

Practical Tips on How to Conduct a Sophisticated Online Psychological Experiment

Jeromy Anglim (jkanglim@unimelb.edu.au)

Department of Psychology
University of Melbourne, Parkville VIC 3010 Australia

Lea Waters (l.waters@unimelb.edu.au)

Department of Management & Marketing
University of Melbourne, Parkville VIC 3010 Australia

Abstract

Researchers interested in greater access to participants and reduced data administration costs are frequently using online software to administer simple surveys. However, many psychological research paradigms are more than just a series of survey items. Psychological research typically has more sophisticated requirements such as the need to use experimental designs, to allocate participants to between-subject conditions, record response latencies, time stimulus presentation in a precise manner, randomise ordering of within-subject conditions, and present feedback. In order to realise the benefits of the online environment while having the control required of experimental psychology, we started using online psychological experimentation software (Inquisit). We have now conducted numerous online psychological experiments on such topics as personality faking, social network analysis and skill acquisition. Drawing on experience we will: 1) display examples of our online experiments; 2) highlight useful features; 3) discuss strategies for overcoming challenges involved with online experiments; and 4) provide tips for learning how to program the experiments in Inquisit. The talk should be of interest to academic, student, and practitioner researchers who are interested in conducting more sophisticated psychological research over the internet.

Introduction

The inspiration for our initial foray into online psychological experiments came from a set of research questions. We wanted to learn about the factors that influenced response latencies to personality test items and what, if anything, response latencies tell us about an individual. Several studies had suggested that there is a curvilinear, (inverted-U) relationship between personality item response latencies and selected response category, such that response latencies are longest for the neutral category and shorter for more extreme responses such as “strongly agree” or “strongly disagree”. There is currently a great deal of interest in the selection and recruitment literature concerning how and why some people fake their responses to

personality tests. Being able to detect ‘faking’ has considerable implications for researchers and practitioners alike. Given the rise in online administration of personality tests, it seemed natural to want to explore the degree to which response latencies were or were not diagnostic of response distortion.

To do this research we knew we needed to have some form of computerised administration of items using a process that accurately recorded response latencies. Test items needed to be presented in randomised order so that any learning effects could be differentiated from effects of item characteristics, such as sentence complexity or length. We were using a student sample and we wanted students to be able to do the exercise in their own time, and preferably from home. An extensive search of online survey providers revealed many good providers of easy to configure and affordable online survey solutions, such as QuestionPro.com and SurveyMonkey.com. However, none of these online survey providers allowed for item level recording of response latencies or the kind of control over stimulus presentation that we desired. Examination of psychological experimentation software revealed many useful tools, such as Media Lab, Direct RT (www.empirisoft.com), E-Prime (www.pstnet.com), SuperLab (www.superlab.com), and Presentation (www.neurobs.com), which, while good in themselves, had at the time of writing limited online capabilities (for a comparative review of psychological software, see Stahl, 2006). We also looked at more generic programming tools, such as Authorware, Flash, Java applets, and Visual Basic. While these tools were immensely flexible, they were not customised to the task of running psychological experiments, and would have required substantial programming to enable features such as accurate timing measurement, precise stimulus presentation, and reliable and secure transmission of data. Eventually, we discovered Inquisit (www.millisecond.com), which seemed to do everything we required. The combination of experimental control and response latency measurement in the online environment was the key factor, which

made the collection of data viable. We have now conducted several online psychological experiments using Inquisit and found it to be an effective means of facilitating the answering of our research questions.

Before we discuss the lessons we have learnt about conducting online psychological experiments, it is important to point out existing articles in the literature on the topic. Arguably the most important document is by the APA Board of Scientific Affairs', where Kraut and colleagues (2004) provide a comprehensive overview of the opportunities and challenges of conducting psychological research on the internet. Kraut and colleagues set out the pros and cons of online data collection and provide advice to researchers, and ethics review panels, regarding the ethical conduct of online psychological research. Nosek, Banaji and Greenwald (2002) discuss similar issues providing many pragmatic strategies for addressing ethical concerns and improving design quality in online experiments. Beyond these two articles, there is now a large quantity of articles that have been published using online data collection methods supporting the idea that online data collection can yield valid results. In particular, McGraw, Tew, and Williams (2000) set up an online psychological laboratory and were able to replicate results from many classic lab-based psychology experiments.

Our article should be read in conjunction with the above articles (i.e., Kraut, et al, 2004; Nosek, et al, 2002). In terms of discussing the pros and cons of online psychological experimentation from ethical and pragmatic perspectives, our article overlaps with the above articles. Our paper differs from the above articles in that the emphasis is on the particular challenges we have encountered in learning how to successfully and ethically program, design, conduct, and analyse our online experiments. The aim is to share some of the strategies that we have adopted to overcome these challenges in the hope that this will be useful to others wishing to conduct online psychological experiments. While some of the material specifically concerns the use of Inquisit, much of the content should also be relevant to researchers adopting other online technologies.

Benefits of Online Experiments

Before outlining the challenges of online data collection it is important to outline some of the benefits to online data collection. Of course, many of these benefits are related to computer administration and could be achieved in the off-line environment. The main reason for outlining these benefits is to motivate researchers to explore online research. What follows is a list of some of the main advantages from our perspective.

Easier Data collection: Computerised administration means that there is no need for manual data entry, which saves time and prevents data entry errors. Online data collection does not require experimenters to conduct the research and does not require laboratory space. Online data collection also makes the pooling of individual participant data files easier by storing data on a centralised server.

Easier Participation: Participants can complete the experiment from a location and time of their choosing. This greater ease of participation further opens up the possibility of greater access to participants.

Experiment Control & Adaptive Experiments: Randomising stimulus ordering (i.e., question items, other stimuli), and randomising ordering of experimental sections are all powerful tools for reducing bias introduced by order effects in an experiment. Similarly, random allocation of participants to groups can be made more sophisticated. The computerised environment facilitates this in ways not possible in paper and pencil experiments.

Computerised experiments can adapt to user behaviour. For example, 1) questionnaire skipping and branching logic can be used to direct a participant to particular questions given responses on previous questions; 2) Performance on a task can be assessed dynamically and used to allocate participants to a particular condition; 3) Participants can input information, such a list of names in a social network application, and subsequent questions can present this user-entered information in a useful way.

The above benefits were sufficient for us to justify the adoption of an online experimental paradigm. We now discuss: 1) strategies for learning Inquisit; 2) strategies for designing and implementing the experiment; 3) challenges in conducting online research; and 4) ethical issues in online experimentation.

Learning Inquisit

Programming an Inquisit experiment involves writing a script. To provide a sense of what a script looks like, a very simple but complete example is presented in the Appendix. Inquisit scripts follow a similar syntax to HTML. They are made up of limited set of elements (e.g., `expt`, `block`, `trial`, etc.). Elements contain attribute-value pairs that determine behaviour of the element and the experiment (e.g., `/blocks = [1=intro]`, where `blocks` is the attribute, and `[1=intro]` is the value of the attribute). The example shows the hierarchical nature of Inquisit experiments with experiments (`expt`) calling blocks (e.g., `block`)

calling trials (e.g., `trial`) calling stimuli (e.g., `text`) calling content items (`item`).

The example script has only one block called `intro` that has three trials (`trials = [1-3=intro.q1]`). Each trial displays two text elements at the start of the trial (`stimulustimes = [0=intro.q1, clickhere]`). One element of text is for the interface (`clickhere`), which needs to be clicked to end the trial (`validresponse = (clickhere)`). The other element of text (`intro.q1`) displays different text on each trial moving sequentially (`/ select = sequence`) through a list of items (`intro.q1.items`).

In general the Inquisit scripting language is simple enough for experimenters to design and program themselves, particularly if they have some experience with programming in HTML, SPSS syntax or an actual computer programming language. The Inquisit website (www.millisecond.com) provides useful resources including detailed specifications of features, sample scripts, and a fully functional demo version of the software.

When learning to write your own Inquisit scripts, there are several strategies which we have found useful. 1) Study sample scripts that are available from the Inquisit website. Think about what each line of code is doing and understand the program flow. In particular identify the details of the syntax conventions of the programming language, such as when to use round brackets, square brackets, commas, and semi-colons. 2) Attempt to modify existing scripts. It can be useful to iterate between the help files and the sample script and examine the consequences of making small changes. 3) Start writing your own simple scripts. When learning any programming language it is generally desirable to make small changes in order to make it easy to identify the reasons why at any point it is not working. In general learning to write Inquisit is like learning any language in the sense that it is important to spend some time reading the language (i.e., reading other people's code). By spending time studying the language, more generally, less time is spent trying to implement some particular technical feature required, because the basics have been consolidated.

Equally, the task of writing an Inquisit script can be outsourced. IT staff are likely to find the language quite straightforward. Equally, there are often post-graduate students and lecturers in academia willing to write an Inquisit script in exchange for some form of authorship on resulting articles.

Programming and Design

Once the basics of writing Inquisit scripts have been learnt, it is important to think about how to design,

code, and process your online experiment in a manner that is efficient and reliable. The following are some strategies that have proved helpful to us in improving the design and programming of online experiments in Inquisit.

Make a Detailed Design: Set out a thorough design before commencing coding, which as a minimum includes: a) exact question/stimuli wording; b) exact specification of response options and interface options; c) specification of display elements; d) specification of sequencing of trials; e) specification of branching and skipping logic. With a clear experimental design, the task of actually programming the experiment becomes less daunting because the only challenge that remains is how to implement the design. Equally, implementation of the design in Inquisit could be outsourced to an external programmer.

Adopt a Naming Convention: Adopting a clear and consistent naming convention for item elements is essential for managing the script as it grows larger. Although many such systems exist, we have found it useful to adopt a dot system, where block elements are given a name and all lower level elements (e.g., `trial`, `likert`, `page`, etc.) are named *blockname.trialname*. For example, if a block was called `intro`, the first trial involving a question could be called `intro.q1`.

Automate Code Creation: A large part of an Inquisit script involves repeated elements. It can save a lot of time, if code writing can be partially automated. One way of doing this involves using Microsoft Excel to generate parts of the script and then copying the text into Inquisit. Content that varies can be placed in one set of columns (e.g., item content and item number) and constant content can be placed in another column. Formulas using concatenation (“&”) and cell references can then be used to create the text in Excel.

Debug: It is important to engage in an appropriate debugging strategy. At first instance, this involves doing basic checks that the experiment runs and the data produced appears appropriate. Inquisit provides the “monkey” to automate the process of randomly responding to the experiment. It is important also to manually check any branching or skipping in your experiment, verify the operation of each between subjects condition, and inspect the data file generated. Once the experiment commences initial data should also be checked.

Pilot Test: In the online environment there is typically no experimenter present to answer questions

from participants during the experiment. There is also no direct observations of participants. Thus, it is especially important that instructions and interface design are clear. To ensure comprehension it is desirable to pilot test the whole experiment with a small group. This can involve a talk-aloud protocol, where the experimenter makes note of any misunderstanding.

Process Data: Processing data is a necessary and potentially frustrating task, although in many ways less frustrating than paper and pencil data collection due to the lack of data entry. Inquisit generally produces a data file in 'long' format, where each row is one trial for one participant. Most statistical packages assume that the data file is in 'wide' format, where each row is one participant. SPSS provides a number of tools for importing text files, aggregating, merging, and restructuring (i.e., converting from long to wide format) data files, and recoding and computing new variables. It is preferable to use syntax for this as the task is often repeated. We have used the free open-source statistics package R (www.r-project.org) to do all the initial data processing. R is excellent but involves a moderate initial learning curve for people used to menu-driven packages such as SPSS.

Challenges and Strategies

Obtaining valid data: Obtaining valid data is important to anyone conducting research. Instructions can be misunderstood and participants can adopt forbidden strategies (e.g., performing web searches to assist answering ability test items). Managing threats to validity requires identification at the study planning phase of all likely threats. Some combination of a threat-avoidance and threat-detection approach can then be adopted. Many of these threats are similar to those encountered in lab-based experiments. The difference is that the ability to avoid and detect these threats is reduced in the online environment (Nosek, et al, 2002). However, relative to the paper and pencil designs computerisation creates many opportunities to improve threat avoidance and detection.

Strategies for **threat-avoidance:** a) instruction screens can be constrained to display for at least a certain amount of time, to reduce the risk of participants not reading instructions; b) Incorrect answers to instruction comprehension checks can trigger additional explanation; c) participants can be told how they should complete the study for their data to be valid (e.g., in a quiet environment, on their own, where they have sufficient time to complete the study; to only complete the study once, etc.); d) responses to test items can be prevented until a minimum amount of time has elapsed in order to prevent item skipping; e) strategies can be

adopted to reduce the motivation and ability of participants to adopt undesirable behaviours during the experiment.

Examples of strategies for **threat-detection:** a) response latencies can detect participants not reading test items (e.g., < 400 millisecond responses); b) items can verify instruction comprehension; c) tests can assess appropriateness of the participant to participate, such as a basic English language test; d) participants can be asked directly whether their participation is valid, such as whether they understood the task instructions or whether they took the experiment seriously.

Maintaining motivation and preventing drop out: The social pressure of the experimenter that discourages drop out in lab-based studies is not typically present online. Thus, it is more important to increasing participant motivation to persist with the study. Some strategies include: offering to provide interesting feedback to participants about their responses; offering prizes and financial incentives to participate; and integrating the study into the aims of some other activity such as university subject. Limiting drop out in long online experiments may require particular strategies. In a lab experiment, the experimenter's presence creates an atmosphere that discourages distraction and dropping out of the experiment. In an online environment, this is less the case. One strategy is to limit online experiments to a certain length, such as 30 minutes. Another strategy is to split the experiment into sessions, although there is then the challenge of ensuring completion of all sessions.

Ethics

Research in the academic environment requires ethical approval. One reason for this is the relative novelty of online experimentation from an ethics reviewer's perspective. While the principles of ethical research do not change, there is less established precedent about what the ethical conduct of online experiments involves. There are also some additional considerations that are important to consider when conducting online experiments. The points below should be read in conjunction with the more comprehensive discussion by Kraut and colleagues (2004).

Data Security: There are several considerations regarding the security of online data. Where identifiable or confidential information is being obtained: a) data should be securely transmitted from the participant's computer to the data storage location; b) data should be securely stored on the data storage location; and c) access to the data storage location should be limited, by password or otherwise, to the authorised investigator.

Inquisit facilitates these goals by using SSL encryption and a secure web server for data storage.

Access to the Experiment: Depending on the nature of the experiment it may be necessary to limit the experiment to authorised participants. Only providing the web link to participants may be sufficient. Other times you may wish to provide a user name and password system that is sent to authorised participants. This can be set up in Inquisit on the first couple of screens of an online experiment.

Consent Forms: Traditional consent forms often involved a piece of paper that is signed by the participant. In the online environment, it is easier to simply have a checkbox to indicate consent. In some senses completion of the experiment represents implied consent to participate. Other options include receiving an email from the participant indicating their consent or even maintaining a paper consent form.

Right to Withdraw Data: While a participant's data remains identifiable, it is usually ethically expected that they have the right to withdraw their data from the study. This can be done by simply providing an email address that the person can send a message indicating their desire to withdraw their data. A question can also be presented on the last screen of the experiment asking the participant whether they wish their data to be used for research purposes.

Additional ethical concerns and strategies for managing them are discussed in Nosek, Banjek, and Greenwald (2002). These include ensuring participant debriefing and limiting unwanted participation by underage participants.

In general, online research does not appear more or less risky than face-to-face research. As Nosek and colleagues (2002) note, the absence of the social pressure by the experimenter to participate may actually reduce the risk of psychological stress being caused. Nonetheless, the ability to assess when adverse participant reactions do arise and the ability to manage these reactions is made more difficult in the online environment. In many senses the same reasoned decision making process of avoidance and detection that is applied to experimental design can be applied to risk management of online experiments.

The Future of Online Experiments

Online psychological experimentation represents a huge potential to increase the efficiency of psychology research, increasing sample sizes and the number of studies that researchers are able to conduct. Online research raises new design and ethical challenges. However, creativity, reasoned decision making, and

technological solutions represent ways of overcoming these challenges. Furthermore, the combination of sharing raw data and sharing experimental scripts has potential for increasing communication and the cumulative nature of psychological research. The emergence and evolution of easy to use online experimental psychology software, such as Inquisit, bodes well for the advancement of psychological knowledge.

Author's Note

Please address correspondence relating to this article to Jeromy Anglim (jkanglim@unimelb.edu.au). Additional Inquisit learning resources are progressively being made available on the first author's website: <http://web.psych.unimelb.edu.au/jkanglim>

References

- Kraut, R., Olson, J., Banaji, M., Bruckman, A., Cohen, J., & Couper, M. (2004). Psychological Research Online: Report of Board of Scientific Affairs' Advisory Group on the Conduct of Research on the Internet. *American Psychologist*, 59 (2), 105-117. url: <http://www.apa.org/science/apainternetresearch.pdf>
- Nosek, B. A., Banaji, M. R., & Greenwald, A. G. (2002). E-research: Ethics, security, design, and control in psychological research on the Internet. *Journal of Social Issues*, 58(1), 161-176.
- McGraw, K. O., Tew, M. D., & Williams, J. E. (2000). The integrity of web-delivered experiments: Can you trust the data? *Psychological Science*, 11(6), 502-506.
- Stahl, C. (2006). Software for generating psychological experiments. *Experimental Psychology*, 53(3), 218-232.

Appendix

Simple example of an Inquisit script

```
<expt myexperiment> / blocks = [1=intro]
</expt>
<block intro> / trials = [1-3=intro.q1]
</block>
<trial intro.q1>
/ stimulustimes = [0=intro.q1, clickhere]
/ validresponse = (clickhere)
/ inputdevice = mouse </trial>
<text intro.q1> / position = (50%, 40%)
/ items = intro.q1.items
/ select = sequence </text>
<text clickhere>
/ items = ("Click here to continue")
/ position = (50%, 75%)
/ fontstyle = ("Verdana" 14pt) </text>
<item intro.q1.items>
/ 1 = "Hello World"
/ 2 = "Hello Again"
/ 3 = "Good Bye"
</item>
```