CSE 591: Data Science
Steven Skiena
Stony Brook University

Lecture 10: Statistical Significance

Talking to Statisticians

Statisticians are primarily concerned with whether observations on data are significant.

Data miners are primarily concerned with whether their observations are interesting.

I have never had a satisfying conversation with a statistician, but...

When is an Observation Meaningful?

Computational analysis readily finds patterns and correlations in large data sets.

But when is a pattern significant?

Sufficiently strong correlations on large data sets may seem ``obviously" significant, but often the effects are more subtle.

Medical Statistics

Evaluating the efficacy of drug treatments is a classically difficult problem.

Drug A cured 19 of 34 patients. Drug B cured 14 of 21 patients. Is B better than A?

FDA approval of new drugs rests on such trials/analysis, and can add/subtract billions from the value of drug companies.

Significance and Classification

In building a classifier to distinguish between two classes, it pays to know whether input variables show a real difference among classes.

Is the length distribution of spam different than that of real mail?

Comparing Population Means

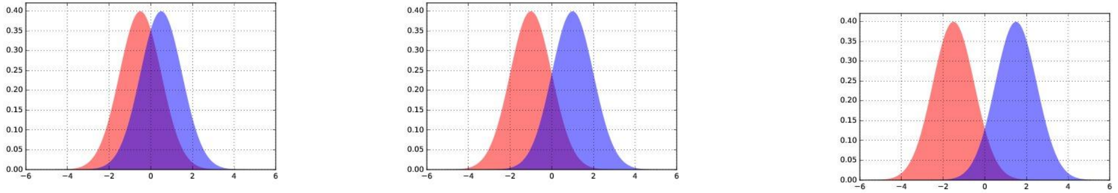
The T-test evaluates whether the population means of two samples are different.

Sample the IQs of 20 men and 20 women. Is one group smarter on average?

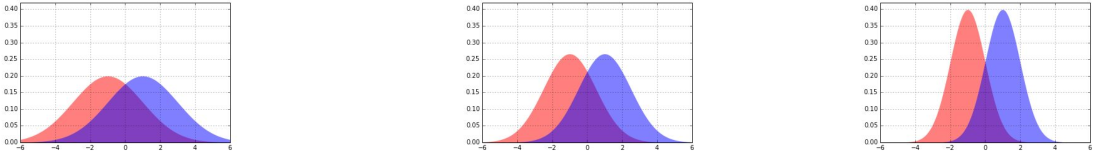
Certainly the sample means will differ, but is this difference significant?

Differences in Distributions

It becomes easier to distinguish two distributions as the means move apart...



... or the variance decreases:



The T-Test

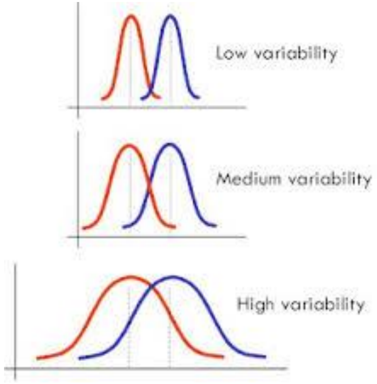
Two means differ significantly if:

- The mean difference is relatively large
- The standard deviations are small enough
- The samples are large enough

Welch’s t-statistic is: $t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{{s_1}^2}{n_1} + \frac{{s_2}^2}{n_2}}}$

where *s*^2 is the sample variance.

Significance is looked up in a table.



Why Significance Tests Can Work

Statistical tests seem particularly opaque (e.g. look up numbers from table), but come from ideas like:

- Probabilities of samples drawn from distributions with given mean and std. dev.
- Bayes theorem converts Pr(data|distribution) to Pr(distribution|data)

The Kolmogorov-Smirnov Test

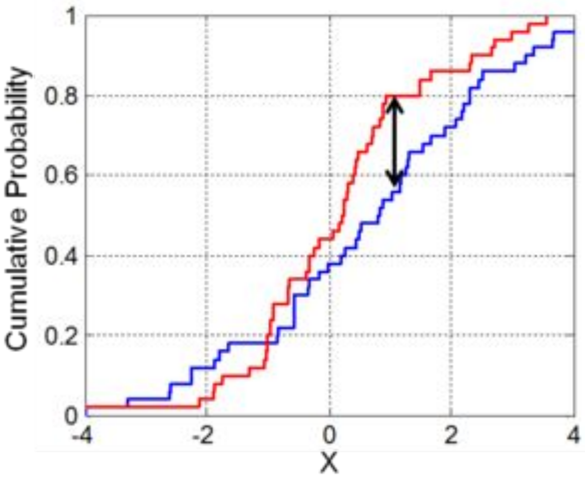
This test measures whether two samples are drawn from same distribution by the maximum difference in their cdf.

The distributions differ if:

$$D_{n,n'} = \sup_x |F_{1,n}(x) - F_{2,n'}(x)|,$$

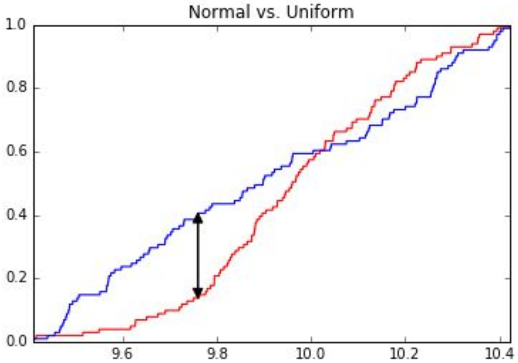
and $D_{n,n'} > c(\alpha)\sqrt{\frac{n+n'}{nn'}}.$

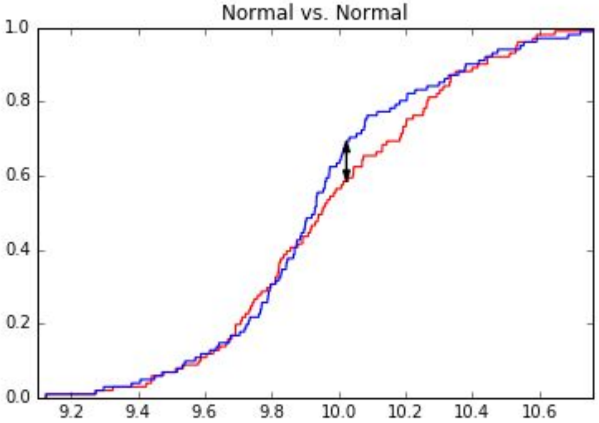
at a significance of alpha.



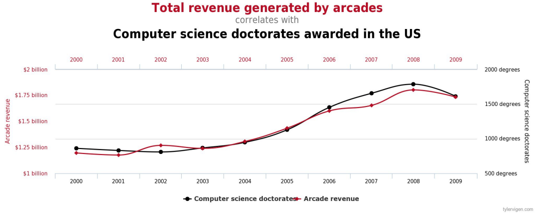
Normality Testing

We can perform the KS-test where one distribution is sampled from the theoretical distribution:





The Bonferroni Correction



Total revenue generated by arcades
correlates with
Computer science doctorates awarded in the US

A statistical significance of 0.05 means there is a probability 1/20 this result came by chance.

Thus fishing expeditions which test millions of hypotheses must be held to higher standards!

In testing *n* hypotheses, one must rise to a level of α / n to be considered significant at the level of *alpha*.

The Significance of Significance

For large enough sample sizes, extremely small differences can register as highly significant.

Significance measures the confidence there is a difference between distributions, not the effect size or importance/magnitude of the difference.

Measures of Effect Size

- *Pearson correlation coefficient:* small effects start at 0.2, medium effects at 0.5, large effects at 0.8
- *Percentage of overlap between distributions:* small effects start at 53%, medium effects at 67%, large effects at 85%
- *Cohen's d* $d = (|\mu - \mu')/\sigma)$: small >0.2, medium > 0.5, large > 0.8

Bootstrapping P-values

Traditional statistical tests evaluate whether two samples came from the same distribution.

Many have subtleties (e.g. one- vs. two-sided tests, distributional assumptions, etc.)

Permutation tests allow a more general, more computationally idiot-proof way to establish significance.

Permutation Tests

If your hypothesis is true, then randomly shuffled data sets should not look like real data.

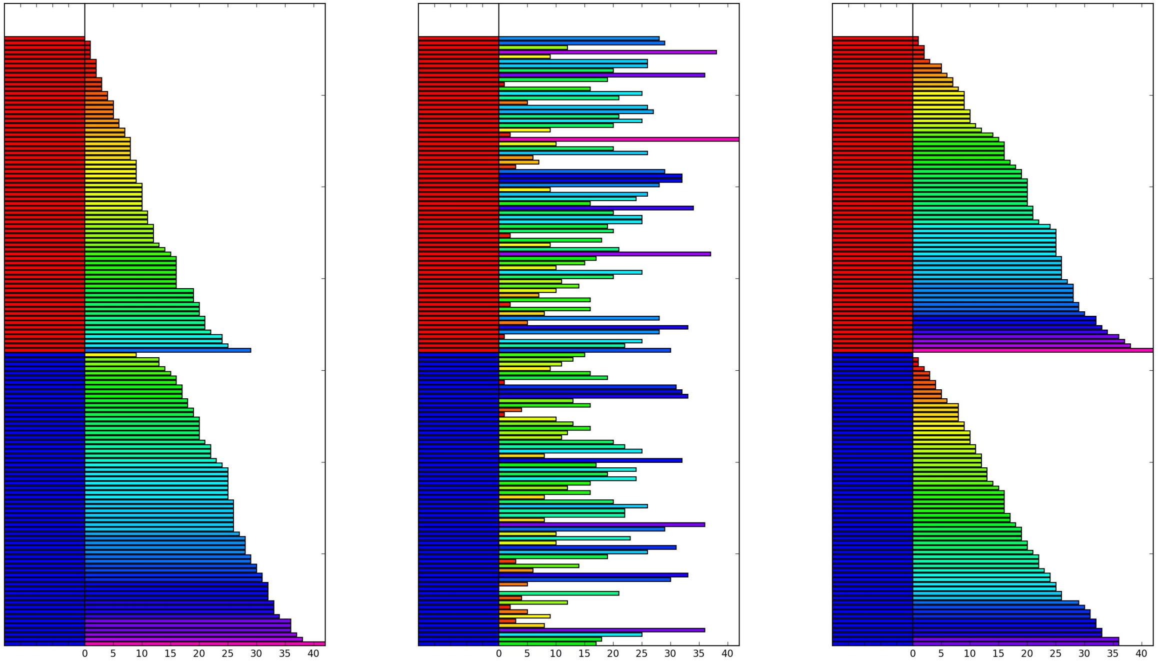
The ranking of the real test statistic among the shuffled test statistics gives a p-value.

You need statistic on your model you believe is interesting, e.g. correlation, std. error, or size.

Permutation Test (Gender Relevant?)

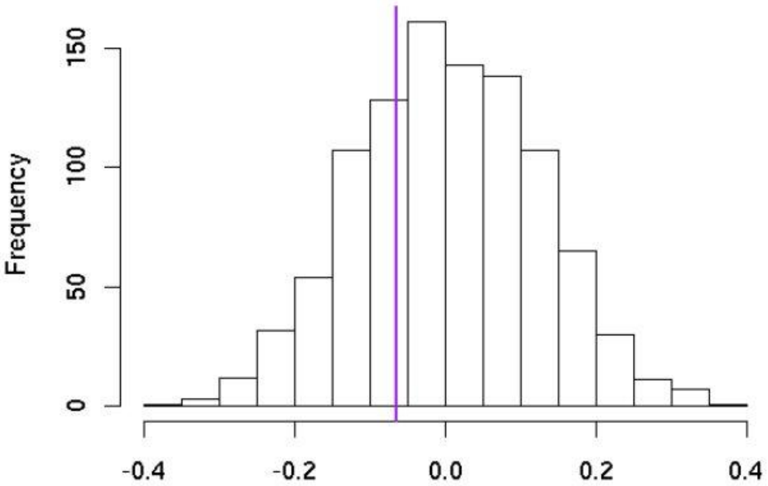
Heights here coded by bar length and color

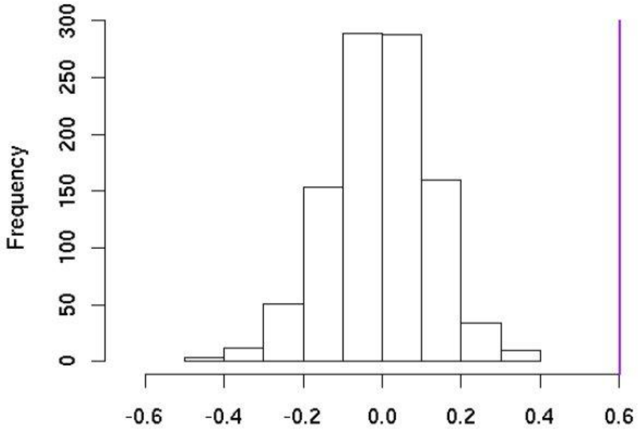
The random permutation (c/r) shows less height difference by gender than the original data (l).



Significance of a Permutation Test

The rank of the real data among the random permutations determines significance:





Performing Permutation Tests

The more permutations you try (at least 1000), the more impressive your significance can be.

Typically we permute the values of fields across records or time-points within a record. Keep comparisons apples-to-apples.

If your model shows decent performance trained on random data, you have a problem.

Permutation Test Caveat!

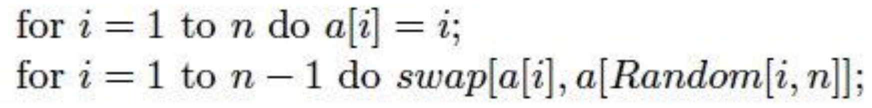
Permutation tests give you the probability of your data given your hypothesis.

This is not the same as the probability of your hypothesis given your data, which is the traditional goal of significance testing.

The real strength of your conclusion does not infinitely increase with more permutations!

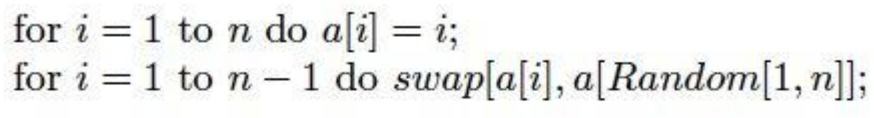
Constructing Random Permutations

Constructing truly random permutations is surprisingly subtle. Which algorithm is right?



```
for $i = 1$ to $n$ do $a[i] = i$;
for $i = 1$ to $n - 1$ do $swap[a[i], a[Random[i, n]]$;
```

or:



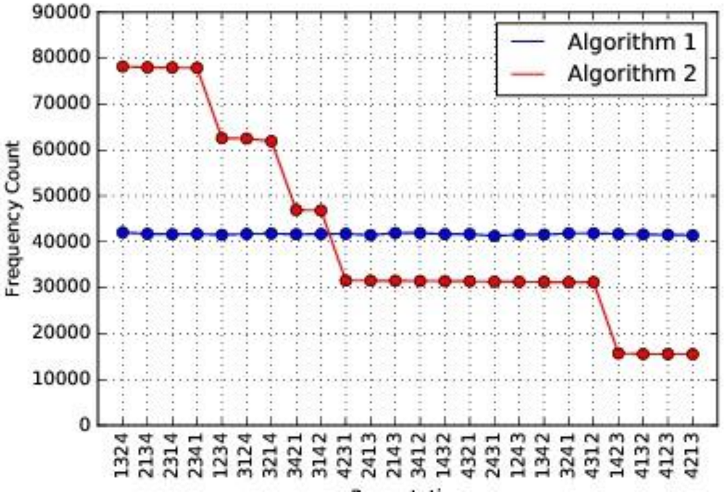
```
for $i = 1$ to $n$ do $a[i] = i$;
for $i = 1$ to $n - 1$ do $swap[a[i], a[Random[1, n]]$;
```

Yes, there is a difference

Experiments constructing 1 million random permutations shows that algorithm 1 is uniform, but algorithm 2 is not.

st. dev. 1 = 166.1

st. dev. 2 = 20,932.9



Why is it Uniform?

The first algorithm picks a random choice for the first position, then leaves it alone and recurs. It generates random permutations.

The second algorithm gives subsequent elements a better chance to end up first. The distribution is not uniform.

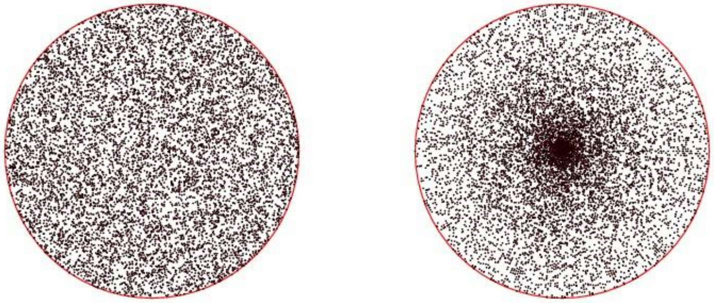
Moral: Random generation can be very subtle.

Sampling from Distributions

A common task is repeatedly drawing random samples from a given probability distribution.

Give me an algorithm to draw uniformly random points from a circle:

The problem is more subtle than it looks.



Drawing Points from a Circle

Each point in a circle is described by a radius *r* and angle *a*, but drawing them uniformly at random picks too many points near the center.

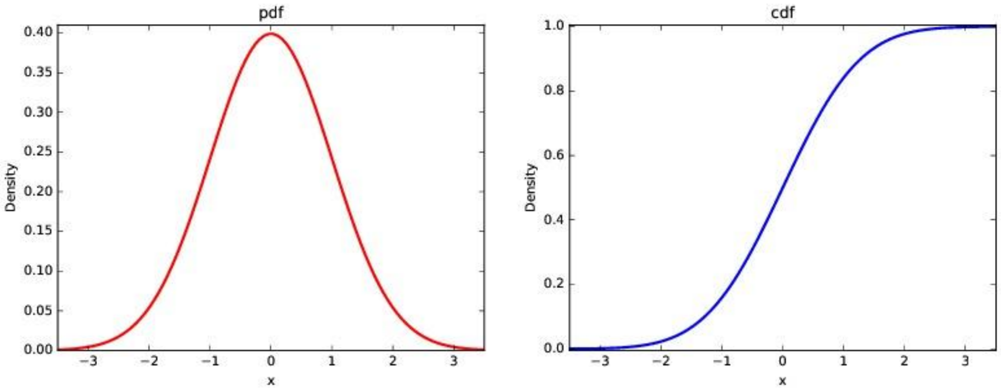
The inner half circle is smaller than the outer half!

Independently sampling *x* and *y* give points uniform in the box, so discarding those outside the circle leaves a uniform distribution.

Sampling in One Dimension

To sample from any probability distribution, convert it to its cumulative distribution (cdf).

Selecting a probability *p* in *[0,1]* now maps to a value in the cdf:



Dimaggio’s Hitting Streak

One of baseball’s most amazing records is Joe Dimaggio’s 56-game hitting streak.

But how unusual is such a long streak in the context of his career?

He played 1736 games, with 2214 hits in 6821 at bats.

Thus he got a hit in roughly 79% of his games.

Monte Carlo Simulation

We can use random numbers to simulate when he got hits in over a synthetic “career”, and compute the length of the longest streak.

After simulating 100,000 Dimaggio career’s, we get a frequency distribution of longest streaks.

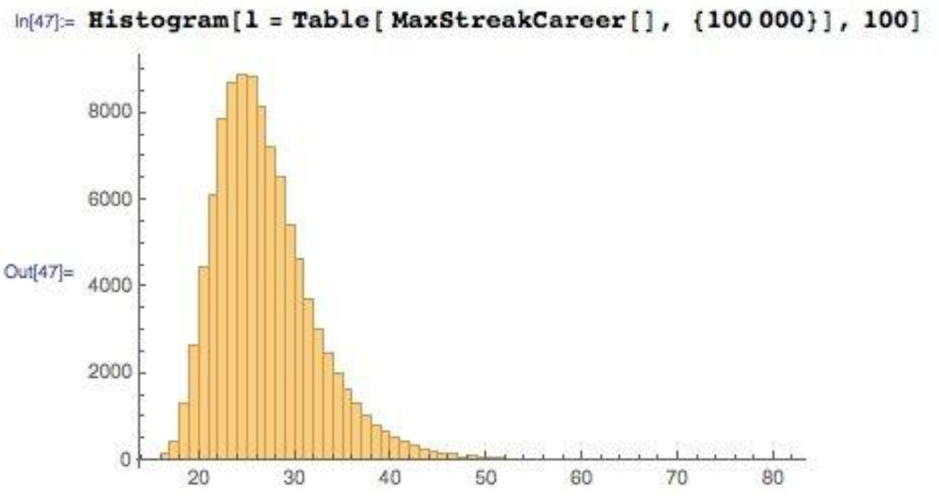
Simulation Results

In only 44/100000 simulated careers (1/2272) did he have a streak of at least 56 games.

Thus the length is quite out of line with what is expected from him, though he hit in 61 straight games in the minors.

The second longest streak of any major league hitter is only 44 games, so it is out of line with everyone else as well.

Closed-form results could presumably follow from analyzing a Poisson distribution, but this requires more skill.



```
In[47]:= Histogram[l = Table[MaxStreakCareer[], {100 000}], 100]
```