time) 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 before
  * But, since we know the Binomial distribution we can just compute the probability for each *k* and sum!

BINOMIAL TEST

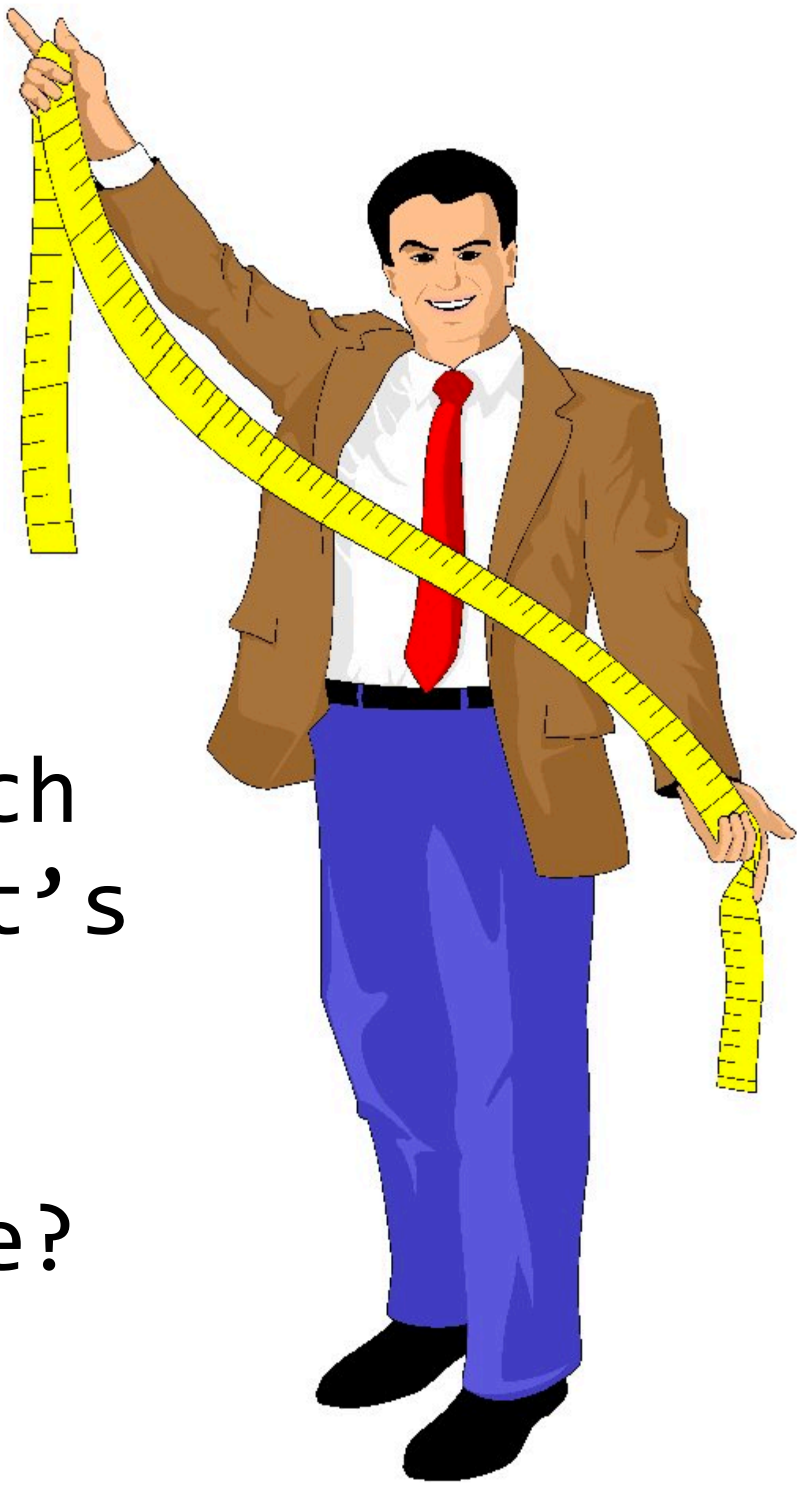
* In reality we would always use **scipy.stats.binom_test**

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 square root of the mean squared deviation

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

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

END