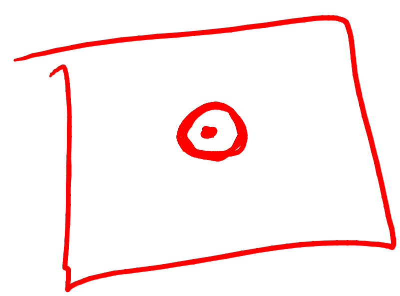
Real Lecture 16: Distributions, Standard deviation , Confidence intervals and levels

Distributions

Assign a probability to sets of possible outcomes of a random variable
(or in our case an experiment)

- For experiments that produce integer or boolean values it's easier to talk about the probability of each discrete value
- For experiments that produce real numbers, the probability of any real number will usually be zero
  - You need to talk about the probability of a neighborhood around a value

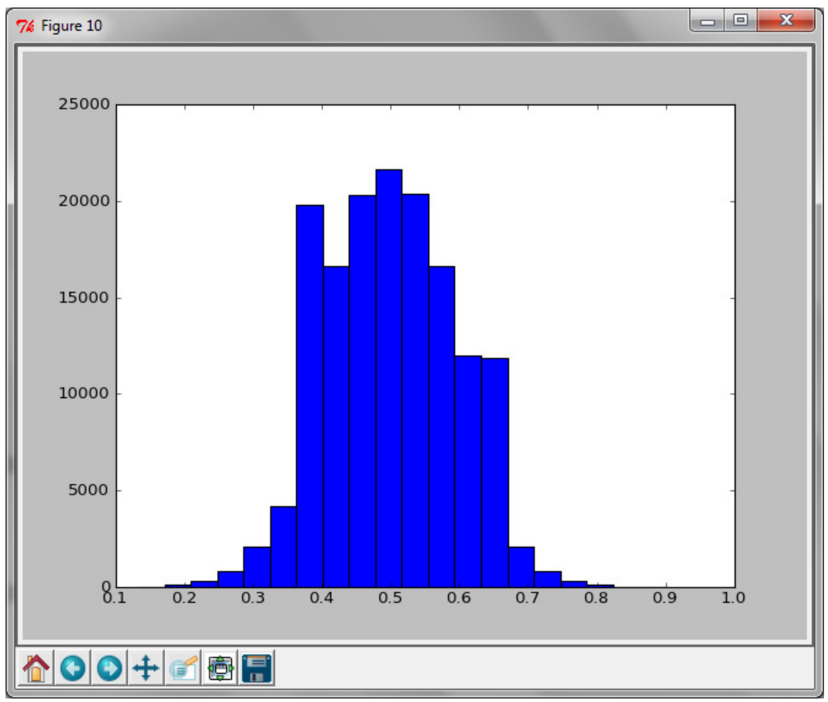


Plotting Histograms

pylab.hist(vals, bins=x)

Example:

- plot of the outcome of the MontyHall Simulation for the random strategy for 150k simulations



Uniform Distribution

That's what random.random() provides

- All values within some domain have equal probability
- All sets of values of the same size have equal probability

Exponential Distributions

Can model time until some random event happens

- assuming the probability of the event is independent of time
- memoryless property
- succinctly described by the half life

Normal Distributions

Completely Specified by two parameters

- Mean and standard deviation

They fall off exponentially

- gives them nice predictive power

Sums of independent variables from uniform distributions give this kind of distribution

Standard deviation

$$\sqrt{E[(E[x]-x)^2]}$$

Provides an estimate of how far values are from the mean

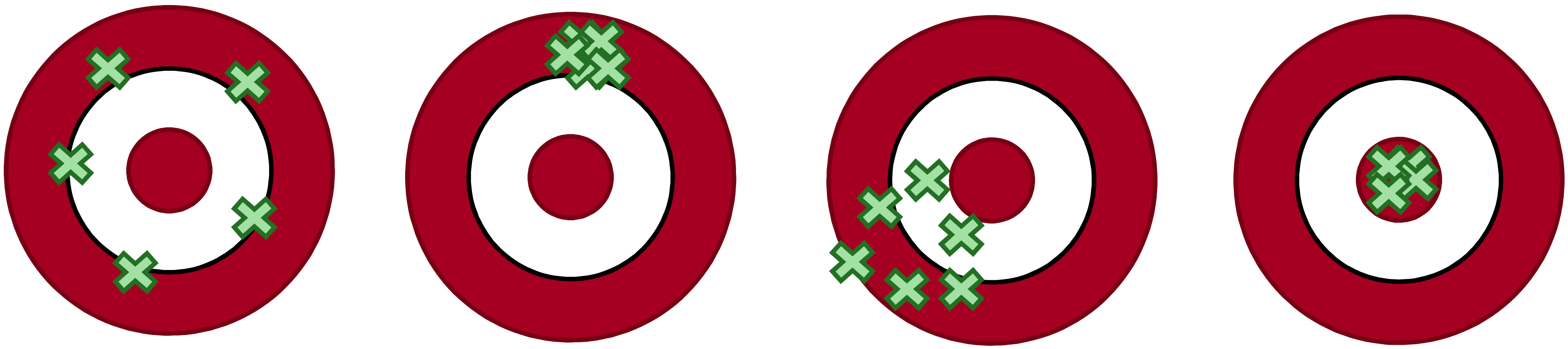
Confidence intervals

What is the probability that a result falls within a given range

- Easy to compute for normal distributions
  - 68% are less than 1 std dev away
  - 95.4% are less than 2 std dev away
  - .2% more than 3 std dev away

Back to Monty Hall example

Accuracy vs. Precision



Accuracy: How far is your mean from the true mean

Precision: How close together are your values