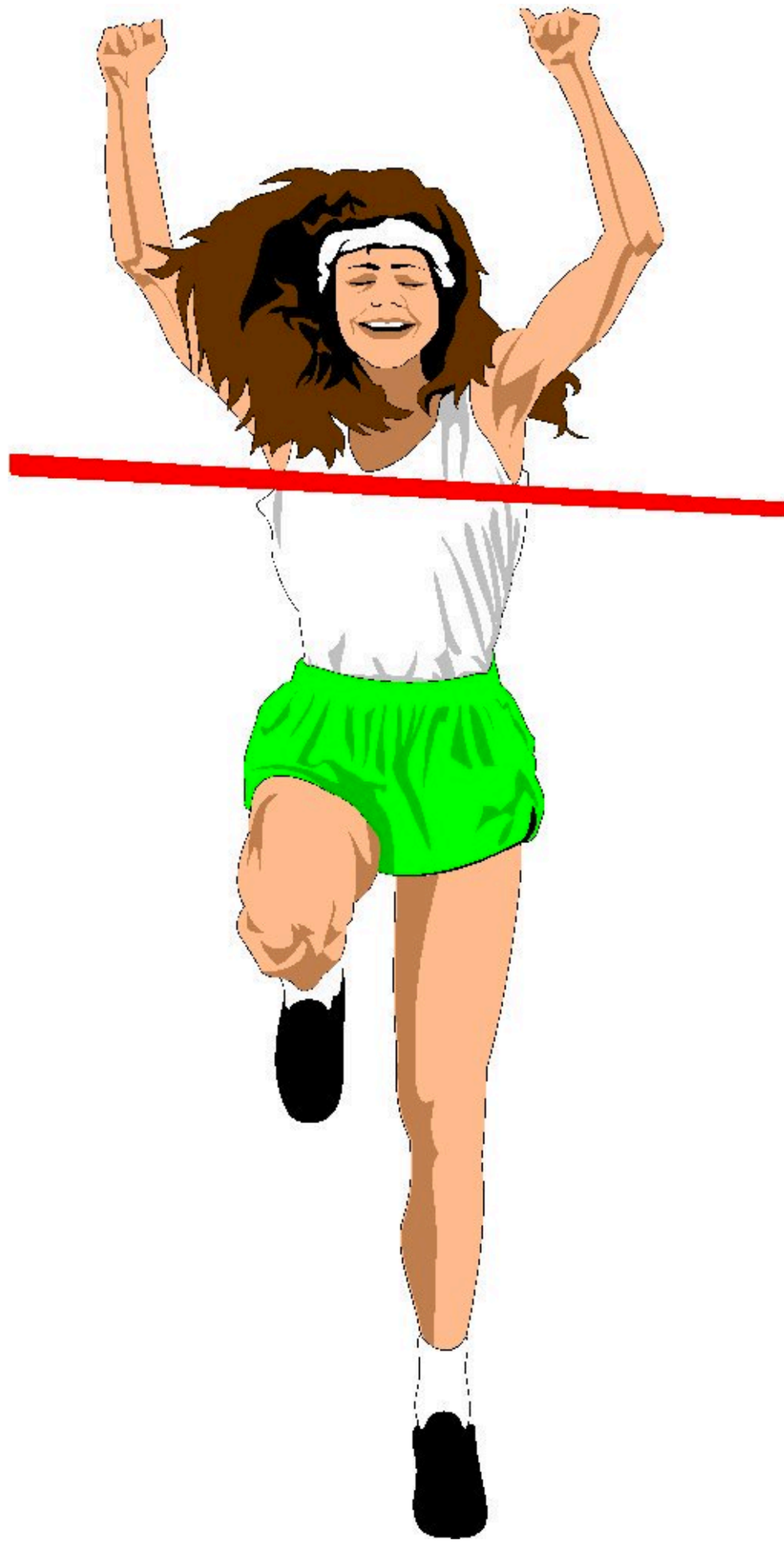
TESTS & DESCRIPTIVE STATISTICS

10.12.2018

PROBLEM SET 2



* Due today!

RECAP

* binomial distribution!

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)

* in numpy you can get the mean two ways:

  * **np.mean(arr)**

  * **arr.mean()**

MEDIAN

* the median is the middle-most element in a dataset (the 50th percentile)

* the median is more robust than the mean (i.e. it is less sensitive to outliers)

* in numpy you can get the median using:

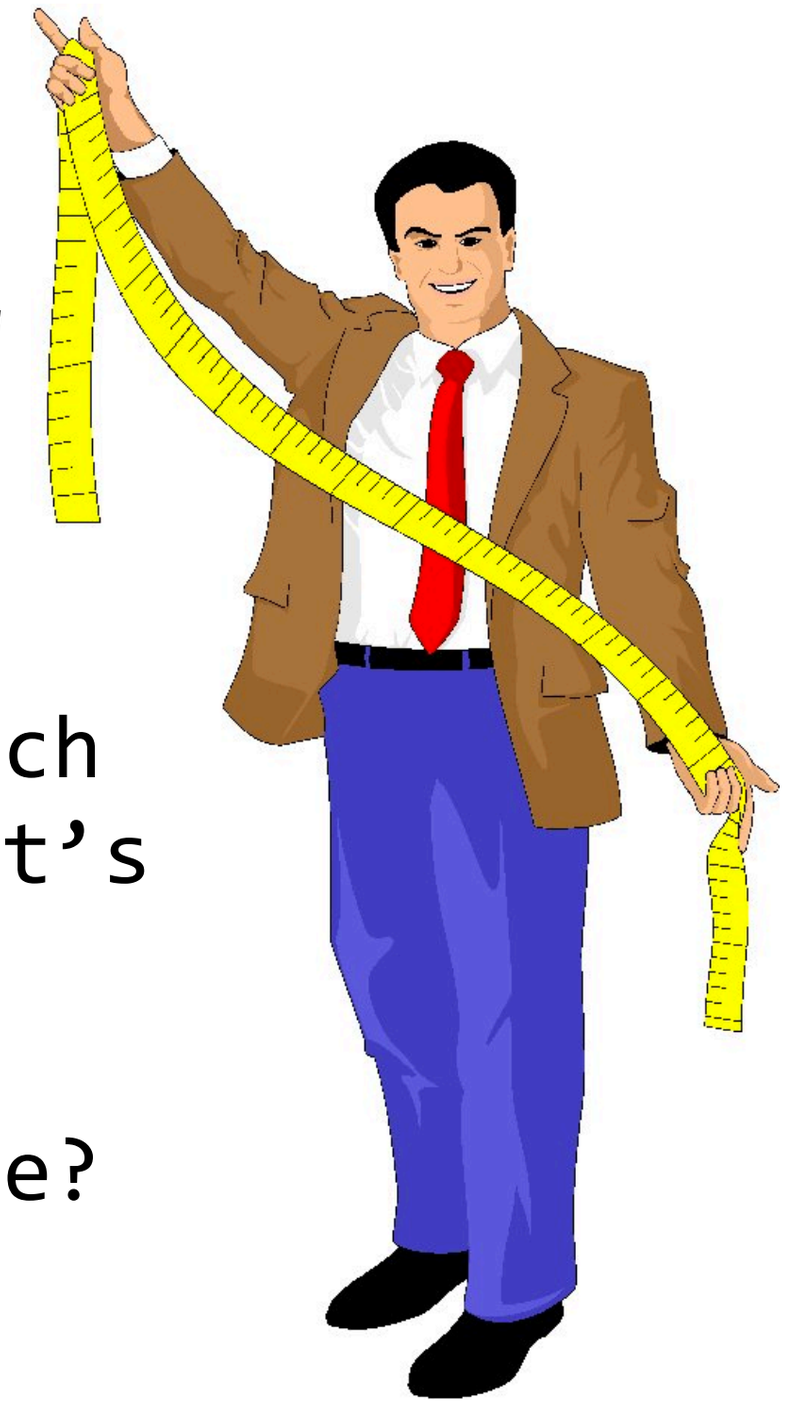
  * **np.median(arr)**

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