

Chapter 2

Pointers on Experiments and Results

This chapter is designed as a refresher on the issues of experiment design and presentation of results. We shall not be presenting an exhaustive survey nor treating the issues in great depth as there are many books which address the points in this chapter already, e.g. [1], although a traditional maths and science secondary school education or science degree ought to cover most of the issues and any university library will have a selection of research methods texts.

2.1 Hypothesis

In science it would be common to start an experiment by defining a hypothesis. This is a “proposition made as a basis for reasoning, without the assumption of its truth; a supposition made as a starting point for further investigation from known facts; a groundless assumption” [4]. The explanation is *proposed* because it does not fit existing models or theories—it may add to them, modify them or replace them. This requires that we have both observed some phenomenon which requires explaining or allows us to make a prediction; and have an explanation that we can test. For some cases the hypothesis will be tested statistically, in others to be true or false. A variation on this is the “null hypothesis”, that there is no difference between a sample and either the whole population or some other population according to a measure—a proposition that this measure is not significant in the definition of the test sample.

Of course rather than the “science” approach, we might adopt the research methods of other disciplines. In particular, ethnography is widely used in pervasive computing—where studies are geared to understanding what people do with the world around them, in a natural (or at least naturalistic) setting. This approach is most commonly used in HCI focused research, which isn’t the focus of this book. However, those expecting to continue with research in this area should be aware of a variety of methods and the other methods that they might encounter. The following may be interesting reading in this area: [2].

We might also try to define a “research question”—wondering “what happens if ...”. This might be “what happens if we give people this system to use” (prompting ethnography) or “what happens if we apply certain stimulus to a system”, typically one which is too complex or whose workings are too opaque for us to make a prediction. As we form a better model of how this class of systems are used, or how the system behaves under a class of stimuli we may move from this open ended approach (which done badly is poking about in the dark and done well is blue sky thinking) towards a traditional hypothesis.

Finally, we may set ourselves an engineering problem and define a set of constraints and evaluate whether a system meets them, or which system meets them at least cost. In such an approach we need to beware of setting arbitrary targets. Existing infrastructure may set constraints and existing competition may set targets, but with research which is further away from deployment it becomes easier to argue for revisions in these constraints.

A question or hypothesis in computer systems is often framed in terms of a comparison, e.g. “system X is better than system Y”. The systems under comparison may take various forms:

- Some existing system, which we believe a different approach can better.
- A single system running with different parameters. Parameters may be tuning variables, number of nodes in a network, deployment hardware, workload data etc.
- Well known base cases, typified by either an exhaustive algorithm, a random algorithm or a theoretical best case. The first two are often encountered in networking literature while the latter is found when evaluating algorithms that use some heuristic or approximation to gain a cost advantage.

A null-hypothesis would suggest that two systems are effectively identical under a given measure, despite whatever difference has been created.

2.1.1 Measures and Conditions

Occasionally it is sufficient to say that an algorithm “works” or meets some criteria, or provide observations of the use of a system, but more often we want to answer some combination of questions, such as:

- Is system x or y better under a certain range of conditions, on average and in the worst case?
- Do algorithms x, y and z scale?
- If we make a change to the configuration of x does this provide an improvement in speed, memory and network use over a range of work-loads?
- In what situations is x better than y?

In each case we need to make *measured* comparisons between systems with equivalent conditions. We use the plural as we generally find that we must consider both

costs and benefits, and claim superiority with a caveat. A measure will typically be numeric, with units. Care must be taken that any noise in the reading is allowed for and that sufficient samples are taken that the result is not due to chance, start-up effects or some external factor. Many experiments need repeating to gain statistical measures of performance, in which case either the conditions must be repeatable or any variation in conditions must be subject to a null hypothesis. Repeatable conditions may require stored rather than live data, e.g. from sensors, or fixed seeds in random number generation for probabilistic simulations.

The necessary dual to measures are conditions. These are the variables that can be controlled in the study—choice of algorithms, parameters for configuration, choice of data sets etc. In some cases conditions have many components, such as hardware, operating systems, other system load, environment, identity of participants etc. Note the system configuration in your experiments as all kinds of parameters might be useful for those that follow to understand how their results and yours are related and might be repeated and in later analysis some aspect may turn out to be more important than you anticipated.

Examples of measures and controls we might consider in this book include:

- CPU and memory use measured for different benchmark algorithms and workloads as controls
- network throughput measured with different packet sizes and loss rates as controls
- distribution of difference (measure, second order) between reported (measured) and actual (controlled) values from signal processing of a given data set (controlled) with different algorithms (another control)

For numeric measures and conditions we might plot a graph, typically with the measures on the y axis, the condition of interest on the x axis, and multiple lines representing discrete conditions—large scale changes in parameters, different algorithms or data sets, as illustrated in Fig. 2.1. Such a graph allows us to identify how the performance of each discrete condition compares to the others. In the example we note experimental test conditions indicated with a point, connected by lines; a theoretical result plotted as a line; “condition two, value B” has no result for $x = 1, 2$ (presumably indicating it does not function under that condition); and the two test systems each have sections of the x axis condition where they are closer to the theoretical result but that “condition two, value A” rises faster as values of “condition one” get higher.

2.2 Method

There are a number of techniques which are used to test a hypothesis:

- Argument—more or less logical, preferably with citations of facts and figures. This is a good basis for setting out a position, but does not verify that the position is correct. One comes across this in the research literature in the form of position papers, which describe early work; and in research proposals, which argue that an idea merits funding.

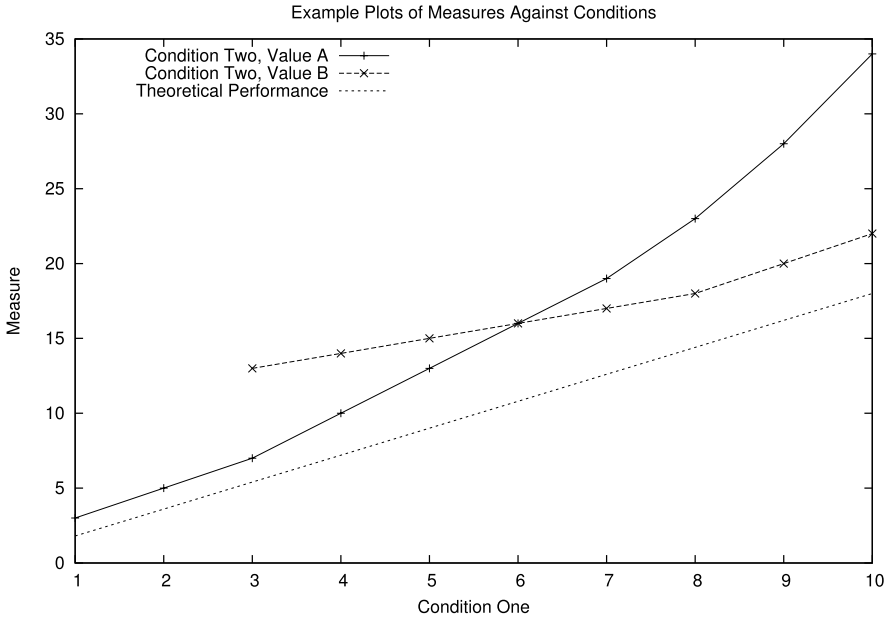


Fig. 2.1 An example plot of a measure under two conditions

- **Proof**—a mathematical process which shows that an idea is correct, often using logic but may use other analysis techniques. A marvellous thing, which any theoretical concept should seek. In addition one can prove properties of algorithms. However, in this text we tend to focus on applications and implementations. In these situations logic gives us confidence in an idea but relies on assumptions which need to be checked; formal techniques in programming can also be used to give confidence in an algorithm's properties and so in the smooth running of the program.
- **Experiment**—setting up controlled conditions and testing an implementation. This is an approach which we tend to promote in this text, as a good basis for examining the properties of sensors and of small pieces of developed code. We would urge the reader to see whether a traditional problem–hypothesis–method–results–conclusion approach to an issue is viable. If so, the approach makes the scientist readers relax and the structure tends to lead to results which are easy to analyse. However, be aware that an “experiment” in this sense usually relies on well controlled conditions and often a control case, for comparison. As a study gets closer to natural human use controlled conditions get harder to argue and complex human and external factors make results harder to analyse (requiring greater numbers of participants and more complex statistical analysis).
- **Simulation**, a form of experiment deserving special mention here—extracting an idealised model of the thing being studied, without the problems of live subjects in an experiment, and performing experiments on this. The models can be sophisticated enough to include some random (usually probabilistic) factors and require

multiple runs to establish average results and variations. Simulation can also be used to exhaustively test software. The limits of this approach are (rather like the mathematical proof) the accuracy of the assumptions made: would the introduction of the thing being tested change behaviour, is some issue being missed?

- Survey—watching what happens, involving techniques including questionnaires, interviews, video, observation and note-taking and instrumented software. The setting is usually as natural as possible (ideally not staged at all) while trying to minimise the involvement of the watcher for fear of altering what happens (the Hawthorne effect). The problems here are the effort required to undertake the survey and analyse the findings; the effect of the observer and the form of questions asked; the general validity of the set of subjects being observed; the problems of creating valid comparisons with different human participants—not to mention issues of funding deployment, research ethics, health and safety etc.

So, *no approach works*? Of course, each has its place and we would urge the reader to consider which is the right way of exploring the qualities of the issue under test. Often there is a progression: argument to convince peers that the idea merits consideration; then analysis to convince yourself the design is sound; then simulation to establish expected sub-system behaviour; then controlled experiment to verify correct function with real users and to review a whole-system design; then limited natural deployment to explore the subtleties that arise in prolonged use; then product for sale to make more general conclusions about a broad range of users—or some sub-set of these. In some cases our hypothesis concerns something smaller than an application or system, and simulation and experiment are the main tools, as testing algorithms and devices is usually best carried out in a physical science / engineering tradition; alternatively if the questions are more open with unanticipated outcomes then surveys and observational experiments drawing from the social sciences are more appropriate. In this text we are mostly concerned with experiments, but many of the comments which follow are applicable to experiment design through simulation and to survey design.

2.3 Collection of Data

Various data collection methods may be found in pervasive computing, including:

- Instrumenting code or log files to note system events
- Physical measurement (controlled experiment)
- Measurement of simulated system
- Timed / monitored activity, either directly, by video, by system logs etc.
- Questionnaire, interview, focus group (less applicable to testing of systems but very relevant to testing the use of systems)

Each has its own strengths and weaknesses and is applicable to different situations. None is “easy” and it is always possible to use more data—although not always possible to meet deadlines if there is more than you can process! In all cases keep

your raw data and process later. It is much easier to generate new statistics and analyses from raw data in log files than it is to re-run an experiment where data is generated and processed at run-time.

If relying on instrumented code in a deployment care needs to be taken that the instrumentation tells you the values you need, with sensible units, accuracy and frequency; that any filtering and aggregation is correct. Testing of the run–data collect–data analysis cycle is time consuming. Tests in the lab may not reveal problems arising from deployment scale, timing from multiple users, environmental factors and unexpected user behaviour. Controlled experiments and simulations do not suffer from the unpredictability of “the wild” but have requirements for careful set up and data collection as above. The observation of- and interaction with- users can give vital information about *systems* research that tests on algorithms and simulators alone cannot answer. While our focus in this book has been on systems pervasive computing lends itself to applied research and the connection with users is a desirable end-point.

2.4 Analysis of Data

There are many analysis techniques and it is not our purpose to address them all. For statistical data the selection of analysis will depend on the number of variables, the type, completeness and volume of data. The results in this case will typically be presented as a correlation, with a given confidence, using a certain test over n data points—all this information is needed when reporting results. Even when reporting correct operation the range of tests and number of runs are vital pieces of information for any non-trivial system. Where sensors are involved then it is important to document the tool chain properly: what sensors, what placement, what stimulus, how many runs, processed with what algorithm, on what platform? For any experiment documenting the tools and data set is important for allowing repeatability; where interfacing with the real world is concerned proper documentation of a rigorous procedure gives confidence in the findings.

2.4.1 Presentation of Results

The results, in particular any graphs, will be read directly after the abstract by some readers. A clear message is vital to putting forward an idea, and a well presented analysis is key. Be sure to address:

- What did you find?
- How can the reader use this finding in their work?

Don’t bury the message in caveats, but do make the bounds of your work clear in the method and any anomalies or unexpected findings clear in the discussion.

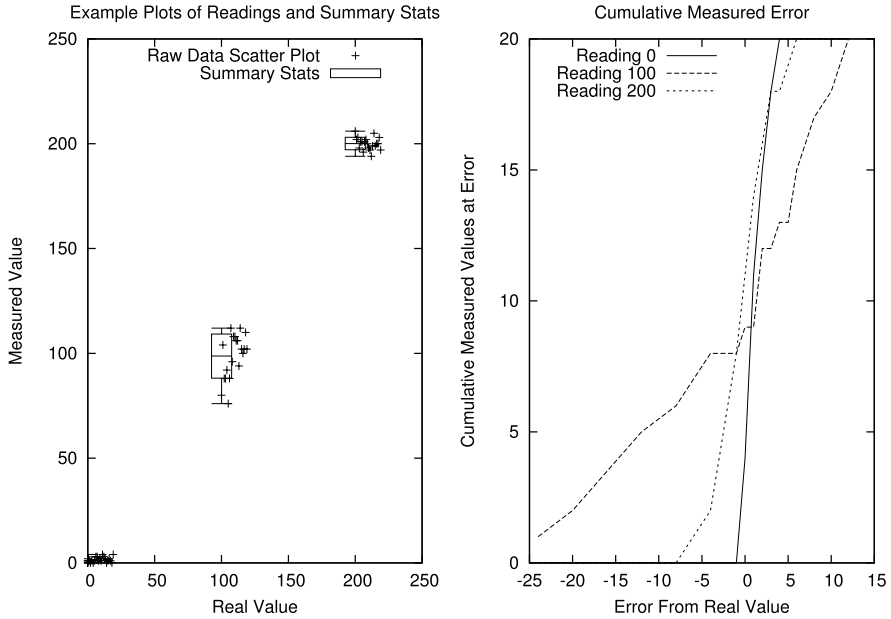


Fig. 2.2 An example of scatter and box and whisker plots for measurements of a known condition, 20 data points at each real value

So, what tools are useful? Of course this depends on the hypothesis / question / observations you made, what the form of the data and analysis is and what the findings are, but common tools include:

- Tables of data. Really only useful for simple summary data. A graph or analysis of correlation is often more useful.
- Histograms, where you have several comparisons and a few conditions to compare on one axis.
- Line graphs, where you have a small number of comparisons over a more numerous (tending to continuous) range of test points (x axis). Often the most useful form of results in considering performance of a number of alternative systems against time or controlled variables.
- Scatter plots, where something close to raw data with an accompanying discussion aids your narrative.
- Be clear about patterns of noise: is it Gaussian, long tail, something else? What have you done to mitigate it? Have you shown standard mean, mode or median, deviation, percentile, limits in e.g. box/whisker plots?
- Textual statement of statistical findings. A simple test of a hypothesis with a clear claim attached, qualified with confidence and number of data points.

Most research methods texts will expand on the presentation of data, and for a particular research area following the norms of that community is fairly safe, but for a deeper read on this Tufte's book is excellent [3].

The graph in Fig. 2.2 illustrates the plotting of raw data with a scatter graph (each value slightly offset on the x axis for clarity) and box and whisker plots of summary statistics: min , $mean - stddev$, $mean$, $mean + stddev$, max . Standard deviation has been used due to an assumption about Gaussian noise, percentiles are also common in this form of plot. It can be seen that where the real value is zero the assumption about normally distributed noise breaks down, as the minimum measurement is zero. The measurements for $x = 100$ have a much wider spread, reflected in their standard deviation and minimum and maximum bars; we also see that although the noise is essentially Gaussian there is a significant outlier below the mean. We also show a graph of the cumulative counts of data points at each difference between the measured and real value, with three lines: one for each real value. The x -axis plots the error to normalise the scales for each plot. We note that this also shows the wider range of the $x = 100$ condition, the positive bias of the $x = 0$ condition, and also the steep rise around zero error which implies that many readings are close to the correct value.

2.5 Research Ethics

Last, but by no means least—when designing experiments which sense human activity and are deployed into the world ethics and safety must be considered. Most research organisations will have a research ethics process, where someone independent will ask questions of research involving people, e.g.:

- Is it physically safe? What has been done to manage any risks, e.g. selection of participants, adjustments to the method, provision of assistance and emergency procedures.
- Might people be upset? What has been done to avoid this and/or handle it when it occurs?
- Is there any risk to the investigators? What has been done to mitigate this?
- Might results be skewed by inappropriate inducements?
- Will participants be aware of the experiment? Before, during, after?
- Is the experiment collecting sufficient data to give good results but not more than is needed?
- Will data about people be treated in a legal, secure and ethical manner? When will it be deleted?
- Will data about participants identify them? If so, can they request its removal?
- Will any publications using the data collected, especially pictures or internet data, allow association of individuals with the study by the reader?

It is not possible to prevent the unexpected (and if the outcome is guaranteed then what kind of experiment is it?), but it is possible to show that you have properly considered any risks and taken care of those that you identify.

2.6 Summary

In this chapter we have not presented any technique in detail, but have given reminders and pointers for a range of tools that will be useful for the lab work in the rest of the book. As a researcher or as an engineer, there is no single “correct” approach to extending the subject or better understanding our products. However, as a professional in these fields one should be aware of what approach we are taking, why we have chosen it and what the alternatives are. This applies to the choice between hypothesis, question and ethnography; to the measures we make and analysis we apply to answer our questions. These choices will inform our methods, the controls and conditions we apply, the data we collect. There are many possible solutions, and to spark ideas and then test them requires the right sort of enquiry to make progress. I wish you many revealing, exciting and enjoyable hours of research—I hope the ideas in this book will inform some of them. Having set the scene, we now delve into the issues in pervasive computing.

References

1. Clarke, G.M., Kempson, R.E.: *Introduction to the Design and Analysis of Experiments*. Arnold, Sevenoaks (1997)
2. Crabtree, A., Benford, S., Greenhalgh, C., Tennent, P., Chalmers, M., Brown, B.: Supporting ethnographic studies of ubiquitous computing in the wild. In: Carroll, J.M., Bødker, S., Coughlin, J. (eds.) *Conference on Designing Interactive Systems*, pp. 60–69. ACM, New York (2006)
3. Tufte, E.R.: *The Visual Display of Quantitative Information*, 2nd edn. Graphics Press, Cheshire (2001)
4. Tulloch, S. (ed.): *The Oxford English Dictionary*. Oxford University Press, Oxford (1995)



<http://www.springer.com/978-0-85729-840-9>

Sensing and Systems in Pervasive Computing

Engineering Context Aware Systems

Chalmers, D.

2011, XXI, 173 p. 27 illus., Softcover

ISBN: 978-0-85729-840-9