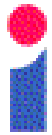
How to do an Informatics PhD

Alan Bundy

School of informatics

University of Edinburgh



What is Informatics?

The study of the structure, behaviour, and interactions of both natural and artificial computational systems.

What are the Big Informatics Questions?

- **What is the nature of computation/information?**
- **What is mind?**
- **How can we build useful ICT products?**

Informatics Techniques

- **Informatics as the space of computational techniques.**
- **Job of Informatics to explore this space.**
  - **Which techniques are good for which tasks?**
  - **What are properties of these techniques?**
  - **What are relationships between these techniques?**

What are Informatics Techniques?

- **Information Representation**: e.g. databases, hash tables, production rules, neural nets.
- **Algorithms**: e.g. quick sort, depth-first search, parser.
- **Architectures**: e.g. von Neumann, parallel, agents.
- **Software Engineering Processes**: e.g. extreme programming, knowledge acquisition/requirements capture.
- **Theories**: e.g. denotational semantics, process algebras, computational logics, hidden Markov models.

The Space of Informatics Techniques

- Multi-dimensional space of techniques,
  - linked by relationships.
- Rival techniques for same task,
  - with tradeoffs of properties.
- Complementary techniques which interact.
- Build systems from/with collections of techniques

Exploration of Techniques Space

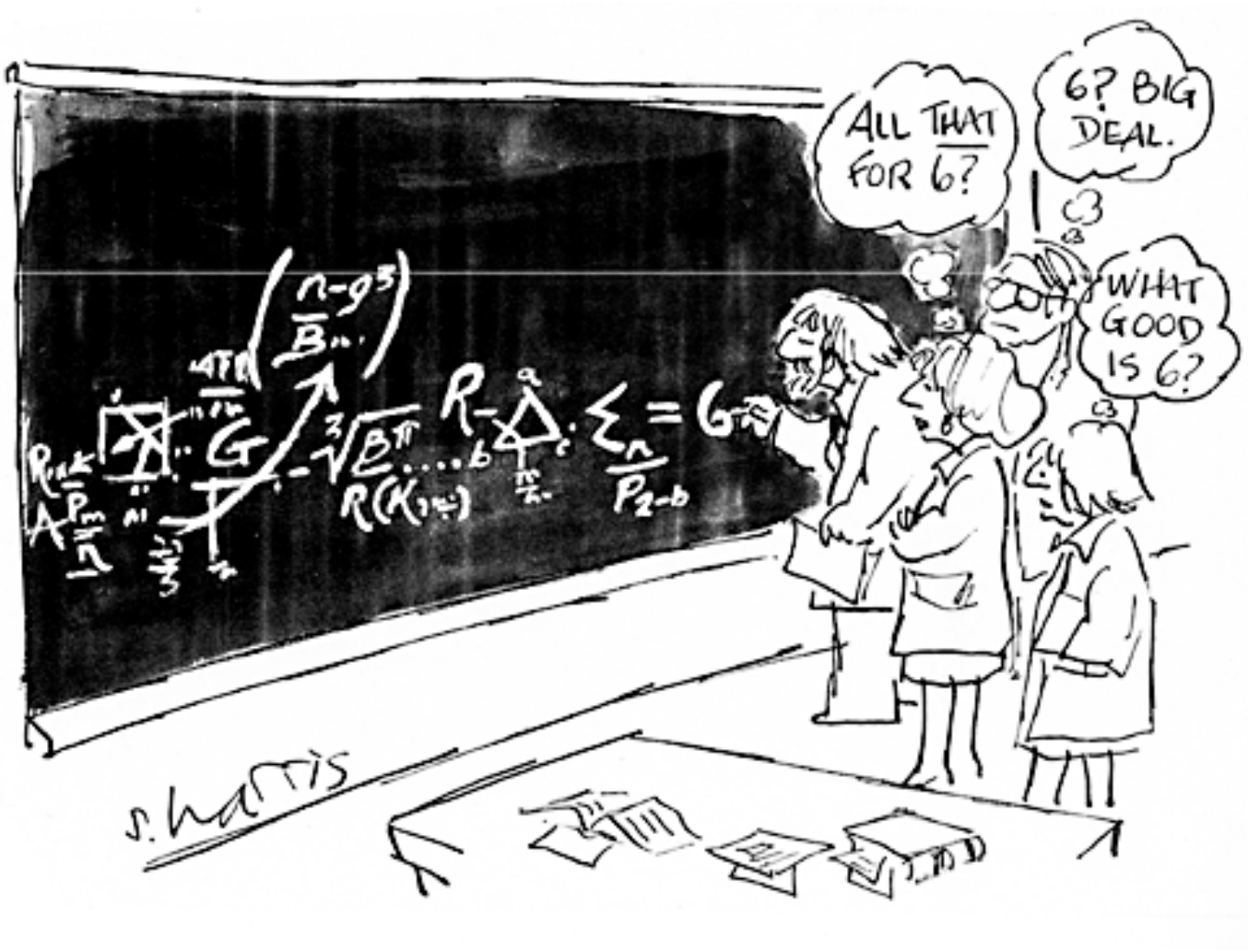
- Invention of new technique,
- Investigation of technique,
  - e.g. discovery of properties of, or relationships between, techniques.
- Extension or improvement of old technique,
- New application of a technique,
  - to artificial or natural systems.
- Combine several techniques into a system.

Exercise: Informatics Techniques

What additional Informatics techniques can you think of?

- Information Representation?
- Algorithms?
- Architectures?
- Software Engineering Processes?
- Theories?
- Other kind?

The Significance of Research



Importance of Hypotheses

- Science and engineering proceed by
  - the formulation of hypotheses
  - and the provision of supporting (or refuting) evidence for them.
- Informatics should be no exception.
- But the provision of explicit hypotheses in Informatics is rare.
- This causes lots of problems.
- My mission – to persuade you to rectify this situation.

Problems of Omitting Hypotheses

- Usually many possible hypotheses.
- Ambiguity is major cause of referee/reader misunderstanding.
- Vagueness is major cause of poor methodology:
  - Inconclusive evidence;
  - Unfocussed research direction.

Hypotheses in Informatics

- Claim about task, system, technique or parameter, e.g.:
  - All techniques to solve task X will have property Y.
  - System X is superior to system Y on dimension Z.
  - Technique X has property Y.
  - X is the optimal setting of parameter Y.
- Ideally, with the addition of a 'because' clause.
- Properties and relations along scientific, engineering or computational modelling dimensions.
- May be several hypothesis in each publication.

Rarely explicitly stated

Scientific Dimensions 1

- **Behaviour**: *the effect or result of the technique,*
  - correctness *vs* quality,
  - need external ‘gold standard’;
- **Coverage**: *the range of application of the technique,*
  - complete *vs* partial;
- **Efficiency**: *the resources consumed by the technique,*
  - e.g. time or space used,
  - usually as approx. function, e.g. linear, quadratic, exponential, terminating.

Scientific Dimensions 2

- Sometimes mixture of dimensions,
  - e.g., behaviour/efficiency poor in extremes of range.
- Sometimes trade-off between dimensions,
  - e.g., behaviour quality vs time taken.
- Property *vs* comparative relation.
- Task *vs* systems *vs* techniques *vs* parameters.

Engineering Dimensions

- **Usability**: *how easy to use?*
- **Dependability**: *how reliable, secure, safe?*
- **Maintainability**: *how evolvable to meet changes in user requirements?*
- **Scalability**: *whether it still works on complex examples?*
- **Cost***: In £s or time of development, running, maintenance, etc.*
- **Portability***: interoperability, compatibility.*

Computational Modelling Dimensions

- **External**: *match to external behaviours,*
  - both correct and erroneous.
- **Internal**: *match to internal processing,*
  - clues from e.g. protocol analysis.
- **Adaptability**: *range of occurring behaviours modelled*
  - *... and non-occurring behaviours not modelled.*
- **Evolvability**: *ability to model process of development.*

All this to some level of abstraction.

Exercise: Hypotheses

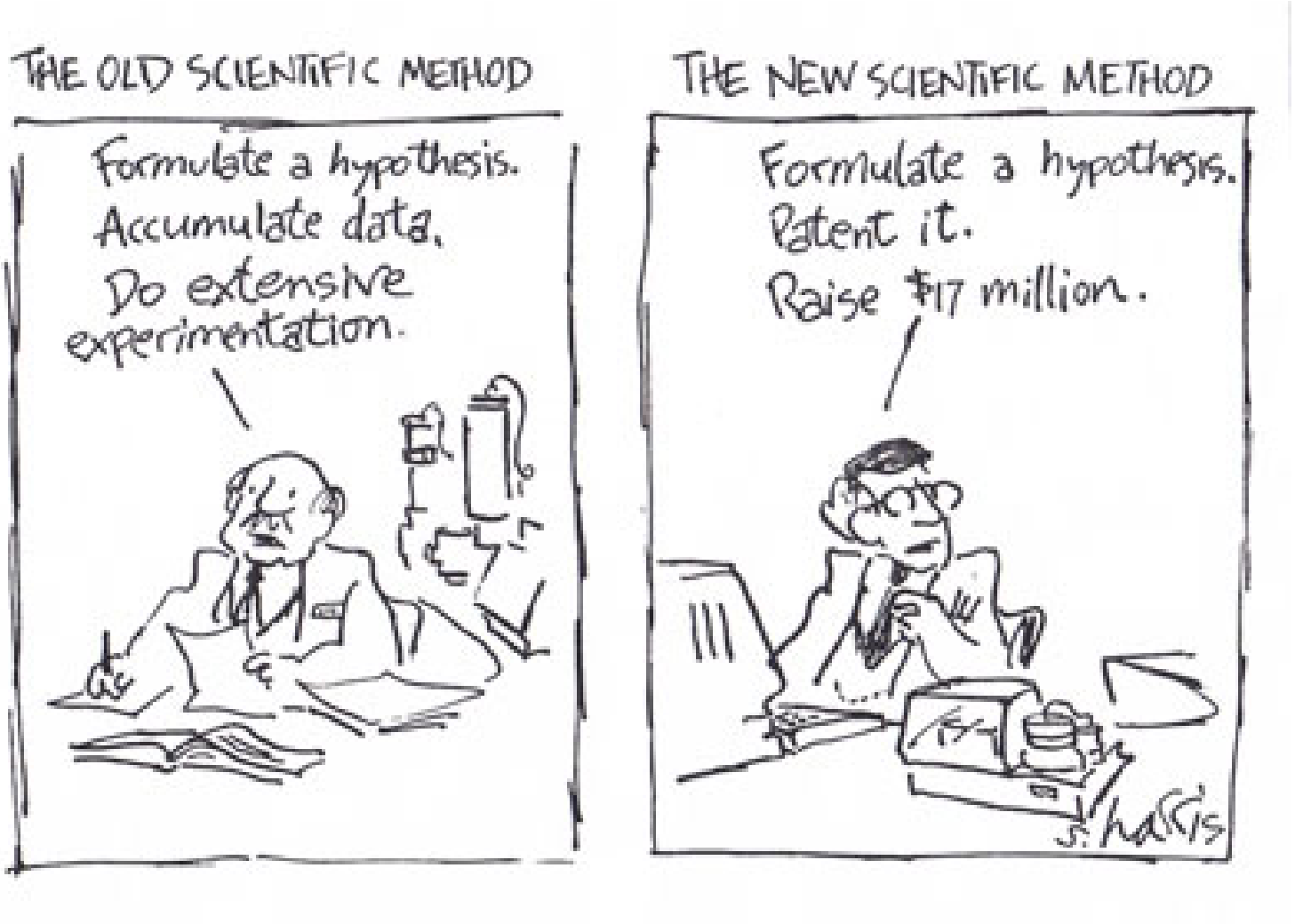
What Informatics hypotheses can you think of?

- Choose system/technique/parameter setting.
- Choose science/engineering/computational modelling dimensions.
- Choose property or relation.
- Has property or is better than rival on property?
- Other?

Theoretical Research

- Use of **mathematics** for definition and proof.
  - or sometimes just reasoned argument.
- **Applies** to task or technique.
- **Theorem** as hypothesis; **proof** as evidence.
- **Advantages**:
  - Abstract analysis of task;
  - Suggest new techniques, e.g. generate and test;
  - Enables proof of general properties/relationships,
    - cover potential infinity of examples;
    - Suggest extensions and generalisations;
- **Disadvantage**:
  - Sometimes difficult to reflect realities of task.

Experimentation



Experimental Research

- **Kinds**:
  - exploratory vs hypothesis testing.
- **Generality of Testing**:
  - test examples are representative.
- **Results Support Hypothesis**:
  - and not due to another cause.

How to Show Examples Representative

- Distinguish development from test examples.
- Use lots of dissimilar examples.
- Collect examples from an independent source.
- Use the shared examples of the field.
- Use challenging examples.
- Use acute examples

How to Show that Results Support Hypothesis

- Vary one thing at a time,
  - then only one cause possible.
  - Unfortunately, not always feasible.
- Analyse/compare program trace(s),
  - to reveal cause of results.
- Use program analysis tools,
  - e.g. to identify cause/effect correspondences

Hypotheses must be Evaluable

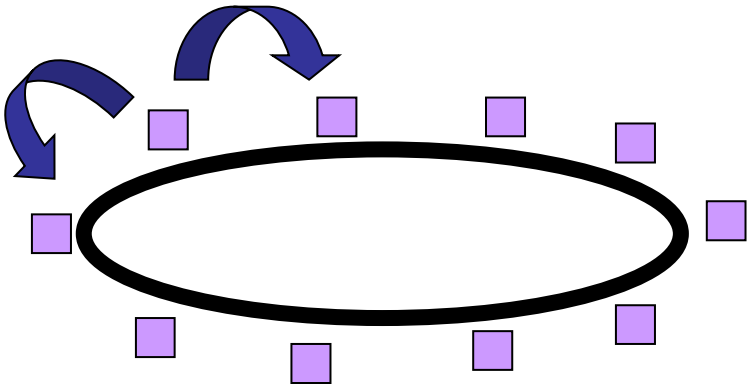
- If hypothesis cannot be evaluated then fails Popper's test of science.
- Obvious hypothesis may be too expensive to evaluate,
  - e.g. programming in MyLang increases productivity,
- Replace with evaluable hypothesis:
  - Strong typing reduces bugs.
  - MyLang has strong typing.

Empirical Methods

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

My thanks to Paul Cohen

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

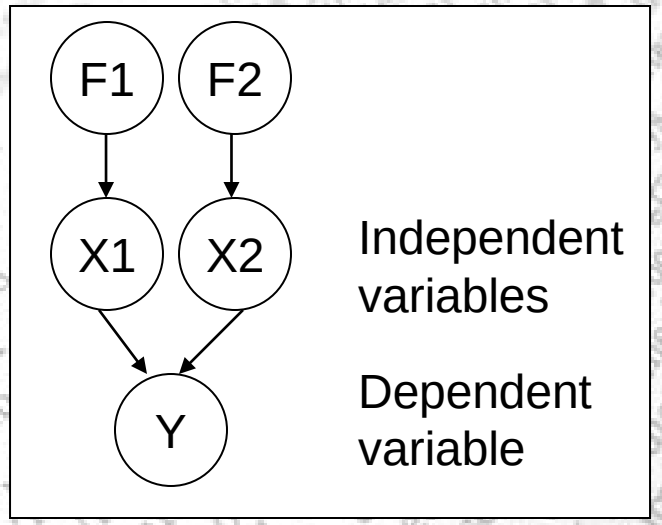
- Hypothesis (or claim): KOSO takes longer than KOSO* *because* KOSO* balances loads better
  - The “because phrase” indicates a hypothesis about why it works. This is a better hypothesis than the "beauty contest" demonstration that KOSO* beats KOSO

- 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



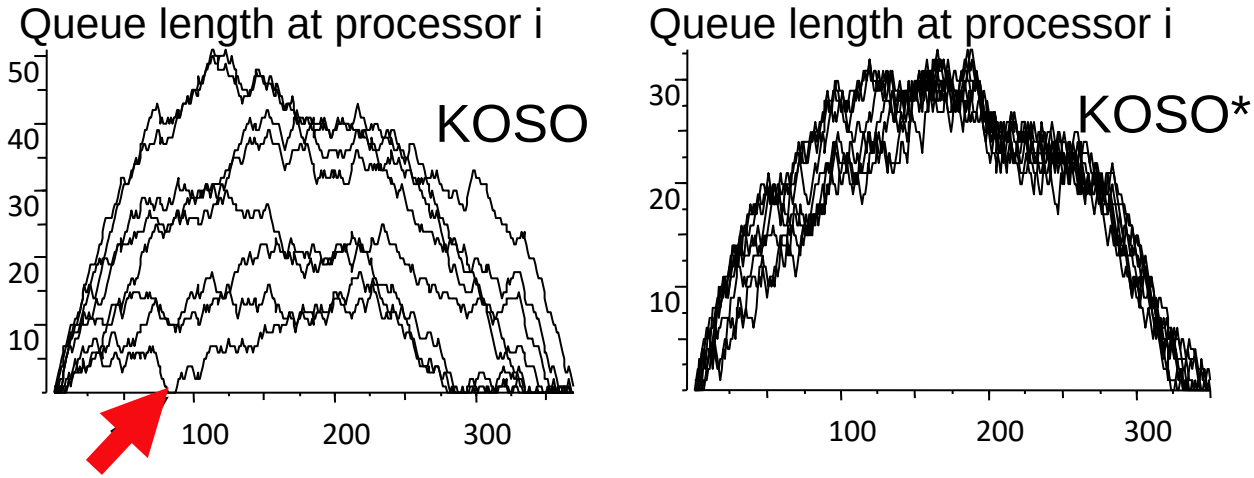
Initial Results

- 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 1: *Exploratory data analysis* means looking beneath results for reasons

- Time series of queue length at different processors:

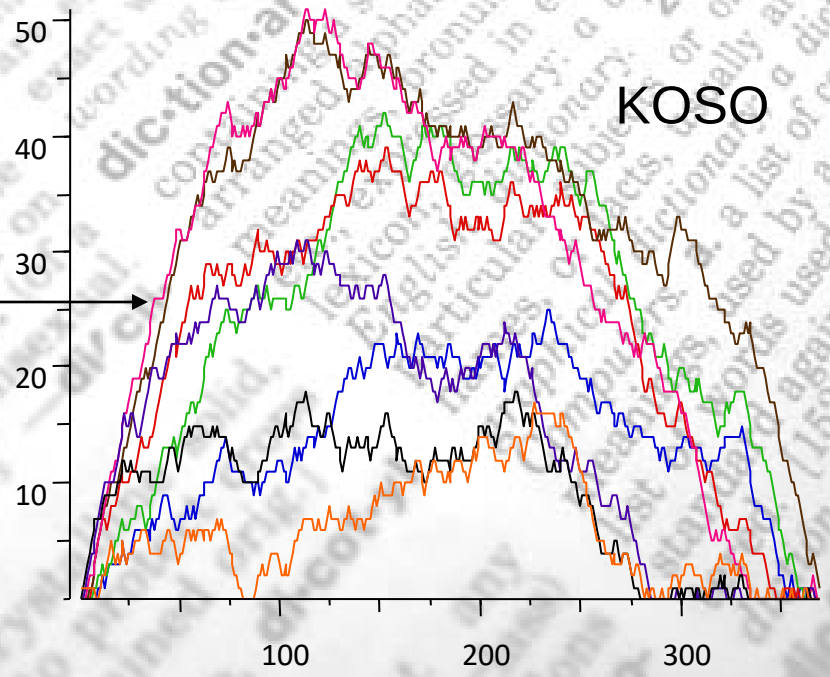


- **Unless processors starve (red arrow) there is no advantage to good load balancing (i.e., KOSO* is no better than KOSO)**

Useful Terms

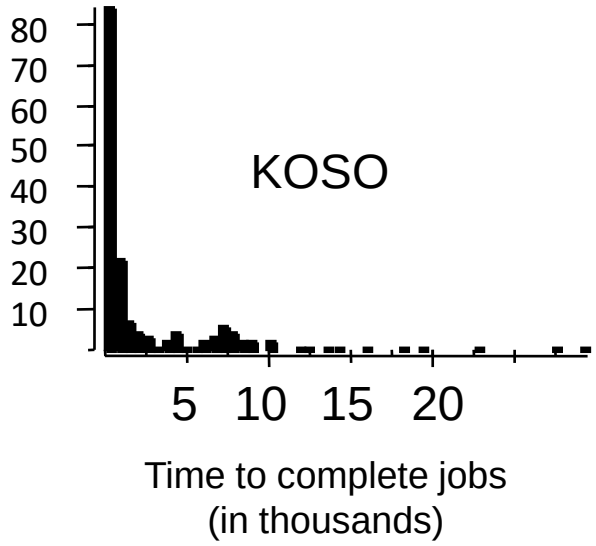
Time series: One or more dependent variables measured at consecutive time points

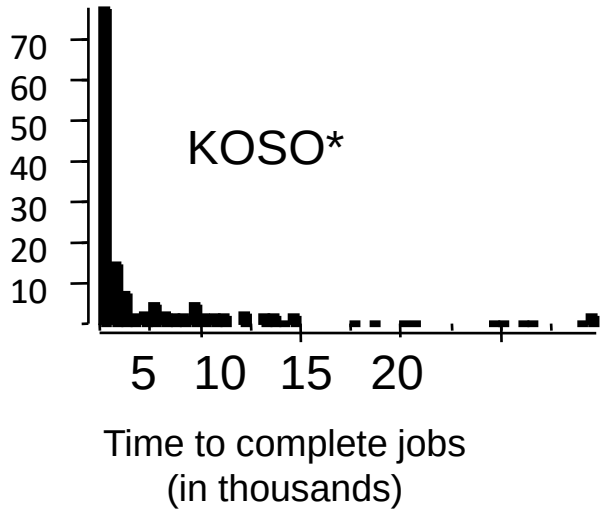
Time series of queue length at processor "red"



Lesson 1: *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**

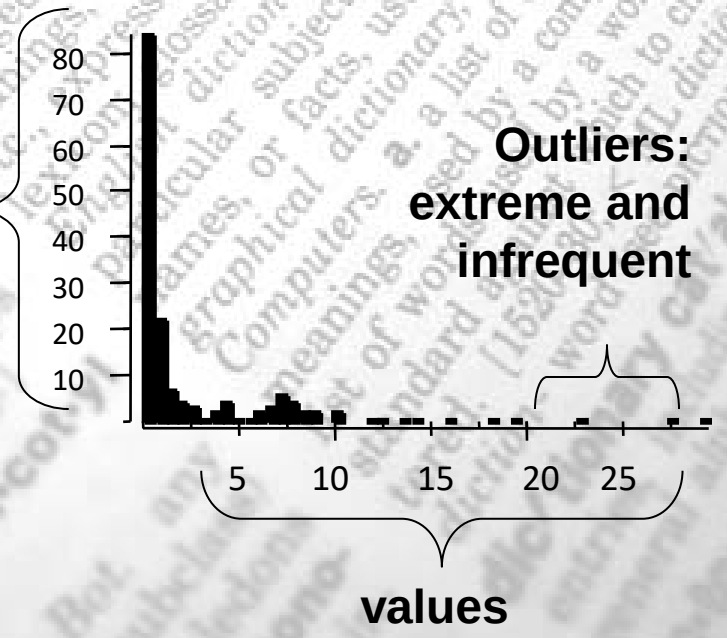
Useful Terms

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

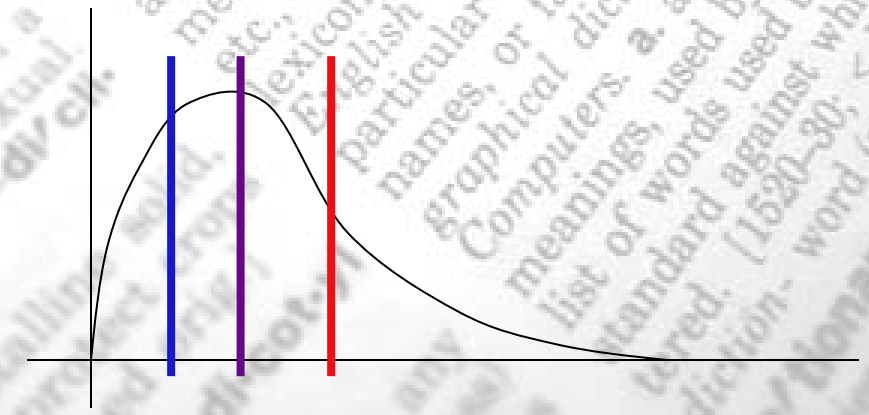
Median: 8

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

Mean: 93.1

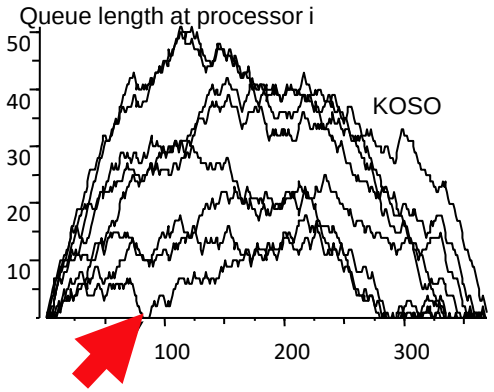
Median: 11

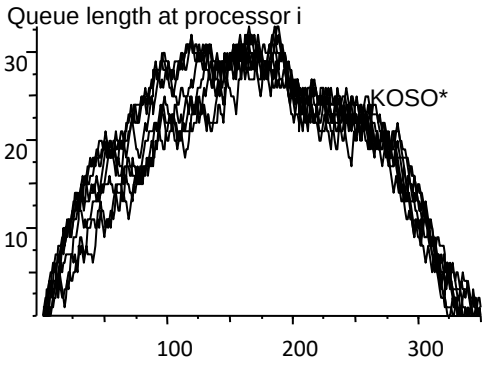
Quantile: A "cut point" q that divides the distribution into pieces of size q/100 and 1-(q/100). Examples: 50th quantile cuts the distribution in half. 25th quantile cuts off the lower *quartile*. 75th quantile cuts off the upper quartile.



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?





Exercise

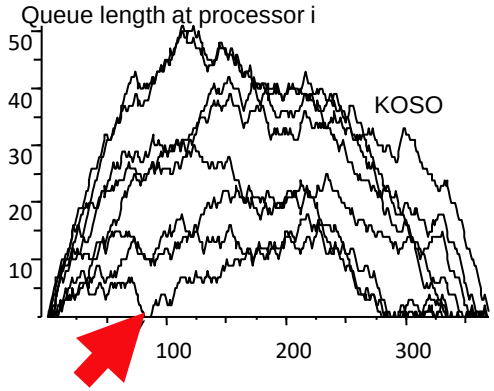
- What do you suggest we do next?
  - How can we change our experimental design so that it evaluates our hypothesis?

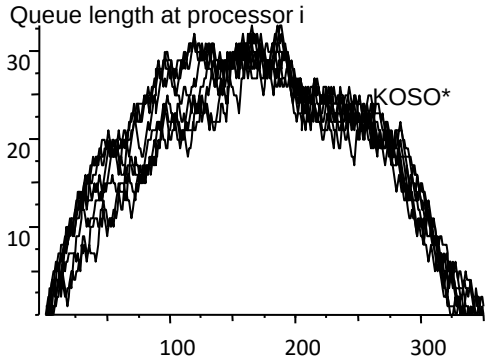
- Hypothesis: KOSO takes longer than KOSO* *because* KOSO* balances loads better.

Lesson 2: Always run pilot experiments

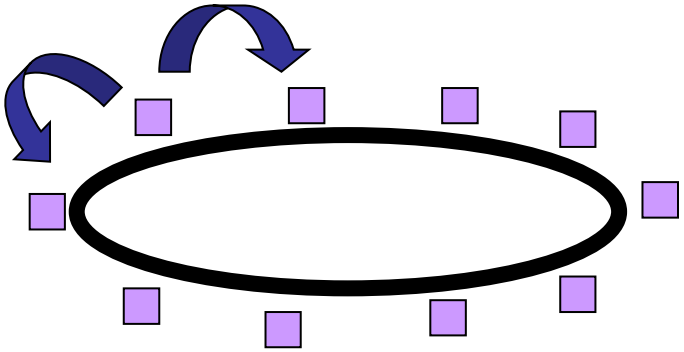
- A pilot experiment tests whether the experimental apparatus *can* test the hypothesis.
- Our independent variables were not set to test our hypothesis.
  - no processor starvation means load balancing not tested.
- Use pilot experiments to:
  - adjust independent and dependent measures,
  - see whether 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…



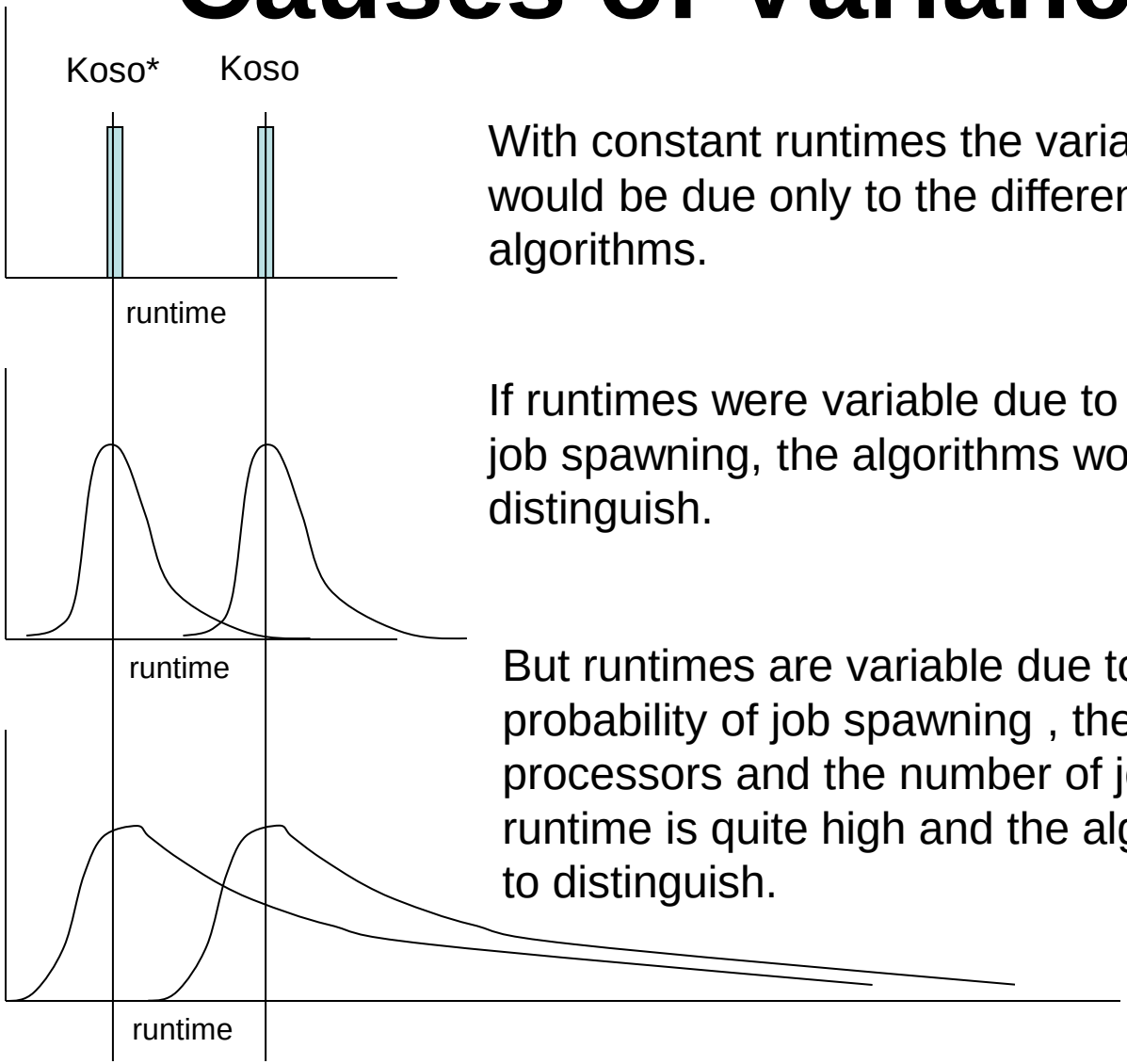


It looks like KOSO* does balance loads better (less variance in the queue length) but without processor starvation, there is no effect on run-time



- 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

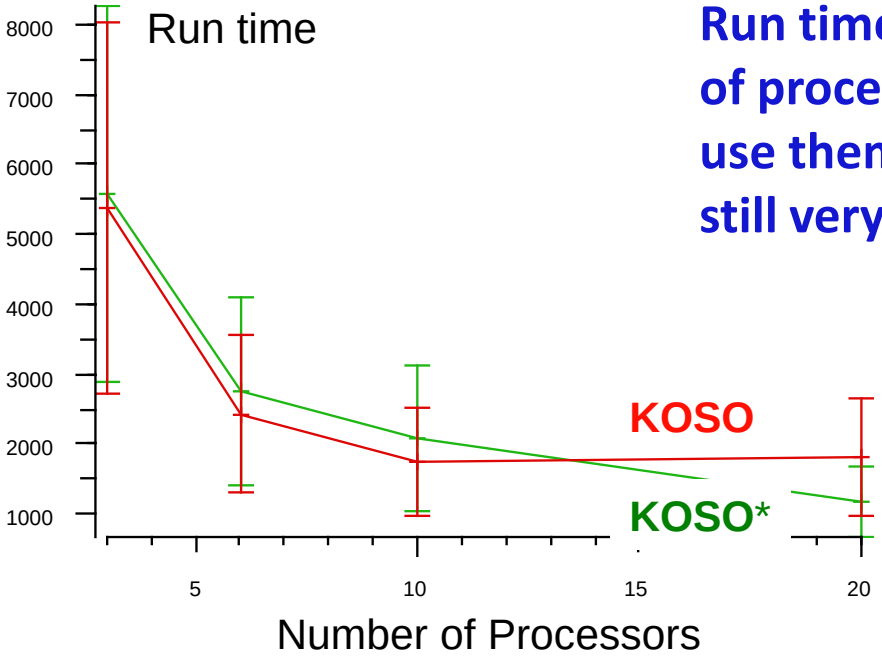


With constant runtimes the variance in runtime would be due only to the difference between the algorithms.

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.

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?

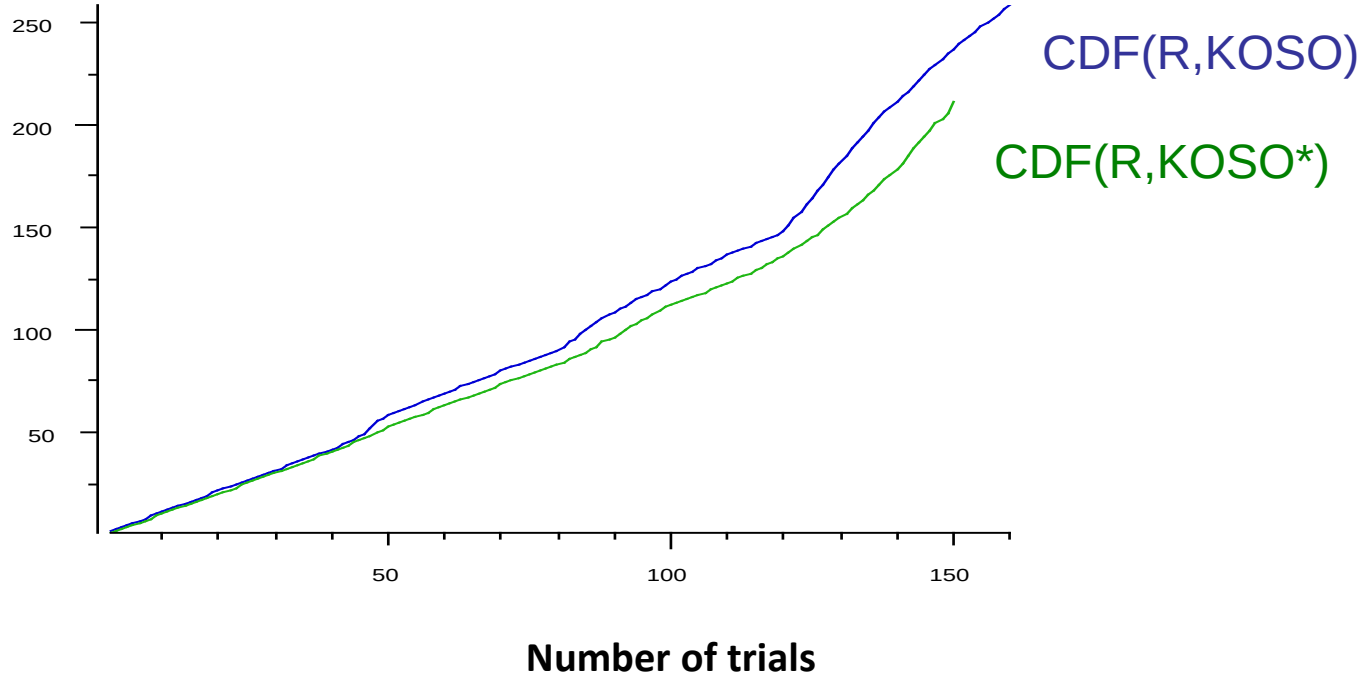
Lesson 3: Control sample variance

- 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 trial 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 4: Check result is significant

	Mean	Median
KOSO	1.61	1.18
KOSO*	1.40	1.03

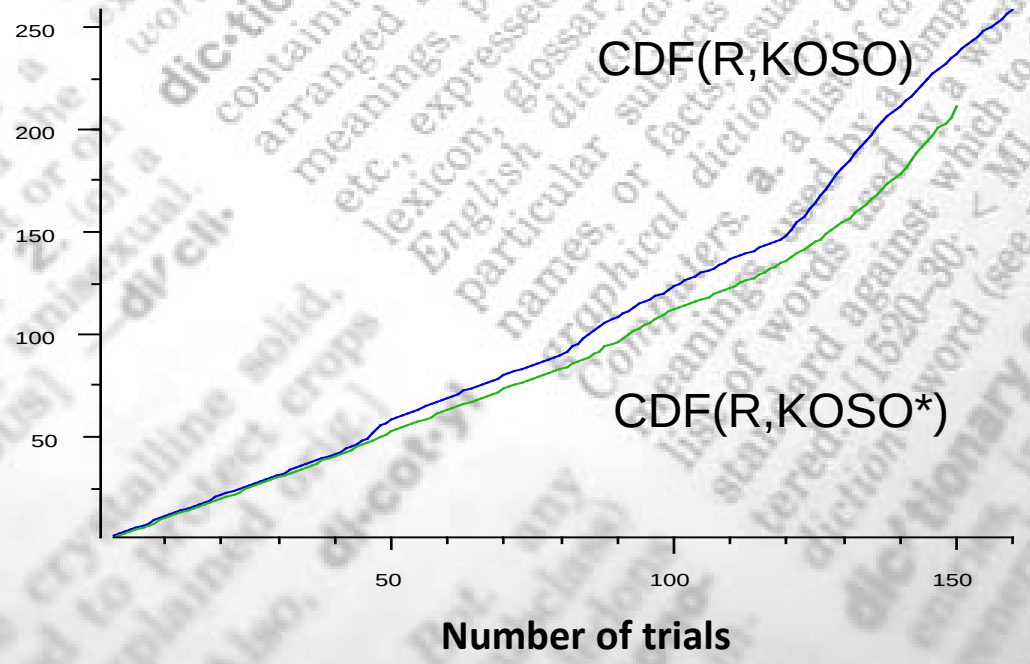
Median KOSO* is almost perfectly efficient



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

Conclusion

- Informatics as exploration of techniques space.
- The importance of hypotheses and their evaluation.
- Use exploratory techniques to understand reasons for experimental results.
- Control for unwanted variance.
- Use statistics to show results significant.

How to Get a PhD in Informatics

A Daunting Prospect?

- Significant and Original Research.
- Creativity is learnable.
- Researchers Bible.
- Anyone can do it:
  - sufficiently bright;
  - work hard;
  - take this advice.

Choosing a Project

- **Criteria project must meet:**
  - inspiring;
  - significant and original;
  - do-able;
  - supervisable.
- **Sources of ideas:**
  - supervisor & other colleagues;
  - read literature of chosen area;
  - further work suggestions of others;
  - previously, badly done work.

Types of Research

- Development of new techniques.
- Exploration of existing techniques:
  - theoretical analysis;
  - ‘rational’ reconstruction;
  - experimental exploration and hypothesis testing;
  - comparison of several techniques;
  - comparison to natural systems.
- Extension and improvement of existing techniques.
- Application of known techniques to new domains.

Hypothesis and Evidence

- What hypotheses will you investigate?
- Along what dimensions will you explore properties or relations of techniques or systems?
- What kind of evidence will you present to support your hypotheses?

When Things Go Wrong

I’m starting to get the impression that you’re not happy here, Jones.

Postgraduate Diseases

- Manna from Heaven.
- Ivory Tower.
- Solving the World.
- Ambitious Paralysis.
- Computer Junky.
- Stamp Collecting.
- Misunderstood Genius.

Psychological Hurdles

- Loneliness of the long distance researcher.
- Self doubt.
- Early morning --- Cold start.
- Theorem envy.
- Fear of exposure.
- Dealing with criticism.

Good Working Habits: Keeping Regular

- **Regular hours:**
  - **get a routine.**
- **Regular reading:**
  - **outer, middle and inner circles.**
- **Regular writing:**
  - **notes, technical reports and journal articles.**
- **Regular talking:**
  - **informal chats, seminars and conference talks.**
- **Regular check-ups:**
  - **where am I going?**
  - **what will it be like when I get there?**
  - **what step should I take next?**

Relations with your Supervisor

- **Meet regularly.**
- **Provide written and oral reports,**
  - **before meeting**
  - **and summary of main actions afterwards.**
- **Talk over problems.**
- **You can swap them.**

Points of Contact in Informatics

- **For practical matters:**
  - **Informatics Graduate School, IF 3.47.**
    - **http://www.inf.ed.ac.uk/admin/IGS/**
  - **College Postgraduate Office.**
- **Otherwise, follow the sequence:**
  - **Principal and assistant supervisors.**
  - **Deputy Head of Graduate School: Alex Lascarides**
  - **Personal tutor: Perdita Stevens or Mike Fourman**
  - **Research institute director.**
  - **Head of Graduate School: Nigel Topham.**
  - **Head of School: Jane Hillston**
  - **Dean of Students: Antony Maciocia.**

Monitoring and Milestones

- **Students have several points of engagement.**
- **Main one: Formal reviews in month 10:**
  - **Student submits written material for review (month 9).**
  - **Presentation to panel (month 10):**
    - **supervisors plus at least 1 external member of staff**
  - **Receives written feedback from the supervisor.**
- **Month 12 (supervisor): Annual report to Graduate School.**
- **Institute review of all its PhD students.**
- **More details: http://web.inf.ed.ac.uk/infweb/student-services/igs/phd/year-timelines**

Annual Milestones

- **Year 1:**
  - **Month 4: Outline proposal and literature review.**
  - **Month 10: Annual review.**
- **Year 2:**
  - **Month 10: Annual review.**
- **Year 3 onwards:**
  - **Month 1: Strategy review meeting.**
  - **Month 4: Complete thesis outline.**
  - **Month 5: Give seminar.**
  - **Month 10: Annual review**

Possible Outcomes of Annual Review

- Confirmation of progression.
- A repeat review within 3 months (1st resort).
- Deferment of decision to the 2nd year.
  - Only for part-time students on 1st review.
- Registration for a different degree:
  - MPhil, MRes, MSc.
- Exclusion from study (last resort).

Conclusion

- **You too can get a PhD ...**
  - **... just by following this simple advice.**
- **Keep doing meta-research.**
- **Keep regular --- stay healthy.**
- **Communicate!**

  Recommended Reading: Researchers Bible.

https://sweb.inf.ed.ac.uk/bundy/how-tos/resbible.html

Exercise

“I can’t answer these questions that I’ve just set.”

Exercise

- Swap your 1000 word project summary with a neighbour.
- Read and critique your neighbour’s summary.
- Provide feedback to your neighbour and vice versa.