Empirical Methods

Alan Bundy

School of informatics

University of Edinburgh

(Slides courtesy of Paul Cohen)

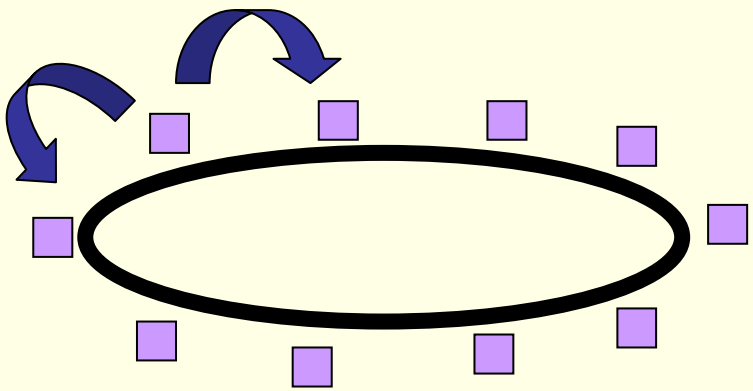
Outline

- **Lesson 1: Evaluation begins with claims**
- **Lesson 2: Exploratory data analysis means looking beneath results for reasons**
- **Lesson 3: Run pilot experiments**
- **Lesson 4: Control sample variance, rather than increase sample size.**
- **Lesson 5: Check result is significant.**

Lesson 1: Evaluation begins with claims

- **The most important, most immediate and most neglected part of evaluation plans.**
- **What you measure depends on what you want to know, on what you claim.**
- **Claims:**
  - **X is bigger/faster/stronger than Y**
  - **X varies linearly with Y in the range we care about**
  - **X and Y agree on most test items**
  - **It doesn't matter who uses the system (no effects of subjects)**
  - **My algorithm scales better than yours (e.g., a relationship between size and runtime depends on the algorithm)**
- **Non-claim: I built it and it runs fine on some test data**

Case Study: Comparing two algorithms



- **Scheduling processors on ring network; jobs spawned as binary trees**
- **KOSO: keep one, send one to my left or right arbitrarily**
- **KOSO*: keep one, send one to my least heavily loaded neighbour**

Theoretical analysis went only so far, for unbalanced trees and other conditions it was necessary to test KOSO and KOSO* empirically

An Empirical Study of Dynamic Scheduling on Rings of Processors" Gregory, Gao, Rosenberg & Cohen, Proc. of 8th IEEE Symp. on Parallel & Distributed Processing, 1996

Evaluation begins with claims

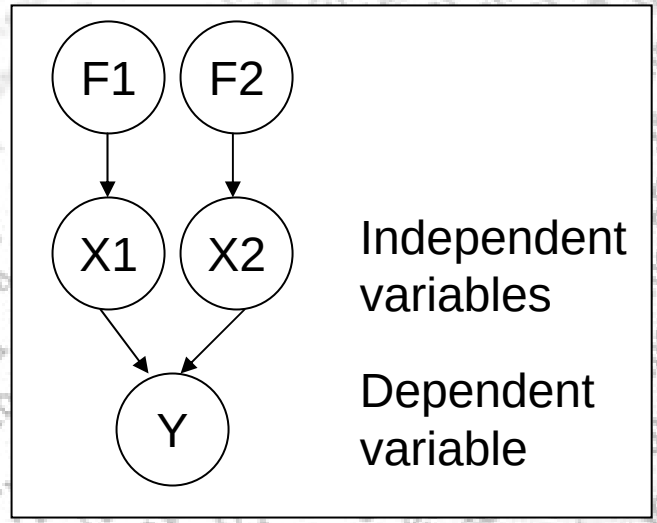
- **Hypothesis (or claim): KOSO takes longer than KOSO* *because* KOSO* balances loads better**
  - **The “because phrase” indicates a hypothesis about why it works. This is a better hypothesis than the "beauty contest" demonstration that KOSO* beats KOSO**

- **Experiment design**
  - ***Independent variables*: KOSO v KOSO*, no. of processors, no. of jobs, probability job will spawn,**
  - ***Dependent variable*: time to complete jobs**

Useful Terms

Independent variable: A variable that indicates something you manipulate in an experiment, or some supposedly causal factor that you can't manipulate such as gender (also called a factor)

Dependent variable: A variable that indicates to greater or lesser degree the causal effects of the factors represented by the independent variables



Initial Results

- **Mean time to complete jobs:**

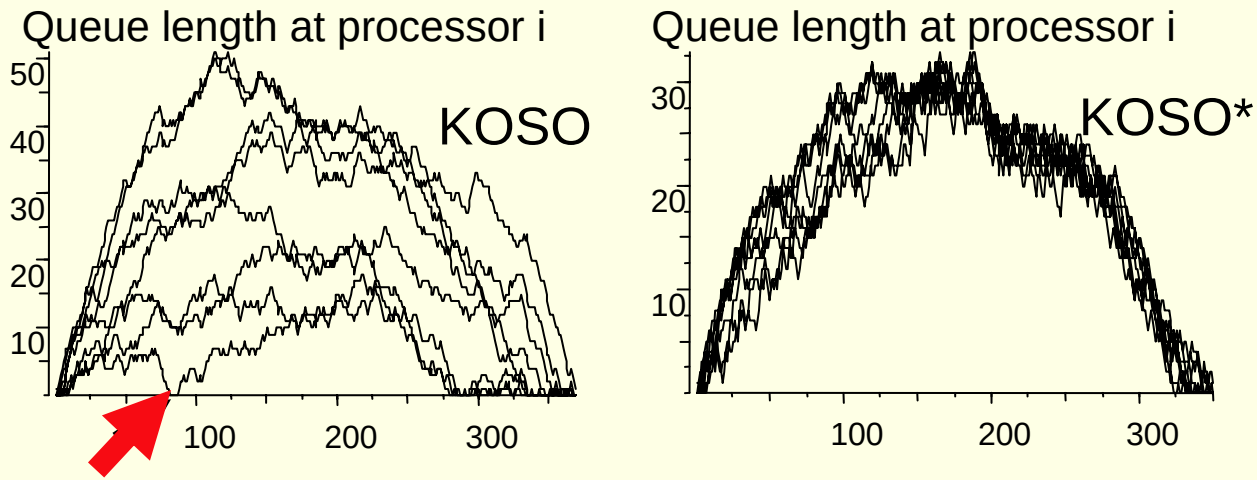
  KOSO: 2825 **(the "dumb" algorithm)**

  KOSO*: 2935 **(the "load balancing" algorithm)**

- **KOSO is actually 4% *faster* than KOSO* !**
- **This difference is not statistically significant (more about this, later)**
- **What happened?**

Lesson 2: *Exploratory data analysis* means looking beneath results for reasons

- **Time series of queue length at different processors:**

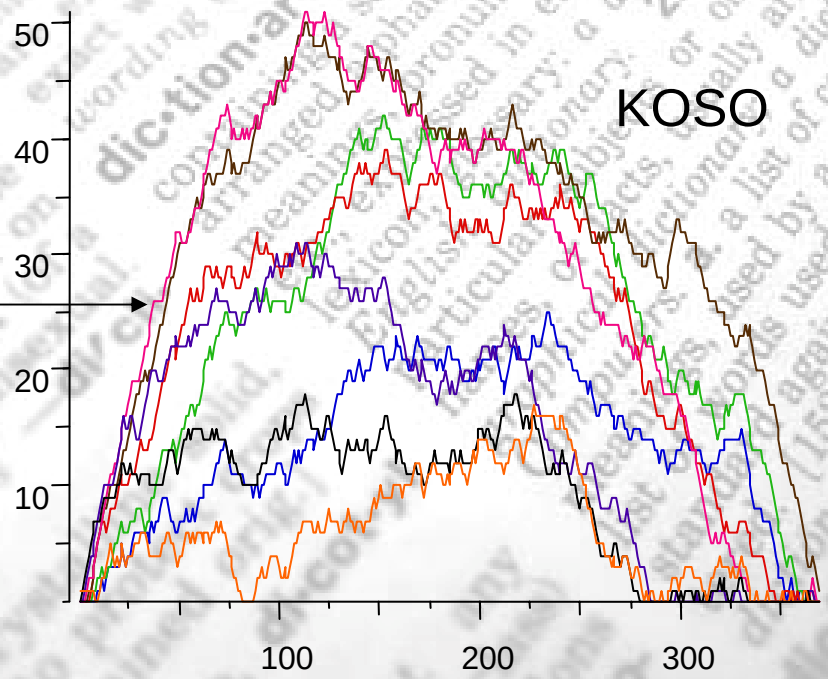


- **Unless processors starve (red arrow) there is no advantage to good load balancing (i.e., KOSO* is no better than KOSO)**

Useful Terms

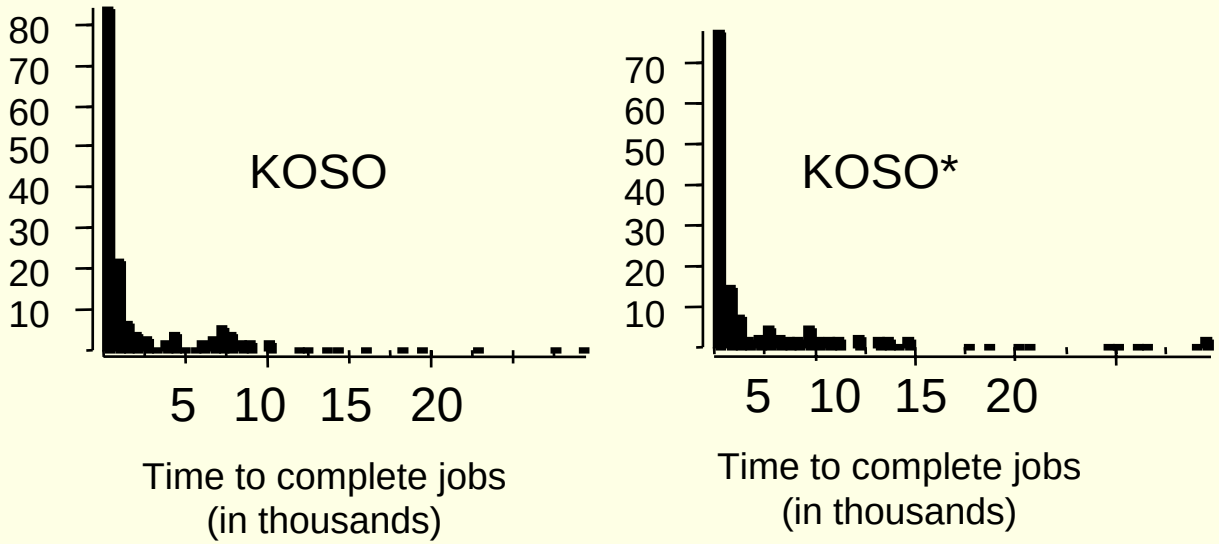
Time series: One or more dependent variables measured at consecutive time points

Time series of queue length at processor "red"



Lesson 2: *Exploratory data analysis* means looking beneath results for reasons

- **KOSO* is statistically no faster than KOSO. Why?**



- **Outliers dominate the means, so test isn't significant**

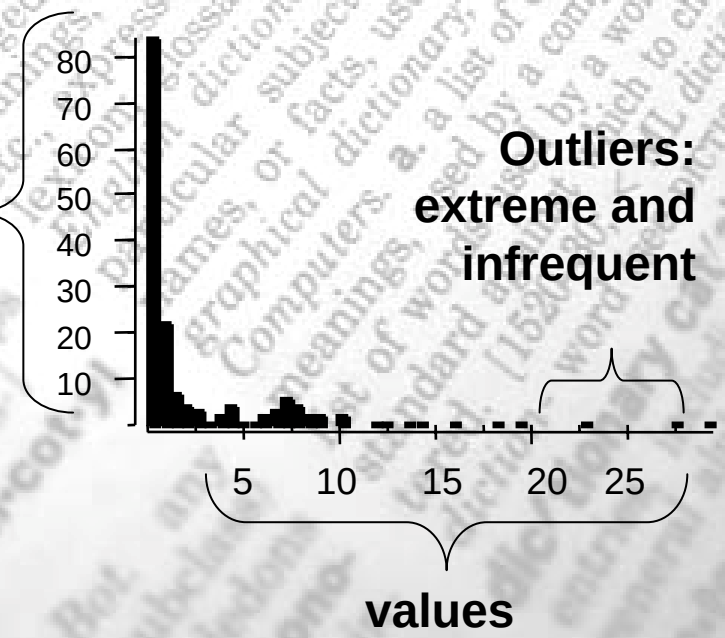
Useful Terms

Frequency distribution: The frequencies with which the values in a distribution occur (e.g., the frequencies of all the values of "age" in the room)

Outlier: Extreme, low-frequency values.

Mean: The average.

Means are very sensitive to outliers.



More exploratory data analysis

- **Mean time to complete jobs:**

  KOSO: 2825

  KOSO*: 2935

- **Median time to complete jobs**

  KOSO: 498.5

  KOSO*: 447.0

- **Looking at means (with outliers) KOSO* is 4% slower but looking at medians (robust against outliers) it is 11% faster.**

Useful Terms

Median: The value which splits a sorted distribution in half. The 50th *quantile* of the distribution.

1 2 3 7 7 8 14 15 17 21 22

Mean: 10.6

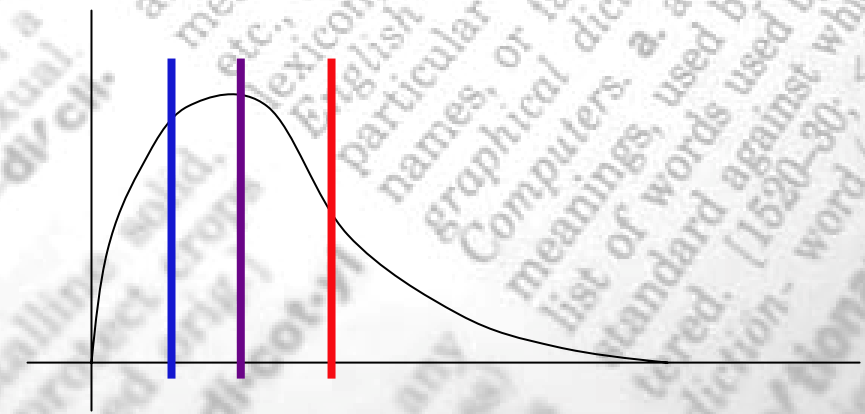
Median: 8

1 2 3 7 7 8 14 15 17 21 22 1000

Mean: 93.1

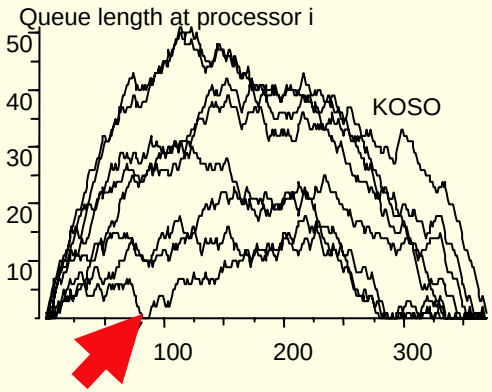
Median: 11

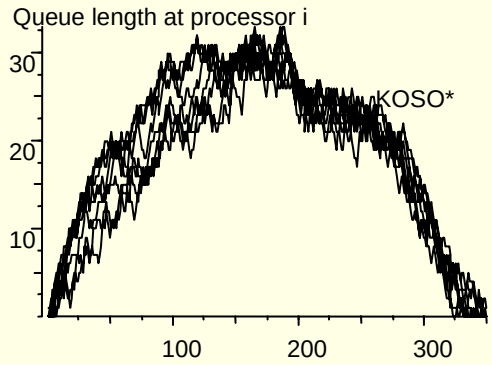
Quantile: A "cut point" q that divides the distribution into pieces of size q/100 and 1-(q/100). Examples: 50th quantile cuts the distribution in half. 25th quantile cuts off the lower *quartile*. 75th quantile cuts off the upper quartile.



How are we doing?

- **Hypothesis (or claim): KOSO takes longer than KOSO* *because* KOSO* balances loads better**
- **Mean KOSO is shorter than mean KOSO*, median KOSO is longer than KOSO*, no evidence that load balancing helps because there is almost no processor starvation in this experiment.**
- **Now what?**

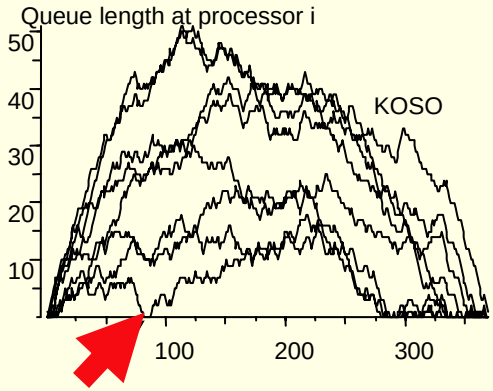


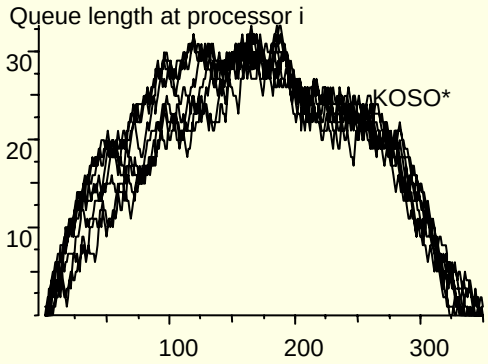


Lesson 3: Always run pilot experiments

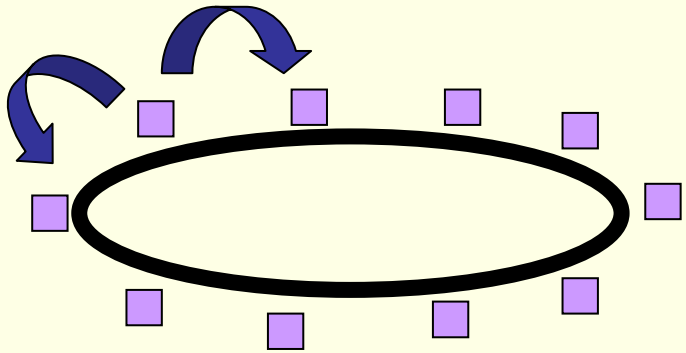
- **A pilot experiment is designed less to test the hypothesis than to test the experimental apparatus to see whether it *can* test the hypothesis.**
- **Our independent variables were not set in a way that produced processor starvation so we couldn't test the hypothesis that KOSO* is better than KOSO because it balances loads better.**
- **Use pilot experiments to adjust independent and dependent measures, see whether the protocol works, provide preliminary data to try out your statistical analysis, in short, test the *experiment design.***

Next steps in the KOSO / KOSO* saga…



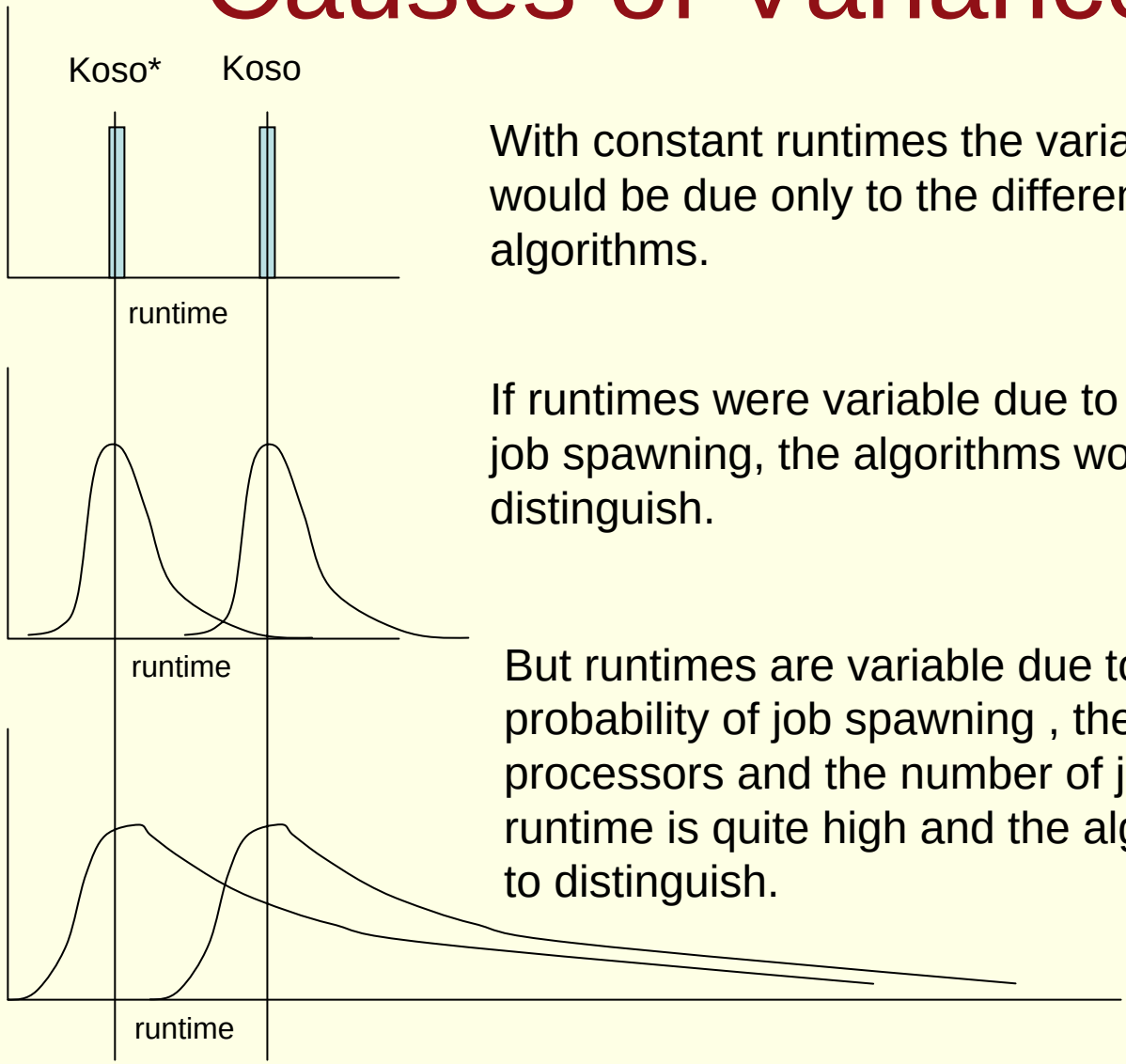


It looks like KOSO* does balance loads better (less variance in the queue length) but without processor starvation, there is no effect on run-time



- Cohen ran another experiment, varying the number of processors in the ring: 3, 9, 10 and 20
- Once again, there was no significant difference in run-time. Why?
- Problem variance dominates algorithm variance.

Causes of Variance



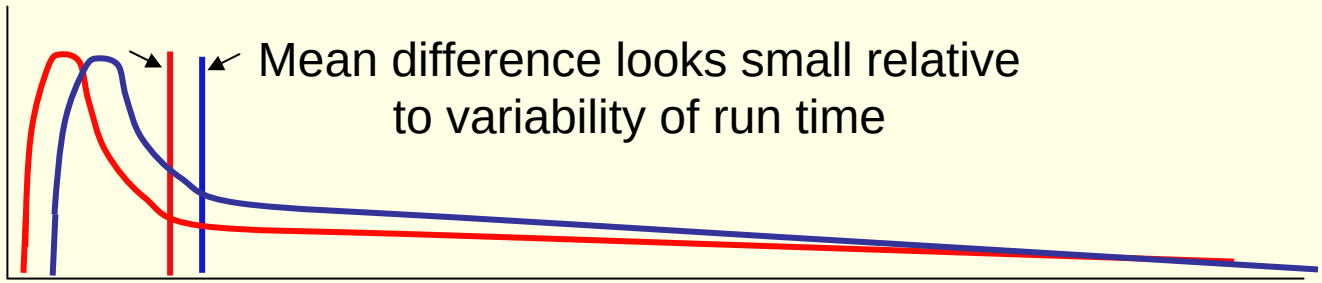
With constant runtimes the variance in runtime would be due only to the difference between the algorithms.

If runtimes were variable due to one cause, say job spawning, the algorithms would still be easy to distinguish.

But runtimes are variable due to several causes, i.e. probability of job spawning , the number of processors and the number of job, so the variance in runtime is quite high and the algorithms are difficult to distinguish.

Lesson 4: Control sample variance

- **Suppose you are interested in which algorithm runs faster on a batch of problems but the run time depends more on the problems than the algorithms**

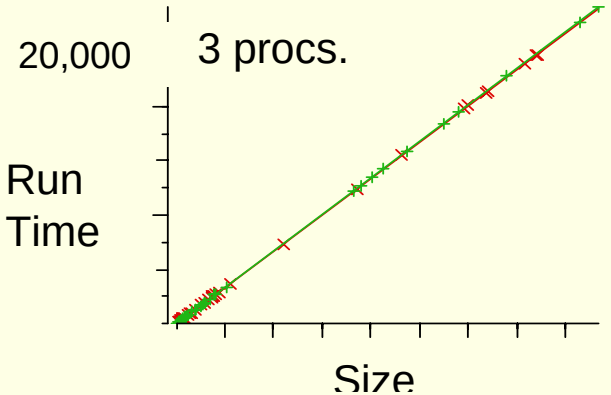


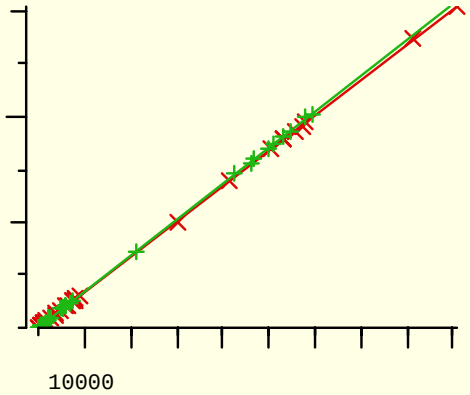
Run times for Algorithm 1 and Algorithm 2

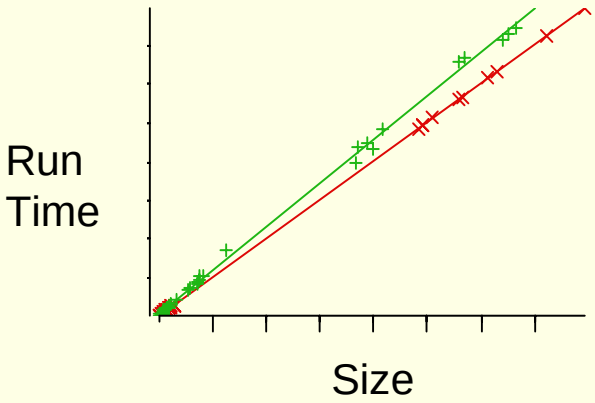
- **You don't care very much about the problems, so you'd like to transform run time to "correct" the influence of the problem. This is one kind of *variance-reducing transform*.**

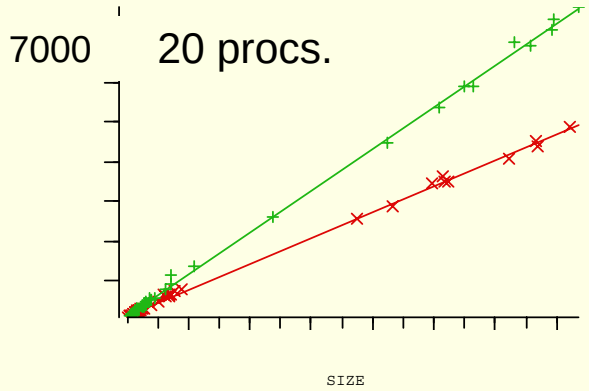
What causes run times to vary so much?

Run time depends on the number of processors and on the number of jobs (size). The relationships between these and run time are different for KOSO and KOSO* Green: KOSO Red: KOSO*

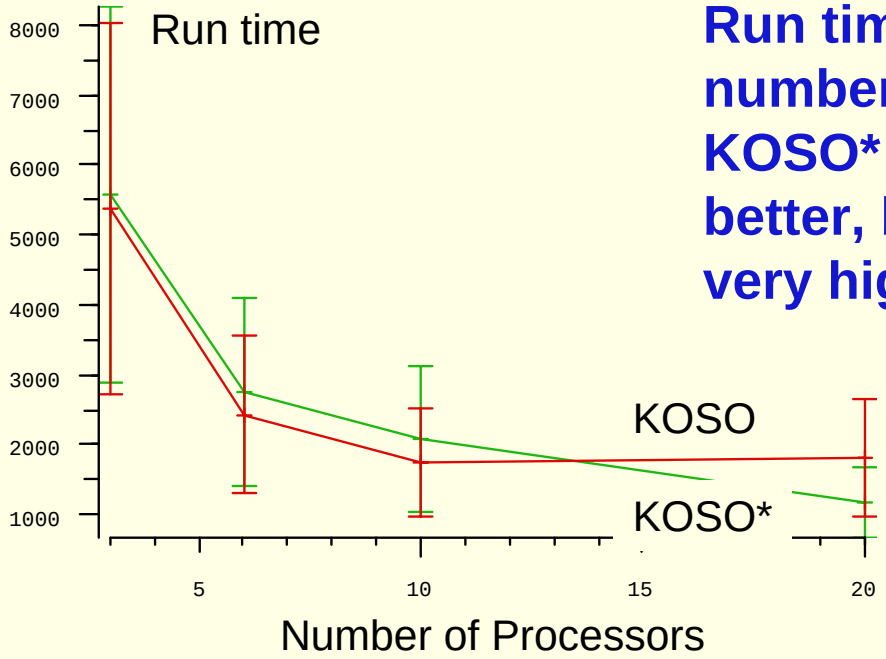








What causes run times to vary so much?



Run time decreases with the number of processors, and KOSO* appears to use them better, but the variance is still very high (confidence intervals)

Can we transform run time with some function of the number of processors and the problem size?

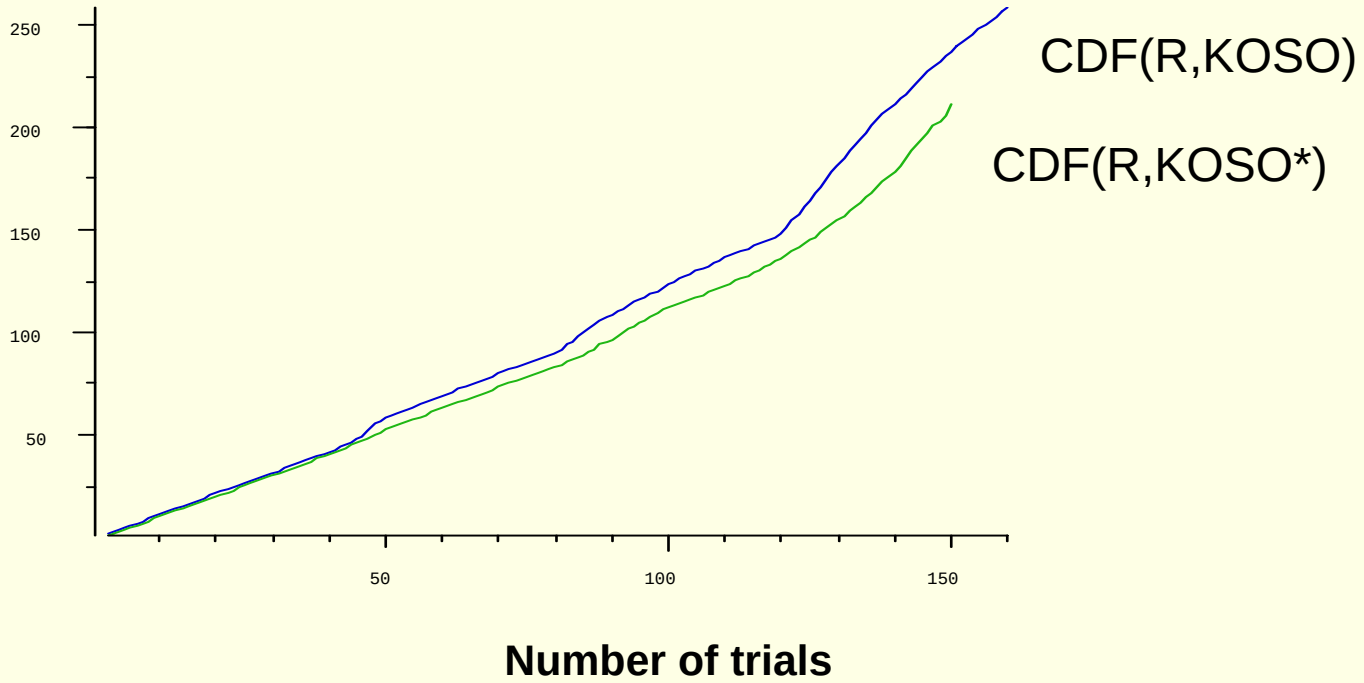
Transforming run time

- **Assume each task takes unit time.**
- **Let S be the number of tasks to be done.**
- **Let N be the number of processors to do them.**
- **Let T be the time required to do them all (run time).**
- **So $k_i = S_i/N_i$ is best possible run time on task i,**
  - **i.e., perfect use of parallelism.**
- **T_i / k_i measures deviation from perfection.**
- **The transform we want is $R_i = (T_i N_i) / S_i$.**
  - **Runtime restated to be independent of problem size and number of processors.**

Lesson 5: Check result is significant

	Mean	Median
KOSO	1.61	1.18
KOSO*	1.40	1.03

Median KOSO* is almost perfectly efficient

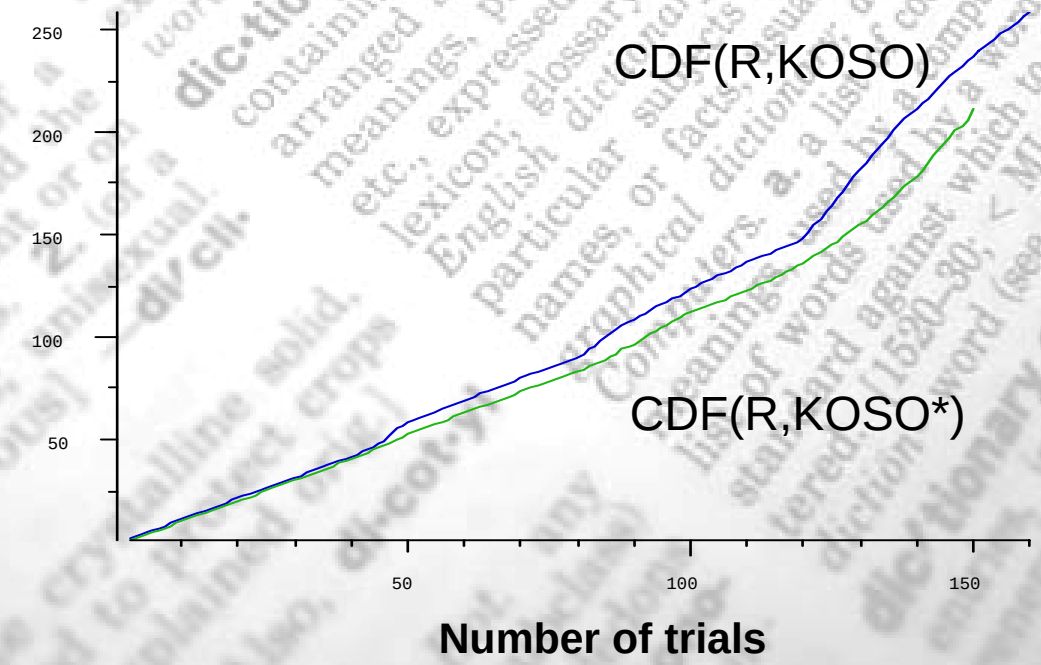


Useful terms

Cumulative Distribution Function:

A "running sum" of all the quantities in the distribution:

7 2 5 3 … => 7 9 14 17 …



A statistically significant difference!

	Mean	Standard deviation
KOSO	1.61	0.78
KOSO*	1.40	0.7

Two-sample t test:

$$t = \frac{\bar{x}_{koso} - \bar{x}_{koso*}}{\hat{\sigma}_{(\bar{x}_{koso} - \bar{x}_{koso*})}}$$

difference between the means

probability of this result if the difference between the means were truly zero

$$t = \frac{1.61 - 1.4}{.084} = 2.49, p < .02$$

estimate of the variance of the difference between the means

The two-sample t test

	Mean	Standard deviation
KOSO	1.61	0.78
KOSO*	1.40	0.7

$$t = \frac{\bar{x}_{koso} - \bar{x}_{koso*}}{\hat{\sigma}_{(\bar{x}_{koso} - \bar{x}_{koso*})}}$$

$$\hat{\sigma}_{(\bar{x}_{koso} - \bar{x}_{koso*})} = \sqrt{\frac{(N_{koso} - 1)s^2_{koso} + (N_{koso*} - 1)s^2_{koso*}}{N_{koso} + N_{koso*} - 2}\left(\frac{1}{N_{koso}} + \frac{1}{N_{koso*}}\right)}$$

$$\hat{\sigma}_{(\bar{x}_{koso} - \bar{x}_{koso*})} = \sqrt{\frac{(159)0.78^2 + (149)0.7^2}{160 + 150 - 2}\left(\frac{1}{160} + \frac{1}{150}\right)} = 0.084$$

$$t = \frac{1.61 - 1.4}{.084} = 2.49, p < .02$$

The logic of statistical hypothesis testing

1. Assume KOSO = KOSO*
2. Run an experiment to find the sample statistics

$$R_{koso}=1.61,\ R_{koso^*} = 1.4, \text{ and } \Delta = 0.21$$

3. Find the distribution of Δ under the assumption KOSO = KOSO*
4. Use this distribution to find the probability p of $\Delta = 0.21$ if KOSO = KOSO*
5. If the probability is very low (it is, p<.02) reject KOSO = KOSO*
6. p<.02 is your residual uncertainty that KOSO *might* equal KOSO*

difference between the means

probability of this result if the difference between the means were truly zero

$$t = \frac{1.61 - 1.4}{.084} = 2.49, p < .02$$

estimate of the variance of the difference between the means

Useful terms

1. Assume KOSO = KOSO*

2. Run an experiment to get the *sample statistics*

   R_{koso}=1.61, R_{koso*} = 1.4, and Δ = 0.21

3. Find the distribution of Δ under the assumption KOSO = KOSO*

4. Use this distribution to find the probability of Δ = 0.21 given H_0

5. If the probability is very low, reject KOSO = KOSO*

6. p is your residual uncertainty

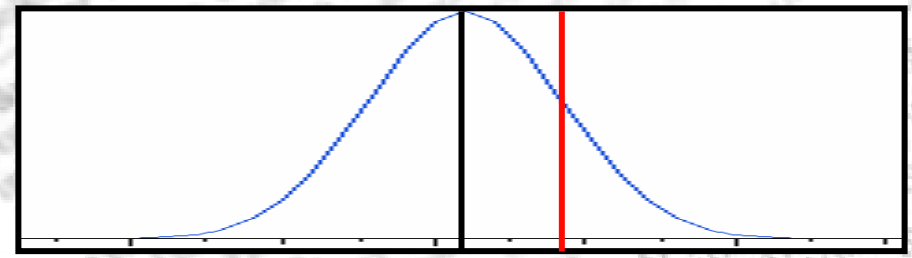
This is called the *null hypothesis* (H_0) and typically is the inverse of the *alternative hypothesis* (H_1) which is what you want to show.

This is called the *sampling distribution* of the statistic under the null hypothesis

This is called *rejecting the null hypothesis*.

This *p value* is the probability of incorrectly rejecting H_0

Useful terms



1. …
2. …
3. Find the distribution of Δ under the assumption KOSO = KOSO*
4. Use this distribution to find the probability of $\Delta = 0.21$ given H_0
5. …
6. …

…the *sampling distribution* of the statistic. Its standard deviation is called the *standard error*

Statistical tests transform statistics like Δ into standard error (s.e.) units

It's easy to find the region of a distribution bounded by k standard error units

E.g., 1% of the normal (Gaussian) distribution lies above 1.96 s.e. units.

Conclusion

- **Clarify your claim before you start.**
- **Make sure your experiment is capable of evaluating your claim (run pilots).**
- **Explore beneath the results to understand what is going on.**
- **Design statistical analysis to control unwanted variance.**
- **Support your claim by showing null hypothesis is very unlikely.**