Statistical Analysis of Data

- Given a set of measurements of a value, how certain can we be of the value?
- Given a set of measurements of two values, how certain can we be that the values are different?
- Given a measured outcome and several condition or treatment values, how can we remove the effect of unwanted conditions or treatments on the outcome?

Measuring CPU Time

I made the 37 measurements of the CPU time required to compute

$$\binom{10000}{500}$$

in Common Lisp on `darwin.cs.orst.edu`.

0.27	0.25	0.23	0.24	0.26	0.24	0.26	0.25	0.24	0.25
0.25	0.24	0.25	0.24	0.25	0.26	0.24	0.25	0.25	0.25
0.25	0.25	0.24	0.25	0.24	0.25	0.25	0.24	0.25	0.25
0.24	0.25	0.24	0.24	0.25	0.25	0.26			

What is the true CPU cost of this computation?

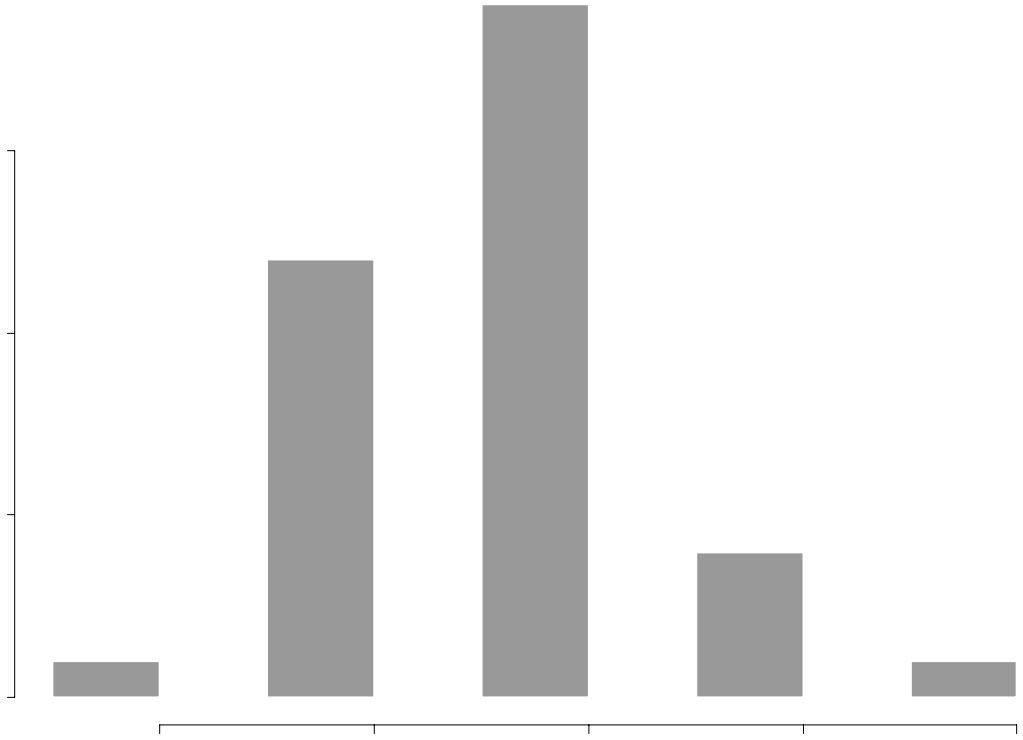
Before doing any calculations with the data,

Always Visualize Your Data

Histogram

Using Splus (on ada), we can issue the commands

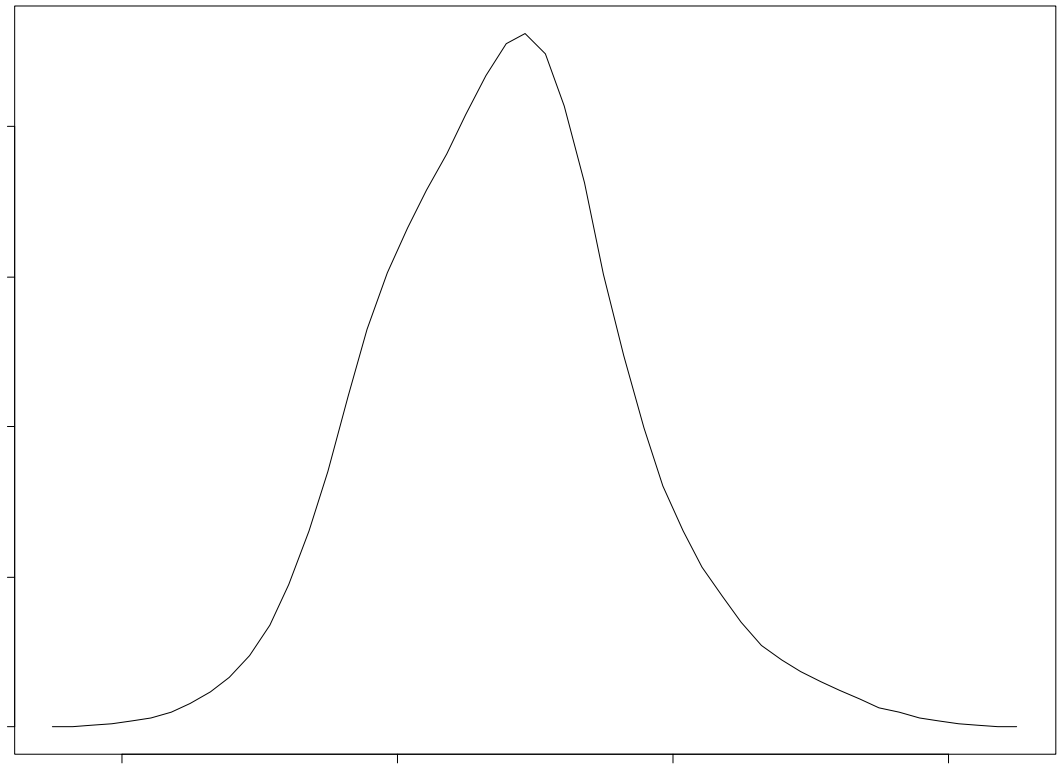
```
% Splus
> comb <- read.table("comb.data")
> hist(comb$V1)
```



Kernel Density Estimate

We can also construct a kernel density estimate. A kernel density estimate places a small normal distribution (the "kernel") at each observed data point and sums them up.

```
> plot(density(comb$V1, width=0.02),type="l")
```



Sample Mean

Based on this visualization, it is reasonable to compute the mean of this distribution:

$$\text{mean} = \overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Mean = 0.248

But how confident can we be that this is the true value? We would like to have a **confidence interval** that would tell us the following:

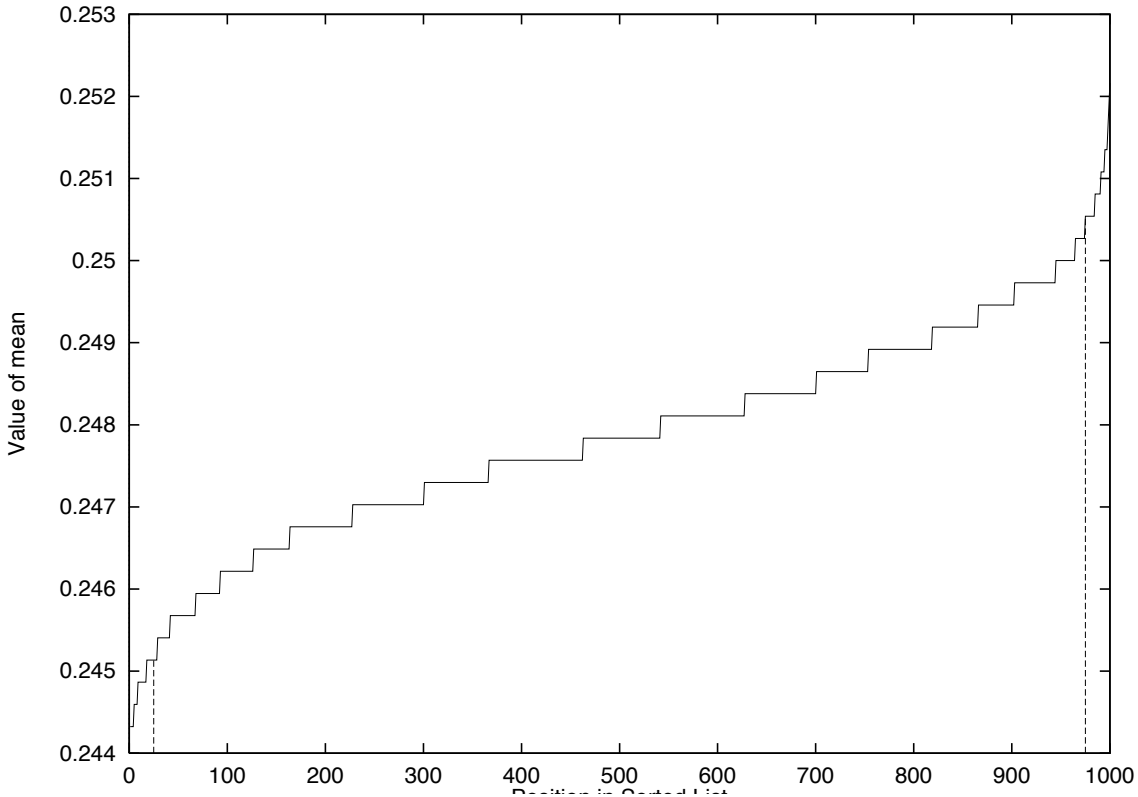
> If we drew random samples of size 37 and took the mean, 95% of the time, the mean would lie between a *lower bound* and an *upper bound*.

Confidence Intervals Via Resampling

Using a computer, we can simulate this. We draw 1000 random subsamples (with replacement) from our original 37 points and compute the mean. Then we sort these means and choose the 26th and 975th values as our lower and upper bounds.

Results: In 950 trials (out of 1000),

$$0.2451 \leq \overline{x} \leq 0.2505$$



Confidence Intervals Via Distributional Theory

If we plot a histogram of the 1000 bootstrap trials, we see that it is very nearly normally distributed. This is called the **sampling distribution of the mean**. The **Central Limit Theorem** says that the sampling distribution of the mean is normally distributed.

The normal distribution has two parameters,

- the *mean* (denoted μ)
- the *standard deviation* (denoted σ).

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

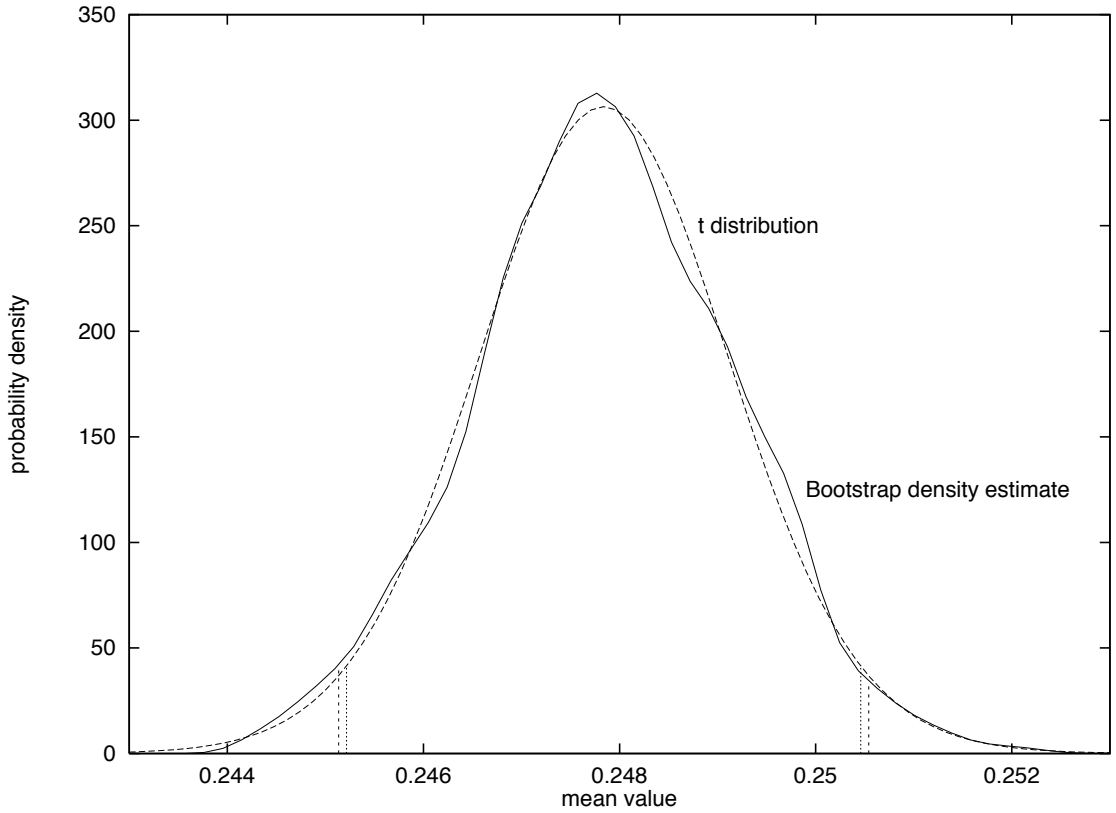
If the original CPU times were distributed with mean μ and standard deviation σ, then the *means* will be distributed with mean μ and standard deviation $\sigma/\sqrt{n}$. (Here $n = 37$.) Unfortunately, we must know the true standard deviation of the CPU times in order to apply the central limit theorem. We don't know this.

The Sample Standard Deviation

$$\text{standard deviation} = s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}$$

The t distribution

Instead of the normal distribution, we can use the t distribution. The t distribution has three parameters: the mean (μ), the standard deviation (σ), and the degrees of freedom ($d.f. = n - 1$).



The 95% confidence limits are slightly tighter according to the central limit theorem using the t distribution.

Distributional confidence intervals

A 95% confidence interval for the mean can be computed via the t distribution as follows:

Let $\overline{x}$ be the sample mean.
Let s be the sample standard deviation.
Let n be the sample size.
Let $t_{0.025}(n-1)$ be the value of t with $n-1$ degrees of freedom such that the probability that $x < t_{0.025}(n-1)$ is 0.975.

Then,

$$\overline{x} - t_{0.025}(n-1)s/\sqrt{n} \leq \mu \leq \overline{x} + t_{0.025}(n-1)s/\sqrt{n}$$

Where μ is the true mean of the CPU times.

The t values can be looked up in a table, or you can use Splus:

```
> qt(0.975,36)
[1] 2.028094
```

Bootstrapping on the Median

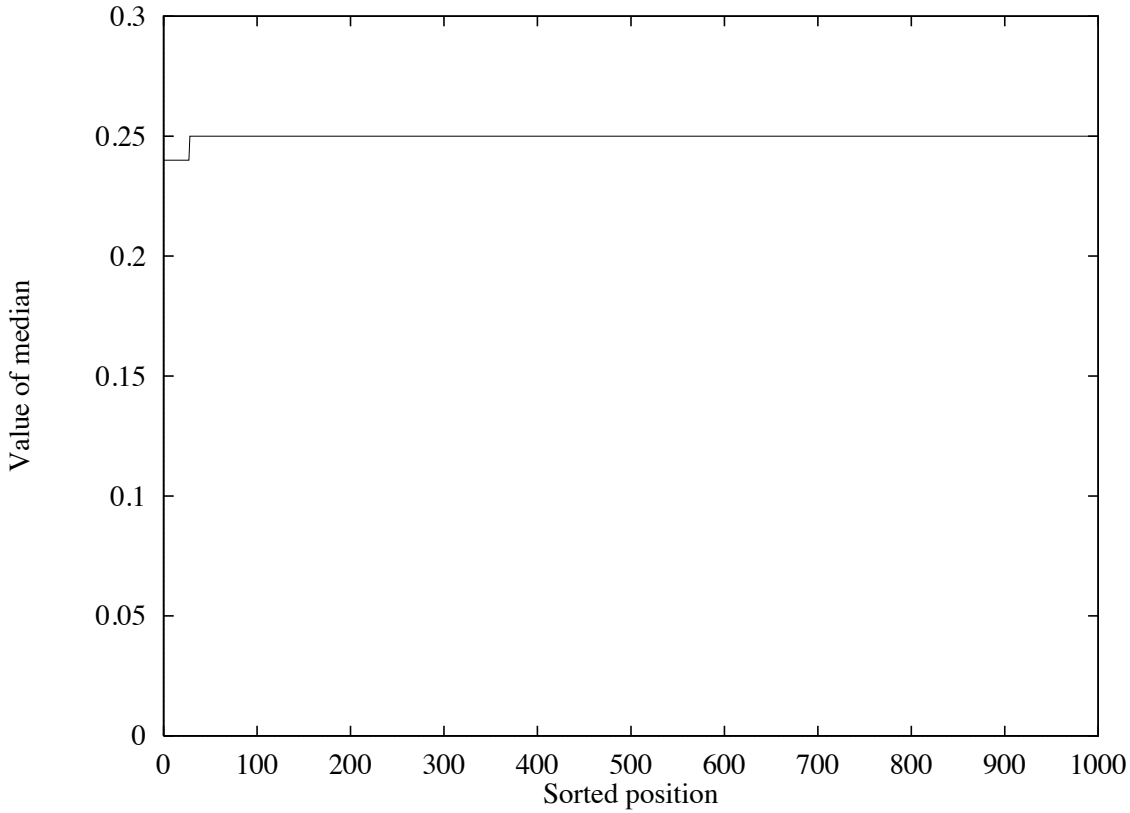
Suppose our goal was to measure the *median* CPU time required for this computation, rather than the average.

We would like to know that 95% of the time, the observed median is within some *bound* of the true median.

While distribution theory can't help us here, we can still apply the bootstrap method:

> Choose 1000 random samples (with replacement) of size 37 from our original 37 points. Take the median value of each sample. Sort and take the value at the 25th and 975th positions.

Bootstrap Median Value



When the Bootstrap Doesn't Work Well

The bootstrap is good for the mean, the median, and other statistics involving the "middle" of a distribution. The bootstrap is not good for estimating the minimum, the maximum, or other statistics involving the "tails" of the distribution.

Measuring Number of Occurrences of Events

In many CS experiments, we count the number of events that occur in n trials. For example, in machine learning, suppose we constructed a decision tree and then evaluated it on a test set of 100 examples and observed 88 correct classifications. We would report the *proportion of correctly classified* test examples as 0.88.

But how uncertain is this quantity? How much might it vary due to the random choice of the test set?

We will say $\hat{\theta} = 0.88$, where θ is the true proportion of correct classifications that our decision tree would make (on an infinite test set).

A Bootstrap Confidence Interval

We can again perform a bootstrapping experiment. Let n be number of test examples.

Repeat 1000 trials:

Draw a random sample of size n with replacement from the test set.

Measure p_i = the proportion correctly classified by the decision tree.

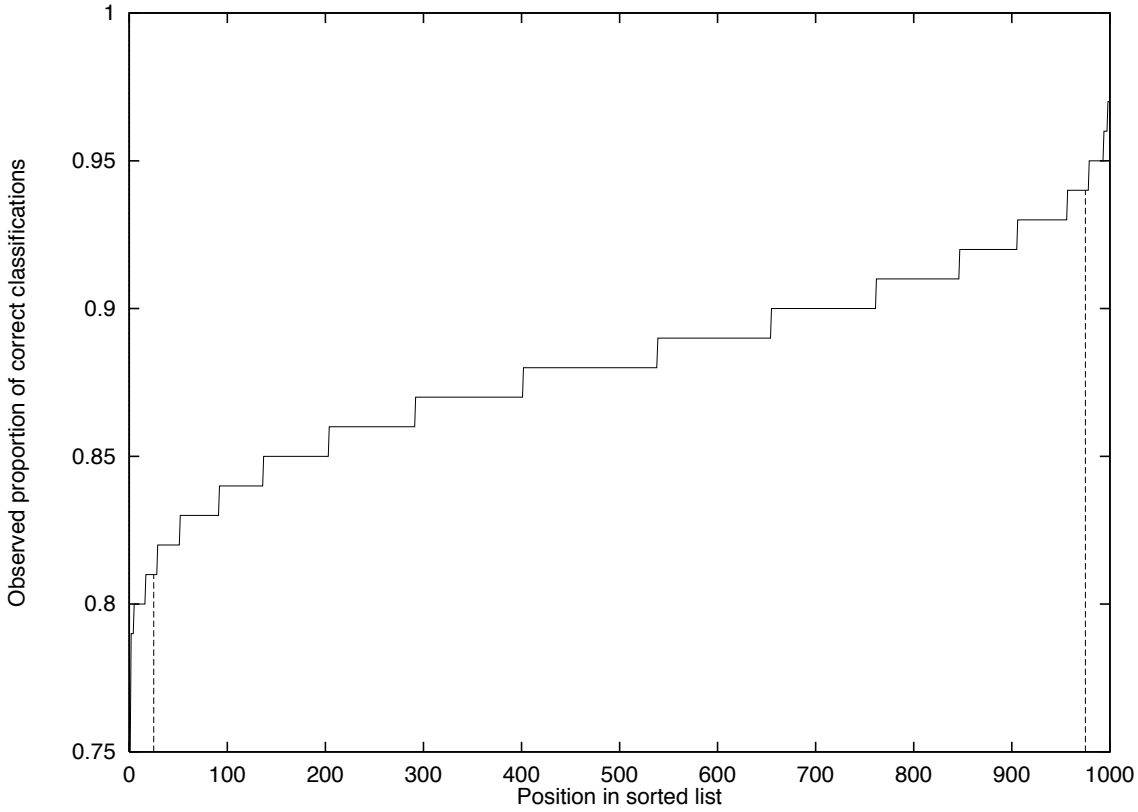
Sort the p_i in increasing order.

Choose lb and ub to be the 26th and 975th elements.

Then, we would say in 1000 trials, the probability is 0.95 that we would observe $lb \leq \hat{\theta} \leq ub$.

Results: $0.81 \leq \hat{\theta} \leq 0.94$ with confidence 0.95.

Bootstrap Graph



Binomial Confidence Interval From Distributional Theory

Suppose we have a biased coin with probability of heads θ. Suppose we take a sample of size n and measure the proportion of successes $\hat{\theta}$. From the central limit theorem, this quantity is approximately normally distributed with mean $\hat{\theta}$ and standard deviation $\sqrt{\hat{\theta}(1-\hat{\theta})/n}$.

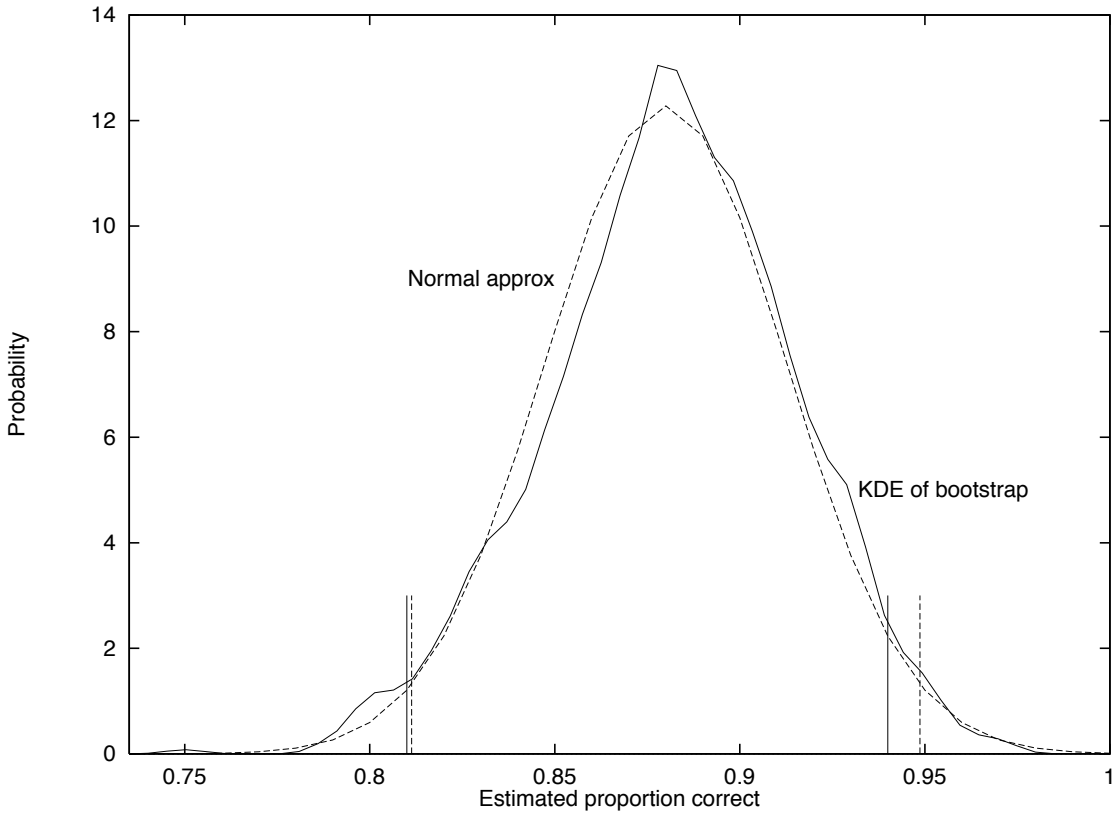
We can therefore use a 95% confidence interval for the mean of the normal distribution to compute a confidence interval for the binomial distribution. We make a slight change (called the "continuity correction") to correct for the discrete nature of the binomial distribution.

$$\hat{\theta} - \left[z_{0.975}\sqrt{\hat{\theta}(1-\hat{\theta})/n} + 1/(2n)\right] \leq \hat{\theta} \leq \hat{\theta} + \left[z_{0.975}\sqrt{\hat{\theta}(1-\hat{\theta})/n} + 1/(2n)\right]$$

Here $z_{0.975}$ is the value of a normally distributed variable z such that $P(z \leq z_{0.975}) = 0.975$. Specifically, $z_{0.975} = 1.96$.

Results: $0.811 \leq \hat{\theta} \leq 0.949$.

Bootstrap and Normal Distributions



The Normal approximation is always symmetrical, so it does not work very well when $\hat{\theta}$ is near 0.0 or 1.0.

Comparing Two Measurements

I performed 33 trials of

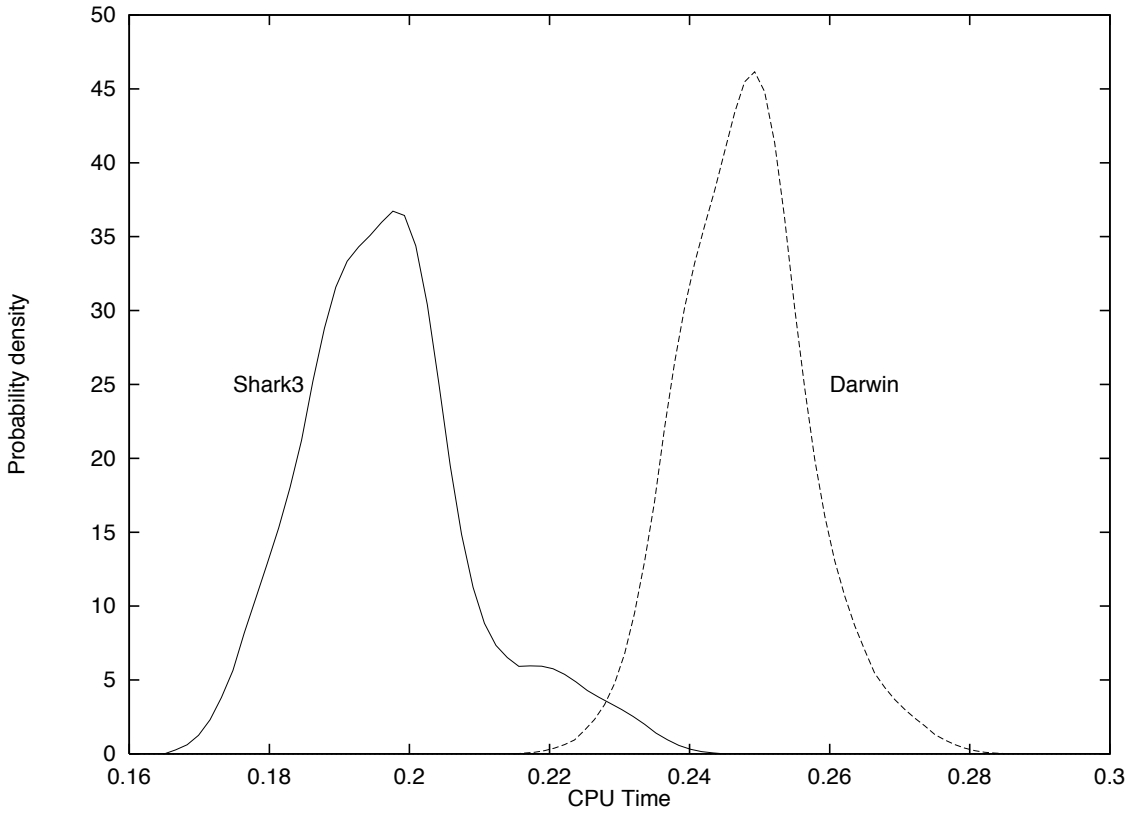
$$\binom{10000}{500}$$

in Common Lisp on `shark3.cs.orst.edu`.

0.21	0.20	0.20	0.19	0.20	0.19	0.18	0.20	0.19	0.19
0.19	0.19	0.20	0.18	0.19	0.20	0.22	0.20	0.20	0.20
0.19	0.20	0.18	0.19	0.19	0.20	0.20	0.22	0.18	0.19
0.21	0.23	0.20							

Can we conclude that shark3 is faster than darwin?

Visualizing



Comparable kernel density estimation plots. Visually, shark3 is much faster than darwin.

Bootstrap Test

Conduct 1000 trials of the following:

Draw bootstrap sample from Darwin, compute mean $\overline{x}_d$

Draw bootstrap sample from Shark3, compute mean $\overline{x}_s$

Count number of times $\overline{x}_d > \overline{x}_s$.

If this is greater than 950, then we can be 95% confident.

Result: All 1000 trials give darwin slower than shark3.

We can also compute a 95% bootstrap confidence interval on the difference $\overline{x}_d - \overline{x}_s$:

$$0.0461 \leq \overline{x}_d - \overline{x}_s \leq 0.0553.$$

Distributional Test

If two random variables are normally distributed, then their difference is normally distributed with mean $\mu = \mu_1 - \mu_2$ and standard deviation $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$.

Now the sampling distribution of the mean $\overline{x}$ is approximately normally distributed (according to the central limit theorem). So we know $\overline{x}_1 - \overline{x}_2$ is also normally distributed. However, because we don't know σ_1 or σ_2, we must use the t distribution instead.

If the two samples have sizes n_1 and n_2, then $\overline{x}_1 - \overline{x}_2$ is has a t distribution with mean $\overline{x}_1 - \overline{x}_2$ and standard deviation

$$s = \sqrt{\left(\frac{\Sigma_{i=1}^{n_1}(x_{1,i} - \overline{x}_1)^2}{n_1 - 1} + \frac{\Sigma_{i=1}^{n_2}(x_{2,i} - \overline{x}_2)^2}{n_2 - 1}\right)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}$$

and $n_1 + n_2 - 2$ degrees of freedom.

Using the data above, we obtain

$\overline{x}_1 - \overline{x}_2 = 0.0509$

$s = 0.0023$

$df = 68.$

A 95% confidence interval for the difference is (0.0463,0.0555).

Hypothesis Testing

Suppose we want to know whether the true difference between the two machines is zero or non-zero. We can formalize this as a statistical decision:

H_0: $\mu_1 - \mu_2 = 0$
H_1: $\mu_1 - \mu_2 \neq 0$

H_0 is the "null hypothesis" and H_1 is the "alternative hypothesis". An hypothesis test determines probablistically whether we can reject H_0 in favor of H_1 by asking "Suppose H_0 is true, what is the probability that we would have observed the given data?"

Specifically, we want to know what is the probability that we would observe $\overline{x}_1 - \overline{x}_2 \geq 0.0509$ when the true difference was zero. This can be determined from the t distribution.

The computed value of t is

$$t = \frac{\overline{x}_1 - \overline{x}_2}{s} = 21.69$$

The probability of seeing a t value greater than or equal to this is virtually 0.0. $t_{0.99999}(68) = 4.59$.

Paired Differences

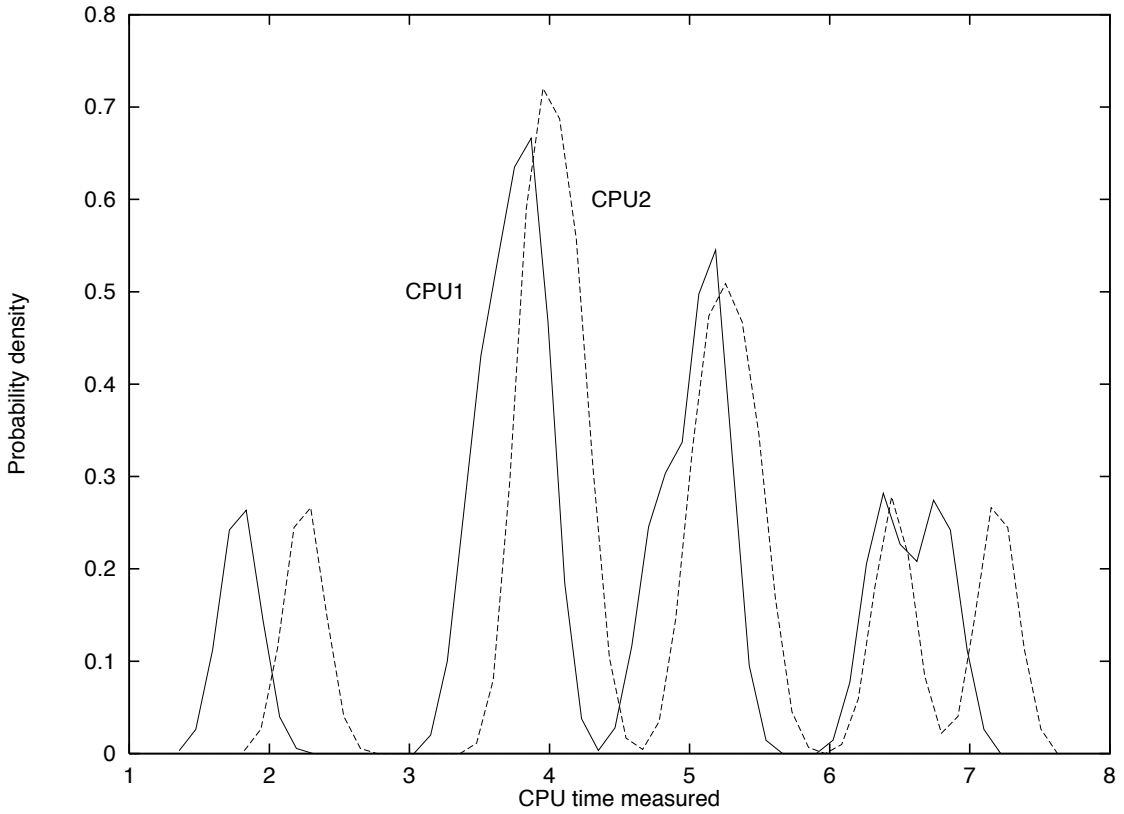
Suppose we had a set of benchmark programs that we were going to run on two machines. We will run each program on each machine to obtain the following data:

Program	CPU1	CPU2
P1	3.482514	3.896850
P2	3.677492	3.866780
P3	3.877525	4.206775
P4	6.787100	7.197257
P5	1.789549	2.250253
P6	5.156133	5.457694
P7	4.777698	5.075136
P8	3.906618	4.095468
P9	6.374434	6.456649
P10	5.152357	5.257691

Notice that the different programs have very different run times (e.g., ranging from 1.78 to 6.79 on CPU1).

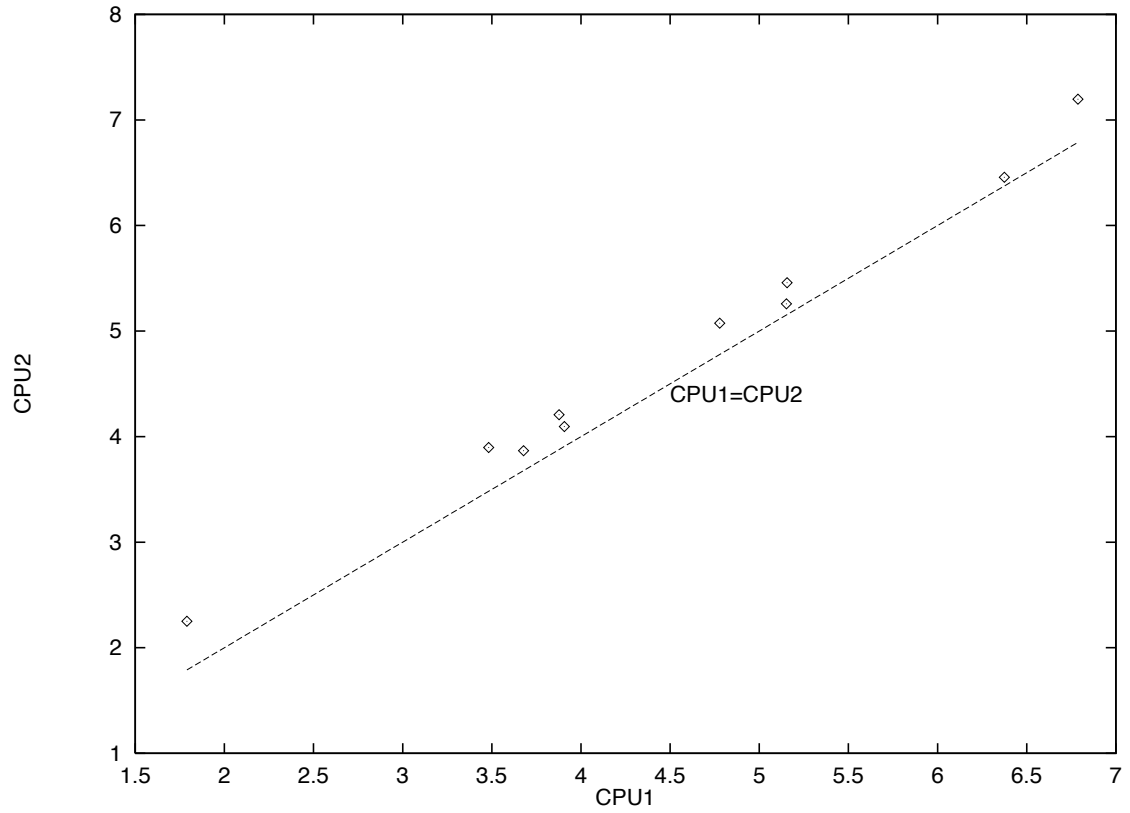
Visualization

There are many ways to visualize the data. We can superimpose a kernel density estimate for each of the CPU's:



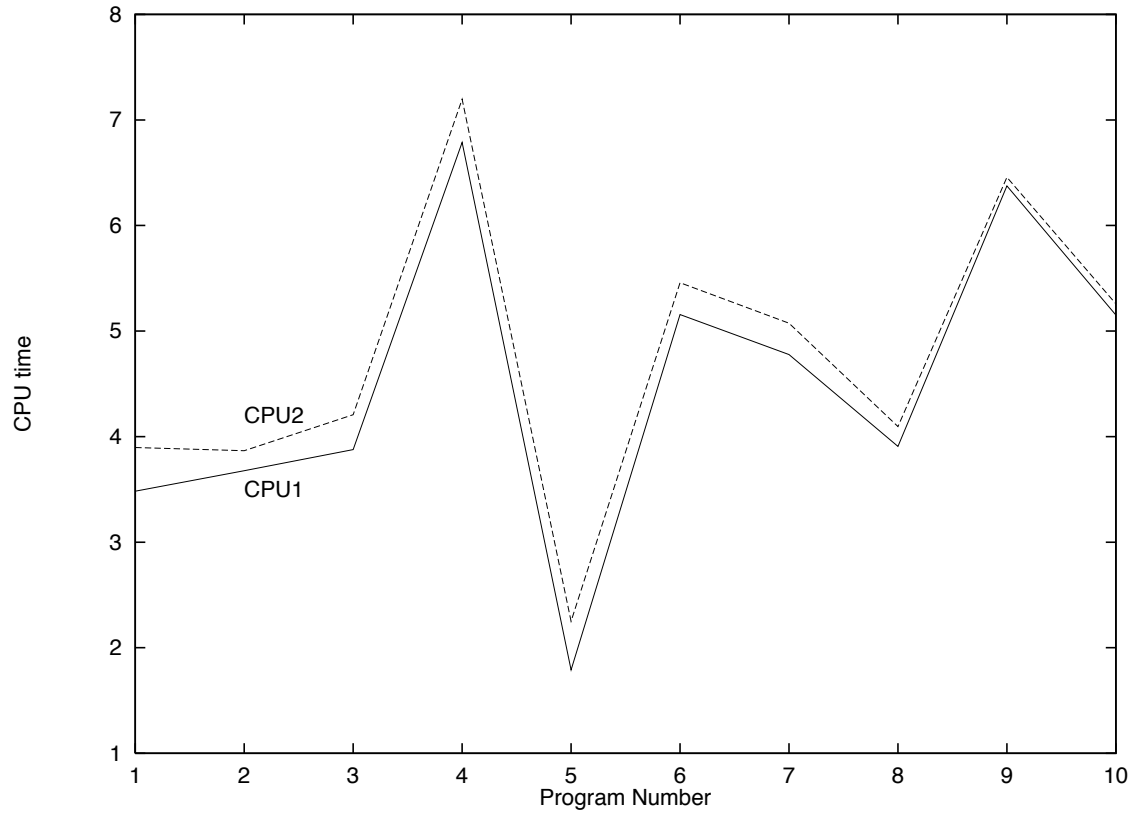
This suggests that CPU1 is systematically offset from CPU2.

Visualization (2)



This plots CPU1 versus CPU2 and also plots the line $y = x$. Notice that the performance of CPU1 is correlated with the performance of CPU2. Notice that all points lie above the line, suggesting that CPU2 is always bigger than CPU1.

Visualization (3)



Here we have plotted the data in sequential order (by program). We can see even more strongly that the CPU times of the programs co-vary.

Analysis of Paired Data

Construct points by subtracting $CPU1_i - CPU2_i$, and analyze this just like the univariate data we analyzed last time.

mean = 0.2779
standard deviation = 0.1321
degrees of freedom = 9
value of $t = 6.6549$

The probability of seeing this value (or greater) if the true mean were 0 is 0.0000466, so we can reject the null hypothesis that the mean is zero in favor of the alternative hypothesis that the mean is greater than zero with confidence at least 0.9999534.

However, in the absence of prior expectation that CPU2 is slower than CPU1, we should use a "two-tailed test". To do this, we must compute the probability that we would have seen a value $t \geq 6.6549$ or $t \leq -6.6549$. Because the distribution is symmetric, this probability is 0.0000932, so we can reject the null hypothesis in favor of the hypothesis that the mean is non-zero with confidence 0.9999068.

Analysis as Unpaired Data

If we had used our previous technique for unpaired data, we would not be able to detect the difference between the two CPU's.

$\overline{x}_1 - \overline{x}_2 = 0.2779$
$s = 0.6473$
$df = 18$
$t = 0.4293$

The probability of observing this t value (or greater) if the true difference is zero is 0.3364 (for a one-tailed test). For a 2-tailed test, it is 0.6728. So we cannot reject the null hypothesis using this analysis.

Bootstrap Analysis

1000-fold Bootstrap 95% Confidence Interval for $\overline{CPU2 - CPU1}$:
$0.2086 \leq \overline{(CPU2 - CPU1} \leq 0.3580$.

1000-fold Bootstrap 95% Confidence Interval for $\overline{CPU2} - \overline{CPU1}$:
$-1.4533 \leq \overline{CPU2} - \overline{CPU1} \leq 1.0277$. (This contains 0, so we cannot reject the null hypothesis.)